

Universidad Católica de Santa María
Facultad de Ciencias e Ingenierías Físicas y
Formales
Escuela Profesional de Ingeniería de Sistemas



**PROPUESTA DE UN SISTEMA DE GESTIÓN DE SÍLABOS EN LA UNIVERSIDAD
CATÓLICA DE SANTA MARÍA**

Tesis presentada por el bachiller:

Meza Luque, Anthony Eduardo

Para optar el Título Profesional de:

**Ingeniero De Sistemas con
especialidad en Sistemas de
Información**

Asesor:

Ing. Rosas Paredes, Karina

Arequipa- Perú

2022

UCSM-ERP

UNIVERSIDAD CATÓLICA DE SANTA MARÍA
INGENIERIA DE SISTEMAS
TITULACIÓN CON TESIS
DICTAMEN APROBACIÓN DE BORRADOR

Arequipa, 15 de Junio del 2022

Dictamen: 002041-C-EPIS-2022

Visto el borrador del expediente 002041, presentado por:

2010100601 - MEZA LUQUE ANTHONY EDUARDO

Titulado:

**PROPUESTA DE UN SISTEMA DE GESTIÓN DE SÍLABOS EN LA UNIVERSIDAD CATÓLICA DE
SANTA MARÍA**

Nuestro dictamen es:

APROBADO

**1568 - ROSAS PAREDES KARINA
DICTAMINADOR**



**1631 - MONTESINOS MURILLO ANGEL FELIPE
DICTAMINADOR**



**1635 - SULLA TORRES JOSE ALFREDO
DICTAMINADOR**



PRESENTACIÓN

Señor Decano de la Facultad de Ciencias e Ingenierías Físicas y Formales

Señor Director de la Escuela Profesional de Ingeniería de Sistemas

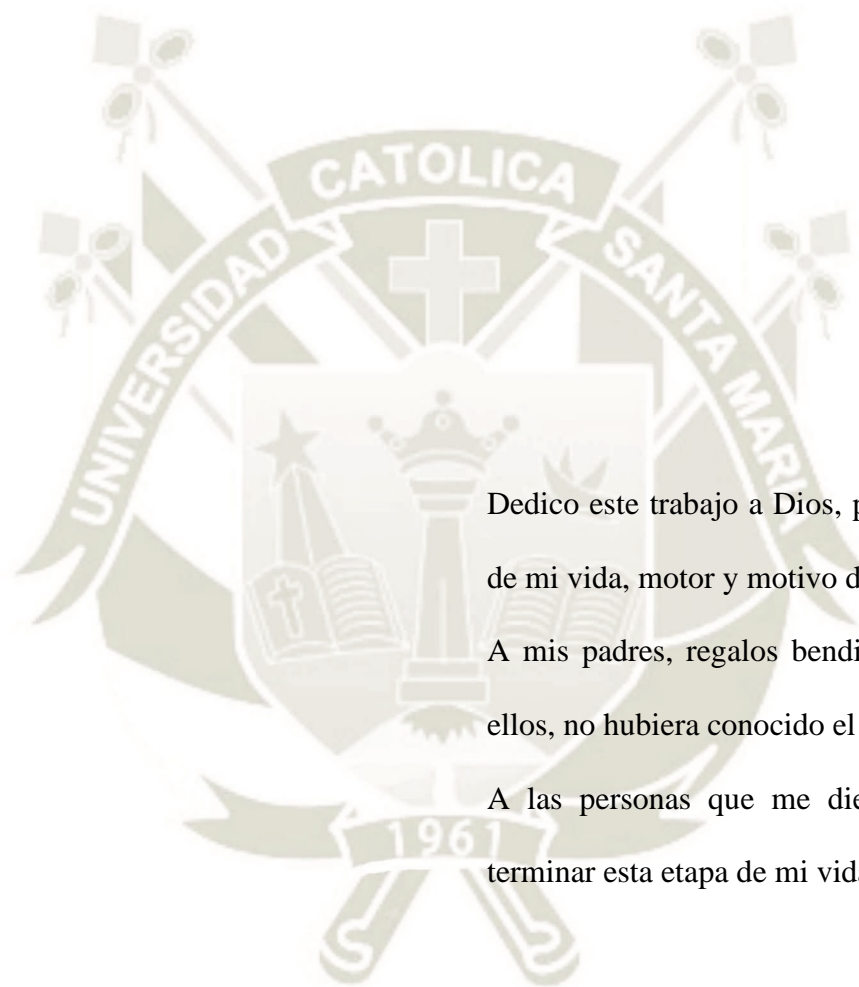
Señores Miembros del Jurado Examinador de Tesis

De conformidad con las disposiciones del Reglamento de Grados y Títulos del Programa Profesional de Ingeniería de Sistemas, pongo a vuestra consideración el presente trabajo de desarrollo de TIC's titulado: **“PROPUESTA DE UN SISTEMA DE GESTIÓN DE SÍLABOS EN LA UNIVERSIDAD CATÓLICA DE SANTA MARÍA”**, el mismo que al ser aprobado me permitirá obtener el Título Profesional de Ingeniero de Sistemas.

Arequipa, junio del 2022

Anthony Eduardo Meza Luque

DEDICATORIA



Dedico este trabajo a Dios, por ser el alfarero
de mi vida, motor y motivo de mi existencia.

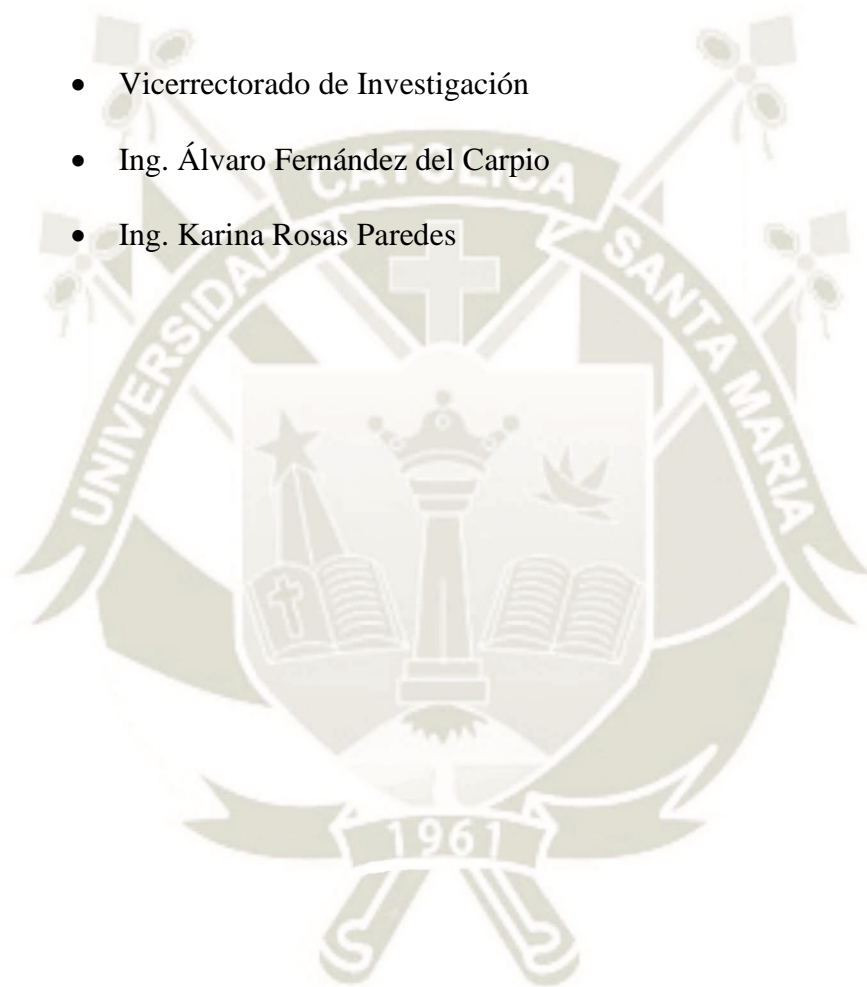
A mis padres, regalos benditos del cielo; sin
ellos, no hubiera conocido el camino correcto.

A las personas que me dieron aliento para
terminar esta etapa de mi vida.

AGRADECIMIENTOS

Agradezco infinitamente a Dios por ayudarme a concluir este proyecto. Así como a los mentores que me ayudaron a culminarlo.

- Vicerrectorado de Investigación
- Ing. Álvaro Fernández del Carpio
- Ing. Karina Rosas Paredes



Resumen

El sílabo es una herramienta que permite planificar, organizar y medir el proceso de enseñanza aprendizaje, que contiene toda la información necesaria sobre la asignatura, es decir la sumilla, competencias, contenidos, mecanismos de evaluación, referencias bibliográficas, entre otros; con el fin de que el estudiante pueda lograr las competencias del perfil de egreso propuestas en las mallas curriculares de las escuelas profesionales de la UCSM definidas en el Modelo Educativo.

Ante la necesidad de mejorar la calidad académica educativa en la Universidad Católica de Santa María frente a la importante tarea de la elaboración de sílabos, el presente trabajo se enfoca en la solución sobre la optimización del proceso de creación de los sílabos para las escuelas profesionales.

Dicho proceso se lleva de manera manual, y está expuesto a errores; la solución planteada se expresa en la creación de un sistema de gestión de sílabos que permita al docente llevar este cometido de una mejor manera, minimizando la existencia de fallas redundantes.

El software se construyó en base a las tecnologías afines usadas en la Universidad Católica de Santa María, basándose en una arquitectura cliente servidor, separando la interfaz (Front-End) de la lógica del negocio (Back-End) utilizando la programación orientada a objetos. Permitiendo la escalabilidad de este al utilizar herramientas open-source. Además, permite la ayuda de supervisión en las etapas de desarrollo, visualizando el avance y actualizando el estado correspondiente del proceso, dando como resultado una mejor elaboración de sílabos.

El trabajo llega a la conclusión de que la solución implementada cumple con las necesidades de la organización, a fin de ser una herramienta de apoyo tanto para docente como para usuarios supervisores en la tarea.

Palabras Clave:

Sistema Gestión de Sílabos, API, Competencias, Software



Abstract

The syllabus is a tool that allows planning, organizing, and measuring the teaching-learning process, which contains all the necessary information about the subject, that is, the summary, skills, content, evaluation mechanisms, bibliographic references, among others; in order that the student can achieve the competencies of the graduate profile proposed in the curricular meshes of the professional schools of the UCSM defined in the Educational Model.

Given the need to improve the educational academic quality at the Catholic University of Santa María in the face of the important task of creating syllables, this paper focuses on the solution on optimizing the process of creating syllables for professional schools.

This process is carried out manually and is exposed to errors; The proposed solution is expressed in the creation of a syllable management system that allows the teacher to carry out this task in a better way, minimizing the existence of redundant failures.

The software was built based on related technologies used at the Catholic University of Santa María, based on a client server architecture, separating the interface (Front-End) from the business logic (Back-End) using object-oriented programming. Allowing the scalability of this when using open-source tools. In addition, it allows the help of supervision in the development stages, visualizing the progress and updating the corresponding state of the process, resulting in a better elaboration of syllables.

The work concludes that the implemented solution meets the needs of the organization, to be a support tool for both teachers and supervisor users in the task.

Keywords:

Syllable Management System, API, Competencies, Software

Índice

PRESENTACIÓN.....	iii
AGRADECIMIENTOS	v
Resumen.....	vi
Abstract	viii
Introducción	1
Capítulo 1: Descripción del proyecto.....	4
1.1 Objetivos	4
1.1.1 Objetivo general.....	4
1.1.2 Objetivos específicos	4
1.2 Alcances y Limitaciones	4
1.2.1 Alcances	4
1.2.2 Limitaciones.....	5
1.3 Fundamentos Teóricos	5
1.3.1 Antecedentes del Proyecto	5
1.3.2 Bases Teóricas del Proyecto.....	8
1.4 Técnicas y Herramientas	15
1.4.1 React.....	15
1.4.2 HTML	16
1.4.3 CSS.....	16
1.4.4 Tailwind CSS	16
1.4.5 PDFMAKE.....	17

1.4.6	Bignumber.....	17
1.4.7	ORM.....	17
1.4.8	Entity Framework Core.....	17
1.4.9	Code First.....	18
1.4.10	LINQ.....	18
1.4.11	JSON.....	19
1.4.12	DTO.....	19
1.4.13	API REST.....	19
1.4.14	Programación orientada a objetos.....	20
1.4.15	Servidor IIS.....	20
1.5	Aspectos Relevantes del Desarrollo.....	20
1.5.1	Arquitectura del Software.....	20
1.5.2	Control de acceso a la aplicación.....	21
1.5.3	Estructura del Silabo.....	22
Capítulo 2: Documentación Técnica.....		29
2.1	Plan del Proyecto Informático.....	29
2.1.1	Planificación temporal del Proyecto.....	29
2.1.2	Estudio de viabilidad del proyecto.....	32
2.2	Especificación De Requisitos Del Proyecto De TIC's.....	36
2.2.1	Historias de Usuario.....	36
2.2.2	Diagrama de casos de Uso.....	45

2.2.3	Requisitos Funcionales.	47
2.2.4	Requisitos No funcionales.	49
2.3	Especificación De Diseño	51
2.3.1	Arquitectura Física	51
2.3.2	Arquitectura Lógica	52
2.3.3	Diseño Estructural	55
2.3.4	Diagrama Dinámico	72
2.3.5	Diseño de Interfaces de Usuarios	96
2.3.6	Diseño de Modelo de Datos	113
2.3.7	Implementación De Funcionalidades Principales	148
2.4	Pruebas de Ejecución	204
2.4.1	Pruebas funcionales.....	204
2.4.2	Pruebas de usabilidad.....	231
2.4.3	Pruebas de accesibilidad	244
2.4.4	Pruebas de seguridad.....	249
2.4.5	Pruebas de Carga.....	252
2.5	Manuales de Usuario	265
CONCLUSIONES		266
RECOMENDACIONES		267
Referencias Bibliográficas		268
APÉNDICE A: MANUAL DE USUARIO		270

1.	RESUMEN DEL PROYECTO.....	271
1.1.	Objetivos.....	271
2.	1.2 Métodos.....	271
3.	PRINCIPALES FUNCIONALIDADES	271
3.1.	Roles de la aplicación	271
3.2.	HU0101 Login de la aplicación	272
3.3.	HU0102 Página principal de la aplicación.....	273
3.4.	HU0401 Pantalla Visualizar Sílabos.....	274
3.5.	HU0402 Pantalla Visualizar Sílabos PDF	275
3.6.	HU0501 Pantalla Habilitar Sílabus.....	276
3.7.	HU0502 Pantalla Habilitar Sílabos.....	277
3.8.	HU0503 Pantalla Habilitar sílabos.....	278
3.9.	HU0201 Pantalla Carga de Archivos	280
3.10.	HU0601 Pantalla Seleccionar Curso.....	281
3.11.	HU0601 Pantalla Silabo del curso	281
3.12.	HU0619 Pantalla de Programa Formativo	297
	APÉNDICE B: ENCUESTAS CON USUARIOS.....	303

Índice de Figuras

Figura 1 Plataforma Scrum	12
Figura 2 <i>Arquitectura Cliente - Servidor</i>	21
Figura 3 <i>Conexión por Azure Active Directory</i>	22
Figura 4 <i>Diseño de Sílabo - Plan de Estudio I</i>	23
Figura 5 <i>Diseño de Sílabo - Plan de Estudio II</i>	24
Figura 6 <i>Diseño de Sílabo - Plan de Estudios III</i>	25
Figura 7 <i>Diseño de Sílabo - Programa Formativo I</i>	26
Figura 8 <i>Diseño de Sílabo - Programa Formativo II</i>	27
Figura 9 <i>Diseño de Sílabo - Programa Formativo III</i>	28
Figura 10 <i>Diseño de Sílabo - Programa Formativo IV</i>	28
Figura 11 <i>Diagrama de Gantt de la Planificación y Estructuración del Proyecto</i>	30
Figura 12 <i>Diagrama de Gantt de Desarrollo de Proyecto de Software</i>	31
Figura 13 <i>Diagrama de Casos de Uso</i>	47
Figura 14 <i>Arquitectura Física</i>	52
Figura 15 <i>Arquitectura Externa</i>	53
Figura 16 <i>Arquitectura Interna</i>	55
Figura 17 <i>Controlador Silabo</i>	62
Figura 18 <i>Diagrama de Clases de Identificación Académica y Sumilla</i>	63
Figura 19 <i>Diagrama de clases de Competencias del Perfil de Egreso</i>	64
Figura 20 <i>Diagrama de clases de Competencias propias de la asignatura</i>	65
Figura 21 <i>Diagrama de clases de los contenidos básicos por unidades de aprendizaje de la asignatura</i>	66
Figura 22 <i>Diagrama de clases de la evaluación de competencias adquiridas</i>	67

Figura 23 <i>Diagrama de Clases Bibliografía</i>	68
Figura 24 <i>Diagrama de clases Bitácora, Transcripción, Firma</i>	69
Figura 25 <i>Diagrama de clases del programa formativo</i>	70
Figura 26 <i>Diagrama de Componentes General</i>	72
Figura 27 <i>Diagrama de Estados - Visualizar Sílabos</i>	75
Figura 28 <i>Diagrama de Estados - Habilitar Sílabos</i>	76
Figura 29 <i>Diagrama de Estados - Desarrollo de Sílabo</i>	77
Figura 30 <i>Diagrama de Estados - Desarrollo de Sílabo Plan de Estudio</i>	78
Figura 31 <i>Diagrama de Estados - Programa Formativo</i>	79
Figura 32 <i>Diagrama de Actividad - Sistema Gestión de Sílabos</i>	81
Figura 33 <i>Diagrama de Secuencia - Visualizar Sílabo</i>	85
Figura 34 <i>Diagrama de Secuencia - Habilitar Sílabo</i>	86
Figura 35 <i>Diagrama de Secuencia - Identificación Académica y Sumilla</i>	87
Figura 36 <i>Diagrama de Secuencia - Competencia por Perfil de Egreso</i>	88
Figura 37 <i>Diagrama de Secuencia - Competencias propias de la asignatura</i>	89
Figura 38 <i>Diagrama de Secuencia - Unidades de Aprendizaje</i>	90
Figura 39 <i>Diagrama de Secuencia - Resultados de Aprendizaje</i>	91
Figura 40 <i>Diagrama de Secuencia - Criterios, Estrategias e Instrumentos</i>	92
Figura 41 <i>Diagrama de Secuencia - Bibliografía</i>	93
Figura 42 <i>Diagrama de Secuencia - Transcripción, Bitácora y Firma</i>	94
Figura 43 <i>Diagrama de Secuencia - Programa Formativo</i>	95
Figura 44 <i>Login Sistema Gestión de Sílabos</i>	96
Figura 45 <i>Login Sistema Gestión de Sílabos API Office 365</i>	97
Figura 46 <i>Interfaz Visualizar Sílabo</i>	98
Figura 47 <i>Interfaz Visualizar Sílabo - Documento PDF</i>	99

Figura 48 <i>Interfaz Habilitar Sílabo</i>	100
Figura 49 <i>Interfaz Cursos</i>	101
Figura 50 <i>Interfaz Sílabo de Asignatura</i>	101
Figura 51 <i>Sílabo de la Asignatura - Identificación Académica</i>	102
Figura 52 <i>Sílabo de la Asignatura - Sumilla</i>	103
Figura 53 <i>Sílabo de la Asignatura - Competencias de la asignatura que apoyan al perfil de egreso</i>	104
Figura 54 <i>Sílabo de la Asignatura - Competencias propias de la asignatura</i>	105
Figura 55 <i>Sílabo de la Asignatura - Unidades teóricas</i>	106
Figura 56 <i>Sílabo de Asignatura - Unidades de Laboratorio</i>	107
Figura 57 <i>Sílabo de Asignatura - Evaluación de competencias adquiridas, títulos de fase</i>	108
Figura 58 <i>Sílabo de Asignatura - Evaluación de competencias adquiridas, resultados de fase</i>	109
Figura 59 <i>Sílabo de Asignatura - Evaluación de competencia adquiridas, Instrumentos de evaluación</i>	110
Figura 60 <i>Sílabo de Asignatura - Evaluación de competencias adquiridas, Criterios</i>	110
Figura 61 <i>Sílabos de Asignatura - Evaluación de competencias adquiridas, estrategias</i>	111
Figura 62 <i>Sílabo de Asignatura - Bibliografía</i>	112
Figura 63 <i>Interfaz Programa Formativo</i>	112
Figura 64 <i>Programa Formativo - Matriz</i>	113
Figura 65 <i>Modelo relacional de Base de Datos SílabosDB</i>	114
Figura 66 <i>Componente para autenticación por Azure</i>	149

Figura 67 <i>Ingreso y redireccionamiento a vista raíz</i>	150
Figura 68 <i>Navegación de Menú</i>	151
Figura 69 <i>Método "generatePDF"</i>	152
Figura 70 <i>Definición de contenido del documento</i>	153
Figura 71 <i>Front - Facultad, Escuela, Identificación Académica</i>	154
Figura 72 <i>Front - Tabla para horas semestrales y semanales</i>	155
Figura 73 <i>Front - Equivalencias, prerrequisitos, competencias propias y que apoyan al perfil</i>	156
Figura 74 <i>Unidades, Evaluación de competencias y Bibliografía</i>	157
Figura 75 <i>Servicio "VerIdentificacionAcademica"</i>	158
Figura 76 <i>Método "ObtenerIdentificacionAcademica" atributos</i>	159
Figura 77 <i>Método "ObtenerIdentificacionAcademica" docentes, prerrequisitos y equivalentes</i>	160
Figura 78 <i>Método "BuscarSumillaPorSilabo"</i>	161
Figura 79 <i>Método "GuardarSumillaEnSilabo"</i>	162
Figura 80 <i>Método "GrabarSumillaSilabo"</i>	163
Figura 81 <i>Método "ListarCompetenciasPerfilEgreso"</i>	165
Figura 82 <i>Método "GuardarAsignaturaCompetenciasPerfilEgreso"</i>	166
Figura 83 <i>Método "ListarCompetenciasPropiasAsignatura"</i>	168
Figura 84 <i>Método "GuardarCompetenciasPropiasDeLaAsignatura"</i>	169
Figura 85 <i>Método "GrabarCompetenciasPAPorDTO"</i>	169
Figura 86 <i>Método "ListarAsignaturaFase"</i>	171
Figura 87 <i>Método "GuardarAsignaturaFase"</i>	172
Figura 88 <i>Método "GrabarAsignaturaFasePorDTO"</i>	172
Figura 89 <i>Método "ListarUnidades"</i>	174

Figura 90 Método " <i>GuardarUnidades</i> "	175
Figura 91 Método " <i>GrabarUnidadesPorDTO</i> "	176
Figura 92 Método " <i>ListarUnidadesLab</i> "	177
Figura 93 Método " <i>GuardarUnidadesLab</i> "	178
Figura 94 Método " <i>GrabarUnidadesLabPorDTO</i> "	179
Figura 95 Método " <i>ListarResultados</i> "	180
Figura 96 Método " <i>GuardarResultados</i> "	181
Figura 97 Método " <i>GrabarResultadosPorDTO</i> " - Parte 1	181
Figura 98 Método " <i>GrabarResultadosPorDTO</i> " - Parte 2	182
Figura 99 Método " <i>GrabarResultadosPorDTO</i> " - Parte 3	184
Figura 100 Método " <i>ListarAsignaturaEstrategia</i> "	185
Figura 101 Método " <i>GuardarAsignaturaEstrategia</i> "	186
Figura 102 Método " <i>GrabarEstrategiasPorDTO</i> "	187
Figura 103 Método " <i>ListarAsignaturaCriterio</i> "	188
Figura 104 Método " <i>GuardarAsignaturaCriterio</i> "	189
Figura 105 Método " <i>GrabarCriteriosPorDTO</i> "	191
Figura 106 Método " <i>ListarAsignaturaInstrumento</i> "	192
Figura 107 Método " <i>GuardarAsignaturaInstrumento</i> "	193
Figura 108 Método " <i>GrabarInstrumentosPorDTO</i> "	194
Figura 109 Método " <i>ListarAsignaturaBibliografía</i> "	195
Figura 110 Método " <i>GuardarAsignaturaBibliografía</i> "	196
Figura 111 Método " <i>GrabarAsignaturaBibliografíaDTO</i> "	197
Figura 112 Método " <i>ListarProyectosAsignatura</i> "	198
Figura 113 Método " <i>GuardarProyectosAsignatura</i> "	199
Figura 114 Método " <i>GrabarProyectosPorDTO</i> " – Parte 1	200

Figura 115 Método "GrabarProyectosPorDTO"	201
Figura 116 Método "TranscribirAsignatura"	202
Figura 117 Método "TranscribirSilabo"	203
Figura 118 Pruebas de usabilidad - Gráfico pregunta 1	232
Figura 119 Pruebas de usabilidad - Gráfico pregunta 2	233
Figura 120 Pruebas de usabilidad - Gráfico pregunta 3	234
Figura 121 Pruebas de usabilidad - Gráfico pregunta 4	235
Figura 122 Pruebas de usabilidad - Gráfico pregunta 5	236
Figura 123 Pruebas de usabilidad - Gráfico pregunta 6	237
Figura 124 Pruebas de usabilidad - Gráfico pregunta 7	238
Figura 125 Pruebas de usabilidad - Gráfico pregunta 8	239
Figura 126 Pruebas de usabilidad - Gráfico pregunta 9	240
Figura 127 Pruebas de usabilidad - Gráfico pregunta 10	241
Figura 128 Pruebas de usabilidad - Gráfico pregunta 11	242
Figura 129 Pruebas de usabilidad - Gráfico pregunta 12	243
Figura 130 Análisis perceptible del sistema usando la herramienta TAW	245
Figura 131 Análisis operable del sistema usando la herramienta TAW	246
Figura 132 Análisis de la comprensibilidad del sistema usando la herramienta TAW	246
Figura 133 Análisis de la robustez del sistema usando la herramienta TAW	247
Figura 134 Gráfico de análisis de resultados	248
Figura 135 Prueba de Seguridad - Ingreso	249
Figura 136 Prueba de Seguridad - Ingreso de correo institucional	250
Figura 137 Prueba de Seguridad - Error en acceso	251
Figura 138 Propiedades de Hilo en herramienta JMeter	252

Figura 139 <i>Resultados de prueba de estrés para Visualizar Sílabo</i>	253
Figura 140 <i>Resultado de hilo para funcionalidad Visualizar Sílabo</i>	254
Figura 141 <i>Tabla resumen de resultados para Visualizar Sílabo</i>	254
Figura 142 <i>Resumen gráfico de resultados de Visualizar Sílabos</i>	255
Figura 143 <i>Resultados de prueba de estrés para Ident. Académ.</i>	256
Figura 144 <i>Resultado de hilo para funcionalidad Ident. Académ.</i>	257
Figura 145 <i>Tabla resumen de resultados para Ident. Académ.</i>	257
Figura 146 <i>Resumen gráfico de resultados de Ident. Académ.</i>	258
Figura 147 <i>Propiedades de Hilo en herramienta JMeter Cursos Docente</i>	259
Figura 148 <i>Resultado de hilo para funcionalidad Cursos Docente</i>	260
Figura 149 <i>Tabla resumen de resultados para Cursos Docente</i>	260
Figura 150 <i>Resumen gráfico de resultados de Cursos Docente</i>	261
Figura 151 <i>Propiedades de Hilo en herramienta JMeter Prog. Formativo</i>	262
Figura 152 <i>Resultado de hilo para funcionalidad Programa Formativo</i>	263
Figura 153 <i>Tabla resumen de resultados para Programa Formativo</i>	263
Figura 154 <i>Resumen gráfico de resultados de Programa Formativo</i>	264

Índice de Tablas

Tabla 1 <i>Actividades de Planificación y Estructuración del proyecto</i>	29
Tabla 2 <i>Desarrollo de proyecto de software GSilabos</i>	31
Tabla 3 <i>Requisitos de Ambiente</i>	33
Tabla 4 <i>Costes Totales</i>	34
Tabla 5 <i>Beneficios Totales</i>	34
Tabla 6 <i>Costes vs Beneficios</i>	35
Tabla 7 <i>Historia de Usuario 1 – Habilitar semestre académico</i>	37
Tabla 8 <i>Historia de Usuario 2 – Ejecutar carga académica</i>	37
Tabla 9 <i>Historia de Usuario 3 – Visualizar el desarrollo de sílabos</i>	38
Tabla 10 <i>Historia de Usuario 4 – Habilitar o cambiar estado de sílabo</i>	38
Tabla 11 <i>Historia de Usuario 5 – Observar lista de cursos asignados</i>	39
Tabla 12 <i>Historia de Usuario 6 – Transcribir sílabo del año pasado</i>	39
Tabla 13 <i>Historia de Usuario 7 – Visualizar sílabo PDF</i>	40
Tabla 14 <i>Historia de Usuario 8 – Visualizar Identificación Académica y Desarrollo de Sumilla</i>	40
Tabla 15 <i>Historia de Usuario 9 – Seleccionar competencias por perfil de egreso</i>	41
Tabla 16 <i>Historia de Usuario 10 – Construir competencias propias de asignatura</i> ..	41
Tabla 17 <i>Historia de Usuario 11 – Desarrollo Unid. de aprendizaje teórica y de laboratorio</i>	42
Tabla 18 <i>Historia de Usuario 12 – Ingresar resultados de aprendizaje por fase</i>	42
Tabla 19 <i>Historia de Usuario 13 – Registrar nombre de fases</i>	43
Tabla 20 <i>Historia de Usuario 14 – Desarrollar criterios, estrategias e instrumentos</i>	43

Tabla 21 <i>Historia de Usuario 15 – Ingresar actividades de investigación y responsabilidad social</i>	44
Tabla 22 <i>Historia de Usuario 16 – Seleccionar Bibliografía</i>	44
Tabla 23 <i>Historia de Usuario 17 – Visualizar Bitácora</i>	45
Tabla 24 <i>Requerimientos Funcionales</i>	48
Tabla 25 <i>Requerimientos no Funcionales</i>	50
Tabla 26 <i>Diccionario de datos de la tabla «Asignatura»</i>	115
Tabla 27 <i>Diccionario de datos de la tabla «AsignaturaBibliografía»</i>	117
Tabla 28 <i>Diccionario de datos de la tabla «AsignaturaCompetenciaPA»</i>	118
Tabla 29 <i>Diccionario de datos de la tabla «AsignaturaCompetenciaPE»</i>	119
Tabla 30 <i>Diccionario de datos de la tabla «AsignaturaCriterio»</i>	120
Tabla 31 <i>Diccionario de datos de la tabla «AsignaturaDocenteFuncion»</i>	121
Tabla 32 <i>Diccionario de datos de la tabla «AsignaturaEstrategia»</i>	122
Tabla 33 <i>Diccionario de datos de la tabla «AsignaturaFase»</i>	123
Tabla 34 <i>Diccionario de datos de la tabla «AsignaturaInstrumento»</i>	124
Tabla 35 <i>Diccionario de datos de la tabla «AsignaturaLog»</i>	125
Tabla 36 <i>Diccionario de datos de la tabla «Bibliografía»</i>	125
Tabla 37 <i>Diccionario de datos de la tabla «CompetenciaPerfilEgreso»</i>	127
Tabla 38 <i>Diccionario de datos de la tabla «Criterio»</i>	128
Tabla 39 <i>Diccionario de datos de la tabla «Departamento»</i>	129
Tabla 40 <i>Diccionario de datos de la tabla «Docente»</i>	130
Tabla 41 <i>Diccionario de datos de la tabla «DocenteFuncion»</i>	131
Tabla 42 <i>Diccionario de datos de la tabla «Equivalente»</i>	132
Tabla 43 <i>Diccionario de datos de la tabla «Escuela»</i>	133
Tabla 44 <i>Diccionario de datos de la tabla «Estado»</i>	134

Tabla 45 <i>Diccionario de datos de la tabla «Estrategia»</i>	135
Tabla 46 <i>Diccionario de datos de la tabla «Facultad»</i>	136
Tabla 47 <i>Diccionario de datos de la tabla «Fase»</i>	137
Tabla 48 <i>Diccionario de datos de la tabla «Funcion»</i>	138
Tabla 49 <i>Diccionario de datos de la tabla «Instrumento»</i>	139
Tabla 50 <i>Diccionario de datos de la tabla «Malla»</i>	140
Tabla 51 <i>Diccionario de datos de la tabla «Parametrica»</i>	141
Tabla 52 <i>Diccionario de datos de la tabla «PreRequisito»</i>	142
Tabla 53 <i>Diccionario de datos de la tabla «Proyecto»</i>	143
Tabla 54 <i>Diccionario de datos de la tabla «Resultado»</i>	144
Tabla 55 <i>Diccionario de datos de la tabla «Sumilla»</i>	145
Tabla 56 <i>Diccionario de datos de la tabla «Unidad»</i>	146
Tabla 57 <i>Diccionario de datos de la tabla «UnidadLab»</i>	147
Tabla 58 <i>Desarrollo de pruebas de aplicación</i>	204
Tabla 59 <i>Caso de Prueba CPGS01</i>	205
Tabla 60 <i>Caso de Prueba CPGS02</i>	206
Tabla 61 <i>Caso de Prueba CPGS03</i>	207
Tabla 62 <i>Caso de Prueba CPGS04</i>	208
Tabla 63 <i>Caso de Prueba CPGS05</i>	209
Tabla 64 <i>Caso de Prueba CPGS06</i>	210
Tabla 65 <i>Caso de Prueba CPGS07</i>	211
Tabla 66 <i>Caso de Prueba CPGS08</i>	212
Tabla 67 <i>Caso de Prueba CPGS09</i>	213
Tabla 68 <i>Caso de Prueba CPGS10</i>	214
Tabla 69 <i>Caso de Prueba CPGS11</i>	215

Tabla 70 <i>Caso de Prueba CPGS12</i>	216
Tabla 71 <i>Caso de Prueba CPGS13</i>	217
Tabla 72 <i>Caso de Prueba CPGS14</i>	218
Tabla 73 <i>Caso de Prueba CPGS15</i>	219
Tabla 74 <i>Caso de Prueba CPGS16</i>	220
Tabla 75 <i>Caso de Prueba CPGS17</i>	221
Tabla 76 <i>Caso de Prueba CPGS18</i>	222
Tabla 77 <i>Caso de Prueba CPGS19</i>	223
Tabla 78 <i>Caso de Prueba CPGS20</i>	224
Tabla 79 <i>Caso de Prueba CPGS21</i>	225
Tabla 80 <i>Caso de Prueba CPGS22</i>	226
Tabla 81 <i>Caso de Prueba CPGS23</i>	227
Tabla 82 <i>Caso de Prueba CPGS24</i>	228
Tabla 83 <i>Caso de Prueba CPGS25</i>	229
Tabla 84 <i>Caso de Prueba CPGS26</i>	230
Tabla 85 <i>Pruebas de usabilidad - Respuestas Pregunta 1</i>	231
Tabla 87 <i>Pruebas de usabilidad - Respuestas Pregunta 2</i>	233
Tabla 88 <i>Pruebas de usabilidad - Respuestas Pregunta 3</i>	234
Tabla 89 <i>Pruebas de usabilidad - Respuestas Pregunta 4</i>	235
Tabla 90 <i>Pruebas de usabilidad - Respuestas Pregunta 5</i>	236
Tabla 91 <i>Pruebas de usabilidad - Respuestas Pregunta 6</i>	237
Tabla 92 <i>Pruebas de usabilidad - Respuestas Pregunta 7</i>	238
Tabla 93 <i>Pruebas de usabilidad - Respuestas Pregunta 8</i>	239
Tabla 94 <i>Pruebas de usabilidad - Respuestas Pregunta 9</i>	240
Tabla 95 <i>Pruebas de usabilidad - Respuestas Pregunta 10</i>	241

Tabla 96 *Pruebas de usabilidad - Respuestas Pregunta 11*242

Tabla 97 *Pruebas de usabilidad - Respuestas Pregunta 12*243



Introducción

La presente tesis: “Propuesta de un sistema de gestión de sílabos en la Universidad Católica de Santa María” es un proyecto de Tecnologías de Información y Comunicación (TIC) que tiene por objetivo el desarrollo de una aplicación web que permita la correcta gestión de los sílabos apoyado en herramientas de software para el desarrollo web y un sistema gestor de bases de datos integrado a la base de datos de la Universidad Católica de Santa María.

En la Universidad Católica Santa María el proceso de elaboración de sílabos se desarrolla de manera manual. Las Escuelas Profesionales son provistas de la información e inmediatamente proceden a la comunicación de las Asignaturas designadas a los docentes, de manera que el flujo inicia con un informe por correo con la lista de asignaturas a desarrollar al finalizar el semestre vigente a cada docente. El periodo de vacaciones sirve a su vez como el tiempo fijado para la elaboración del sílabo. Cada docente maneja de manera independiente la plantilla provista por la Oficina Académica.

Al culminar el desarrollo del sílabo, el documento final mantiene los elementos esenciales del formato definido por el centro de desarrollo académico, pero también contiene detalles que el docente agrega; observando así una diversidad en el resultado final.

Aproximadamente un mes antes del inicio del semestre impar o dos semanas antes del inicio del semestre par, se habilita la carga de los sílabos en el repositorio virtual que el centro de desarrollo académico maneja. Este repositorio solo tiene como fin almacenar las versiones finales de sílabos de cada semestre. El tiempo de finalización de carga culmina pasadas dos semanas del inicio de las labores universitarias. Los directores de cada programa profesional se encargan de supervisar que todos los docentes que están dictando alguna asignatura hayan

cumplido la carga del sílabo en su versión final; de no ser así, se envía recordatorios para realizar la tarea de manera pronta.

Cargada la información, el Centro de Desarrollo Académico apoyada por el jefe de departamento revisan de manera general lo ingresado en los sílabos. Las observaciones se transmiten al docente para poder subsanarlas. El trabajo no puede ser muy exhaustivo debido a la cantidad de asignaturas manejadas por escuela, la revisión no toca mucho contenido, solo información referente a: año académico, fechas de fases inicio y fin, créditos, nombre del curso, nombre del docente, código del curso.

El contenido puede considerar falencias en la elaboración de las competencias, resultados y demás conceptos que son parte del enfoque de Competencias que la universidad ha adoptado. Se requiere de la experiencia adquirida por el tiempo para perfeccionar el ingreso correcto de las secciones del sílabo. Para docentes que se incorporan recientemente a la labor universitaria es complicado adoptar este enfoque debido al desconocimiento de su realización, a menos que ya tengan pericia por trabajar en otras universidades.

Periódicamente se dictan cursos de didáctica educativa para que los docentes estén actualizados en la forma de enseñanza manejada por la universidad sobre: Instrumentos, elaboración de sílabos, etc. El docente nuevo recibía un libro-manual para entender los conceptos y además que está basado en sílabos de años anteriores para elaborar el suyo. Pero no garantizaba que después de la supervisión de los sílabos por parte de la comisión el contenido impreso en el sílabo tenga una correcta formulación.

Las acreditaciones en diversas escuelas profesionales ocasionaron que se genere un esfuerzo extra en cada aspecto del ejercicio de la docencia y por ende en la elaboración de los sílabos.

Dado que el sílabo es la herramienta base de planificación para la formación de los estudiantes universitarios, el proceso no puede seguir siendo manual en todas sus instancias. Debe recurrirse a automatizar aspectos dentro del proceso de la elaboración del sílabo que no requieran la intervención necesaria del docente.



Capítulo 1: Descripción del proyecto

1.1 Objetivos

1.1.1 *Objetivo general.*

Implementar una propuesta de sistema de Gestión de Sílabos en la Escuela Profesional de Ingeniería de Sistemas de la Universidad Católica de Santa María.

1.1.2 *Objetivos específicos*

- Diseñar un sistema que facilite la elaboración y la gestión de los sílabos de manera online.
- Implementar la propuesta del sistema de Gestión de Sílabo a través de las tecnologías libres de REACT y .Net Core.
- Realizar la validación del sistema piloto en la Escuela Profesional de Ingeniería de Sistemas de la Universidad Católica de Santa María.
- Mejorar la experiencia del docente en el uso y gestión de sílabos académicos.

1.2 Alcances y Limitaciones

1.2.1 *Alcances*

- El presente sistema se encuentra definido según la gestión de sílabos definido por el centro de desarrollo académico de la universidad Católica de Santa María.
- Está implementado en la plataforma utilizada por la Universidad Católica de Santa María, que usa un servidor IIS, framework .net 4.8 y base de datos SQL Server 2018.
- Se considera a todos los programas profesionales y sus docentes de la Universidad Católica de Santa María como los usuarios finales a usar el sistema de Gestión de Sílabo, iniciando el piloto en la Escuela profesional de Ingeniería de Sistemas.
- El piloto satisface las necesidades esenciales de la gestión de sílabos y aspira a ser la base para un sofisticado sistema de gestión de sílabos.

1.2.2 Limitaciones

- Cualquier valor agregado al sistema debe coexistir con el diseño original planteado e integrarse.
- Se considera el alojamiento de la base de datos en la Universidad Católica de Santa María.

1.3 Fundamentos Teóricos

1.3.1 Antecedentes del Proyecto

Las propuestas presentadas en las últimas décadas muestran el interés de plantear diversos enfoques, alternativas y herramientas que apoyados por la tecnología ayuden a la construcción de programas de estudio eficientes.

El portal desarrollado por Gluga, Kay & Lever (2012) llamado CUSP, permite a través de un mapeo semántico liviano, brindar una forma flexible y escalable de relacionar los objetivos de aprendizaje establecidos por la institución educativa, el programa formativo que será diseñado por el docente y a su vez presentar esta información a los estudiantes que ayudarán con la retroalimentación de si la información es clara.

Así mismo, el modelo jerárquico abstracto para la gestión de sílabos basada en Arquitectura dirigida por Gerbé, Raynauld & Camarero (2010), especificando un conjunto de atributos como niveles, horas, créditos, etc. Genera varios tipos de informes, como para acreditación y propósito general. El modelo está vinculado al sistema Open Syllabus, un sistema para crear, editar y publicar programas de estudio.

Por otro lado, Molontay, R., Horváth, N., Bergmann, J., Szekrényes, D. & Szabó, M. (2020) a través de una red de requisitos secuencial que va de la mano con el programa STEM, dónde los nodos son las asignaturas entrelazadas, siendo un nodo previo, requisito para un nodo subsecuente. Además de implementar un análisis probabilístico que indica el flujo de estudiantes, el tiempo determinado a culminar la materia, la deserción de estos y la tasa de

finalización de los cursos. Se apoya también en modelos matemáticos para medir el impacto relativo en tiempo de graduación de cada curso.

Del mismo modo, Kawaba, T., Tsuchiya, T., & Koyanagi, K. (2012) desarrolló un modelo de servicio para el programa de estudios para estandarizar el método de clasificación del programa de estudios de artículos y permiten la colaboración entre entornos. Para hacer esto, modelaron el esquema de sílabas para facilitar la búsqueda entre diferentes dominios, desarrolló la generación automática de la sílaba basada en un lenguaje de marcado e implementado agentes de búsqueda multicapa. Además, el modelo incluye datos de funciones de creación, almacenamiento y recuperación.

También, Ota, S. & Mima, Hideki (2011), desarrollaron un sistema semiautomático que permite la clasificación de planes de estudios basados en machine learning para la elaboración de sílabos interdisciplinarios orientados a temas, así buscan apoyar a los estudiantes a encontrar sílabos sobre temas que se adapten a su necesidad. Para esta tarea, utilizaron un reconocimiento de términos propios de la asignatura más sustantivos compuestos o morfemas para la búsqueda de los temas, además del aprendizaje de patrones por interacción con los alumnos. Ambas fuentes ayudan a las máquinas de vectores de soportes a adquirir el conocimiento para clasificar con mayor precisión los temas y a su vez para que la búsqueda sea cada vez más precisa.

En sí, las tecnologías de la World Wide Web han transformado el diseño, desarrollo, implementación y despliegue de sistemas de soporte de decisiones. La literatura académica basada en la web del DSS se centra en aplicaciones e implementaciones, y solo unos pocos artículos examinan cuestiones arquitectónicas o proporcionan pautas de diseño basadas en evidencia empírica como lo mencionan Bhargava, H. K., Power, D. J., & Sun, D. (2007). Para el desarrollo de propuestas tecnológicas, como es el caso de la presente investigación, el enfoque de la arquitectura de software es importante, ya que mejora la accesibilidad y la usabilidad de los productos de software en beneficio de los sistemas de educación avanzada de

alta calidad en todo el mundo (Morales, G. R. & Benedí, J. P., 2017). También permite mejorar las experiencias de aprendizaje a través de servicios colaborativos que son conscientes del contexto: arquitectura de software y sistema prototipo (Dimakis, N., Polymenakos, L., & Soldatos, J., 2006).

Los cursos de los programas de estudio de archivo en América del Norte fueron sometidos a un análisis, el cual produjo una identificación de las convergencias y divergencias de los temas, buscaron comprender la relación entre los dos cursos y obtener información sobre cómo estos cursos continúan sirviendo como un componente integral de la educación en estudios de archivo. La investigación examinó tres aspectos diferentes del programa de estudios: libros de texto, artículos requeridos y temas semanales. Los programas se analizaron como conjuntos de datos separados (programas de RM y programas de ERM), que fue seguido por un análisis comparativo de los dos tipos de programas de estudio (Force, D. C. & Zhang, J., 2016). Esto puede permitir el diseño del plan de estudios de gestión del conocimiento (Yucel, R., 2012).

Esta gestión de conocimiento se puso en práctica en la Universidad Multimedia de Malasia al identificar requisitos del M.Q.A. que es el ente de acreditación, el compromiso de los académicos y los requisitos de la universidad como factores que contribuyen a la gestión del conocimiento entre los académicos. Esto fue plasmado en un prototipo de sistema de gestión de plan de estudios que ayuda a crear y gestionar la práctica de gestión de conocimiento en el programa de tecnologías de información, todo desarrollado en la herramienta libre Protége 2000. Esta herramienta se basa en la creación, consulta, actualización y eliminación de ontologías (Chelvarayan, A., Jayothisa, C., Hashim, H. & Nizat, K., 2011).

También se indica que la arquitectura de software como producto de desarrollo es útil para actividades técnicas, como describir las opiniones y preocupaciones de futuros productos de software, así como para actividades de gestión, incluida la asignación de tareas a cada equipo

y como entrada para actividades de gestión de proyectos. Un problema importante al describir la arquitectura del software es saber qué elementos deben incluirse en la arquitectura y con qué nivel de detalle (Júnior, A. A. C., Misra, S., & Soares, M. S., 2019), hacia una Arquitectura de referencia (Vidoni, M. & Vecchiatti, A., 2016).

Por esta razón, el desarrollo de herramientas tecnológicas orientadas a una arquitectura para el aprendizaje y gestión de aprendizaje se vuelve de primera necesidad en centros académicos superiores, es por ellos que la Universidad Católica de Santa María plasma esta realidad como característica que un docente debe dominar, dentro de la gestión del conocimiento, el uso de las TIC's; y no solamente el uso, si no también contar con un entorno preparado para la generación de conocimiento y tecnología original (Martines, H., 2013).

1.3.2 Bases Teóricas del Proyecto

El proyecto se construye sobre conceptos de educación superior, gestión y de tecnología para poder ser colaborador en la misión de la universidad y constituir un apoyo tecnológico en el camino. A continuación, se detallan:

- **Enfoque basado en competencias:** La UCSM adoptó el enfoque por competencias desarrollado por la UNESCO para su proceso de desarrollo académico, esta está definida de la siguiente manera como: El desarrollo de las capacidades complejas que permiten a los estudiantes pensar y actuar en diversos ámbitos. Consiste en la adquisición de conocimiento a través de la acción, resultado de una cultura de base sólida que puede ponerse en práctica y utilizarse para explicar qué es lo que está sucediendo (Braslavsky, Cecilia citada en la UNESCO, 2007).

El sistema trabaja con el formato definido por la universidad basado en el enfoque por competencias. Permite de esa manera que cada sección conocida pueda ser desarrollada por el docente según este enfoque.

- **SÍLABO:** Herramienta o instrumento de aprendizaje que permite la planificación general del ciclo de estudio que presentará el docente a los estudiantes de la asignatura que se cursará en el semestre académico. Este documento cumple una función pedagógica, de comunicación y de gestión en la labor del docente.

Dentro de la Universidad Católica de Santa María, el sílabo tiene dos grandes divisiones, el plan de estudio y el programa formativo. Se detalla el contenido de ambos:

A) Plan de Estudio:

Planifica los puntos, métodos, estrategias y resultados que se desarrollarán a lo largo de la asignatura, sus partes son:

- 1. Identificación Académica:** contiene los datos básicos de la asignatura como:
 - 1.1. Nombre de asignatura, Código de la asignatura y semestre académico en que se desarrolla.
 - 1.2. Peso académico de la asignatura: Se detallan cuántos créditos vale la asignatura y las horas que se utilizarán para impartir el curso. Dentro de las horas se divide en semanales y semestrales.
 - 1.3. Código, nombre y créditos de las asignaturas equivalentes.
 - 1.4. Código y nombre de las asignaturas prerequisites.
- 2. Sumilla:** Es desarrollada por el o los docentes que estén a cargo de la asignatura.
- 3. Competencias del Perfil de Egreso a las que contribuye la asignatura:** Competencias definidas por el Centro de Desarrollo Académico que están destinadas para determinados grupos de facultades y de las cuales seleccionarán los docentes para implementar en la asignatura.

4. **Competencias propias de la asignatura:** Competencias desarrolladas por el mismo docente como objetivos para cada fase de aprendizaje.
5. **Contenidos Básicos por unidades de aprendizaje:** Unidades y subunidades teóricas que se dictarán en las sesiones a lo largo del semestre por cada fase. También se implementan las unidades de laboratorio que indican el ejercicio práctico de la enseñanza.
6. **Evaluación de competencias adquiridas:** Se detallan los resultados de aprendizaje para las evidencias de conocimiento, desempeño y/o producto. También detalla los instrumentos de evaluación que se usarán en la asignatura.
7. **Bibliografía:** Referencia que utiliza el docente para la elaboración del contenido.

B) Programa Formativo de Asignatura:

Indica la sistematización y estructuración del contenido desarrollado en el plan de estudio en cada fase de aprendizaje, detallando de forma precisa los elementos que se usarán para la asignatura.

I. Identificación Académica

1. Facultad: Facultad al que pertenece la asignatura
2. Departamento Académico: Departamento al que pertenece la asignatura.
3. Nombre de la asignatura y código
4. Escuela profesional donde se desarrolla la asignatura
5. Docentes y jefes de práctica
6. Ubicación y Peso Académico de la asignatura
7. Ambiente donde se realiza el aprendizaje

II. Lineamiento Académico Profesional

1. **Sumilla:** Es desarrollada por el o los docentes que estén a cargo de la asignatura.
2. **Competencias del perfil de Egreso a las que contribuye la asignatura:** Competencias definidas por el Centro de Desarrollo Académico que están destinadas para determinados grupos de facultades y de las cuales seleccionarán los docentes para implementar en la asignatura.
3. **Competencias propias de la asignatura:** Competencias desarrolladas por el mismo docente como objetivos para cada fase de aprendizaje.

III. Programación por fase de aprendizaje: Estructura con el detalle de ejecución de competencias, resultados de aprendizaje, temas de la fase, estrategias de aprendizaje – enseñanza, evaluación de aprendizaje (criterios, instrumentos de evaluación y ponderación teórico - práctico) y bibliografía por fase de aprendizaje y la especificación de su duración.

IV. Programación de actividades de investigación y responsabilidad

social: Especificación de proyectos de investigación o responsabilidad social que se llevarán a lo largo del semestre.

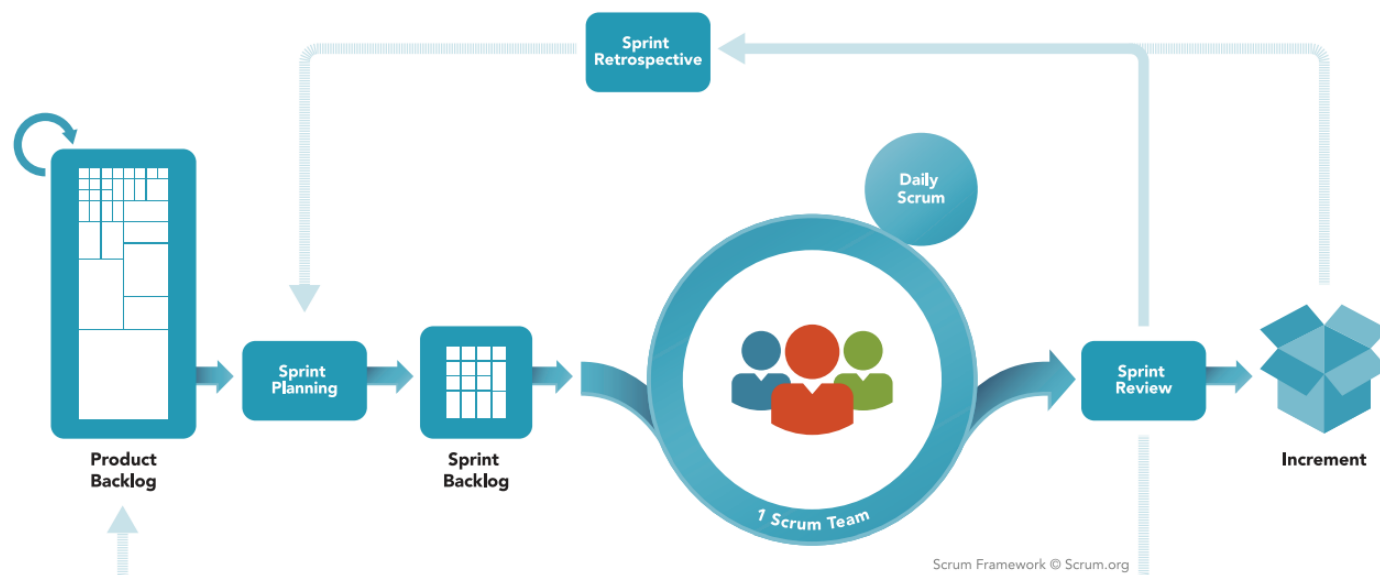
- **ISO/IEC/IEEE 42010:2011:** Estándar que permite establecer las descripciones de la arquitectura, los marcos que la arquitectura optará al desarrollarse y la descripción de las arquitecturas implementadas, que significa su ciclo de vida y los procesos que la conforman. La arquitectura del sistema requiere, mantener las directrices del enfoque previamente explicado y debe estar soportado en la arquitectura que la UCSM tiene implementada.

- **El marco de Trabajo SCRUM:** Es una metodología de desarrollo ágil y está caracterizado por:
 - Equipos que se autogestionan y que son autónomos, el conocimiento que producen es compartido de una manera abierta y aspiran a un aprendizaje en conjunto.
 - Apunta a una estrategia de desarrollo incremental, en vez de una planificación rígida y completa del objeto o producto.
 - Por medio del conocimiento personal y tácito de los integrantes del grupo, generar un resultado de calidad, apoyado en la creatividad y no en procesos secuenciales.
 - Utilizar un solapamiento de las fases de desarrollo, y dejar de lado el ciclo en cascada.

Tiene como proceso de elaboración lo descrito en la Figura 1.

Figura 1 Plataforma Scrum

SCRUM FRAMEWORK



Nota: Plataforma Scrum, por Scrum.Org, 2020, ([https://www.scrum.org/resources/what-](https://www.scrum.org/resources/what-is-scrum)

is-scrum)

Dentro de Scrum se manejan roles, artefactos y eventos, se detalla a continuación:

Roles: Para Scrum se definen los siguientes roles.

- **Product Owner (Propietario del Producto):** Indica cuales serían las prioridades y es una única persona.
- **Software Developers (Desarrollador):** Constructor del producto.
- **Scrum Master:** Realiza la gestión y ejecución de las reglas de Scrum
- **Stakeholders (Interesados):** Resto de personas que están implicadas, pueden realizar una tarea de asesoramiento o solo observar el desarrollo.

Artefactos: Para Scrum se definen los siguientes Artefactos:

- **Product BackLog (Pila del Producto):** El Product Owner define un documento donde tiene la lista completa de las necesidades del cliente y expresa en él las ideas, requisitos y funcionalidades que darán cumplimiento a lo solicitado.
- **Sprint BackLog (Pila del Sprint):** lista de funcionalidades. Son elaborados a partir del Product BackLog
- **Product (Producto o incremento):** Producto funcional donde el cliente pueda interactuar y revisar el avance del proyecto.

Eventos: Para Scrum se definen los siguientes eventos:

- **Sprint Planning Meeting (Planificación del Sprint):** Es una reunión de primera fase donde se tratará de dar solución al documento con los requisitos del cliente y definir que tareas se trabajarán en los sprints.
- **Sprint:** Proceso de desarrollo o construcción de la necesidad del cliente. Contiene un tiempo definido entre 1 y 4 semanas donde se dará solución a la funcionalidad deseada descrita en el Sprint Backlog.

- **Daily Scrum (Scrum diario):** Reuniones diarias que tienen como objetivo la supervisión y seguimiento. Se plantean preguntas como: ¿Qué se hizo ayer? ¿Qué se hará hoy? ¿Qué se hará mañana? ¿Qué problemas se encontraron? Esto se extiende a todos los integrantes del Team Developer. Y su duración no debe ser mayor a 15 minutos.
- **Sprint Review (Revisión del Sprint):** Reunión que abarca a los tres roles mencionados para revisar los objetivos de los sprints en cuestión, y para que no se vea comprometido el desarrollo del sprint.
- **Sprint Retrospective (Retrospectiva del Sprint):** reunión que busca analizar cuáles fueron los resultados del sprint anterior para encontrar alguna problemática, falencias o mejoras que puedan agilizar al siguiente sprint.
- **Sistemas de Información:**
Conjunto formal de procesos que operando sobre una colección datos estructurada según la entidad que lo maneje, recopila información para hacer uso en los diferentes niveles de la entidad.

Los componentes de un sistema de información son:

- **Recursos Físicos:** Como por ejemplo archivos, documentos, ordenadores, etc
- **Reglas:** Normas establecidas para transferencias de información entre los componentes
- **Personal:** Recursos humanos que utilizan la información

Las características de un sistema de información es que se organiza generalmente en una estructura piramidal y que se dividen en los niveles:

- **Transaccional:** Procesamiento de las actividades diarias u operaciones que afectan a la empresa.
- **Operativo:** Es el que se ocupa de analizar el resultado del nivel transaccional
- **Tácito:** Se encarga de gestionar los recursos de la empresa de manera efectiva a medio plazo
- **Estratégico:** Es el encargado de trabajar y analizar cuál es el futuro de la empresa a largo plazo.

Existe un flujo de datos ascendente y descendente de la información entre los diferentes niveles.

Los tipos de sistemas de información:

- **Sistemas informáticos**
 - **Sistemas Informáticos transaccionales:** Automatización de las operaciones y transacciones que se llevan en la entidad.
 - **Sistemas informáticos de gestión:** Se ocupan de los datos que se manejan en una empresa y del almacenamiento de estos.
- **Sistemas de ayuda a la decisión:** Ayudan al directivo y al personal con responsabilidad a tomar decisiones estratégicas.

1.4 Técnicas y Herramientas

1.4.1 React

Librería Open Source de JavaScript creada por un equipo de Facebook para la creación de interfaces de usuario de una sola página o S.P.A. Nace para solucionar problemas de rendimiento utilizando un D.O.M virtual en memoria para que solo se enfoque en aquellos componentes que tuvieron un cambio en el estado, de tal modo que los algoritmos internos de

REACT al reconocer el cambio del estado realizan el cambio en el D.O.M real. La solución no solo incrementó eficiencia de la renderización, si no que permitió un nuevo enfoque de desarrollo para la vista basado en componentes que encapsulan la presentación y la lógica. Actualmente es mantenida por Facebook y por la comunidad de software libre.

1.4.2 HTML

Lenguaje de marcado de hipertexto, tecnología creada para poder construir y estructurar la información que se aprecia en las páginas web. Su desarrollo se compone a través de etiquetas que se escriben de manera lógica para colocar un contenido. Además, se utiliza para publicar, enlazar y compartir información a través de sitios web que vinculan a sus diferentes páginas, las cuales se presentan a los visitantes del sitio web. Esta tecnología ha ido evolucionando desde la década de los 90' hasta la actualidad, su más reciente versión es la número 5.

1.4.3 CSS

Cascading Style Sheets, u Hojas de estilo en cascada; es un lenguaje de diseño visual que se utiliza para darle estilo a las páginas web. Su creación revolucionó la forma en como presentar la interfaz gráfica web, en donde el contenido estructurado en la página web es completamente independiente de su presentación, resultando en una optimización de la construcción de la parte visual al concentrar las propiedades de estilos en un documento que puede ser reutilizado. Esta tecnología está en su versión 3 hoy.

1.4.4 Tailwind CSS

Framework para CSS, que contiene las herramientas para construir los componentes necesarios para desarrollar la interfaz web. Está compuesto por diferentes clases ya desarrolladas para poder dar forma y estilo al proyecto. Este Framework está orientado a utilidades, que significa que no contiene los componentes ya desarrollados, si no que permite

la personalización de estos a través de clases específicas. Esto permite un desarrollo más ágil de la interfaz.

1.4.5 PDFMAKE

Librería que permite la creación de PDF a través de código JavaScript. Su implementación se puede realizar en el lado del cliente, construcción en memoria; o en el lado del servidor, PDF generado solamente para descargar. A la construcción del PDF se le puede adherir tablas, imágenes, textos, etc.; Además de la personalización de la fuente.

1.4.6 BigNumber

Librería para JavaScript que permite realizar operaciones aritméticas de precisión arbitraria. Al recurrir a esta librería se pueden aprovechar métodos para todas las operaciones aritméticas necesarias como suma, resta, multiplicación, división, potencia, etc.

1.4.7 ORM

Object-relational mapping o mapeo de objeto relacional es una técnica que permite convertir los tipos de datos de un lenguaje de programación a objetos a los tipos de datos de un gestor de bases de datos relacional, esto significa que una fila de una tabla de la base de datos en un objeto en el lenguaje de programación y las respectivas columnas se convierten en las propiedades de dicho objeto. A su vez reciben el nombre de ORM los frameworks que utilizan esta técnica. Estos utilizan una capa de abstracción entre el desarrollador y la base de datos, debido a que el mapeo de los objetos permite utilizar las funciones de insertar, actualizar, modificar o eliminar la información por medio de los métodos que el framework maneje.

1.4.8 Entity Framework Core

Es un ORM creado por Microsoft que facilita el trabajo del mapeo de las entidades permite la implementación de la base de datos usando LINQ para entity, permite mantener el diseño de la base de datos permitiendo que la aplicación sea mantenible y extensible. Entity

Framework tiene varios enfoques como Code First, Model First y DataBase First, y contiene un conjunto de tecnologías de ADO.NET que admite el desarrollo de aplicaciones de software orientadas a datos y permite la solución de problemas empresariales, además que resuelve y automatiza la operación estándar de CRUD (creación, lectura, actualización y eliminación) de la información sin que se realice una escritura explícita de las sentencias. Permite la creación de la base de datos a través de metadatos que es un grupo de datos que permiten la escritura del contenido informativo de un objeto.

1.4.9 Code First

Enfoque dentro de Entity Framework que permite la creación de la base de datos a partir de un modelo de objetos. Este enfoque apunta al dominio de la aplicación e inicia la creación de las clases que definirán las entidades de dominio. La definición de clases y sus propiedades coinciden con el diseño de la base de datos. Una vez definidas, Entity Framework toma la configuración implementada por medio de Fluent-API o atributos de Data Annotation en las clases para realizar una migración, para que por último se cree o actualice el esquema de base de datos usando la migración automática o migración basada en código.

1.4.10 LINQ

Consulta de lenguaje integrado o LINQ, es un conjunto de tecnologías que permite la comprobación de tipos de consultas en compilación. Utiliza una sintaxis declarativa para escribir las consultas que apuntan a obtener la información de diferentes orígenes de datos: Base de datos SQL, conjunto de datos ADO.NET, documentos XML, varios servicios web y cualquier colección de objetos. Además, es compatible con la tecnología IntelliSense. LINQ evita aprender el lenguaje para tecnologías por separado y consolida mediante las palabras clave y operadores propios las funcionalidades de filtrado, ordenación y agrupamiento de los resultados de los diversos orígenes de datos; cabe resaltar que se pueden utilizar los mismos patrones de expresión para las consultas y transformación de los datos.

1.4.11 JSON

Notación de objeto de JavaScript, o JavaScript Object Notation; es un estándar de formato simple de intercambio de datos. Es independiente del lenguaje de programación que se utilice, pero utiliza algunas convenciones similares de lenguajes conocidos como C, C++, java, C#, JavaScript y otros. Está compuesto por dos estructuras: primeramente, utiliza el concepto de colección de pares de nombre y valor, que puede entenderse como un objeto, registro, estructura, etc; y como segunda estructura, una lista ordenada de valores, que es similar a un vector, o matriz, o lista.

1.4.12 DTO

Objeto de transferencia de datos, o Data Transfer Object; Es un objeto de una clase simple que contiene atributos a manera de resumida de la información precisa que el desarrollador retornará como respuesta a la consulta recibida. Este patrón nace con la necesidad de recuperar la información del servidor con una sola invocación. En un DTO no viaja lógica de negocio, y omite información innecesaria, esto significa que debe ser de solo lectura y serializable.

1.4.13 API REST

API (Application Programming Interface o Interfaz de programación de aplicaciones) REST (Representational State Transfer o Representación de transferencia de estado), es un tipo de API que está construida en Back-end y que tiene uno o múltiples rutas o métodos configurados para resolver diferentes funcionalidades de la lógica de negocio, además consultan a la información de la base de datos y que puede ser consumido por el protocolo HTTP por cualquier cliente a través de diferentes dispositivos como una aplicación web, móvil, etc. La arquitectura REST permite el almacenamiento de datos en caché, que el estado no se envíe en las peticiones y que el acceso a los datos pueda estar condicionado a permisos cuando otra aplicación acceda. REST tiene para cada recurso un identificador único llamado URI que,

al momento de contestar a la consulta del servidor, utiliza una codificación de respuestas (200, respuesta exitosa; 300, redirección; 400 Petición inválida; 500, errores en el servidor).

1.4.14 Programación orientada a objetos

Paradigma de programación que surgió en la década del 70' que utiliza objetos como elementos fundamentales en la construcción de la solución. El objeto es la abstracción de un hecho o ente en el mundo real, y está compuesta de atributos que representa sus características principales, así como también de métodos, que expresan o emulan las acciones o comportamientos que el objeto tendrá.

1.4.15 Servidor IIS

Servidor web donde se pueden alojar aplicaciones para el sistema operativo de Microsoft. Las aplicaciones pueden ser publicadas en internet como de manera local para ser utilizadas en intranet. Puede ofrecer servicios web, SMTP, FTP entre otros.

1.5 Aspectos Relevantes del Desarrollo

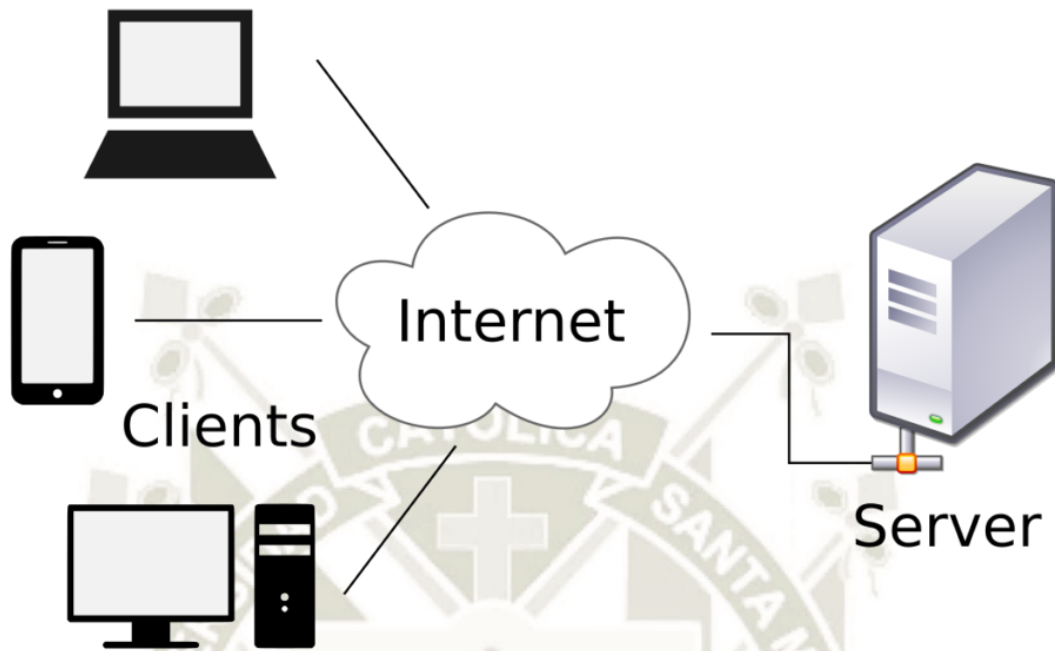
1.5.1 Arquitectura del Software

La arquitectura de software que se manejará en este proyecto es la de cliente – servidor como se muestra en la Figura 2, dividiendo el desarrollo en:

La parte del cliente que está orientada a la interfaz con la que interactúa el usuario final, siendo desarrollada principalmente con la tecnología de la librería React.Js.

La parte del servidor estará integrada por los diferentes servicios que realizarán las funcionalidades requeridas para el desarrollo y gestión de sílabos. La tecnología principal para el desarrollo es a través del framework .Net Core mediante el lenguaje de programación C#.

Figura 2 *Arquitectura Cliente - Servidor*



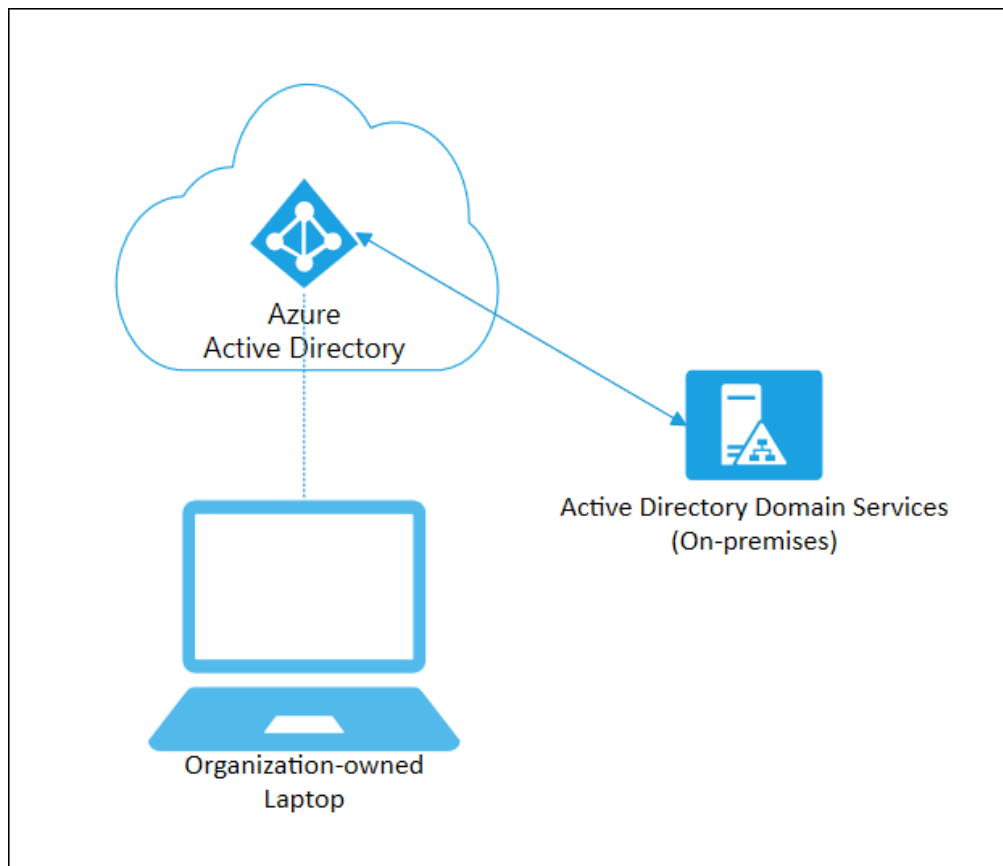
Fuente: Client-server model, Wikipedia

(https://en.wikipedia.org/wiki/Client%E2%80%93server_model)

1.5.2 Control de acceso a la aplicación

Para el control de acceso a la aplicación se utiliza el API – Azure Active Directory, que es un servicio para la administración de identidades y accesos basados en la nube. Según la configuración del administrador, solo se brinda acceso por las credenciales institucionales de la Universidad Católica de Santa María. La conexión se puede apreciar cómo se visualiza en la Figura 3.

Figura 3 *Conexión por Azure Active Directory*



Fuente: Conexión por Azure AD, por Microsoft, 2022, (<https://docs.microsoft.com/es-es/azure/active-directory/devices/concept-azure-ad-join>)


1.5.3 Estructura del Silabo

La estructura del sílabo que se manejará en la aplicación web se basa en el diseño para sílabos elaborado por el centro de desarrollo académico de la Universidad Católica de Santa María.

Para la parte del Plan de Estudios, el modelo es como se muestra en las Figuras 4,5 y

6.

Figura 4 *Diseño de Sílabo - Plan de Estudio I*

 UNIVERSIDAD CATÓLICA DE SANTA MARÍA AREQUIPA- PERÚ								
FACULTAD: _____								
ESCUELA PROFESIONAL: _____								
PLAN DE ESTUDIOS								
SÍLABO DE ASIGNATURA								
1. IDENTIFICACIÓN ACADÉMICA								
1.1.- Nombre de la Asignatura: Código de la Asignatura: Semestre Académico en que se desarrolla:						El desarrollo de las actividades académicas se distribuye en tres fases.		
1.2.- Peso Académico de la Asignatura.								
CRÉDITOS	HORAS SEMANALES				HORAS SEMESTRALES			Cada semestre académico comprende dieciocho semanas. (Resolución M ^o 6199-CU-2016)
	Horas Teóricas	Horas Prácticas		Horas Virtuales	Horas Teóricas	Horas Prácticas	Horas Virtuales	
		Práctica de Aula	Jefe de Prácticas					
1.3.- Código, nombre y créditos de Asignaturas Equivalentes:								
1.4.- Código y nombre de Asignaturas <u>Pre-requisito</u> :								

Fuente: Diseño de sílabo desarrollado por el Centro de Desarrollo académico de la Universidad Católica de Santa María

Figura 5 *Diseño de Sílabo - Plan de Estudio II*

2. SUMILLA
3. COMPETENCIAS DEL PERFIL DE EGRESO A LAS QUE CONTRIBUYE LA ASIGNATURA
4. COMPETENCIAS PROPIAS DE LA ASIGNATURA

Fuente: Diseño de sílabo desarrollado por el Centro de Desarrollo académico de la Universidad Católica de Santa María



Figura 6 *Diseño de Sílabo - Plan de Estudios III*

5. CONTENIDOS BÁSICOS DE LA ASIGNATURA
6. EVALUACIÓN DE COMPETENCIAS ADQUIRIDAS
RESULTADOS DE APRENDIZAJE
Evidencias de conocimiento:
Evidencias de desempeño y/o producto:
INSTRUMENTOS DE EVALUACIÓN
7. BIBLIOGRAFÍA

Fuente: Diseño de sílabo desarrollado por el Centro de Desarrollo académico de la Universidad Católica de Santa María

El programa formativo comprende las secciones mostradas en las figuras 7, 8, 9 y 10.

Figura 7 Diseño de Sílabo - Programa Formativo I



UNIVERSIDAD CATÓLICA DE SANTA MARÍA

PROGRAMA FORMATIVO DE ASIGNATURA

I. IDENTIFICACIÓN ACADÉMICA

1. Facultad: _____

2. Departamento Académico: _____

3. Nombre de la Asignatura: _____

_____ Código: _____

4. Programa Profesional donde se desarrolla la asignatura

5. Docente (s) y /o Jefe (s) de Práctica (s)

Código	Apellidos y Nombres	Función	Categoría

6. Ubicación y Distribución Horaria

AÑO ACADEMICO	SEMESTRE	CRÉDITOS	HORAS SEMANALES			HORAS SEMESTRALES		
			Horas Teóricas	Horas Prácticas		Horas Virtuales	Horas Teóricas	Horas Prácticas
				Práctica Docente	Jefe de Prácticas			

7.

Ambiente donde se realiza el aprendizaje
Teoría: _____
Práctica: _____

*Fuente: Diseño de sílabo desarrollado por el Centro de Desarrollo académico de la
Universidad Católica de Santa María*

Figura 8 *Diseño de Sílabo - Programa Formativo II*

II.- LINEAMIENTO ACADÉMICO PROFESIONAL

1. Sumilla:

2. Competencias del perfil de egreso a las que contribuye la asignatura

3. Competencias propias de la asignatura

*Fuente: Diseño de sílabo desarrollado por el Centro de Desarrollo académico de la
Universidad Católica de Santa María*

Figura 9 Diseño de Sílabo - Programa Formativo III

III. PROGRAMACIÓN POR FASE DE APRENDIZAJE					
FASE	Título de Fase	Total de Horas de Fase	Cronograma de la Fase	Desde Hasta	
COMPETENCIAS					
RESULTADOS DE APRENDIZAJE	TEMAS DE LA FASE	ESTRATEGIAS DE APRENDIZAJE-ENSEÑANZA	EVALUACIÓN DE LOS APRENDIZAJES		
			Criterios	Instrumentos de Evaluación	Porcentaje % Teoría Práctica
De Conocimiento					
De Desempeño					
y/o Producto					
Actividad de Investigación Formativa, y/o Responsabilidad Social					

BIBLIOGRAFÍA:

Fuente: Diseño de sílabo desarrollado por el Centro de Desarrollo académico de la Universidad Católica de Santa María

Figura 10 Diseño de Sílabo - Programa Formativo IV

PROGRAMACIÓN DE ACTIVIDADES DE INVESTIGACIÓN FORMATIVA Y RESPONSABILIDAD SOCIAL

Área	Denominación de la actividad	Propósito	Indicadores de evaluación	Beneficiarios	Responsables	Cronograma
Investigación Formativa						
Responsabilidad Social						

FIRMA: _____

NOMBRES Y APELLIDOS: _____

CODIGO: _____

Fuente: Diseño de sílabo desarrollado por el Centro de Desarrollo académico de la Universidad Católica de Santa María

Capítulo 2: Documentación Técnica

2.1 Plan del Proyecto Informático

2.1.1 Planificación temporal del Proyecto

El proyecto de Gestión de Sílabos se encuentra dividido en las etapas de Planificación y Estructuración del proyecto y Desarrollo del Proyecto. Estas etapas están expresadas al cronograma moldeado en forma de “Diagrama de Gantt”.

2.1.1.1 Planificación y estructuración del proyecto

Esta etapa consiste en analizar los requerimientos basados en la solicitud del Centro Académico por medio de las entrevistas y la construcción de la Base de datos. La programación en términos de fechas y duración de días hábiles se puede verificar en la Tabla 1:

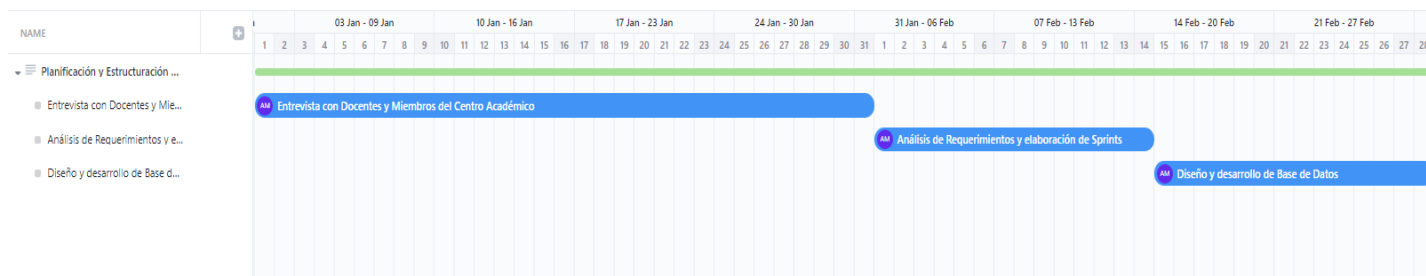
Tabla 1 *Actividades de Planificación y Estructuración del proyecto*

Actividad	Inicio	Fin	Duración (Días)
Entrevista con Docentes y Miembros del Centro Académico	01/01/2021	31/01/21	21
Análisis de Requerimientos y elaboración de Sprints	01/02/21	14/02/21	10
Diseño y Desarrollo de Base de Datos	15/02/21	28/02/21	10

Fuente: Elaboración propia.

A continuación, se muestra el diagrama de Gantt, de las actividades anteriormente mencionadas.

Figura 11 Diagrama de Gantt de la Planificación y Estructuración del Proyecto



Fuente: Elaboración propia.

Las actividades en esta etapa:

A) Entrevista con Docentes y Miembros del Centro Académico.

Reuniones con docentes y miembros del centro académico para poder recolectar la información sobre el funcionamiento actual de la gestión de sílabos, las necesidades que presentan, la proyección de una solución esperada y los requisitos tecnológicos a usar.

B) Análisis de Requerimientos y elaboración de Sprints

Revisión posterior a la recolección de información, en la cual se estructura los datos, se analizan, se ordenan con el fin de diseñar la solución de acuerdo con los requerimientos y proponer las fases de entregas o Sprints.

C) Diseño y Desarrollo de Base de Datos

Diseño de la base de datos, según el entendimiento de los requerimientos, fundamentados en las reuniones y entrevistas.

2.1.1.2 Desarrollo del proyecto de Software

Etapa de construcción, implementación, pruebas y ejecución integral del sistema. La codificación del sistema en sus frentes de FrontEnd y Backend se desarrollará por los lenguajes y frameworks coordinados en las reuniones. La programación de las actividades en términos de fecha y en duración de días hábiles, se aprecia en la Tabla 2:

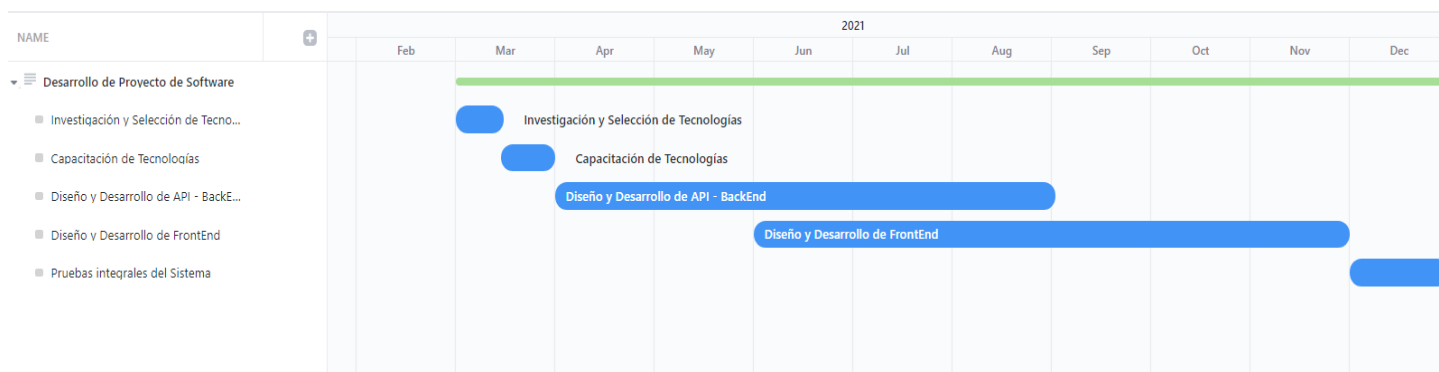
Tabla 2 Desarrollo de proyecto de software GSilabos

Actividad	Inicio	Fin	Duración (Días)
Investigación y selección de tecnologías	01/03/2021	15/03/21	11
Capacitación de Tecnologías	16/03/21	31/03/21	12
Diseño y Desarrollo de API – BackEnd	01/04/21	30/08/21	109
Diseño y Desarrollo de FrontEnd	01/06/21	30/11/21	131
Pruebas Integrales del Sistema	01/12/21	31/12/21	23

Fuente: Elaboración propia.

El diagrama de Gantt de la Figura 12 muestra la distribución de las actividades en la línea de tiempo en los meses que demorará desarrollar esta etapa.

Figura 12 Diagrama de Gantt de Desarrollo de Proyecto de Software



Fuente: Elaboración propia.

Las actividades de esta etapa son:

A) Investigación y Selección de Tecnologías

Investigar las tecnologías actuales y determinar cuál es la que se adecue a las necesidades de la UCSM y que permita la instalación ágil en los servidores privados.

B) Capacitación de Tecnologías

Seleccionadas las tecnologías para el backend y frontend se procede a tener un periodo de capacitación que permita instruirse en la utilización de estas.

C) Diseño y Desarrollo de API – Backend

Periodo de diseño de la arquitectura y codificación del API. Estableciendo los métodos o servicios necesarios para las acciones requeridas dentro del sistema.

D) Diseño y Desarrollo de Frontend

Periodo de diseño y selección del estilo a adoptar para la vista del sistema. Utilización de componentes que permitan el dinamismo sobre textos, secciones, botones, etc.

E) Pruebas integrales del Sistema

Integración final de Backend y FrontEnd con la finalidad de realizar pruebas integrales sobre las funcionalidades permitidas en el sistema (Desarrollo de sílabo, visualización de archivo en formato PDF, actualización de estados, selección de cursos).

2.1.2 Estudio de viabilidad del proyecto

La viabilidad del proyecto se determina por las siguientes divisiones:

2.1.2.1 Viabilidad Técnica

Los requisitos técnicos para el desarrollo y puesta en marcha del sistema son

A) Ambiente para poder realizar la publicación del aplicativo, siendo el detalle lo descrito en la tabla 3.

Tabla 3 *Requisitos de Ambiente*

Elemento	Nombre del Software	Función del software
Servidor del SGBD	SQL Server 2019 (15.x)	Almacenamiento de la base de datos del Sistema de Gestión de Sílabos
Administrador del SGBD	SQL Server Management Studio 15.0.2000.5	Administrador de la Base de Datos en un servidor web
Servidor Web	IIS v1809	
Runtime para la ejecución de la aplicación	ASP.NET Core Runtime 5.0.13	Permite la ejecución de la aplicación codificada en C#
Gestor de Servidores	Microsoft Windows Server manager 10.0.17763.168	Administrador de servidores

Fuente: Elaboración propia.

- B) Sistema operativo Windows de 64 bits para la máquina que servirá de herramienta de trabajo.
- C) IDE Visual Studio 2019 v 16.9.5 para el desarrollo del API que contendrá los servicios.
- D) Google Chrome Versión 97.0.4692.71 para la ejecución del aplicativo web.
- E) SQL Server Management Studio 15.0.2000.5 para el uso de base de datos.
- F) Librerías React para el desarrollo de front, Tailwind CSS para aspecto y bignumber para cálculos matemáticos.

2.1.2.2 Viabilidad Económica

El proyecto tendrá el siguiente coste y beneficio. En la tabla 4 se aprecia el costo de la implementación del proyecto.

Tabla 4 Costes Totales

Costos	Total (S/
Personal	S/. 10000.00
Equipo Computacional	S/. 6500.00
Licencia de Software	S/. 00.00
Suministros	S/. 113.00
Servicios	S/. 1080.00
Costo total Estimado	S/. 17693.00
Monto estimado de contingencia	S/. 2000.00
Total de Presupuesto	S/. 19693.00

Fuente: Elaboración propia.

La tabla 5, expresa los beneficios en ahorro de los siguientes elementos:

Tabla 5 Beneficios Totales

Beneficios Anuales	Descripción	S/.
Ahorro de costos en Suministros	Papel Bond A4 Millar x 96 Tinta de Impresora x 48	S/. 1,440.00 S/. 720.00
Ahorro en Espacio	Almacén de documentos	S/. 3,000.00
Ahorro de Tiempo	Jefe de Departamento tiempo (72h x 4m x S/.50)	S/. 14,400.00
	Comité Tiempo (100h x 4m x S/.150)	S/. 60,000.00
	Docentes Tiempo (48h x 2m x S/.50 x 15 - Cant. De Docentes)	S/. 72,000.00
Automatización de Tareas	Docentes sueldo (2m x S/.50 x 15 - Cant. De Docentes)	S/. 15,00.00
Total		S/. 153,060.00

Fuente: Elaboración propia.

La tabla 6, expresa el contraste entre el costo y el beneficio:

Tabla 6 *Costes vs Beneficios*

Descripción	Costo	Beneficio
Costo de Software	S/. 19,693.00	
Beneficios del Software		S/. 153,060.00
Total	S/. 19,693.00	S/. 153,060.00

Fuente: Elaboración propia.

El costo total del software será aproximadamente S/. 19,693.00 y otorgará una serie de beneficios calculados en S/ 153,060.00 dando como resultado una ganancia de S/. 133,367.00 en un 1 año. Sextuplicando así la inversión inicial.

2.1.2.3 Viabilidad Operativa

Para la UCSM, la necesidad de implementar un cambio en la forma en que los docentes desarrollan los sílabos, para después ser gestionados al ser cargados en el repositorio; es imperativo.

No solo debido al excedente de tiempo implementado en un desarrollo manual del sílabo, en dónde se invierte tiempo en redacción, investigación de conceptos, verificación de datos esenciales en la identificación académica, no tener una diversidad de opciones para instrumentos, estrategias o criterios, entre otros. Además, para los jefes de departamento aguardar la carga del documento en el repositorio, incrementa el lapso de consolidación del sílabo al no poder revisar su desarrollo en una etapa temprana y sugerir algunos cambios.

El sistema evita el gasto de tiempo en detalles no trascendentes y enfoca el esfuerzo en el contenido, apoyando al docente con:

- Disminución de tiempo en el proceso de construcción por parte del docente.
- Disminución de tiempo en el proceso de revisión por parte del Jefe de departamento.

- Aumenta la confianza en el docente y jefe de departamento por ser automatizada la carga académica.
- Reduce carga de trabajo administrativo, lo que mejora la eficiencia las partes involucradas.
- Los datos almacenados están guardados de manera segura en el servidor institucional de la UCSM

2.2 Especificación De Requisitos Del Proyecto De TIC's

Los requisitos del proyecto se establecen a través de la información recolectada. Se inicia con la estructuración de las historias de usuario, para luego formular los requisitos funcionales y no funcionales.

2.2.1 Historias de Usuario

Las historias de usuario se desarrollan a partir de las entrevistas con los usuarios los cuales expresan la necesidad para luego consolidarlo en el siguiente formato expresado en la siguiente lista de tablas:

Tabla 7 *Historia de Usuario 1 – Habilitar semestre académico*

Historia de Usuario	
Número: 1	
Usuario: Administrador	
Nombre Historia: Habilitar semestre académico	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 30	Iteración asignada: 0
Programador responsable: Anthony Meza	
Descripción: El administrador desea tener la opción de iniciar el semestre académico.	
Validación:	
<ul style="list-style-type: none"> - Permite al usuario mediante la ejecución de alguna opción, iniciar con la carga de la información correspondiente del nuevo semestre. 	

Fuente: Elaboración propia.

Tabla 8 *Historia de Usuario 2 – Ejecutar carga académica*

Historia de Usuario	
Número: 2	
Usuario: Administrador	
Nombre Historia: Ejecutar Carga académica	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 30	Iteración asignada: 0
Programador responsable: Anthony Meza	
Descripción: El administrador desea poder ejecutar la carga académica.	
Validación:	
<ul style="list-style-type: none"> - Ejecución de carga académica al iniciar semestre académico. - Información de cursos para docentes cargados. - Relación entre docentes y asignaturas. 	

Fuente: Elaboración propia.

Tabla 9 *Historia de Usuario 3 – Visualizar el desarrollo de sílabos*

Historia de Usuario	
Número: 3	
Usuario: Jefe de Departamento	
Nombre Historia: Visualizar el desarrollo de sílabos	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 60	Iteración asignada: 1
Programador responsable: Anthony Meza	
Descripción: El jefe de departamento desea visualizar los sílabos en PDF	
Validación:	
<ul style="list-style-type: none"> - Jefe de departamento realiza búsqueda de los sílabos por escuela profesional, código de docente, nombre de docente, nombre de asignatura, año en que fue se desarrolló y semestre académico. - Se visualiza el resultado con la información en columnas de escuela profesional, curso, docente, estado y semestre académico. - Los sílabos que aparecen en el resultado pueden tener la opción de generar archivo PDF. 	

Fuente: Elaboración propia.

Tabla 10 *Historia de Usuario 4 – Habilitar o cambiar estado de sílabo*

Historia de Usuario	
Número: 4	
Usuario: Jefe de Departamento	
Nombre Historia: Habilitar o cambiar estado de Sílabo	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 60	Iteración asignada: 1
Programador responsable: Anthony Meza	
Descripción: El jefe de departamento desea actualizar estado del sílabo a revisar	
Validación:	
<ul style="list-style-type: none"> - Jefe de departamento realiza búsqueda de los sílabos por escuela profesional, código de docente, nombre de docente, nombre de asignatura, año en que fue se desarrolló y semestre académico. - Se visualiza el resultado con la información en columnas de escuela profesional, curso, docente, estado y semestre académico. - Los sílabos sufren actualización de estado: “Cerrado” o devuelvo a “En Proceso”. 	

Fuente: Elaboración propia.

Tabla 11 *Historia de Usuario 5 – Observar lista de cursos asignados*

Historia de Usuario	
Número: 5	
Usuario: Docente	
Nombre Historia: Observar lista de cursos asignado	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 30	Iteración asignada: 1
Programador responsable: Anthony Meza	
Descripción: Docente tiene una lista de los cursos asignados en el semestre académico y puede seleccionar cualquiera para poder desarrollar.	
Validación:	
<ul style="list-style-type: none"> - Docente observa lista de cursos. - Tiene una opción en cada curso para iniciar desarrollo. 	

Fuente: Elaboración propia.

Tabla 12 *Historia de Usuario 6 – Transcribir sílabo del año pasado*

Historia de Usuario	
Número: 6	
Usuario: Docente	
Nombre Historia: Transcribir sílabo del año pasado	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 125	Iteración asignada: 6
Programador responsable: Anthony Meza	
Descripción: Docente puede copiar información ingresada del sílabo del año anterior.	
Validación:	
<ul style="list-style-type: none"> - Docente tiene una opción para solicitar transcripción. - Puede visualizar la información del sílabo anterior en PDF. - Realiza la transcripción al confirmar la acción. 	

Fuente: Elaboración propia.

Tabla 13 *Historia de Usuario 7 – Visualizar sílabo PDF*

Historia de Usuario	
Número: 7	
Usuario: Docente	
Nombre Historia: Visualizar sílabo PDF	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 137	Iteración asignada: 6
Programador responsable: Anthony Meza	
Descripción: Docente puede visualizar sílabo en PDF a través de una opción.	
Validación:	
<ul style="list-style-type: none"> - Docente tiene una opción para solicitar la visualización del sílabo. - Docente observa una ventana emergente con el sílabo cargado y en formato PDF. 	
<i>Fuente:</i> Elaboración propia.	

Tabla 14 *Historia de Usuario 8 – Visualizar Identificación Académica y Desarrollo de Sumilla*

Historia de Usuario	
Número: 8	
Usuario: Docente	
Nombre Historia: Visualizar identificación académica y Desarrollar Sumilla	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 50	Iteración asignada: 2
Programador responsable: Anthony Meza	
Descripción: Docente tiene una opción para visualizar identificación y un casillero para ingresar sumilla.	
Validación:	
<ul style="list-style-type: none"> - Docente se encuentra en el espacio de desarrollo de sílabo, con la posibilidad de seleccionar el apartado de identificación académica y ver datos esenciales de la asignatura. - Docente se encuentra en el espacio de desarrollo de sílabo, con la posibilidad de seleccionar el apartado de sumilla. - Al elegir el apartado de sumilla, el docente puede redactar una nueva sumilla, o modificar la sumilla transcrita. 	
<i>Fuente:</i> Elaboración propia.	

Tabla 15 *Historia de Usuario 9 – Seleccionar competencias por perfil de egreso*

Historia de Usuario	
Número: 9	
Usuario: Docente	
Nombre Historia: Seleccionar competencias por perfil de egreso	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 30	Iteración asignada: 2
Programador responsable: Anthony Meza	
Descripción: Docente tiene una lista de las competencias desarrolladas por el centro académico.	
Validación:	
<ul style="list-style-type: none"> - Docente se encuentra en el espacio de desarrollo de sílabo, con la posibilidad de seleccionar el apartado de competencias por perfil de egreso. - Puede seleccionar las competencias que acompañarán al desarrollo del sílabo. 	

Fuente: Elaboración propia.

Tabla 16 *Historia de Usuario 10 – Construir competencias propias de asignatura*

Historia de Usuario	
Número: 10	
Usuario: Docente	
Nombre Historia: Construir competencias propias de asignatura	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 35	Iteración asignada: 3
Programador responsable: Anthony Meza	
Descripción: Docente puede redactar las competencias propias de la asignatura y asignar a la fase que corresponderán.	
Validación:	
<ul style="list-style-type: none"> - Docente se encuentra en el espacio de desarrollo de sílabo, con la posibilidad de seleccionar el apartado de competencias propias de la asignatura. - Puede elaborar la competencia redactándola y asignándole la fase en la que participará. - Docente asigna el orden en el que se mostrarán las competencias en caso de que exista más de una. 	

Fuente: Elaboración propia.

Tabla 17 *Historia de Usuario 11 – Desarrollo Unid. de aprendizaje teórica y de laboratorio*

Historia de Usuario	
Número: 11	
Usuario: Docente	
Nombre Historia: Desarrollo unidades de aprendizaje teóricas y de laboratorio.	
Prioridad en negocio: Media	Riesgo en desarrollo: Alta
Puntos estimados: 70	Iteración asignada: 3
Programador responsable: Anthony Meza	
Descripción: Docente puede redactar las unidades teóricas y de laboratorio para cada fase	
Validación:	
<ul style="list-style-type: none"> - Docente se encuentra en el espacio de desarrollo de sílabo, con la posibilidad de seleccionar el apartado de contenidos básicos de aprendizaje. - Puede desarrollar las unidades, y subunidades teóricas, además de las unidades prácticas o de laboratorio. - Docente asigna el orden en el que se mostrarán las subunidades de cada unidad en caso de que existan más de una. 	

Fuente: Elaboración propia.

Tabla 18 *Historia de Usuario 12 – Ingresar resultados de aprendizaje por fase*

Historia de Usuario	
Número: 12	
Usuario: Docente	
Nombre Historia: Ingresar resultados de aprendizaje por fase	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 60	Iteración asignada: 4
Programador responsable: Anthony Meza	
Descripción: Docente puede ingresar los resultados de aprendizaje según orden requerido y asignando la fase	
Validación:	
<ul style="list-style-type: none"> - Docente se encuentra en el espacio de desarrollo de sílabo, con la posibilidad de seleccionar el apartado de evaluación de competencias adquiridas. - Puede elaborar ingresar los resultados de conocimiento, desempeño y producto, asignándole la fase en los que participarán. - Docente asigna el orden en el que se mostrarán los resultados en caso de que exista más de uno. 	

Fuente: Elaboración propia.

Tabla 19 *Historia de Usuario 13 – Registrar nombre de fases*

Historia de Usuario	
Número: 13	
Usuario: Docente	
Nombre Historia: Registrar nombre de fases	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 30	Iteración asignada: 4
Programador responsable: Anthony Meza	
Descripción: Docente registra el nombre de cada fase que tendrá la asignatura.	
Validación:	
<ul style="list-style-type: none"> - Docente se encuentra en el espacio de desarrollo de sílabo, con la posibilidad de seleccionar el apartado de evaluación de competencias adquiridas. - Puede ingresar el nombre de cada fase. 	

Fuente: Elaboración propia.

Tabla 20 *Historia de Usuario 14 – Desarrollar criterios, estrategias e instrumentos*

Historia de Usuario	
Número: 14	
Usuario: Docente	
Nombre Historia: Desarrollar criterios, estrategias e instrumentos de aprendizaje	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 80	Iteración asignada: 4
Programador responsable: Anthony Meza	
Descripción: Docente desarrolla criterios, estrategias e instrumentos de aprendizaje por cada fase, con la posibilidad de tener opciones y registrar cada concepto de manera independiente.	
Validación:	
<ul style="list-style-type: none"> - Docente se encuentra en el espacio de desarrollo de sílabo, con la posibilidad de seleccionar el apartado de evaluación de competencias adquiridas. - Tiene la posibilidad de elegir la fase a la que ingresará cada concepto - Tiene las opciones de ingresar un criterio de manera manual; o seleccionar de una lista de opciones ingresadas previamente. - Tiene las opciones de ingresar una estrategia de manera manual; o seleccionar de una lista de opciones ingresadas previamente. - Tiene las opciones de ingresar un instrumento de manera manual; o seleccionar de una lista de opciones ingresadas previamente. 	

Fuente: Elaboración propia.

Tabla 21 *Historia de Usuario 15 – Ingresar actividades de investigación y responsabilidad social*

Historia de Usuario	
Número: 15	
Usuario: Docente	
Nombre Historia: Ingresar actividades de investigación y responsabilidad social	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 70	Iteración asignada: 5
Programador responsable: Anthony Meza	
Descripción: Docente ingresa actividades de investigación y responsabilidad social para cada fase.	
Validación:	
<ul style="list-style-type: none"> - Docente se encuentra en el espacio de desarrollo de sílabo, con la posibilidad de seleccionar el apartado de programación de actividades de investigación formativa y responsabilidad social. - Puede ingresar el texto de cada columna y seleccionar a qué fase pertenecerá la actividad, además de elegir el área. 	

Fuente: Elaboración propia.

Tabla 22 *Historia de Usuario 16 – Seleccionar Bibliografía*

Historia de Usuario	
Número: 16	
Usuario: Docente	
Nombre Historia: Seleccionar bibliografía	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 60	Iteración asignada: 5
Programador responsable: Anthony Meza	
Descripción: Seleccionar la bibliografía que se utilizará como base para la asignatura	
Validación:	
<ul style="list-style-type: none"> - Docente se encuentra en el espacio de desarrollo de sílabo, con la posibilidad de seleccionar bibliografía. - Puede agregar bibliografía para cada fase. - Se despliega una ventana emergente para buscar bibliografía por nombre, autor, editorial, lugar y año de publicación. 	

Fuente: Elaboración propia.

Tabla 23 *Historia de Usuario 17 – Visualizar Bitácora*

Historia de Usuario	
Número: 17	
Usuario: Docente	
Nombre Historia: Visualizar Bitácora	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 35	Iteración asignada: 6
Programador responsable: Anthony Meza	
Descripción: Observar la bitácora de cambios de la asignatura.	
Validación:	
<ul style="list-style-type: none"> - Docente se encuentra en el espacio de desarrollo de sílabo, con la posibilidad de seleccionar opción para visualizar bitácora. - Se despliega ventana emergente con filtros por docente y acción en la lista de la bitácora ordenada por fecha descendente. 	

Fuente: Elaboración propia.

2.2.2 *Diagrama de casos de Uso.*

El diagrama de casos de uso se desarrolla a partir historias de usuarios como muestra gráfica sobre los casos de uso que tendrá cada rol que existe en el sistema, esto se especifica en la Figura 13.

Lo roles tendrán los siguientes casos de uso:

A) Actor Administrador

- a) Habilitar semestre académico
- b) Ejecuta carga académica

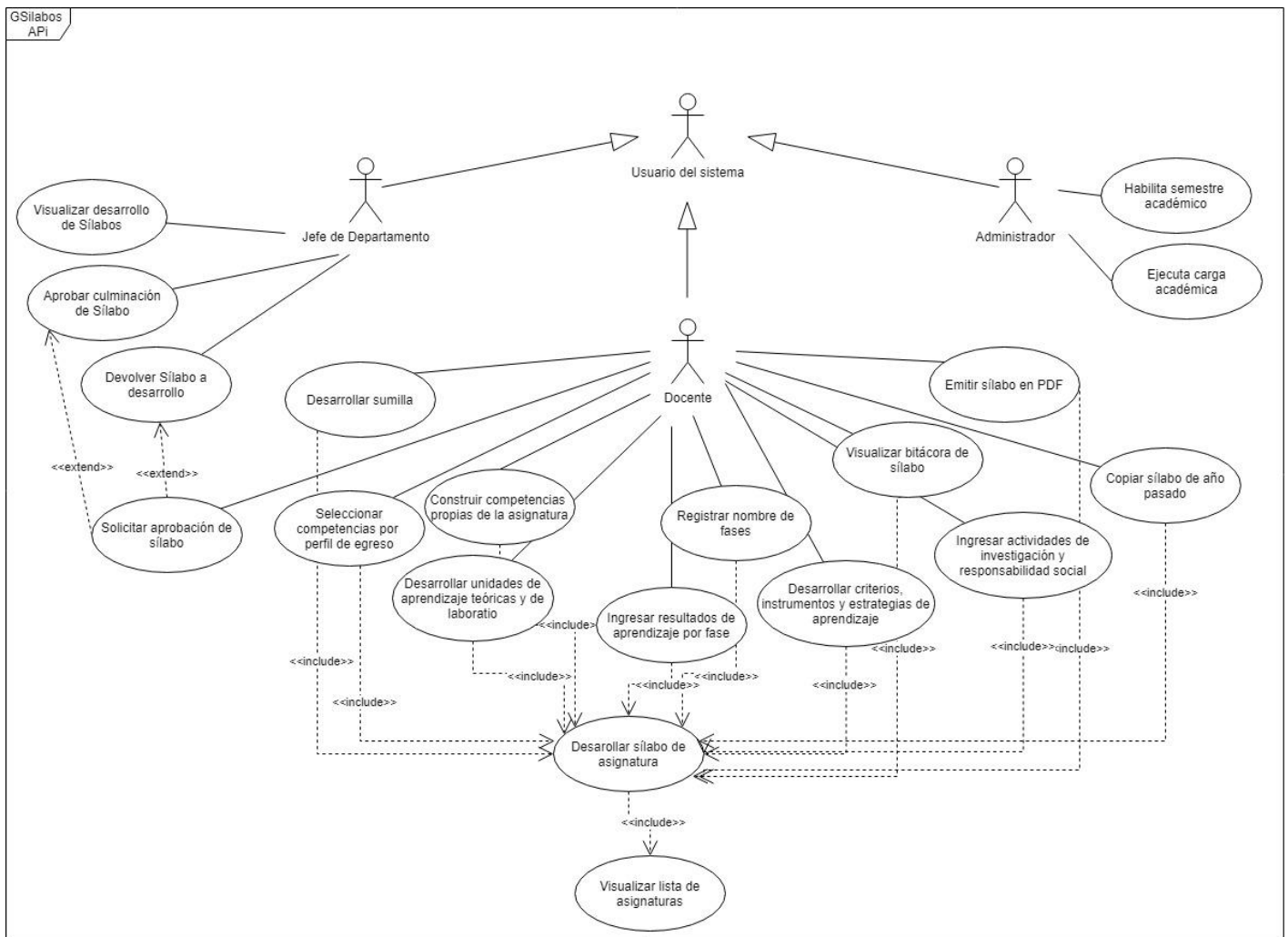
B) Actor Jefe de Departamento

- a) Visualizar desarrollo de sílabos
- b) Aprobar culminación de sílabo
- c) Devolver sílabo a desarrollo

C) Actor Docente

- a) Visualizar lista de asignaturas
- b) Desarrollar sílabo de asignatura
- c) Copiar sílabo de año pasado
- d) Desarrollar sumilla
- e) Seleccionar competencias por perfil de egreso
- f) Construir competencias propias de asignatura
- g) Desarrollar unidades de aprendizaje teóricas y de laboratorio
- h) Ingresar resultados de aprendizaje por fase
- i) Registrar nombre de fases
- j) Desarrollar criterios, estrategias e instrumentos de aprendizaje
- k) Visualizar bitácora de sílabo
- l) Emitir sílabo en PDF
- m) Solicitar aprobación de culminación de sílabo

Figura 13 Diagrama de Casos de Uso



Fuente: Elaboración propia.

2.2.3 Requisitos Funcionales.

Con las historias de usuario se elaboran los requerimientos funcionales. Los mismos son definidos y colocados en los sprints que involucra la parte inicial dentro de la metodología ágil SCRUM. Se plantea como elementos de análisis a la complejidad, prioridad y tiempo en horas de desarrollo. Esta estimación es en base a la creación tanto de la parte del API (backend) y la parte visual (Frontend) expresados en la tabla 24.

Tabla 24 *Requerimientos Funcionales*

REQUERIMIENTOS FUNCIONALES

Sprint	Historia	Descripción	Complejidad	Prioridad	Tiempo (h)
0	1	El administrador desea tener la opción de iniciar el semestre académico a través de la ejecución de un script en la tabla “Paramétricas”.	Media	Alta	30
0	2	El Sistema de Gestión de Sílabos tiene una interfaz para que el administrador pueda Ejecutar Carga académica.	Media	Alta	30
1	3	El Sistema de Gestión de Sílabos, permite al jefe de departamento tener una interfaz donde a través de determinados filtros pueda visualizar el desarrollo de sílabos en el semestre académico.	Media	Media	60
1	4	El Sistema de Gestión de Sílabos, permite al jefe de departamento tener una interfaz donde a través de determinados filtros pueda habilitar o cambiar estado de Sílabos que se encuentren terminados por parte de los docentes. Podrá cerrarlos o devolverlos para corrección.	Media	Media	60
1	5	El Sistema de Gestión de Sílabos permite tener para el docente una interfaz donde observa la lista de cursos asignados en el semestre académico.	Media	Media	30
2	8	El Sistema de Gestión de Sílabos Visualizar cuenta con la visualización de la identificación académica y un apartado para el Desarrollo de la Sumilla.	Alta	Alta	50
2	9	El Sistema de Gestión de Sílabos proporciona la opción de seleccionar competencias por perfil de egreso.	Media	Media	30
3	10	El Sistema de Gestión de Sílabos permite construir competencias propias de asignatura, según el criterio del docente.	Media	Media	35
3	11	El Sistema de Gestión de Sílabos contiene las opciones para el desarrollo unidades y subunidades de aprendizaje teóricas y unidades de laboratorio.	Alta	Media	70

4	12	El Sistema de Gestión de Sílabos contiene las opciones para el desarrollo el ingreso de resultados de aprendizaje por fase.	Alta	Alta	60
4	13	El Sistema de Gestión de Sílabos contiene opciones para registrar nombre de las fases del semestre académico.	Media	Alta	30
4	14	El Sistema de Gestión de Sílabos contiene opciones para desarrollar criterios, estrategias e instrumentos de aprendizaje.	Alta	Alta	80
5	15	El Sistema de Gestión de Sílabos contiene opciones para ingresar actividades de investigación y responsabilidad social.	Alta	Alta	70
5	16	El Sistema de Gestión de Sílabos contiene opciones para seleccionar bibliografía para la asignatura en cada fase.	Alta	Alta	60
6	17	El Sistema de Gestión de Sílabos contiene la opción para visualizar Bitácora	Media	Media	35
6	6	El Sistema de Gestión de Sílabos contiene la opción de transcribir sílabo del año pasado,	Alta	Alta	125
6	7	El Sistema de Gestión de Sílabos contiene la opción de visualizar sílabo PDF.	Alta	Alta	137

Fuente: Elaboración propia.

2.2.4 Requisitos No funcionales.

Los requisitos no funcionales describen los atributos de calidad que contendrá el Sistema de Gestión de Sílabos, esto expresado en la tabla 25.

Tabla 25 *Requerimientos no Funcionales*

REQUERIMIENTOS NO FUNCIONALES				
Tipo	Nombre	Descripción	Prioridad	
Eficiencia	Búsqueda ágil de información histórica de sílabos	Obtención inmediata de la información histórica al momento de visualizar los sílabos en la pantalla Visualizar y Habilitar los Sílabos.	ALTA	
Desempeño	Obtener información histórica consumiendo los recursos necesarios	El Sistema de Gestión de Sílabos utiliza el código adecuado para realizar las consultas y tratamiento de la información de modo que el consumo de recursos sea el necesario e indispensable.	ALTA	
Confiabilidad	Solicitud de información necesaria.	Asegurar la integridad de la información que haya sido ingresada al sistema. No se permitirá información redundante o innecesaria	ALTA	
Usabilidad	Manual de usuario	Facilitar el entendimiento del sistema con la ayuda de manuales de usuario, que permitirán a los docentes tener una guía del flujo implementado en el sistema.	MEDIA	
Desarrollo	Lenguajes utilizados para programación del Sistema	El Sistema de Gestión de Sílabos utilizará las librerías React para el desarrollo visual, tailwindcss para aspecto y bignumber para cálculos matemáticos en la parte de FrontEnd. Y para la parte de BackEnd se utilizará C#9 en la creación del API.	ALTA	
Desarrollo	Gestor de Base de Datos	El gestor de base de datos que albergará la información del sistema es SQL SERVER.	ALTA	
Ético	Tratamiento adecuado de la información	La información ingresada en el sistema estará protegida de acceso no autorizado y divulgación.	MEDIA	
Espacio	Almacenamiento necesario	El Sistema de Gestión de Silabos utiliza el almacenamiento requerido en el alojamiento de los archivos publicados de FrontEnd y BackEnd además de la base de datos.	MEDIA	
Seguridad	Garantizar el acceso adecuado a los usuarios	El Sistema de Gestión de Sílabos brindará accesos por medio de los usuarios que los docentes ya tienen asignados con autenticación de Windows a través del API de Offices 365	ALTA	
Operacional	Mantenimiento operativo del Sistema	Permitir que se realicen soportes óptimos realizando documentación del diseño del sistema que ayudará al usuario de mantenimiento el entendimiento ágil del sistema.	MEDIA	

Fuente: Elaboración propia.

2.3 Especificación De Diseño

En la especificación de diseño se desarrolla en base a los requerimientos definidos.

Para ello, se desglosa en los siguientes puntos:

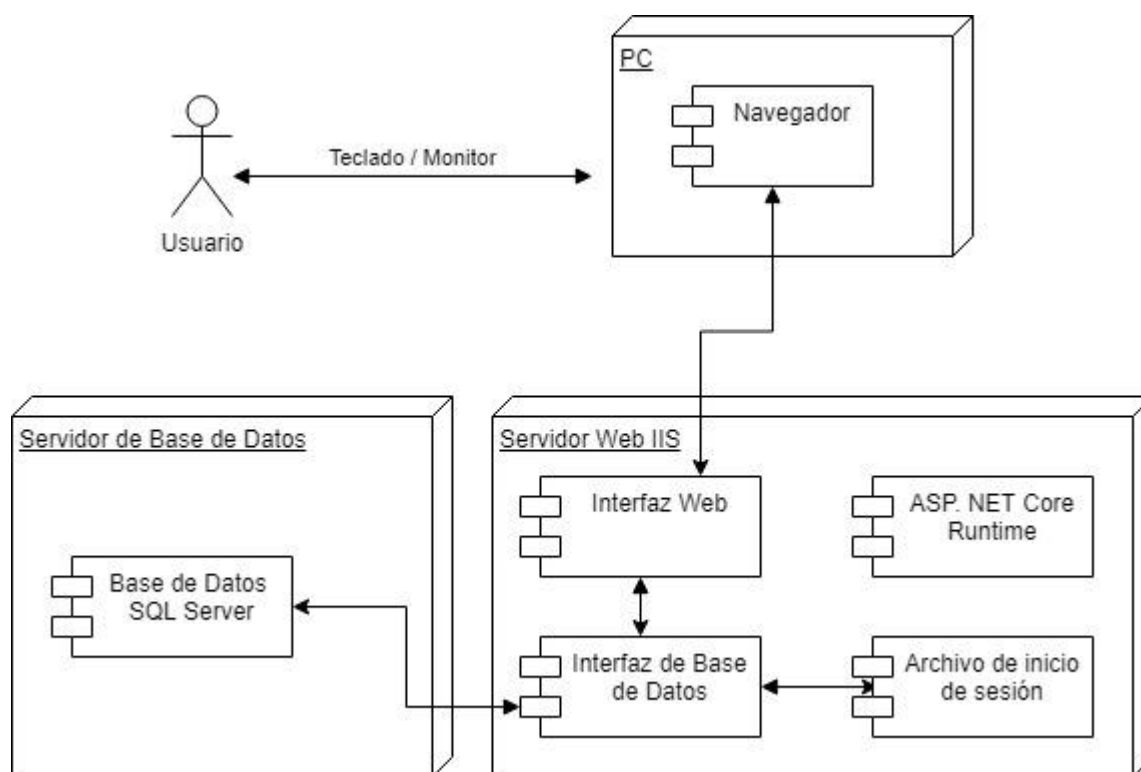
- a) Arquitectura física
- b) Arquitectura Lógica
- c) Diseño estructural
- d) Diseño Dinámico
- e) Diseño de Interfaces de usuario
- f) Diseño de Modelo de Datos
- g) Implementación de funcionalidades principales
- h) Mecanismos de reuso de software utilizados

2.3.1 *Arquitectura Física*

La arquitectura física mostrada en la figura 14, necesaria para la ejecución del sistema de gestión de sílabos está compuesta por la participación de:

- A) Computadora Personal o Dispositivo:** Equipo que el usuario utiliza para el ingreso al Sistema de Gestión de Sílabos.
- B) Servidor Web:** Contiene el servidor de aplicaciones en este caso IIS con los servicios que permiten al Sistema Operativo de Microsoft Windows la posibilidad de operar, alojar y ejecutar la aplicación de Sistema de Gestión de Sílabos.
- C) Servidor de Base de Datos:** Permite la gestión de la base de Datos SilabosDB utilizada para el Sistema de Gestión de Sílabos.

Figura 14 *Arquitectura Física*



Fuente: Elaboración propia.

2.3.2 *Arquitectura Lógica*

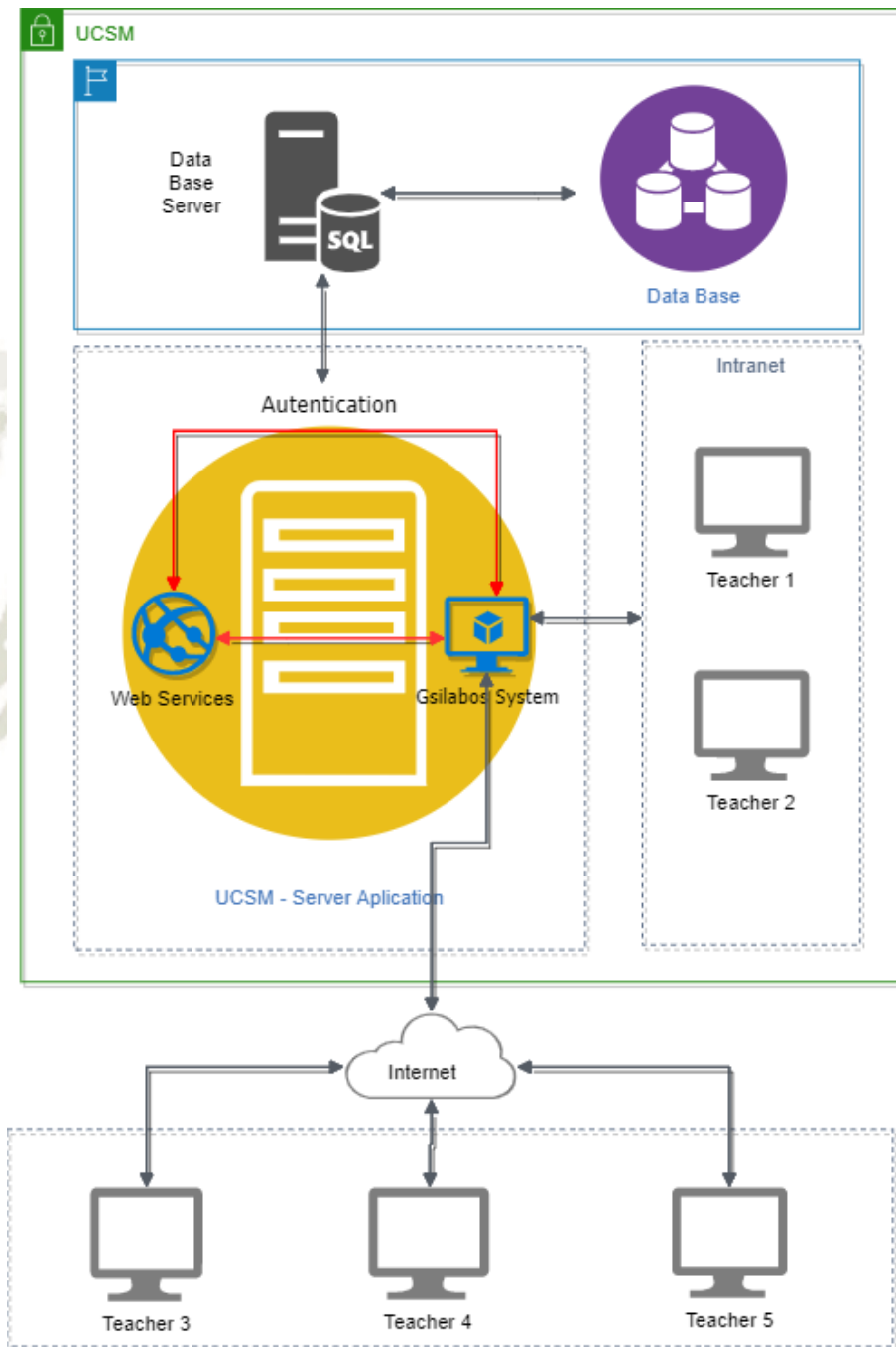
2.3.2.1 *Arquitectura Externa del Sistema*

La arquitectura externa del sistema será Web. La interacción con ella se podrá realizar por medio de internet, o de la intranet de la universidad. Los docentes podrán conectarse por medio de la autenticación de Windows a través del API Office 365 con los correos institucionales con que fueron provistos. Se validará la información y podrá acceder sin inconvenientes. Se generó una relación entre la base de datos de la UCSM y la base de datos que servirá para gestionar los sílabos.

El servidor que soportará la aplicación web, es el mismo donde se alojan las diversas soluciones que la UCSM tienen puestas en producción para su desenvolvimiento. Se establecen

dos repositorios o carpetas, una contiene la solución de Back-End y la otra la solución de Front-End. Se puede observar la arquitectura externa en la figura 15.

Figura 15 *Arquitectura Externa*



Fuente: Elaboración propia.

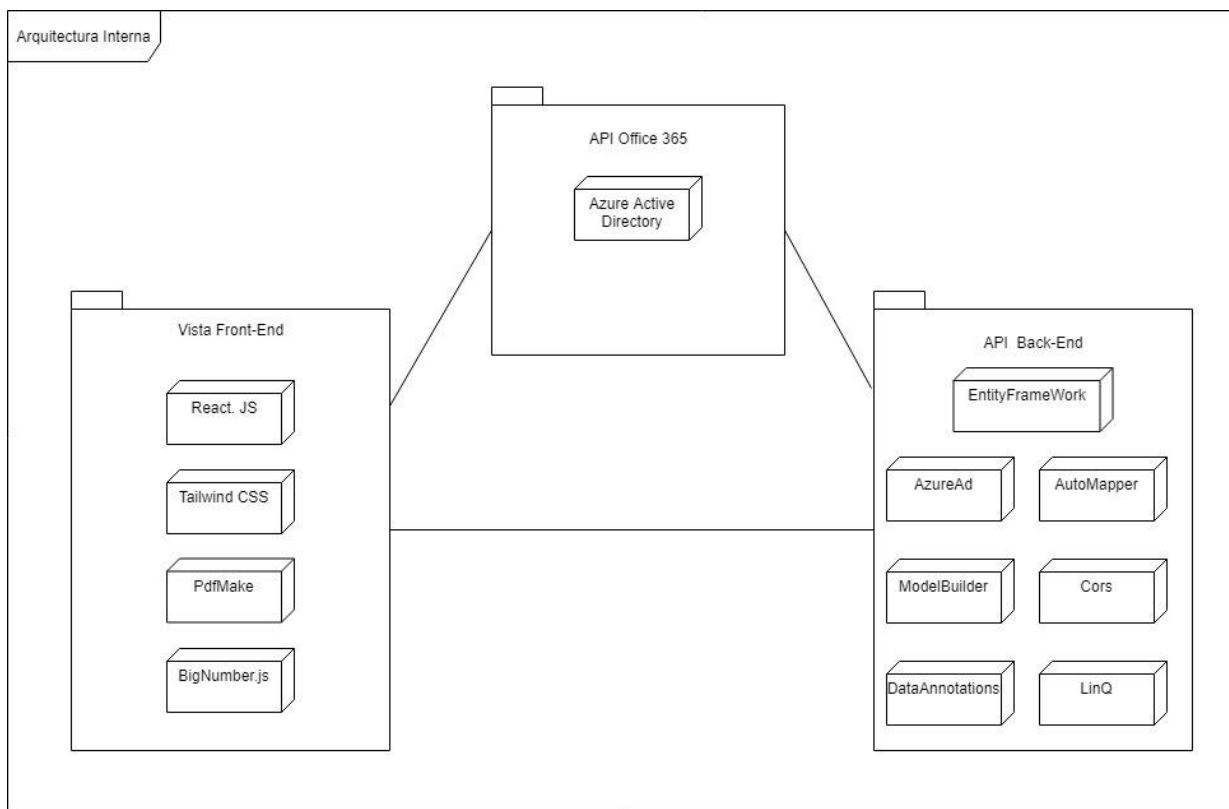
2.3.2.2 Arquitectura Interna del Sistema

La arquitectura interna del sistema es cliente – servidor. Se grafica la relación de las tecnologías utilizadas para el desarrollo del sistema. La parte del cliente donde se encuentra la presentación o FrontEnd está desarrollada en con la tecnología de la librería REACT para la creación de los componentes visuales y la interacción entre estos. Por otro lado, se utiliza el framework Tailwind CSS que está orientado a utilidades, permitiendo que el desarrollo del estilo sea específico al definir las clases que definirán los textos, color, etc. Así también como el API bignumber.js, que permite realizar operaciones aritméticas utilizando en este caso para el cálculo del peso académico. Y la librería PDFMAKE para la generación de archivos .PDF en javascript.

La parte de Servidor se maneja mediante API Rest, por el protocolo HTTP. El API está desarrollado con el lenguaje de programación C#, utilizando la tecnología de Entity Framework para mapeo de las entidades en la capa de datos, por el enfoque de Code First; permitiendo a través del código crear las clases que representarán a las entidades en la base de datos. El Gestor de Base de Datos donde estará alojada la información, será SQL SERVER 2019.

Se muestra la relación interna de las tecnologías en el diagrama de la arquitectura interna en la figura 16.

Figura 16 *Arquitectura Interna*



Fuente: Elaboración propia.

2.3.3 *Diseño Estructural*

El diseño estructural abarca el modelado de la arquitectura de la API Gestión de Sílabo, que se modelará en UML en los diagramas de clases, casos de uso y componente. Con estos diagramas se pretende mostrar de manera simple y conciso el diseño.

2.3.3.1 **Diagrama de Clases.**

En la Figura 17, se muestran los diagramas de clases que se utilizan en la API de gestión de Sílabos. Estos diagramas estarán divididos primeramente por los controladores definidos para el funcionamiento ordenado y a su vez por las secciones o categorías que tiene el documento del sílabo, para el caso del segundo controlador.

A) Controlador Sílabos

El controlador Sílabos tiene los métodos o servicios que, al consumirlos; resuelven:

- <Post> ListarSilabos (): Obtener lista de sílabos en la vista “Habilitar Sílabos”.
- <Post> ListarSilabos (): Obtener lista de sílabos en la vista “Visualizar Sílabos”.
- <Get> ListarSemestres(): Obtener los semestres que se utilizarán para los filtros.
- <Put> GuardarEstadoSilab(): Guardar el estado determinado por el jefe de departamento.
- <Get> VerSilabo(): Obtener el estado del sílabo actualmente.
- <Get> VerCurso(): Actualizar el estado del sílabo cuando este se encuentra de “Abierto” a “En proceso.”
- <Get> ListarEstados(): Obtener listado de estados para los sílabos.
- <Get> ListarCursosDocente(): Obtener los cursos que se encuentran asignados a los docentes.
- <Post> ReaperturaAsignatura(): Reapertura del Sílabo en caso de que el docente necesite realizar una edición después de solicitar el cierre del sílabo.
- <Post> VerEstadoSilabo (): Obtener el estado actual del sílabo.

B) Controlador Gestión de Sílabos

El controlador de gestión de sílabos contiene los métodos o servicios que resuelven:

a) Identificación académica y Sumilla

La identificación académica y la sumilla se encuentran representadas por el diagrama de clases mostrado en la Figura 18.

Los métodos utilizados en esta sección resuelven:

- <Post> VerIdentificacionAcademica(): Obtiene todo los datos para la parte de identificación académica del sílabo.
- <Post> BuscarSumillaPorSilabo(): Obtiene la sumilla del sílabo.
- <Post> GuardarSumillaEnSilabo(): Guarda el texto ingresado para la sumilla del sílabo desarrollado.

b) Competencias del perfil de egreso a las que contribuye la asignatura

Las competencias del perfil de egreso se encuentran representados por el diagrama de clases de la Figura 19.

Los métodos utilizados en esta sección resuelven:

- <Post> ListarCompetenciasPerfilEgreso(): Obtiene todo las competencias definidas por el centro académico para la Escuela Profesional.
- <Post> GuardarAsignaturaCompetenciasPerfilEgreso(): Guarda las competencias seleccionadas asociadas al sílabo.

c) Competencias propias de la asignatura

Las competencias propias de la asignatura ingresadas por el docente se encuentran desarrolladas en el diagrama de clases de la Figura 20.

Los métodos usados en esta sección son:

- <Post> ListarCompetenciasPropiasAsignatura(): Obtiene las competencias ingresadas por los docentes que sean propias de la asignatura.
- <Post> GuardarCompetenciasPropiasDeLaAsignatura(): Guarda las competencias desarrolladas por el docente para la asignatura.

d) Contenidos básicos por unidades de aprendizaje

Para los contenidos básicos por unidades de aprendizaje, se detalla en el diagrama de clases de la Figura 21.

Los métodos usados en esta sección son:

- <Post> ListarUnidadesFase(): Obtener las unidades de aprendizaje teóricas tratadas en las fases de la asignatura.
- <Post> GuardarUnidades(): Guardar las unidades establecidas por el docente para el estudio de la asignatura.

- <Post> ListarUnidadesLabFase(): Obtener unidades de laboratorios, o prácticas que se realizarán a lo largo del desarrollo de la asignatura en las diferentes fases.
- <Post> GuardarUnidadesLab(): Guardar las unidades de laboratorio establecidas por el docente para el estudio de la asignatura

e) Evaluación de competencias adquiridas

Diagrama de clases de la sección de evaluación de competencias adquiridas detallada en la Figura 22.

Los métodos usados en esta sección son:

- <Post> ListarAsignaturaFase(): Obtiene los títulos de las fases que el docente estableció para las fases de la asignatura.
- <Post> GuardarAsignaturaFase(): Guarda los títulos de las fases que el docente considere para la asignatura.
- <Post> ListarResultados(): Obtiene los resultados de aprendizaje (De conocimiento, de desempeño, de producto) que el docente ingresó para la asignatura.
- <Post> GuardarResultados(): Almacena los resultados de aprendizaje que el docente establecerá para la asignatura.
- <Post> ListarEstrategias(): Lista las estrategias configuradas para que el docente pueda elegir las requeridas para el desarrollo de la asignatura.
- <Post> ListarCriterios (): Lista los criterios configurados para que el docente pueda elegir los requeridos para el desarrollo de la asignatura.
- <Post> ListarInstrumentos (): Lista los instrumentos configurados para que el docente pueda elegir los requeridos para el desarrollo de la asignatura.
- <Post> ListarAsignaturaEstrategia(): Obtiene las estrategias que previamente el docente ingresó para la asignatura.

- <Post> GuardarAsignaturaEstrategia(): Guarda las estrategias ingresadas por los docentes.
- <Post> ListarAsignaturaCriterio(): Obtiene los criterios que previamente el docente ingresó para la asignatura.
- <Post> GuardarAsignaturaCriterio(): Guarda los criterios ingresados por los docentes.
- <Post> ListarAsignaturaInstrumento(): Lista los instrumentos que previamente el docente ingresó para la asignatura.
- <Post> ListarAsignaturaInstrumentoAgrupados(): Lista los instrumentos que previamente el docente ingresó para la asignatura pero de manera agrupada en caso de que algún valor se repita por estar en una fase diferente o asociado a algún resultado diferente.
- <Post> GuardarAsignaturaInstrumento(): Guarda los instrumentos ingresado por los docentes.

f) Evaluación de competencias adquiridas

Diagrama de clases de la sección de la bibliografía que es seleccionada para la asignatura detallada en la Figura 23.

Los métodos usados en esta sección son:

- <Post> ListarBibliografia (): Lista el contenido bibliográfico almacenado en la tabla de bibliografía en caso de que se utilice una base local.
- <Post> GuardarAsignaturaBibliografia (): Guarda la bibliografía seleccionada para la asignatura que se encuentra en desarrollo.
- <Post> ListarAsignaturaBibliografia(): Lista el contenido bibliográfico que fue seleccionado y guardado para la asignatura que se encuentra en desarrollo.

g) Aprobación, Transcripción, Filtros y Firma

Diagrama de clases para la solicitud de aprobación, momento de transcripción de un sílabo, listado de la bitácora y sus filtros; y los docentes que figuran en la sección de firma, detallada en la Figura 24.

Los métodos usados en esta sección son:

- <Post> FiltrosBitacora(): Obtiene los filtros para la bitácora generada en el desarrollo del sílabo. Esta bitácora comprende los docentes que realizaron modificaciones o conceptos que fueron desarrollados.
- <Post> ListarBitacora(): Lista la bitácora al día de desarrollo, esta bitácora está ordenada de la fecha más reciente a la fecha anterior.
- <Post> AprobarCurso(): Solicitud de aprobación de los docentes al terminar el desarrollo del sílabo y entre en un periodo de revisión por parte del jefe de departamento.
- <Post> ListarDocentesFirma(): Lista los docentes que aparecerán en el apartado de la firma. No se incluyen a los jefes de prácticas.
- <Post> ListarSilaboPeriodoAnterior(): Investiga el desarrollo de la asignatura de años pasados, obteniendo el sílabo más reciente.
- <Post> TranscribirAsignatura (): Servicio que transcribe el contenido de un sílabo anterior al sílabo que se encuentra en desarrollo. Esto permite tener una base en la planificación del nuevo sílabo.

C) Controlador Programa Formativo

Para el controlador del programa formativo se tiene el diagrama de clases en la Figura 25.

Para esta sección los métodos son los siguientes:

- <Post> ListarProyectosAsignatura(): Obtiene la lista de los proyectos ingresados para la asignatura
- <Post> GuardarProyectosAsignatura(): Guarda los proyectos designados por el docente para las fases de la asignatura.
- <Post> ListarProgramaPorFaseDeAprendizaje(): Obtiene la lista de los programas para cada fase.
- <Post> ListarProgramacionPorFaseDeAprendizaje(): Obtiene toda la información para la matriz de la programación por fase de aprendizaje.

Diagramas de clase:

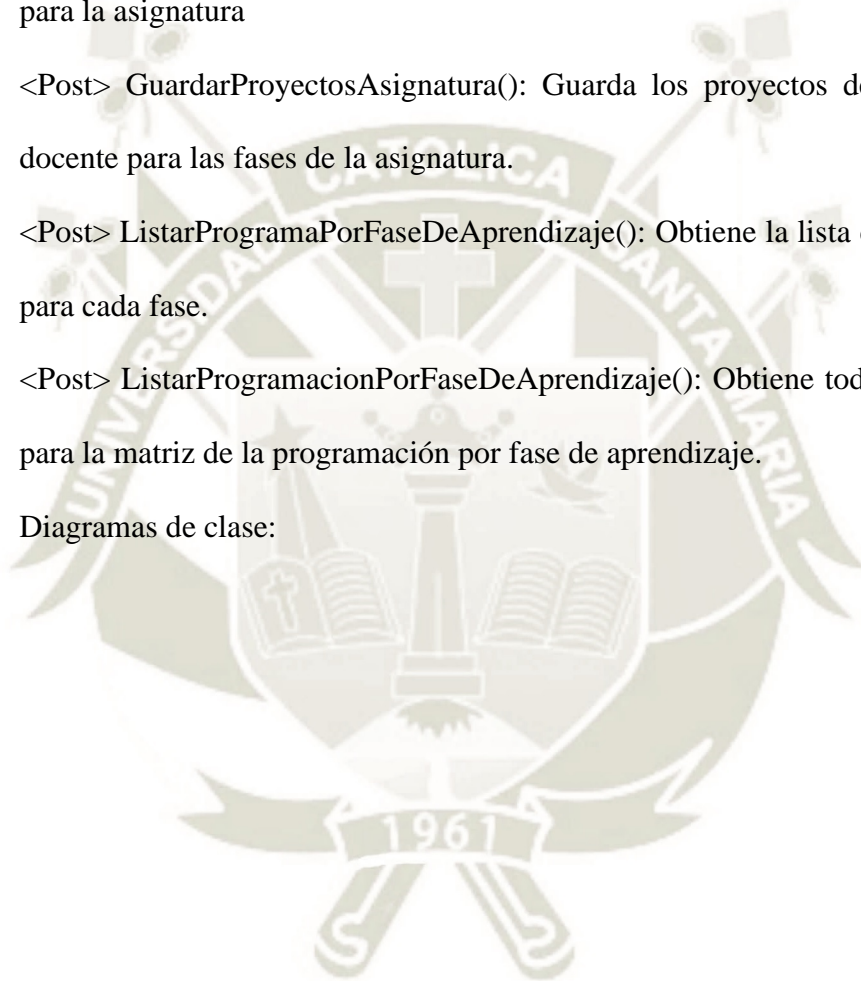
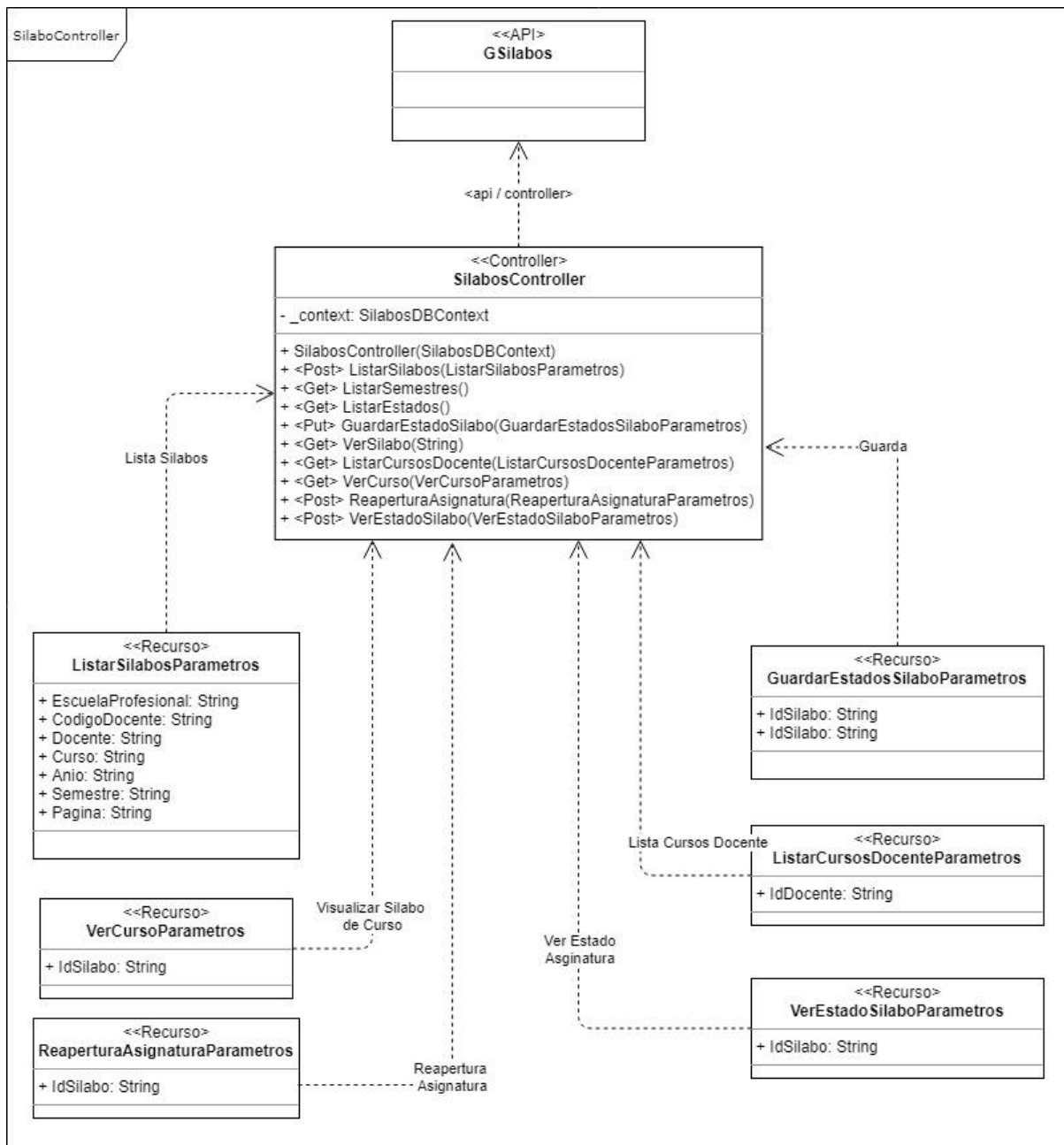
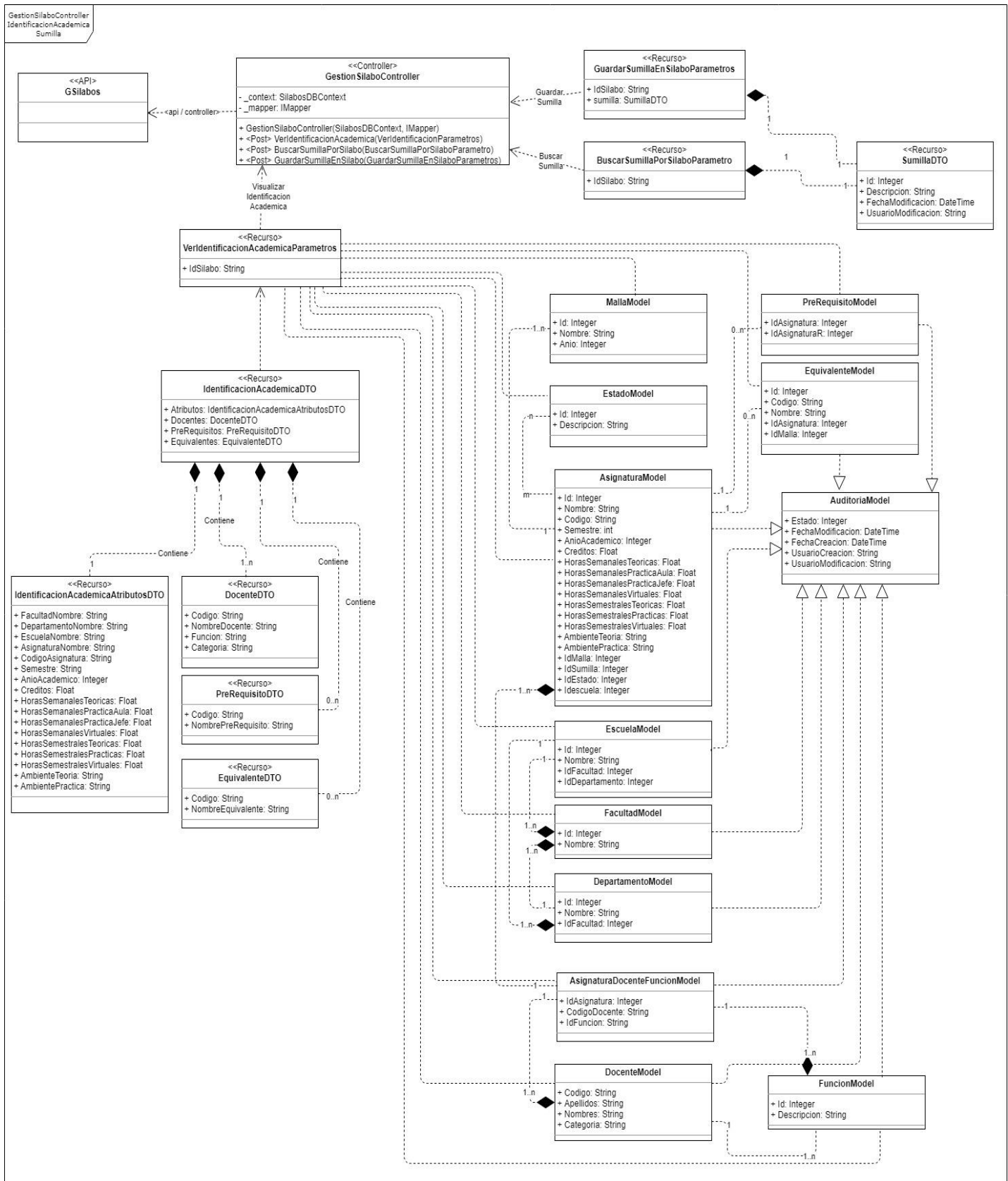


Figura 17 Controlador Silabo



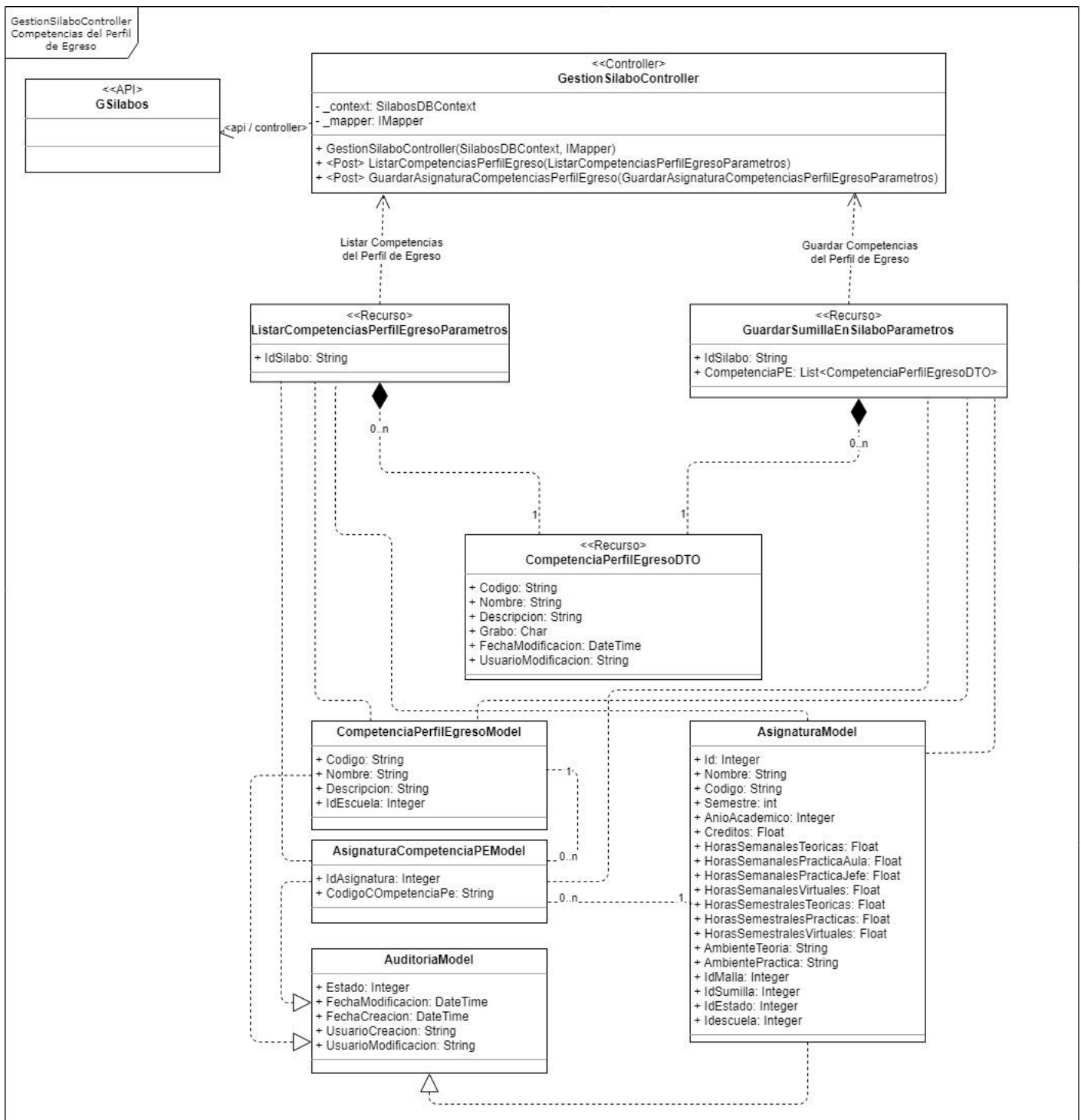
Fuente: Elaboración propia.

Figura 18 Diagrama de Clases de Identificación Académica y Sumilla



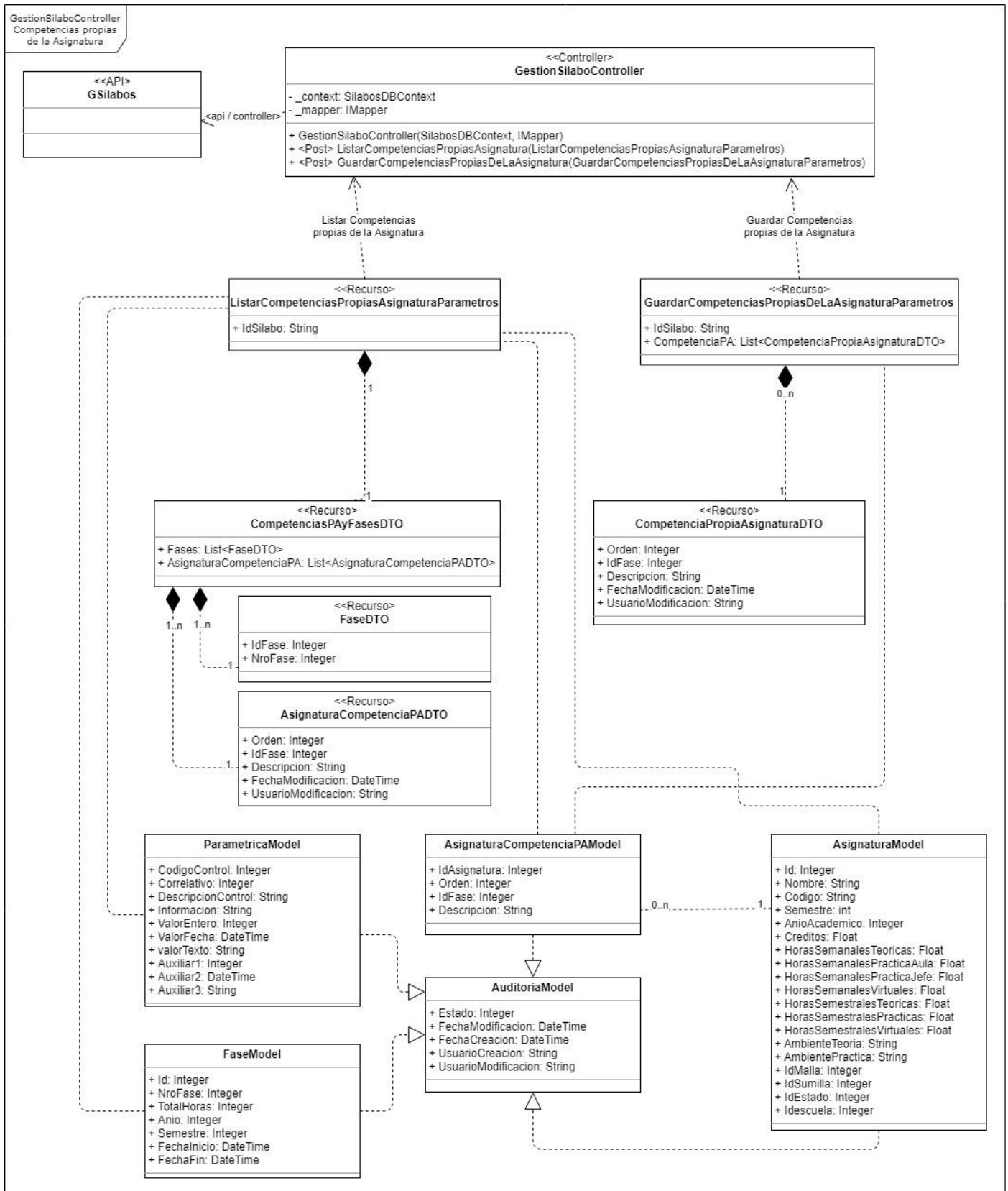
Fuente: Elaboración propia.

Figura 19 Diagrama de clases de Competencias del Perfil de Egreso



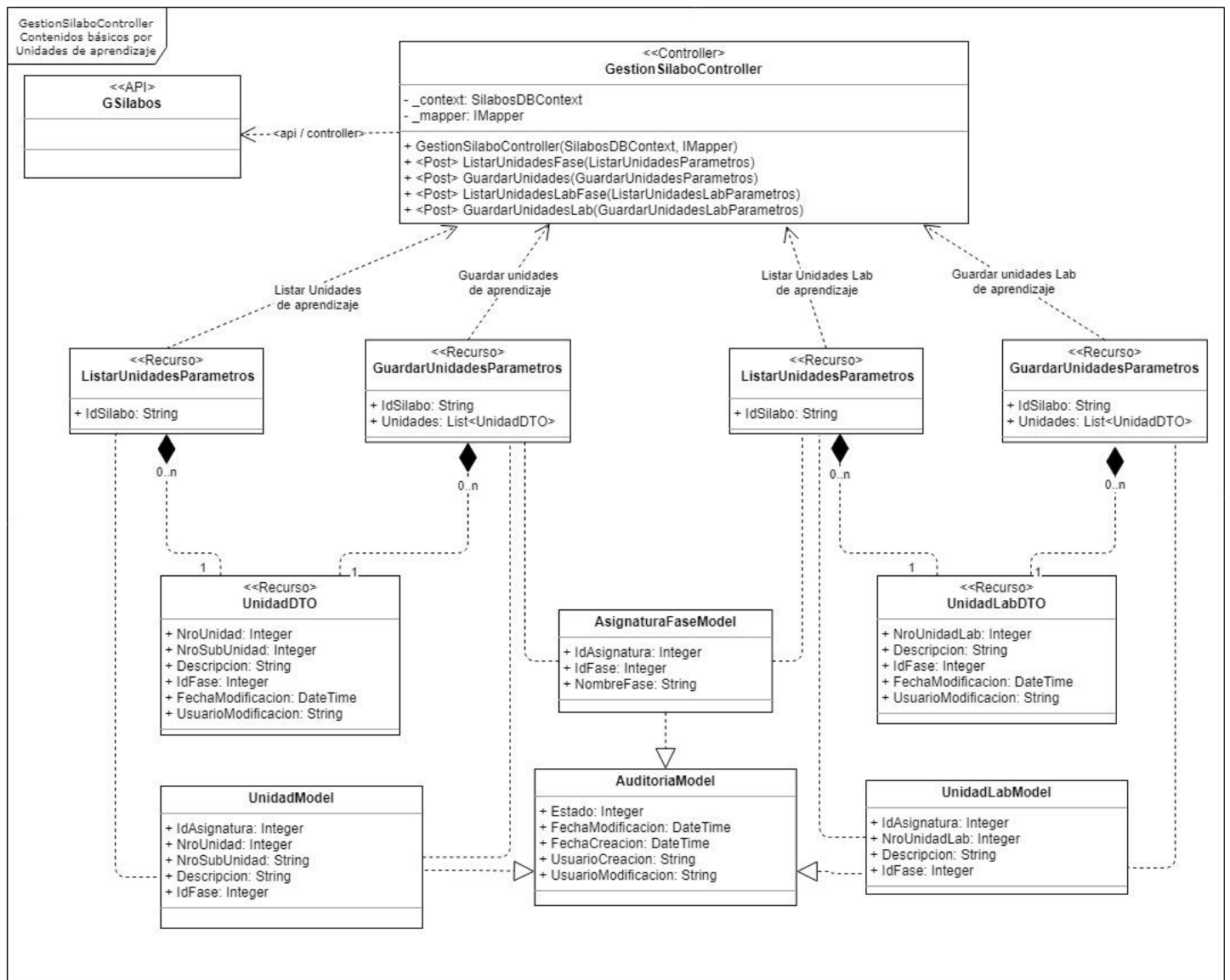
Fuente: Elaboración propia.

Figura 20 Diagrama de clases de Competencias propias de la asignatura



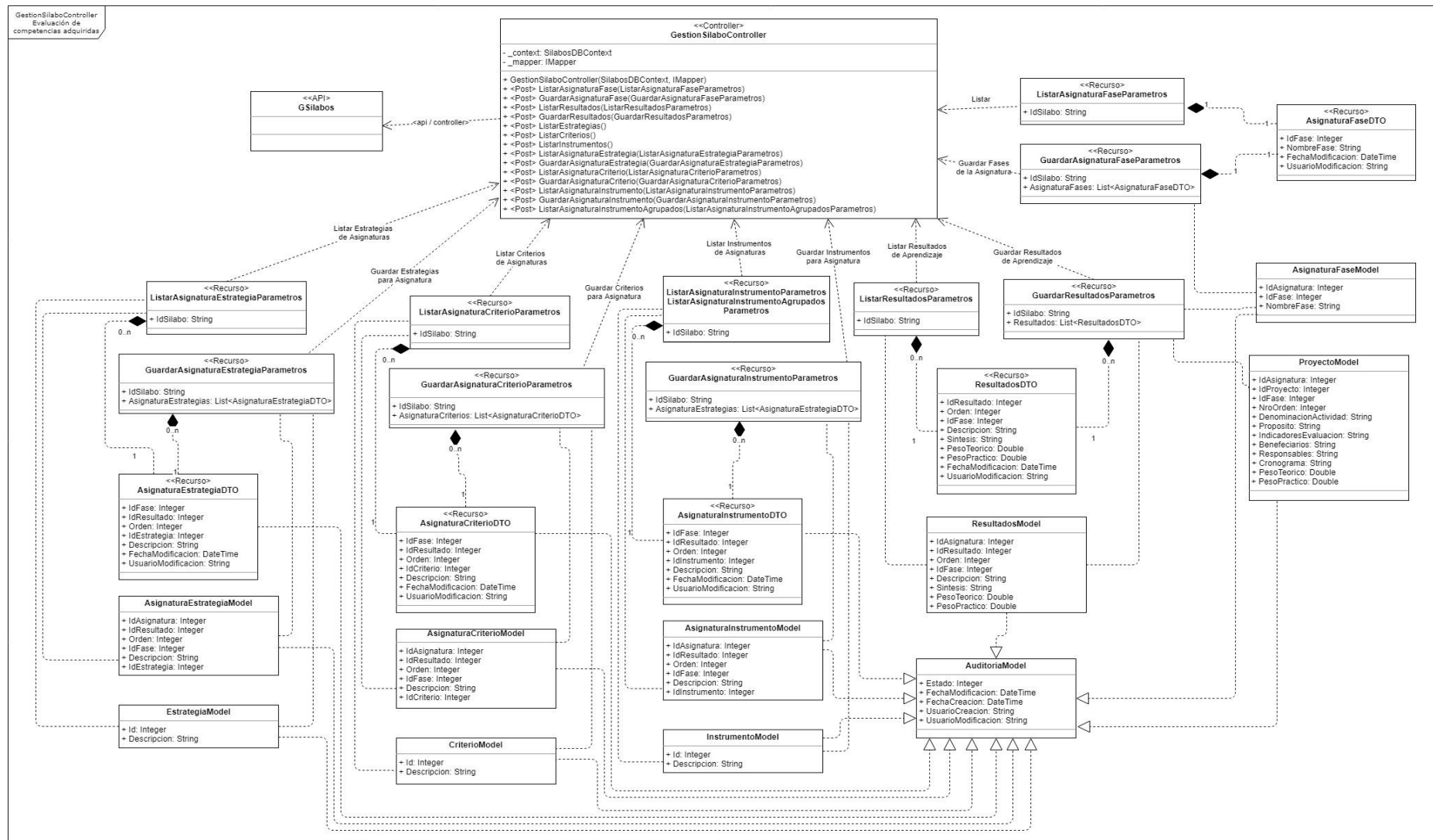
Fuente: Elaboración propia.

Figura 21 Diagrama de clases de los contenidos básicos por unidades de aprendizaje de la asignatura



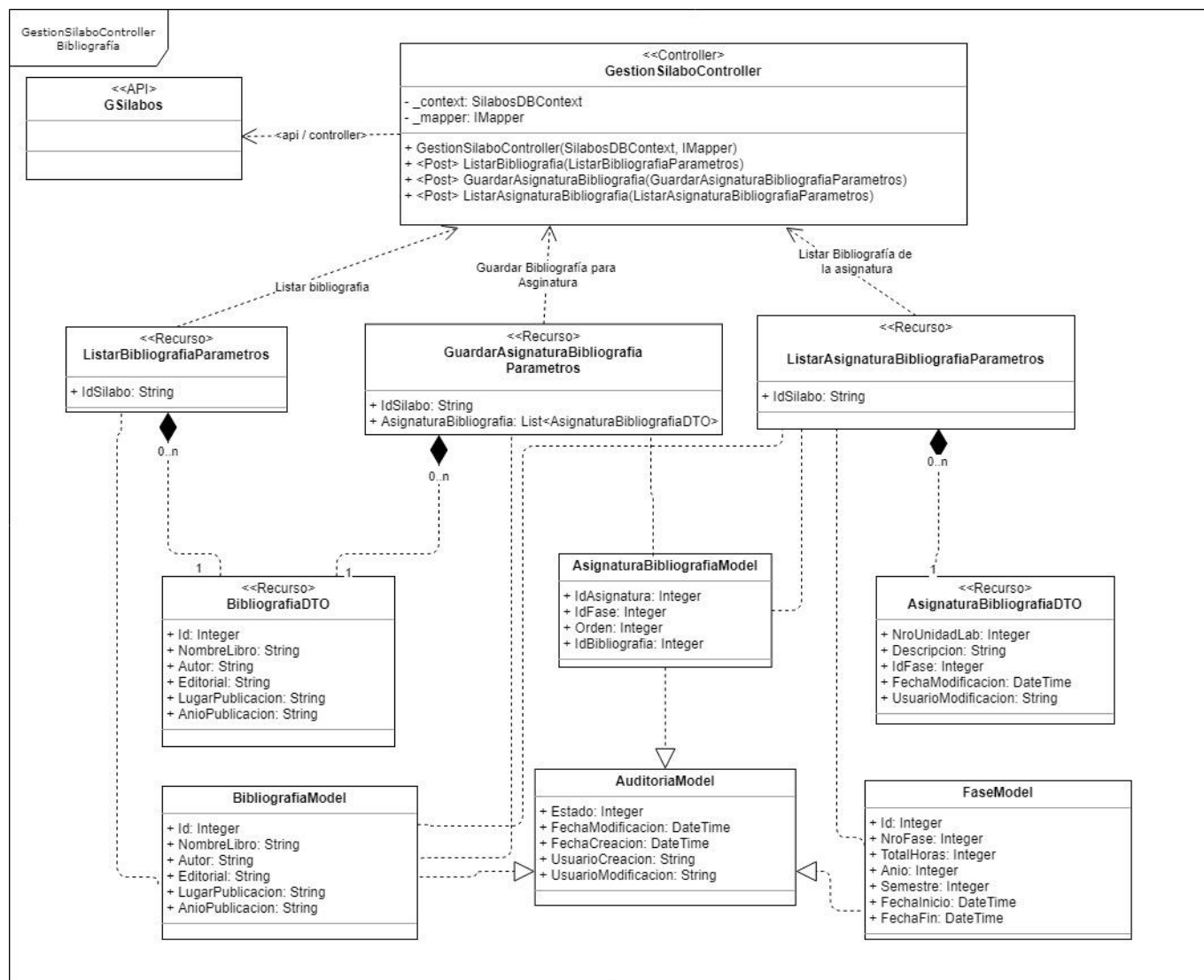
Fuente: Elaboración propia.

Figura 22 Diagrama de clases de la evaluación de competencias adquiridas



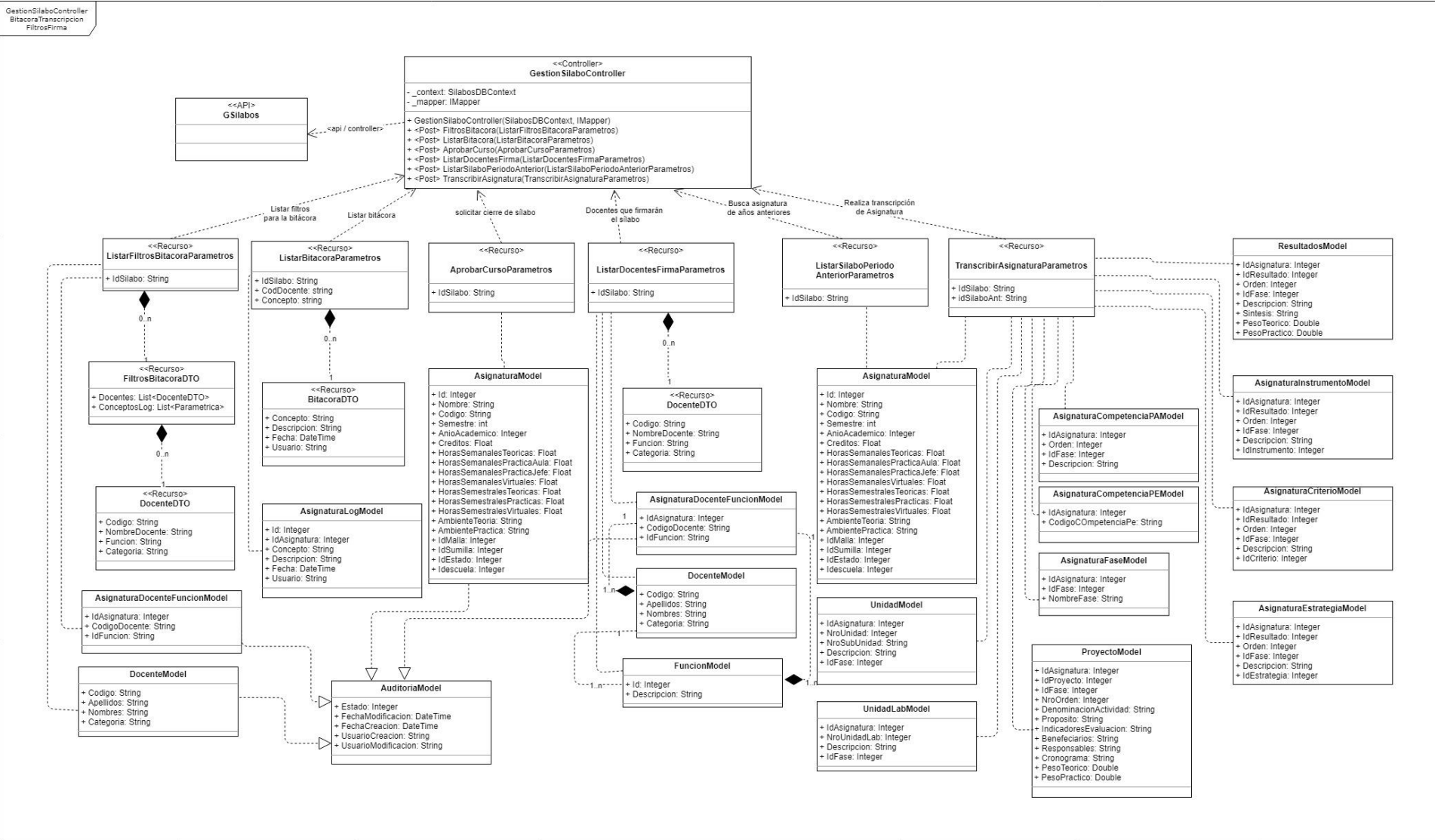
Fuente: Elaboración propia.

Figura 23 Diagrama de Clases Bibliografía



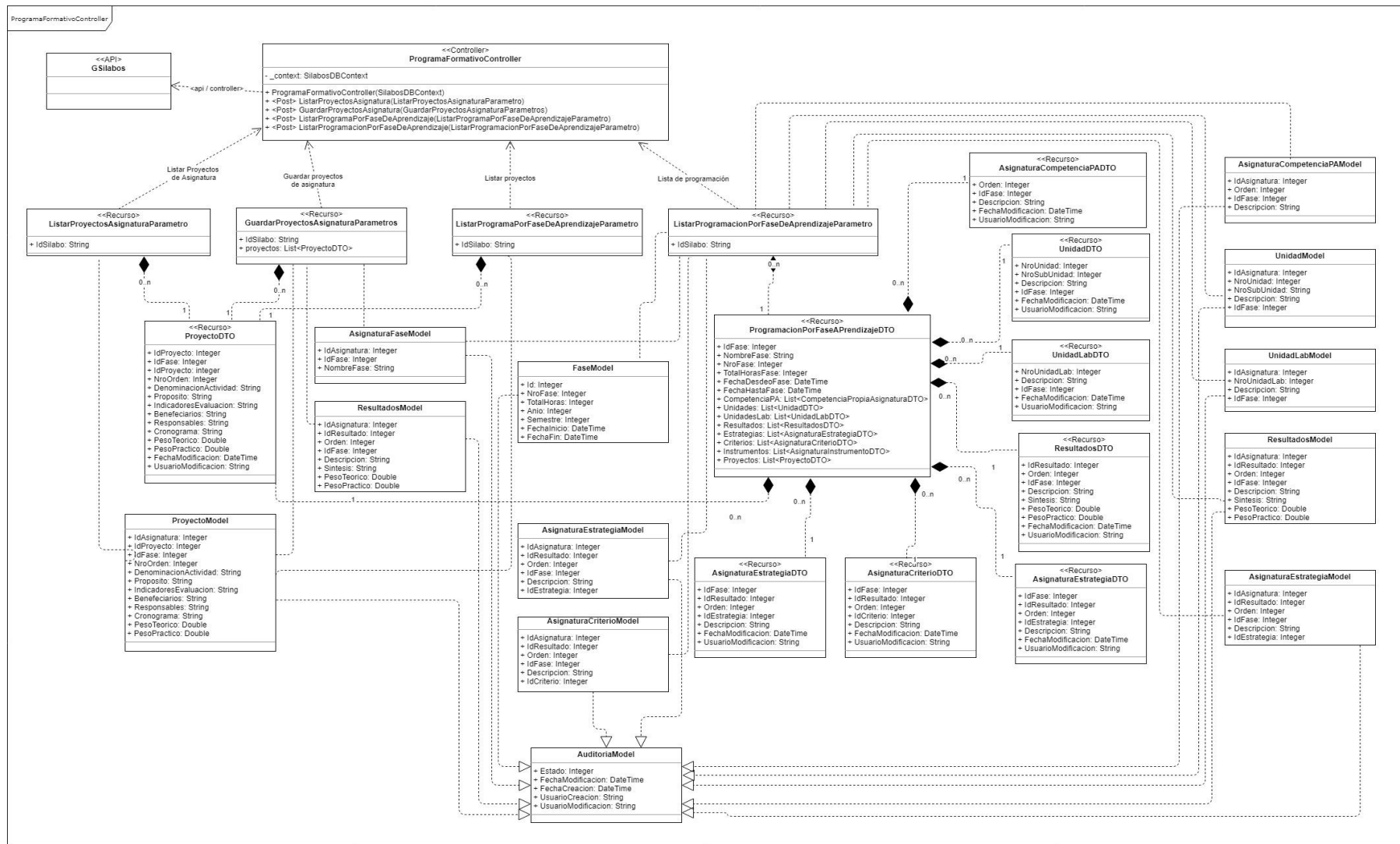
Fuente: Elaboración propia.

Figura 24 Diagrama de clases Bitácora, Transcripción, Firma



Fuente: Elaboración propia.

Figura 25 Diagrama de clases del programa formativo



Fuente: Elaboración propia.

2.3.3.2 Diagrama de Componentes.

Para el sistema de gestión de sílabos, se establecen los componentes que sirven para el correcto funcionamiento del sistema como se aprecia en la Figura 26. Se divide la elaboración del sistema en los elementos Front-End, Back-End y Base de datos, y se detalla los componentes que estarán conformándolos:

A) Front-End

- a) Componente para docente
- b) Jefe de departamento
- c) Administrador

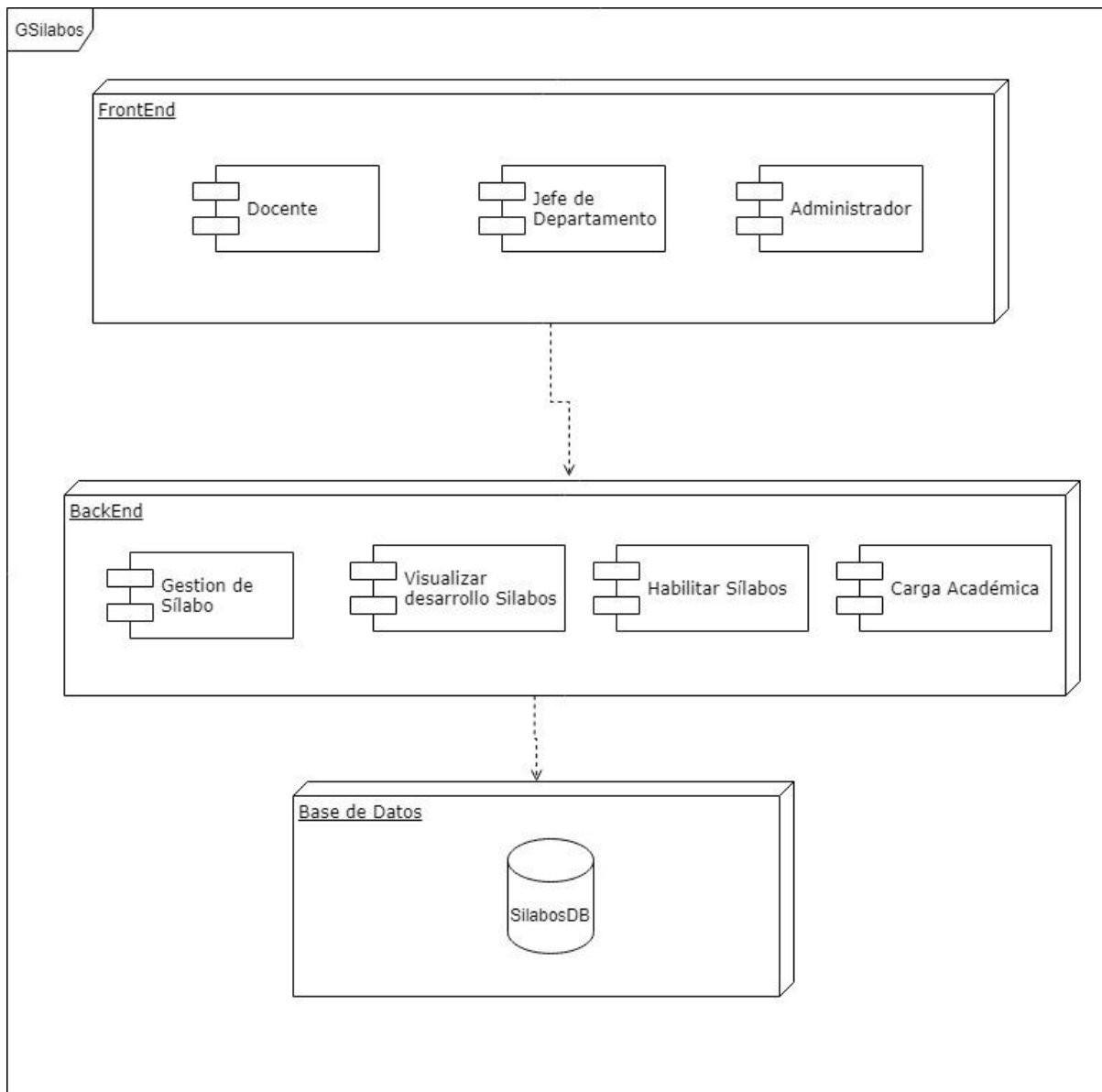
B) Back-End - API

- a) Gestión de sílabos
- b) Visualizar desarrollo de sílabos
- c) Habilitar sílabos
- d) Carga Académica

C) Base de Datos

- a) Gestor de base de datos

Figura 26 Diagrama de Componentes General



Fuente: Elaboración propia.

2.3.4 Diagrama Dinámico

El modelo dinámico del sistema representa los aspectos del sistema de gestión de sílabos que se relacionan con tiempo y cambios en los objetos que lo conforman, además de sus relaciones. Para esto se formulan los diagramas de: Casos de Uso, Estado, Actividad y Secuencia.

Los diagramas expuestos, permitirán explicar la interacción dinámica en los elementos del sistema.

2.3.4.1 Diagramas de Estados.

En el diagrama de estados, se visualiza la situación en de la vida de un objeto. Esta situación se le denomina “estado”, que es alcanzado a través de alguna condición, o después de haberse realizado alguna actividad o evento.

A) Objeto Visualizar Sílabo

La secuencia de ingreso del usuario docente a la pantalla Visualizar Sílabo comprende los estados expuestos en el diagrama de la Figura 27.

B) Objeto Habilitar Sílabos

La secuencia de ingreso del usuario docente a la pantalla Habilitar Sílabo comprende los estados expuestos en el diagrama de la Figura 28.

Se aprecia además la aprobación o rechazo del sílabo como estados posibles ante la solicitud de revisión por parte del docente.

C) Objetos en Controlador Gestión Silabo

a) Desarrollo de sílabo general en el plan de estudio

La secuencia de ingreso del usuario docente a la pantalla Desarrollo de Sílabo comprende los estados expuestos en el diagrama de la Figura 19.

b) Desarrollo de sílabo carga de plan de estudio

La secuencia de ingreso del usuario docente al desarrollo específico del Sílabo en cada una de sus secciones comprende los estados expuestos en el diagrama de la Figura 30.

La carga de información sucede al momento del ingreso al plan de estudio, lo que significa que se lista cada valor posible ingresado del sílabo hasta el momento. Por otro lado, al iniciar o

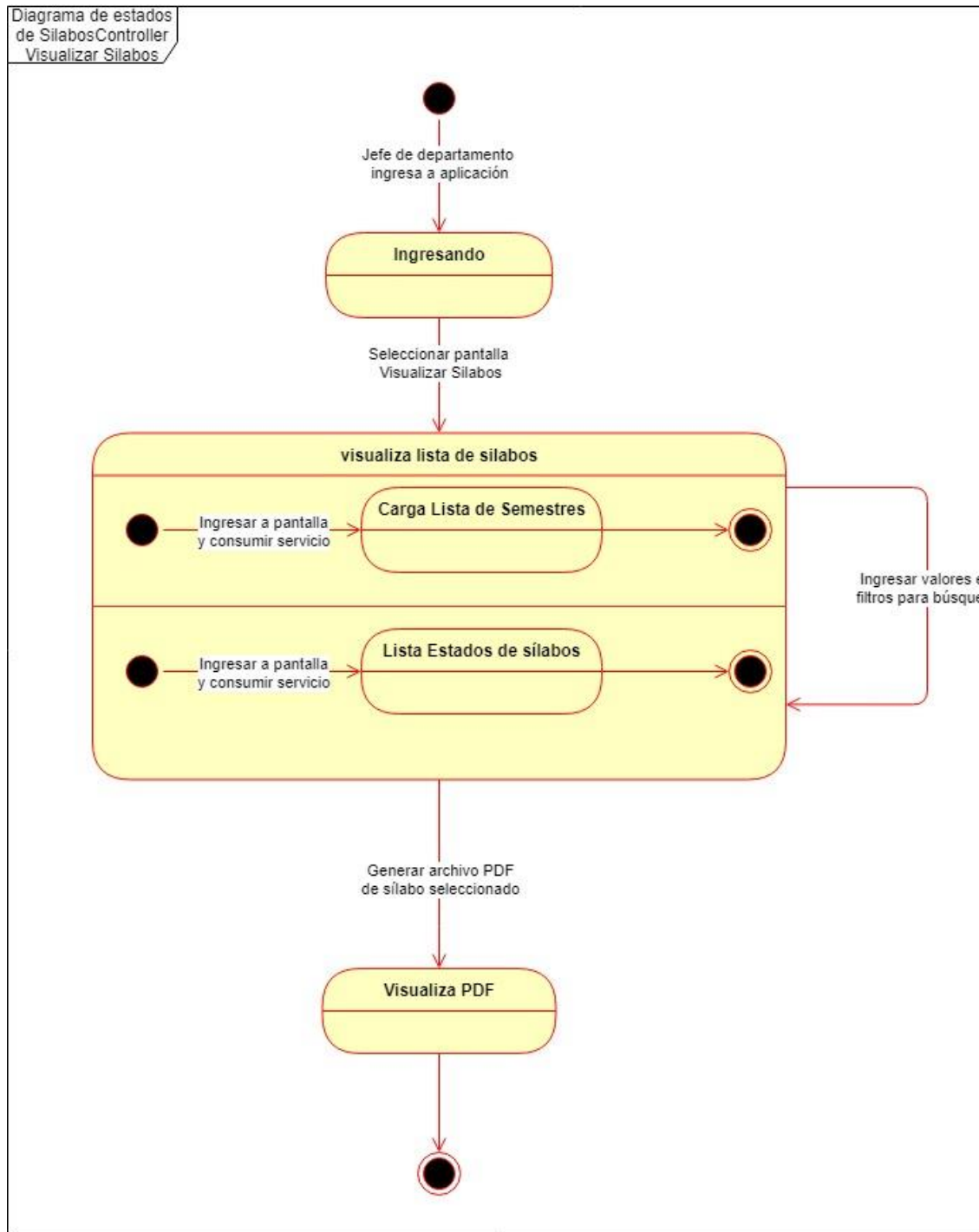
continuar el desarrollo del sílabo los valores nuevos valores irán insertándose progresivamente hasta la culminación y generación del documento PDF del sílabo.

D) Objeto Programa Formativo

La secuencia de ingreso del usuario docente al desarrollo del programa formativo, este comprende los estados expuestos en el diagrama de la Figura 31.

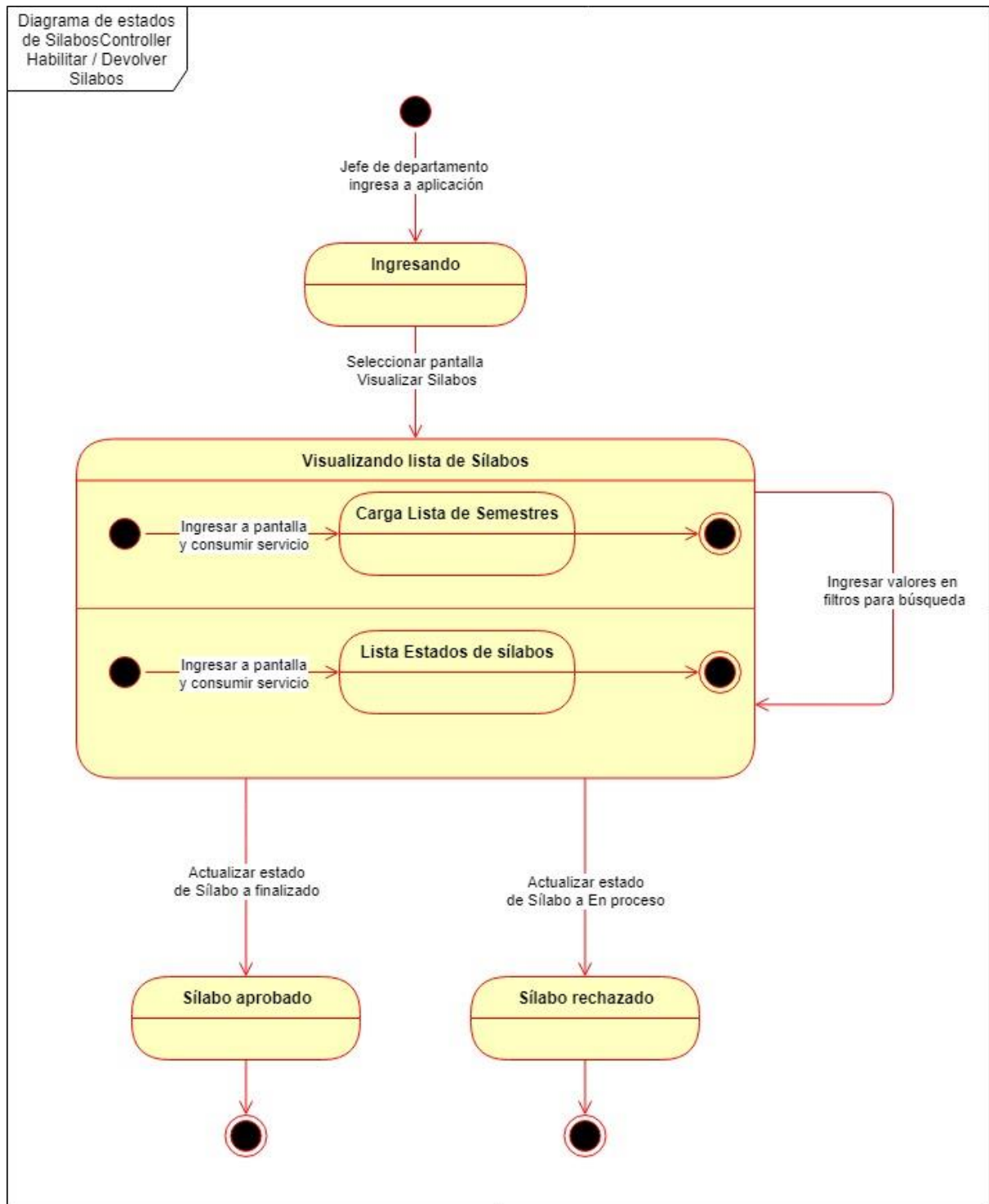


Figura 27 Diagrama de Estados - Visualizar Sílabos



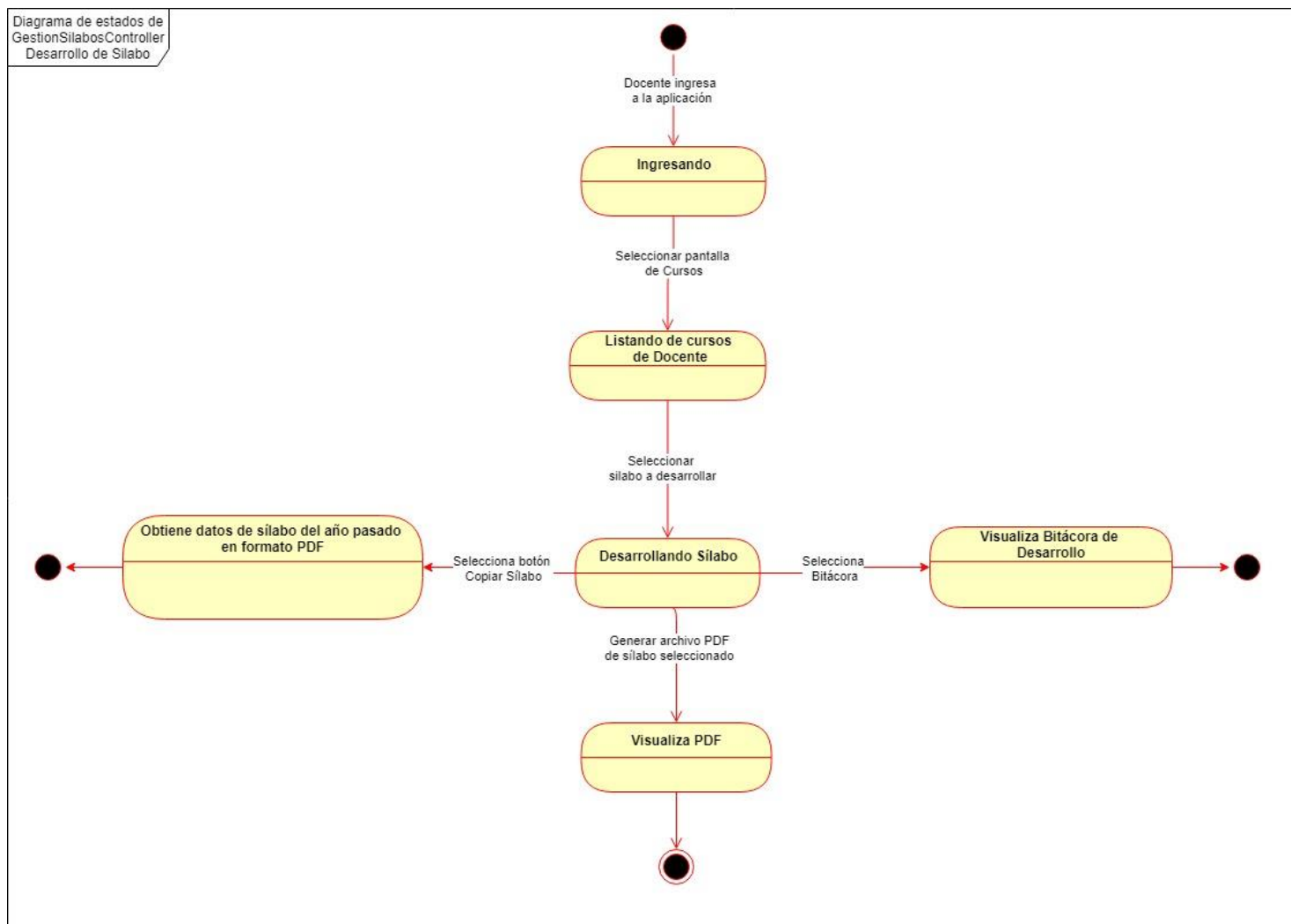
Fuente: Elaboración propia.

Figura 28 Diagrama de Estados - Habilitar Sílabos



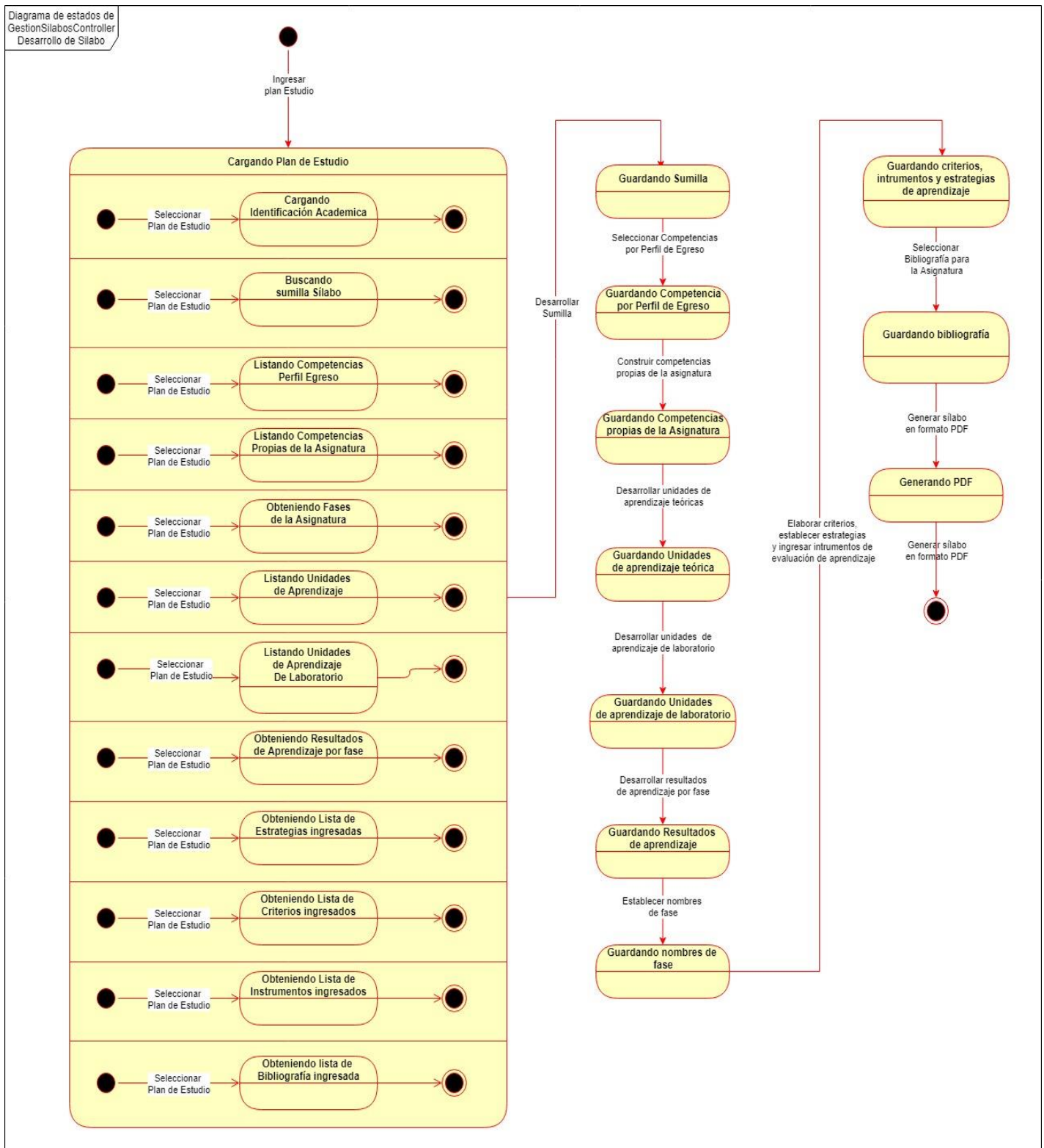
Fuente: Elaboración propia.

Figura 29 Diagrama de Estados - Desarrollo de Sílabo



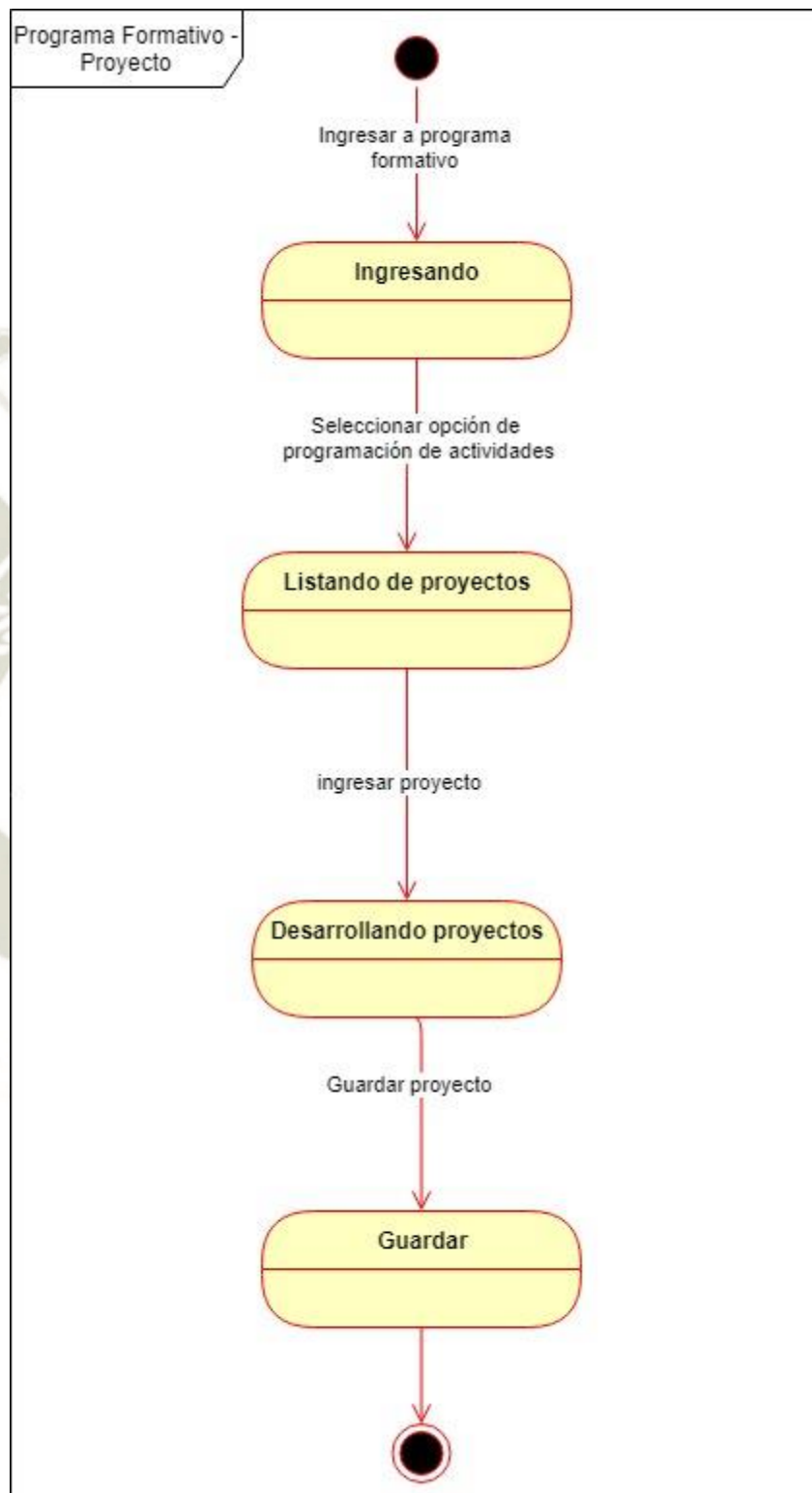
Fuente: Elaboración propia.

Figura 30 Diagrama de Estados - Desarrollo de Sílabo Plan de Estudio



Fuente: Elaboración propia.

Figura 31 Diagrama de Estados - Programa Formativo



Fuente: Elaboración propia.

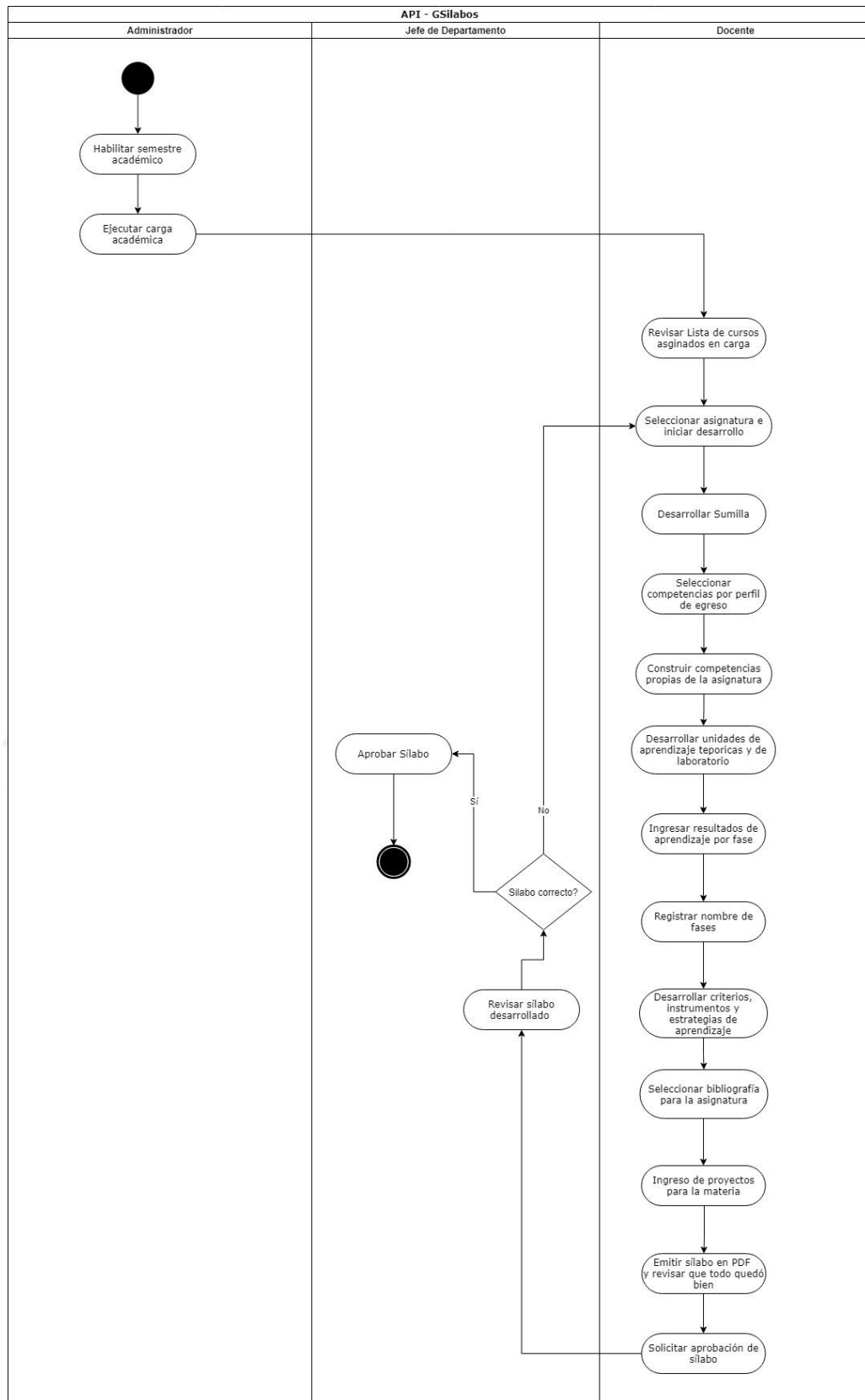
2.3.4.2 Diagrama de Actividad.

El diagrama de actividad nos permite tener el panorama del comportamiento del flujo de control o del flujo de objetos. Este diagrama muestra una secuencia de tareas o acciones que indican el trabajo realizado por el negocio de principio a fin.

Se presenta el diagrama de actividad de acuerdo con la secuencia de tareas en las que se llevaría a cabo el funcionamiento del Sistema gestión de Sílabos con la participación de los roles involucrados en la Figura 32.



Figura 32 Diagrama de Actividad - Sistema Gestión de Sílabos



Fuente: Elaboración propia.

2.3.4.3 Diagramas de Secuencia.

El diagrama de secuencia permite conocer la interacción entre los diversos componentes u objetos que tienen entre sí a través de los métodos o servicios para completar el proceso de desarrollo. Se detalla sobre los diversos casos de uso presentados anteriormente y tomando como apoyo las secciones del sílabo.

Los componentes que participan a partir del usuario que ingresa al Sistema de gestión de sílabos son:

- Interfaz de usuario – Front-End
- Controlador – Back-End
- Entidades – Modelo de Base de Datos

A) Controlador Sílabo

Se muestra la interacción del controlador encargado de las pantallas utilizadas por el jefe de departamento, administrador y la lista de cursos del docente.

a) Pantalla Visualizar Sílabo

Interfaz que verifica la información de los sílabos desarrollados por el sistema de gestión de sílabos. Se aprecia la invocación de los métodos para interfaz de usuario y servicio que sirve para listar la información, esto descrito en la Figura 33.

b) Pantalla Habilitar Sílabo

Interfaz que verifica el estado de los sílabos desarrollados por el sistema de gestión de sílabos. Se aprecia la invocación de los métodos para interfaz de usuario y servicio que sirve para la aprobación o devolución del sílabo para corrección del docente, esto se visualiza en la Figura 34.

B) Controlador Gestión Silabo

Se muestra la interacción del controlador encargado de las pantallas utilizadas por el docente para el desarrollo del sílabo.

a) Identificación Académica y Sumilla

Interfaz que verifica la información de la identificación académica mostrada en el plan de estudio, así como en el programa formativo. Se aprecia la invocación de los métodos para interfaz de usuario y servicio que sirve para listar la información en la Figura 35.

b) Competencias por Perfil de Egreso

Interfaz que gestiona la información de las competencias por perfil de egreso en el plan de estudio. Se aprecia la invocación de los métodos para interfaz de usuario y servicio que sirve para listar y almacenar la información en la Figura 36.

c) Competencias propias de la Asignatura

Interfaz que gestiona la información de las competencias propias de la asignatura en el plan de estudio. Se aprecia la invocación de los métodos para interfaz de usuario y servicio que sirve para listar y almacenar la información en la Figura 37.

d) Unidades de aprendizaje

Interfaz que gestiona la información de las unidades de aprendizaje teóricas o de laboratorio en el plan de estudio. Se aprecia la invocación de los métodos para interfaz de usuario y servicio que sirve para listar y almacenar la información en la Figura 38.

e) Resultados de Aprendizaje

Interfaz que gestiona la información resultados de aprendizaje en el plan de estudio. Se aprecia la invocación de los métodos para interfaz de usuario y servicio que sirve para listar y almacenar la información en la Figura 39.

f) Estrategias, Criterios e Instrumentos

Interfaz que gestiona la información de las estrategias, criterios e instrumentos en el plan de estudio. Se aprecia la invocación de los métodos para interfaz de usuario y servicio que sirve para listar y almacenar la información en la Figura 40.

g) Bibliografía

Interfaz que gestiona la información de las bibliografías seleccionadas para asociar a la asignatura. Se aprecia la invocación de los métodos que listan y guardan la información en la Figura 41.

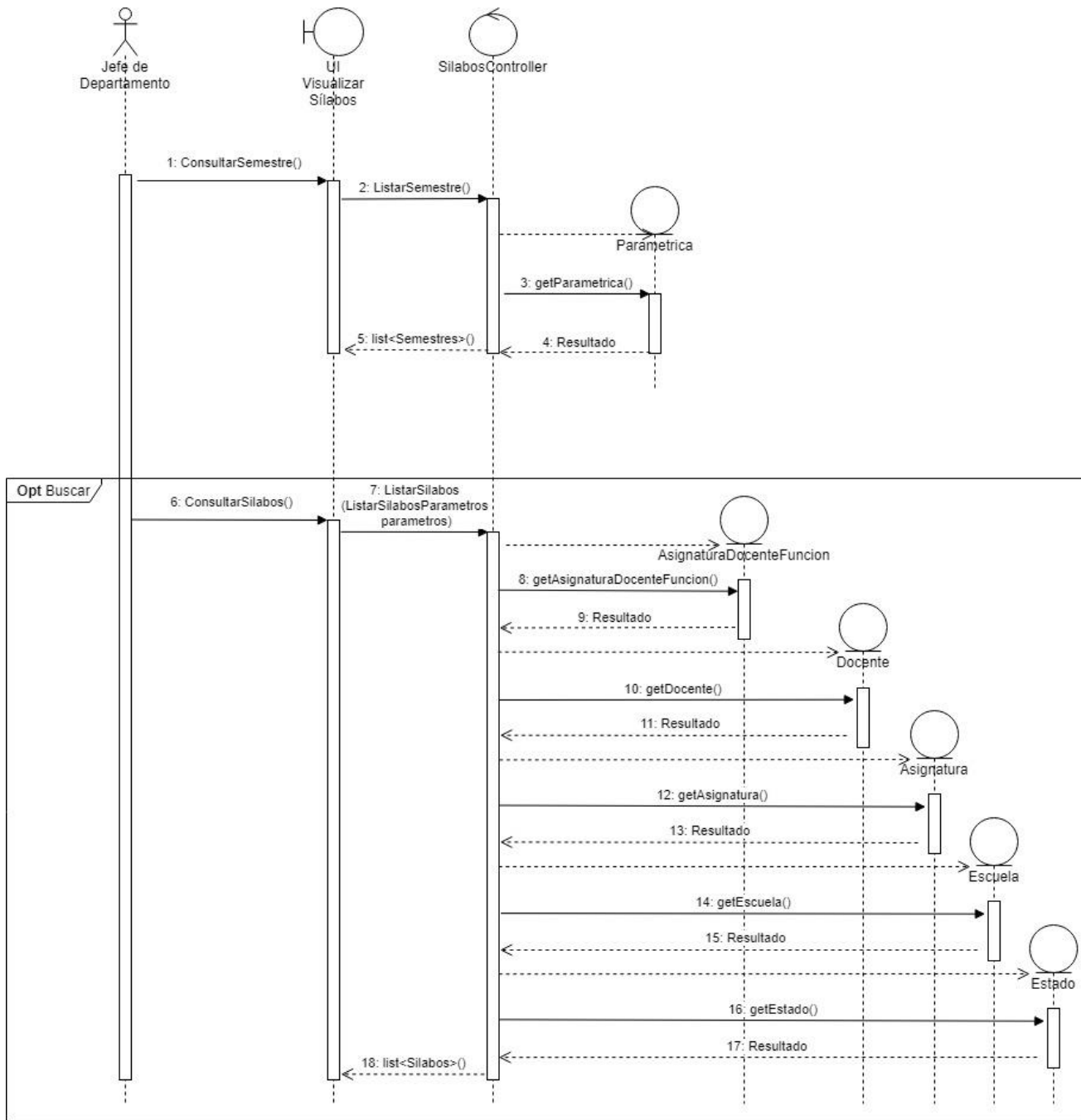
h) Transcripción, Bitácora y Firma

Interfaz que resuelve la información para la transcripción de un sílabo anterior a otro, además de mostrar la información almacenada en la bitácora y los docentes que figuran en la firma. Se aprecian los métodos que listan y guardan información en la Figura 42.

C) Controlador Programa Formativo

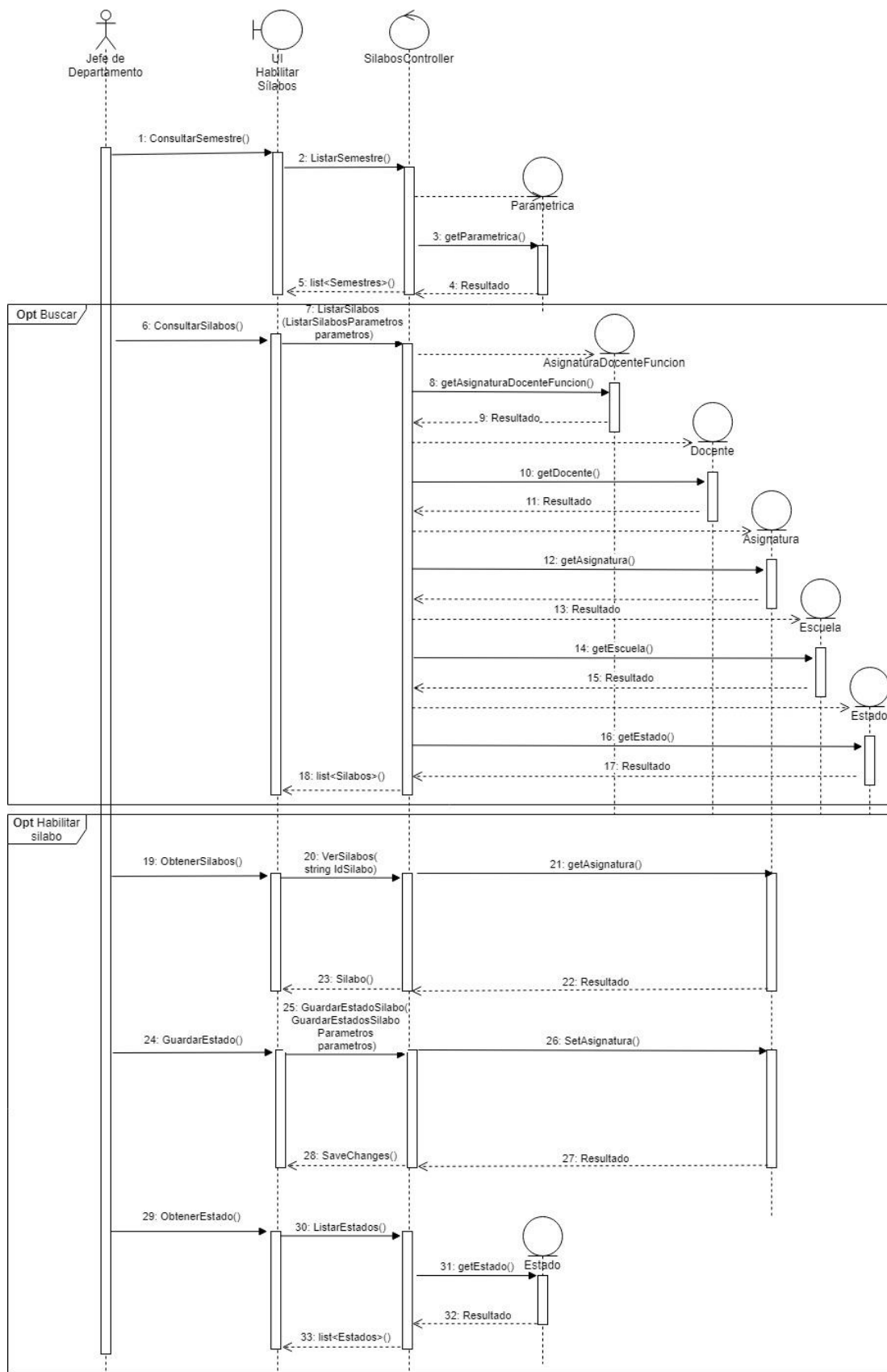
Interfaz que gestiona la información del programa formativo. Se aprecia la invocación de los métodos para interfaz de usuario y servicio que sirve para listar y almacenar la información en la Figura 43.

Figura 33 Diagrama de Secuencia - Visualizar Sílabo



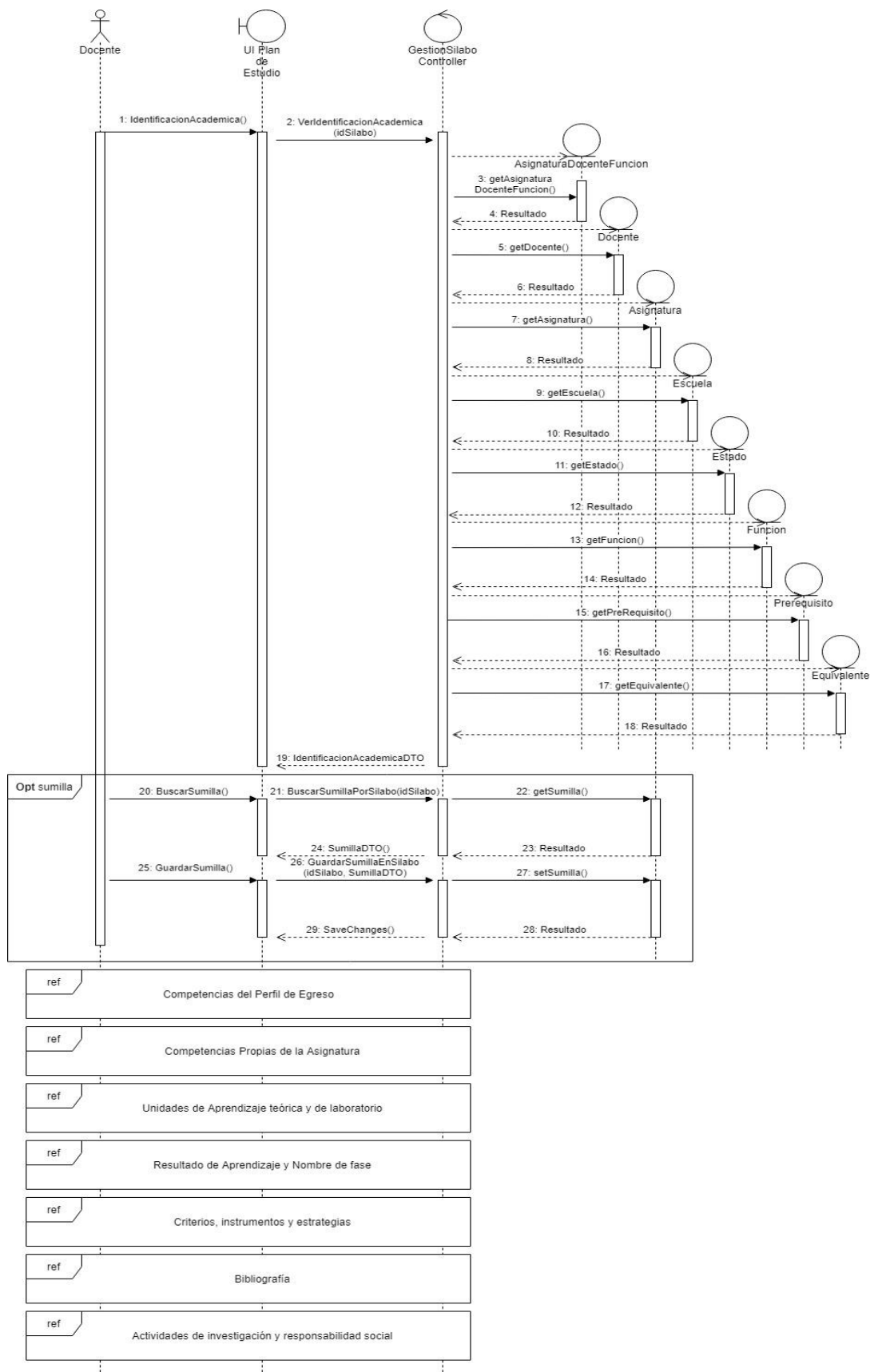
Fuente: Elaboración propia.

Figura 34 Diagrama de Secuencia - Habilitar Sílabo



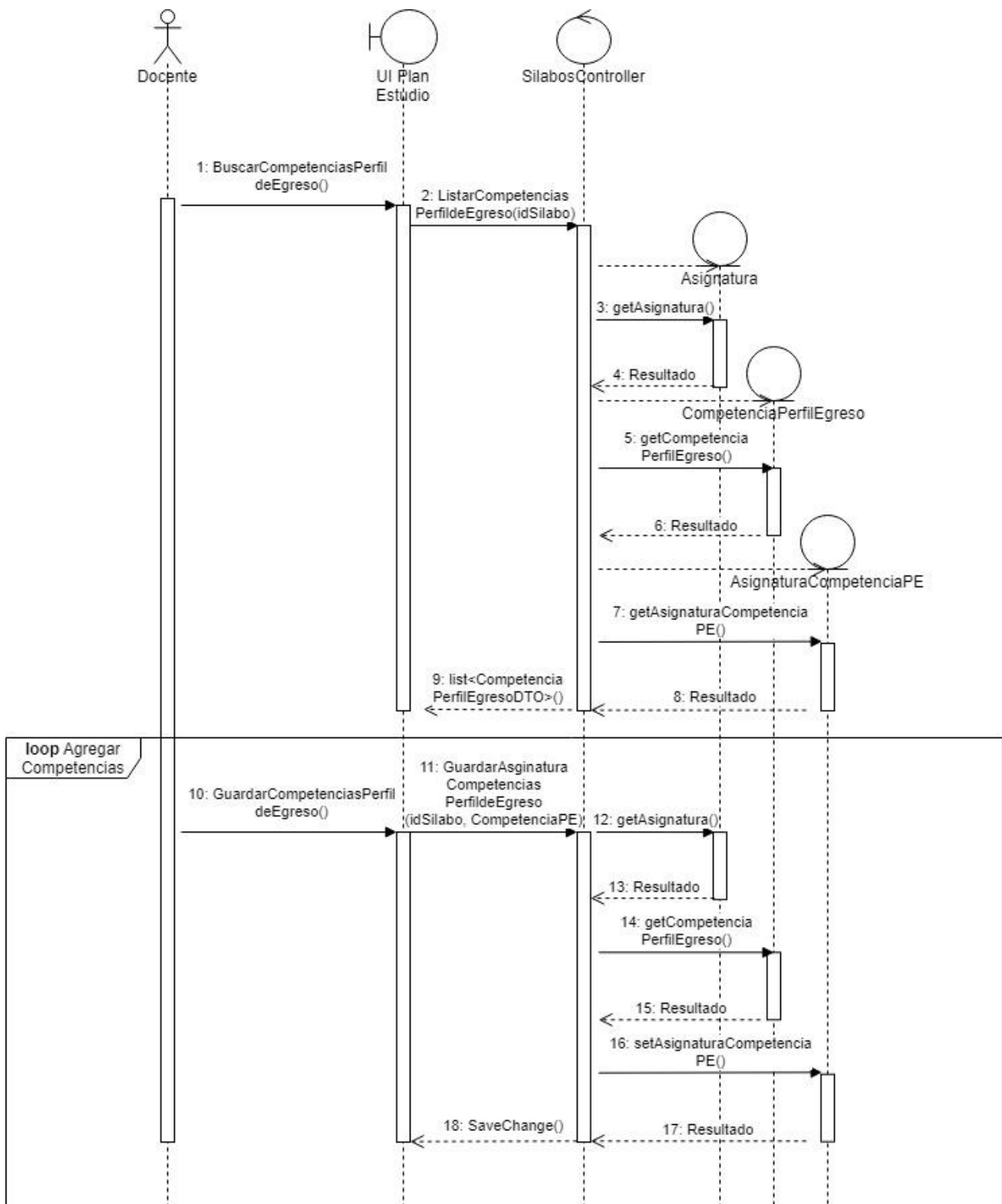
Fuente: Elaboración propia.

Figura 35 Diagrama de Secuencia - Identificación Académica y Sumilla



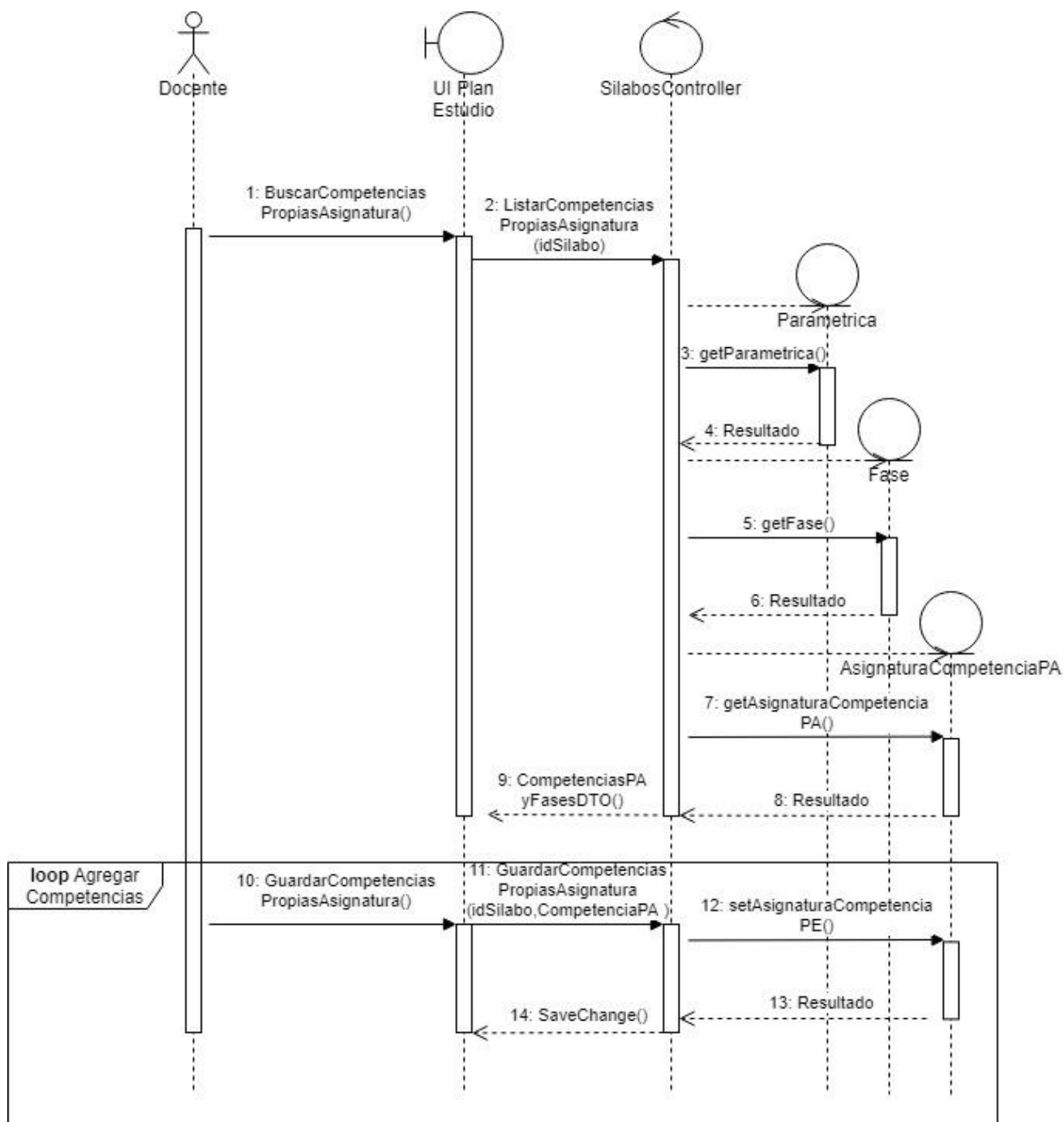
Fuente: Elaboración propia.

Figura 36 Diagrama de Secuencia - Competencia por Perfil de Egreso



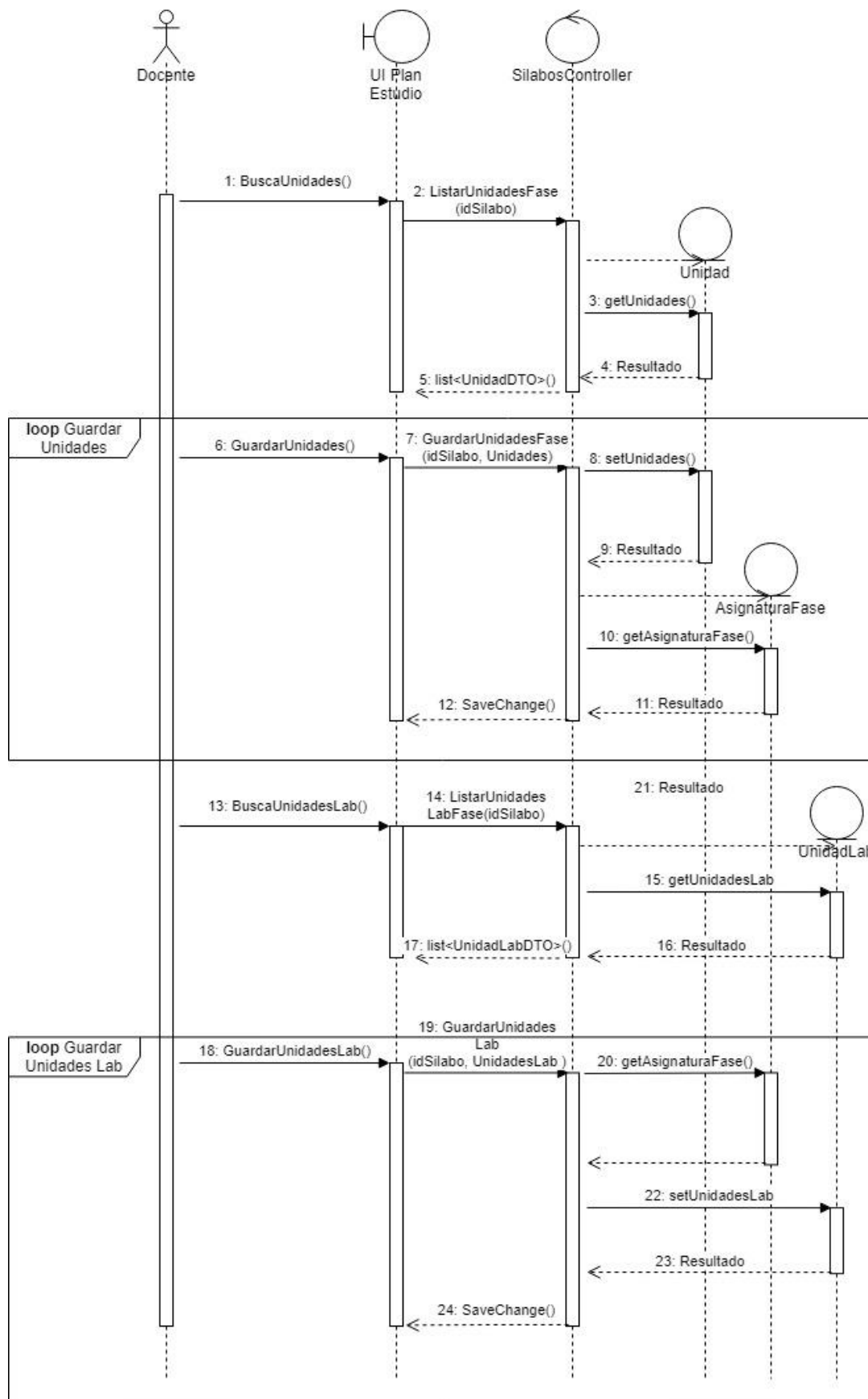
Fuente: Elaboración propia.

Figura 37 Diagrama de Secuencia - Competencias propias de la asignatura



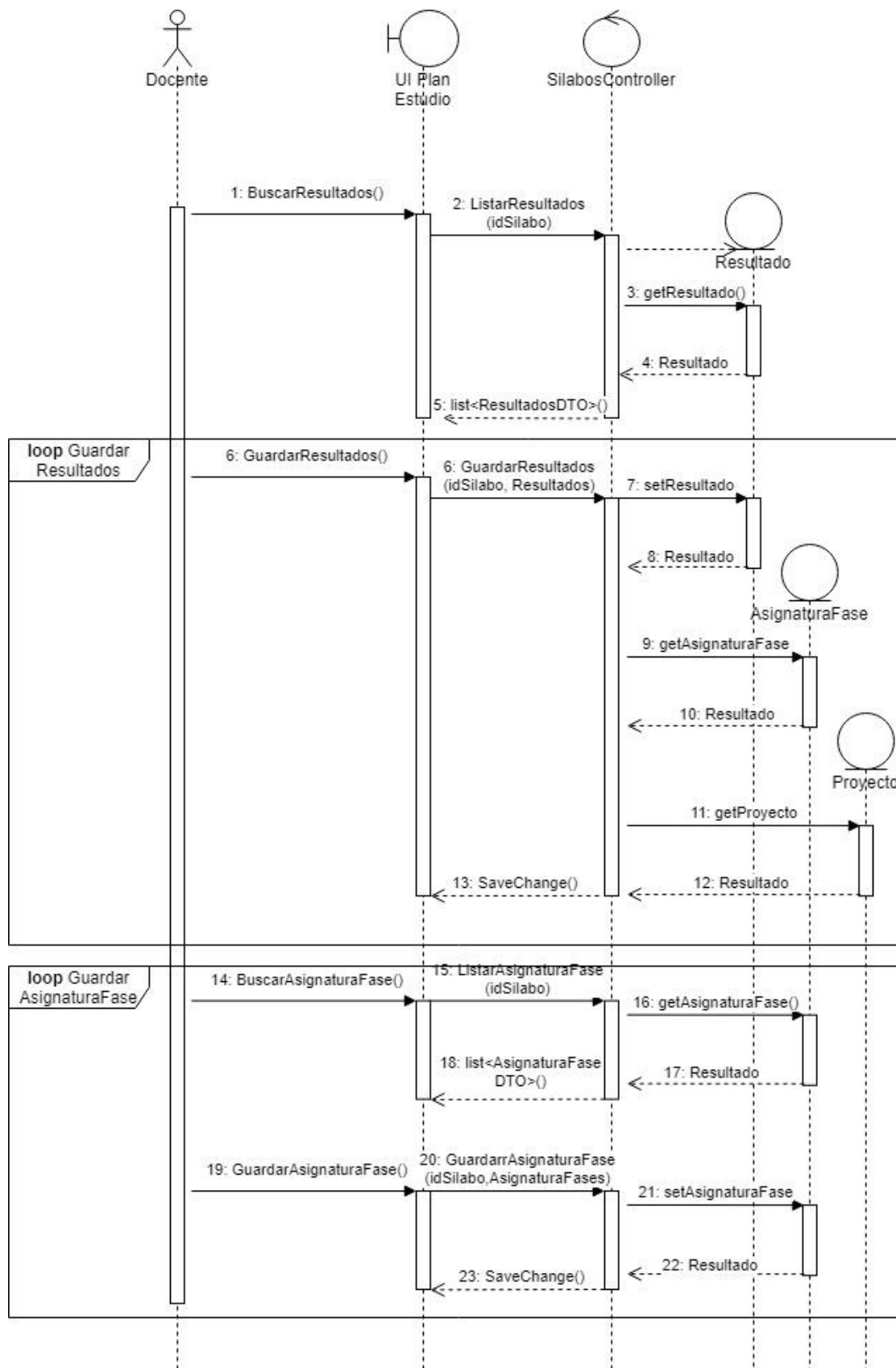
Fuente: Elaboración propia.

Figura 38 Diagrama de Secuencia - Unidades de Aprendizaje



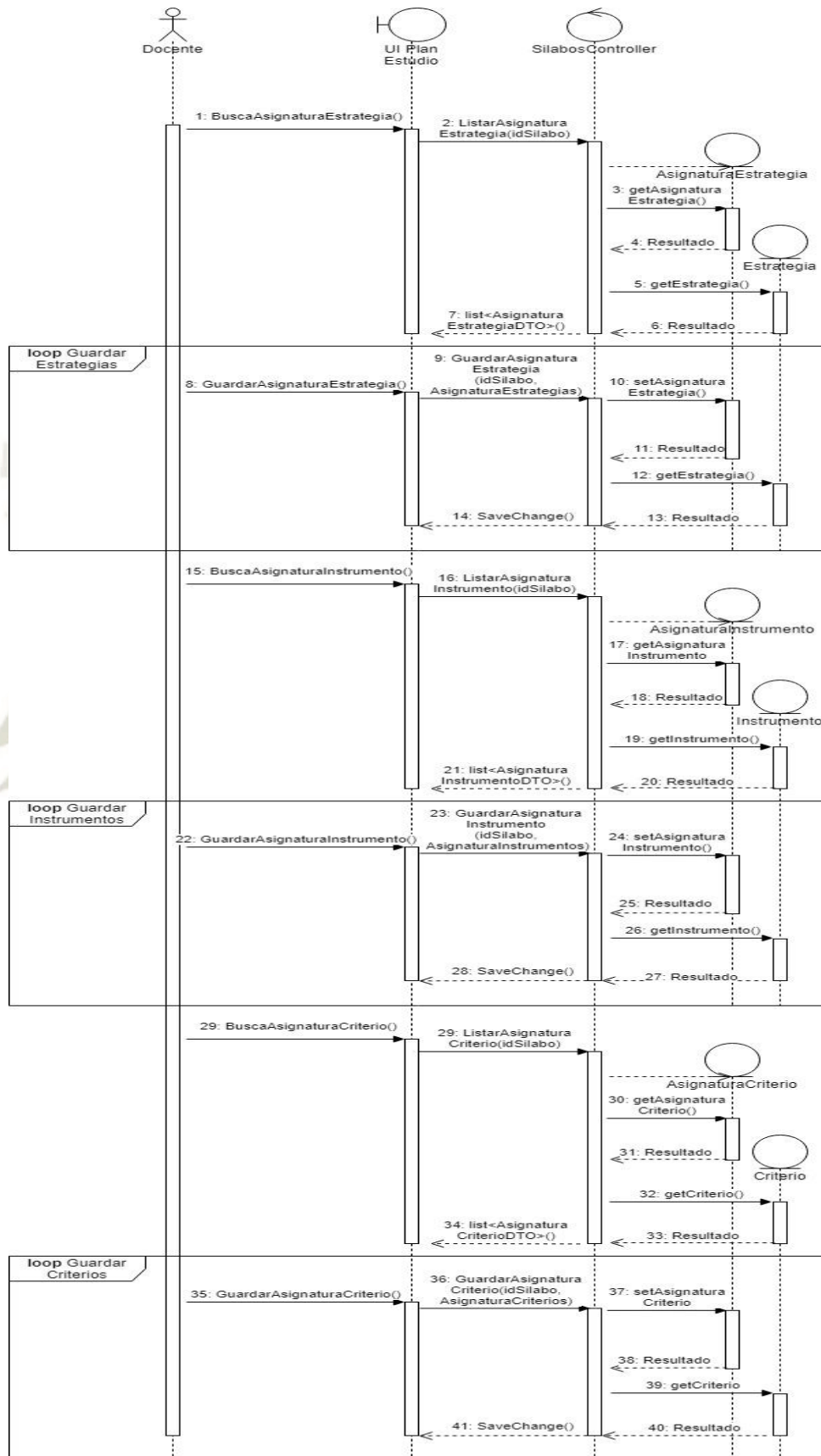
Fuente: Elaboración propia.

Figura 39 Diagrama de Secuencia - Resultados de Aprendizaje



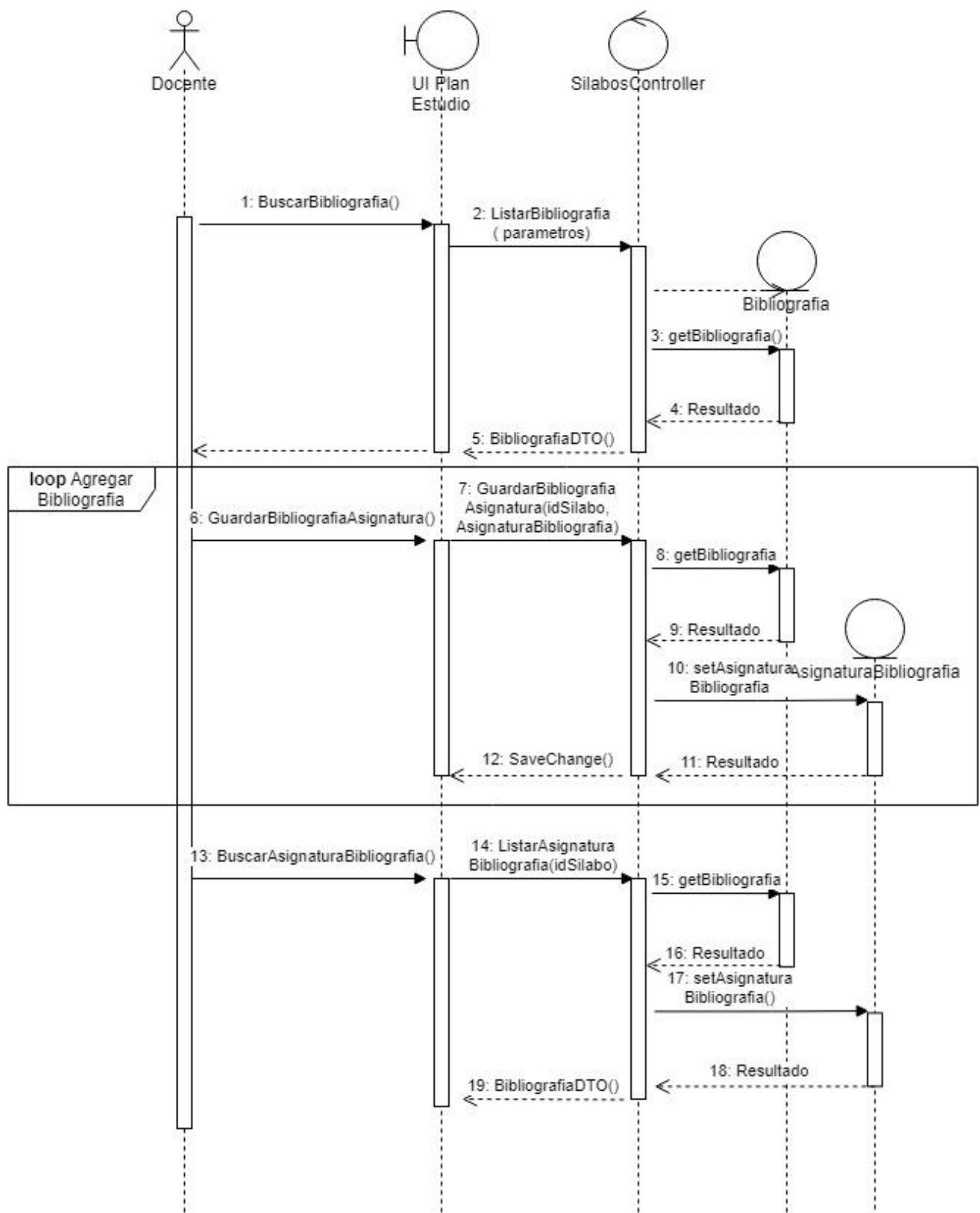
Fuente: Elaboración propia.

Figura 40 Diagrama de Secuencia - Criterios, Estrategias e Instrumentos



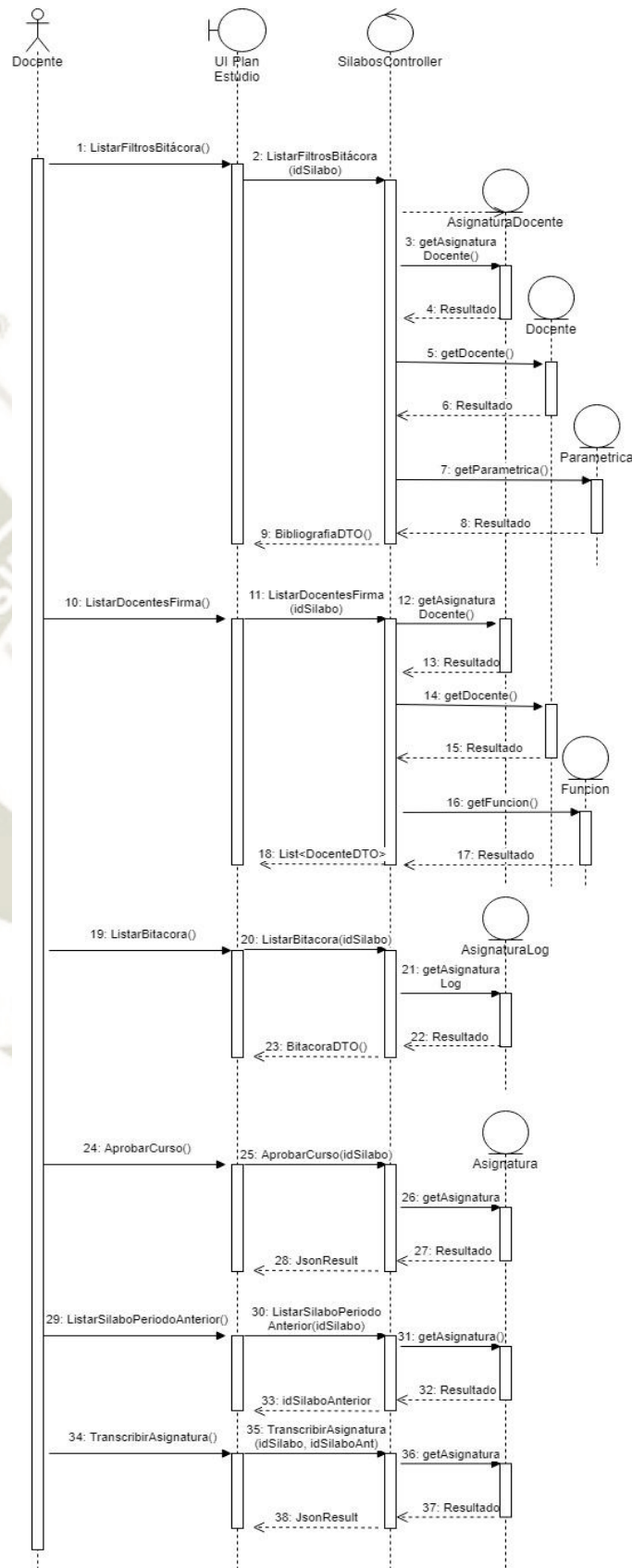
Fuente: Elaboración propia.

Figura 41 Diagrama de Secuencia - Bibliografía



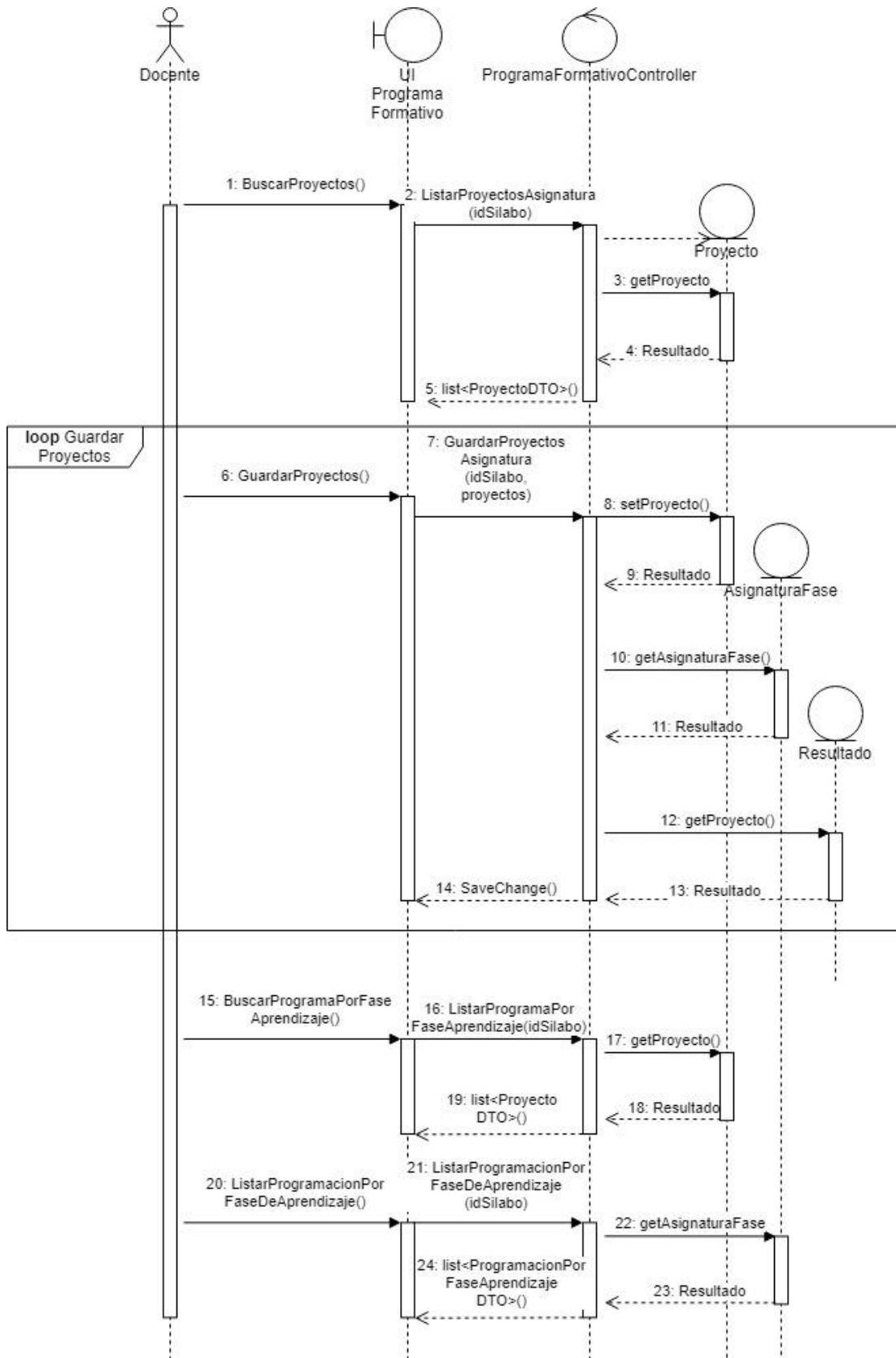
Fuente: Elaboración propia.

Figura 42 Diagrama de Secuencia - Transcripción, Bitácora y Firma



Fuente: Elaboración propia.

Figura 43 Diagrama de Secuencia - Programa Formativo



Fuente: Elaboración propia.

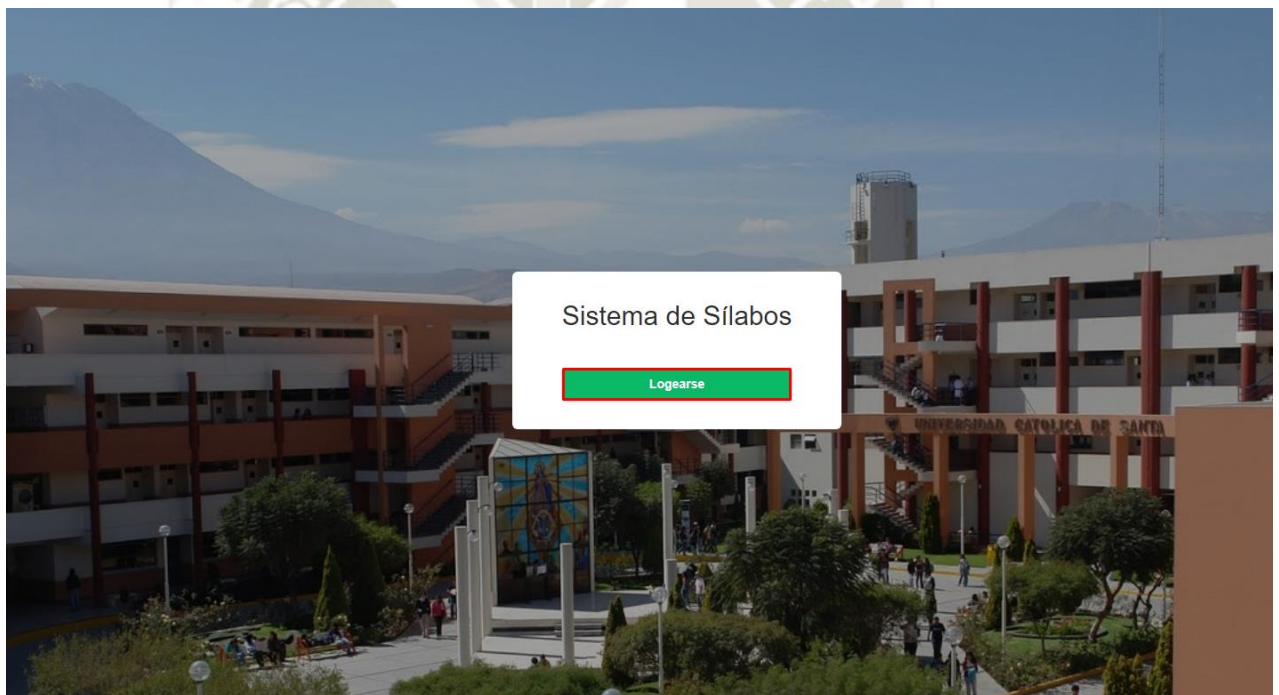
2.3.5 *Diseño de Interfaces de Usuarios*

Las interfaces utilizadas en el proyecto e implementación del sistema de gestión de sílabo tienen como principales:

A) Login de la aplicación

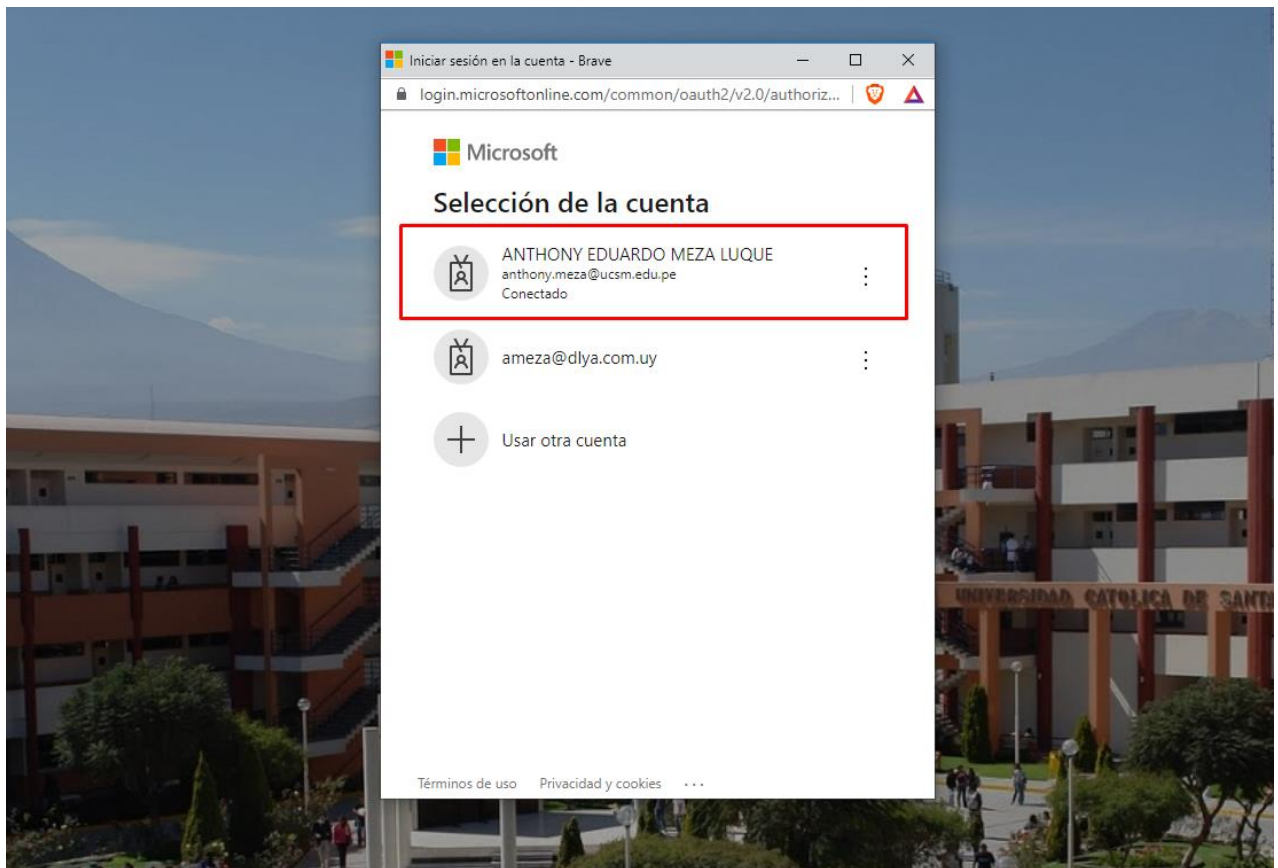
El login de la aplicación está soportado en la API de Microsoft Office 365 para la validación por medio del correo institucional y el rol desempeñado del usuario. Se muestra pantalla de inicio en la Figura 44.

Figura 44 *Login Sistema Gestión de Sílabos*



Fuente: Sistema Gestión de Sílabos UCSM

Esta autenticación permite controlar de manera segura por medio la configuración que se realice en AZURE bajo la jurisdicción de la UCSM. Asignando los roles específicos que cada usuario utilizará; sea docente, jefe de departamento o administrador de aplicativo.

Figura 45 Login Sistema Gestión de Sílabos API Office 365

Fuente: Sistema Gestión de Sílabos UCSM

B) Visualizar Sílabo

La pantalla o interfaz para visualizar sílabo, tiene como objetivo permitir al jefe de departamento, consultar el desarrollo de la asignatura a través de la generación de documento en formato PDF, con el contenido que el docente encargado ingresó en la construcción del sílabo, además de observar el estado en el que se encuentra.

Figura 46 *Interfaz Visualizar Sílabo*

Visualizar Sílabo

Escuela profesional

Código Docente Docente

Curso Año

Semestre
 Seleccione una opción

ESCUELA	CURSO	DOCENTE	SEMESTRE	ESTADO	IR
INGENIERIA DE SISTEMAS	ANÁLISIS Y DISEÑO DE SISTEMAS	JORGE MARTINEZ MUÑOZ	V	EN PROCESO	Ir
INGENIERIA DE SISTEMAS	ANÁLISIS Y DISEÑO DE SISTEMAS	JOSÉ ALFREDO SULLA TORRES	V	EN PROCESO	Ir
INGENIERIA DE SISTEMAS	ANÁLISIS Y DISEÑO DE SISTEMAS	KARINA ROSAS	V	EN PROCESO	Ir
INGENIERIA DE SISTEMAS	CÁLCULO DIFERENCIAL	FREDY QUISPE	III	ABIERTO	Ir

Fuente: Sistema Gestión de Sílabos UCSM

Una vez ubicado el sílabo a consultar, se procede a generar el archivo por el enlace en la columna “IR”. El jefe de departamento podrá observar en una ventana emergente el documento.

Figura 47 Interfaz Visualizar Sílabo - Documento PDF

The screenshot shows a web interface for viewing a syllabus. The main content is a PDF document with the following text:

UNIVERSIDAD CATÓLICA DE SANTA MARÍA
AREQUIPA- PERÚ

FACULTAD: FACULTAD DE CIENCIAS FÍSICAS Y FORMALES
ESCUELA PROFESIONAL: INGENIERIA DE SISTEMAS

PLAN DE ESTUDIOS
SÍLABO DE ASIGNATURA

1. IDENTIFICACIÓN ACADÉMICA
1.1.- Nombre de la Asignatura: ANÁLISIS Y DISEÑO DE SISTEMAS
Código de la Asignatura: 7105037
Semestre Académico en que se desarrolla: V
1.2.- Peso Académico de la Asignatura:

CRÉDITOS	HORAS SEMANALES			HORAS SEMESTRALES			
	Horas Teóricas	Horas Prácticas Práctica de Aula	Jefe de Prácticas	Horas Virtuales	Horas Teóricas	Horas Prácticas	Horas Virtuales
4	2	2	2	0	36	72	0

1.3.- Código, nombre y créditos de Asignaturas Equivalentes:
7105037 Análisis y Diseño de Sistemas
7107188 Laboratorio de Análisis y Diseño de Sistemas
1.4.- Código y nombre de Asignaturas Pre-requisito:

Below the PDF viewer, there is a table with the following data:

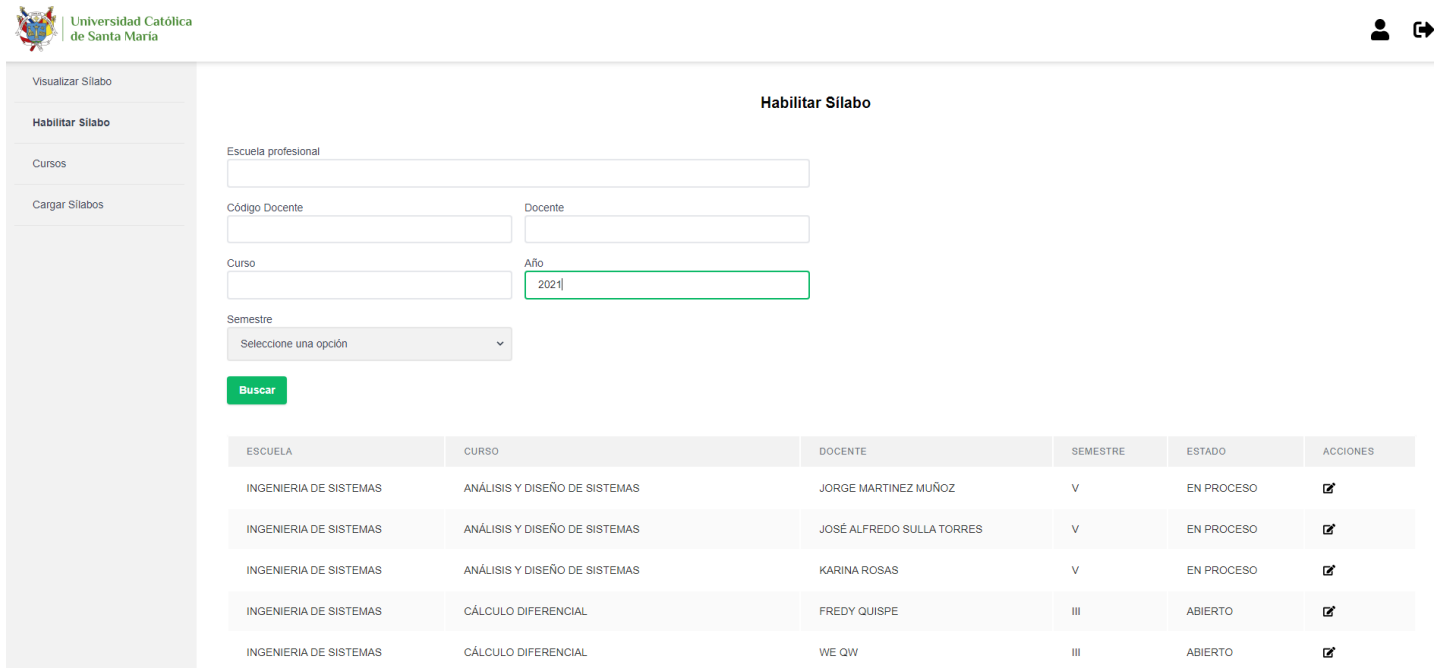
INGENIERIA DE SISTEMAS	COMPUTACIÓN EN RED II	KARINA ROSAS	V	APROBADO	Ir
INGENIERIA DE SISTEMAS	DESARROLLO I	MARIA GRACIA AYALA	I	ABIERTO	Ir






Fuente: Sistema Gestión de Sílabos UCSM

C) Habilitar Sílabo

La interfaz de habilitar sílabo permite al jefe de departamento devolver al sílabo al estado “En proceso” en caso de que encuentre algún punto a corregir, o considere culminado el proceso de desarrollo para establecer la finalización actualizando a el estado de “Cerrado”.

Figura 48 Interfaz *Habilitar Sílabo*



ESCUELA	CURSO	DOCENTE	SEMESTRE	ESTADO	ACCIONES
INGENIERIA DE SISTEMAS	ANÁLISIS Y DISEÑO DE SISTEMAS	JORGE MARTINEZ MUÑOZ	V	EN PROCESO	
INGENIERIA DE SISTEMAS	ANÁLISIS Y DISEÑO DE SISTEMAS	JOSÉ ALFREDO SULLA TORRES	V	EN PROCESO	
INGENIERIA DE SISTEMAS	ANÁLISIS Y DISEÑO DE SISTEMAS	KARINA ROSAS	V	EN PROCESO	
INGENIERIA DE SISTEMAS	CÁLCULO DIFERENCIAL	FREDY QUISPE	III	ABIERTO	
INGENIERIA DE SISTEMAS	CÁLCULO DIFERENCIAL	WE QW	III	ABIERTO	

Fuente: Sistema Gestión de Sílabos UCSM

D) Cursos

La interfaz de Curso permite al docente observar los cursos que tiene asignados en el año académico y en el periodo (semestre par o impar) en el que se encuentra.

Una vez finalizada la migración, todos los cursos o asignaturas se encuentran en estado “Abierto”, listos para poder ser construidos. Al momento de seleccionar el curso a trabajar, este actualizará inmediatamente el estado a “En proceso”, para registrar el inicio.

Si el curso entró en una etapa de “Aprobado” o “Cerrado”, se emitirán mensajes emergentes para indicar la situación.

Figura 49 *Interfaz Cursos*

CURSO	ESCUELA	SEMESTRE	TOTAL HORAS	EDITAR
DISEÑO DE SOFTWARE A BAJO NIVEL	INGENIERIA DE SISTEMAS	V	120	
COMPUTACIÓN EN RED II	INGENIERIA DE SISTEMAS	V	108	
ANÁLISIS Y DISEÑO DE SISTEMAS	INGENIERIA DE SISTEMAS	V	108	
TECNOLOGÍAS MÓVILES	INGENIERIA DE SISTEMAS	VII	108	

Fuente: Sistema Gestión de Sílabos UCSM

E) Silabo de Asignatura o Plan de Estudio

La interfaz de Sílabo de Asignatura contiene la mayor cantidad de secciones en las que se ingresará el contenido propuesto por el docente para la generación del sílabo.

Figura 50 *Interfaz Sílabo de Asignatura*

Seleccionar curso

Sílabo de asignatura

- Identificación académica
- Sumilla
- Competencias de la asignatura que apoyan al perfil de egreso
- Competencia propias de la asignatura
- Contenidos básicos por unidades de aprendizaje
- Evaluación de competencias adquiridas
- Bibliografía

Transcribir Sílabo Aprobar Sílabo Visualizar sílabo

Fuente: Sistema Gestión de Sílabos UCSM

En caso de que el docente quiera utilizar la base de los sílabos de años anteriores, utilizará la opción de Transcribir por medio del botón “Transcribir Sílabo”.

Si desea observar cómo va construyéndose el sílabo, puede observar el contenido en PDF por medio del botón “Visualizar Sílabo”.

Si el docente considera que la construcción fue exitosa y es prudente solicitar el cierre, solicitará la aprobación del sílabo por medio del botón “Aprobar Sílabo”.

En esta interfaz se tiene como vistas parciales a:

a) Identificación académica

Información básica de la asignatura acerca de: código, nombre, horas, créditos, equivalencias, prerequisites.

Figura 51 *Silabo de la Asignatura - Identificación Académica*

The screenshot shows the 'Silabo de asignatura' interface. On the left is a sidebar with navigation options: 'Visualizar Sílabo', 'Habilitar Sílabo', 'Cursos', and 'Cargar Sílabos'. The main content area is titled 'Silabo de asignatura' and includes a 'Seleccionar curso' button. Below this, the 'Identificación académica' section provides the following details:

- Facultad:** FACULTAD DE CIENCIAS FÍSICAS Y FORMALES
- Departamento:** JEFATURA DE LA FACULTAD DE CIENCIAS FÍSICAS Y FORMALES
- Nombre de la asignatura:** ANÁLISIS Y DISEÑO DE SISTEMAS
- Código:** 7105037
- Escuela profesional:** INGENIERÍA DE SISTEMAS

The 'Ubicación y peso académico de la asignatura' section contains a table with the following data:

AÑO ACADÉMICO	SEMESTRE	CRÉDITOS	HORAS SEMANALES				HORAS SEMESTRALES		
			HORAS TEÓRICAS	HORAS PRÁCTICAS		HORAS VIRTUALES	HORAS TEÓRICAS	HORAS PRÁCTICAS	HORAS VIRTUALES
				PRÁCTICA DE AULA	JEFE DE PRÁCTICAS				
2021	V	4	2	2	2	0	36	72	0

Below the table, the 'Ambiente donde se realiza el aprendizaje' section specifies:

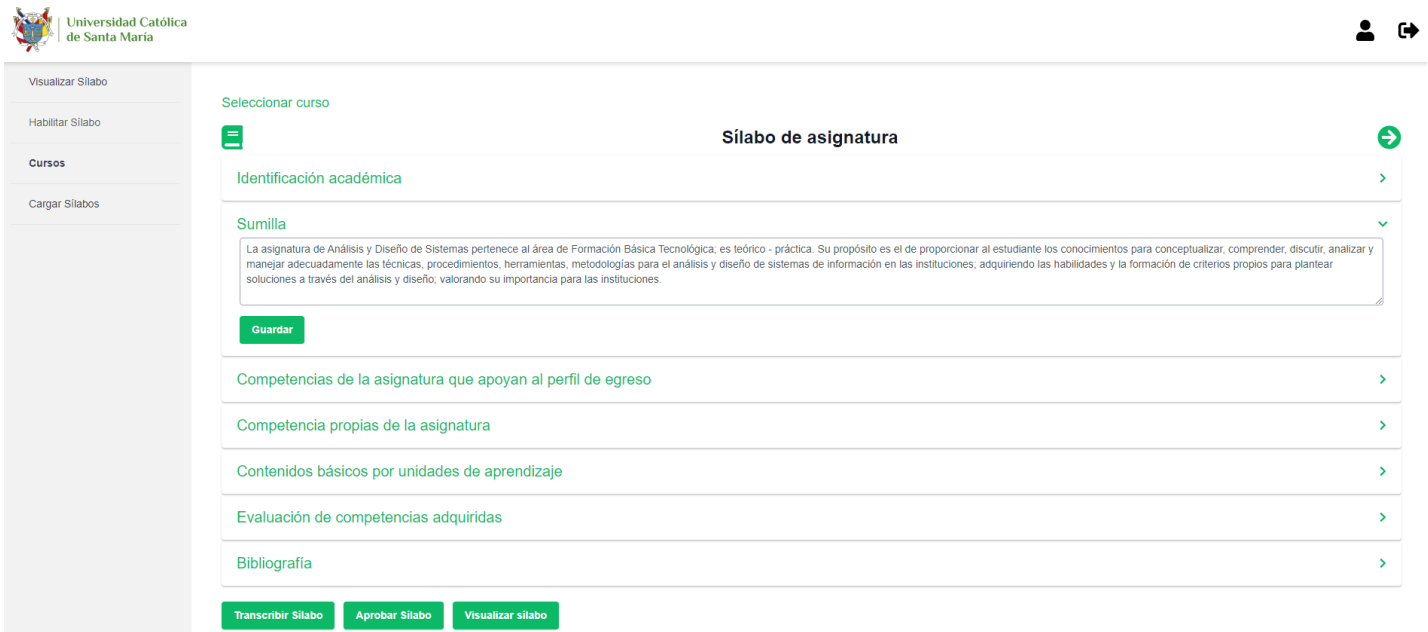
- Teoría:** MS Teams
- Práctica:** MS Teams

Fuente: Sistema Gestión de Sílabos UCSM

b) Sumilla

Sumilla de la asignatura, desarrollada por el docente.

Figura 52 *Sílabo de la Asignatura - Sumilla*



The screenshot shows a web interface for managing syllabi. On the left is a sidebar with navigation options: 'Visualizar Sílabo', 'Habilitar Sílabo', 'Cursos', and 'Cargar Sílabos'. The main area is titled 'Selección de curso' and 'Sílabo de asignatura'. It contains several expandable sections: 'Identificación académica', 'Sumilla', 'Competencias de la asignatura que apoyan al perfil de egreso', 'Competencia propias de la asignatura', 'Contenidos básicos por unidades de aprendizaje', 'Evaluación de competencias adquiridas', and 'Bibliografía'. The 'Sumilla' section is expanded, showing a text box with the following content: 'La asignatura de Análisis y Diseño de Sistemas pertenece al área de Formación Básica Tecnológica; es teórico - práctica. Su propósito es el de proporcionar al estudiante los conocimientos para conceptualizar, comprender, discutir, analizar y manejar adecuadamente las técnicas, procedimientos, herramientas, metodologías para el análisis y diseño de sistemas de información en las instituciones, adquiriendo las habilidades y la formación de criterios propios para plantear soluciones a través del análisis y diseño; valorando su importancia para las instituciones.' Below the text box is a 'Guardar' button. At the bottom of the main area are three buttons: 'Transcribir Sílabo', 'Aprobar Sílabo', and 'Visualizar sílabo'.

Fuente: Sistema Gestión de Sílabos UCSM

c) Competencias de la asignatura que apoyan al perfil de egreso

Competencias que el centro académico elaboró para que puedan ser utilizadas en las diferentes escuelas profesionales de la UCSM.

Son seleccionables y agregadas al sílabo.

Figura 53 *Silabo de la Asignatura - Competencias de la asignatura que apoyan al perfil de egreso*



Universidad Católica de Santa María

Visualizar Sílabo

Habilitar Sílabo

Cursos

Cargar Sílabos

Seleccionar curso

Sílabo de asignatura

Identificación académica

Sumilla

Competencias de la asignatura que apoyan al perfil de egreso

- Capacidad de aplicar conocimientos de matemáticas, ciencias e ingeniería en la solución de problemas complejos de ingeniería.
- La capacidad de comunicarse eficazmente, mediante la comprensión y redacción de informes eficaces y documentación de diseño, la realización de exposiciones eficaces, y la transmisión y recepción de instrucciones claras.
- La capacidad de desenvolverse eficazmente como individuo, como miembro o líder en diversos equipos, y en entornos multidisciplinarios.
- El reconocimiento de la necesidad del aprendizaje permanente y la capacidad para encararlo en el más amplio contexto de los cambios tecnológicos.
- La capacidad de crear, seleccionar y utilizar técnicas, habilidades, recursos y herramientas modernas de la ingeniería y las tecnologías de la información, incluyendo la predicción y el modelamiento, con una comprensión de las limitaciones.
- La capacidad de demostrar el conocimiento y la comprensión de los principios de gestión en ingeniería y la toma de decisiones económicas, y su respectiva aplicación.

Guardar

Fuente: Sistema Gestión de Sílabos UCSM

d) Competencias propias de la asignatura

Competencias que el docente elabora para las diferentes fases del semestre académico.

Figura 54 Silabo de la Asignatura - Competencias propias de la asignatura

Competencia propias de la asignatura

Fase 1

COMPETENCIA

Efectúa, explica y desarrolla los conceptos sobre la administración del desarrollo de sistemas de información, así como del análisis de sistemas a partir en primer lugar del análisis estructurado.



Agregar competencia

Guardar

Fase 2

COMPETENCIA

Establece, explica y exhibe las metodologías para el análisis de sistemas orientado a objetos y por prototipos como alternativas viables.



Agregar competencia

Guardar

Fase 3

COMPETENCIA

Diseña, elabora, propone el diseño desde el punto de vista de entradas, salidas y otros del módulo o sistema de información para su implementación.



Agregar competencia

Guardar

Fuente: Sistema Gestión de Sílabos UCSM

e) Contenidos básicos por unidad de aprendizaje

Los contenidos básicos albergan las unidades teóricas y prácticas o de laboratorio que se desarrollarán en el semestre académico.

Cada unidad teórica permite subunidades para un mayor nivel de detalle.

Figura 55 *Silabo de la Asignatura - Unidades teóricas*

Unidades clases teóricas

Fase 1

FASE	UNIDAD	SUB-UNIDAD	DESCRIPCIÓN	
1	1	0	1ra. Unidad: Introducción al Análisis y Diseño de Sistemas.	↑ ↓ 🗑️
1	1	1	1.1 Objetivos, problemas y problemática del análisis, responsabilidades, relación an:	↑ ↓ 🗑️
1	1	2	1.2 Equipo de desarrollo, organización, funciones	↑ ↓ 🗑️
1	1	3	1.3 Ciclo de vida del SW, modelos	↑ ↓ 🗑️
1	1	4	1.4 Metodologías de desarrollo de SW, características, clasificación	↑ ↓ 🗑️
1	1	5	1.5 Análisis de Costos y Beneficios del proyecto	↑ ↓ 🗑️
1	1	6	1.6 Preparación de la Propuesta de Sistemas	↑ ↓ 🗑️ +
FASE	UNIDAD	SUB-UNIDAD	DESCRIPCIÓN	
1	2	0	2da. Unidad: Gestión de Proyectos.	↑ ↓ 🗑️
1	2	1	2.1 Identificación de Proyectos	↑ ↓ 🗑️
1	2	2	2.2 Análisis de Factibilidad	↑ ↓ 🗑️
1	2	3	2.3 Selección de Proyectos	↑ ↓ 🗑️
1	2	4	2.4 Herramientas de gestión de proyectos	↑ ↓ 🗑️
1	2	5	2.5 Creación y gestión del plan de trabajo	↑ ↓ 🗑️ +

Agregar unidad

Fuente: Sistema Gestión de Sílabos UCSM

Las unidades de laboratorio no contienen subunidades, aunque se ingresan por cada fase del semestre.

Figura 56 Sílabo de Asignatura - Unidades de Laboratorio

Unidades laboratorio

Fase 1

FASE	UNIDAD	DESCRIPCIÓN	
1	1	Práctica 1: Repaso de configuración de routers y switches.	↑ ↓ 🗑️
1	2	Práctica 2: Configuración de VLAN, enrutamiento inter Vlan.	↑ ↓ 🗑️
1	3	Práctica 3: Configuración STP y Etherchannel.	↑ ↓ 🗑️
1	4	Práctica 4: Habilidades de configuración.	↑ ↓ 🗑️

Agregar unidad

Fase 2

FASE	UNIDAD	DESCRIPCIÓN	
2	5	Práctica 4: Configuración servidor y clientes DHCP con IPv4 e IPv6.	↑ ↓ 🗑️
2	6	Práctica 5: Configuración de FHPR, seguridad del switch.	↑ ↓ 🗑️
2	7	Práctica 6: Habilidades de configuración.	↑ ↓ 🗑️

Agregar unidad

Fuente: Sistema Gestión de Sílabos UCSM

f) Evaluación de Competencias Adquiridas

En esta vista, se encuentra el ingreso de los títulos de la fase, el desarrollo de los resultados de aprendizaje, la síntesis y la ponderación que se les da. Además de los criterios, estrategias e instrumentos que se utilizarán para la asignatura.

Figura 57 Sílabo de Asignatura - Evaluación de competencias adquiridas, títulos de fase

☰
Sílabo de asignatura
➔

- Identificación académica ➤
- Sumilla ➤
- Competencias de la asignatura que apoyan al perfil de egreso ➤
- Competencia propias de la asignatura ➤
- Contenidos básicos por unidades de aprendizaje ➤
- Evaluación de competencias adquiridas ✓

Titulos fase

FASE	TÍTULO
1	SWITCHING
2	DHCP, FHRP y SEGURIDAD LAN
3	WLAN Y ENRUTAMIENTO

Guardar

Fuente: Sistema Gestión de Sílabos UCSM

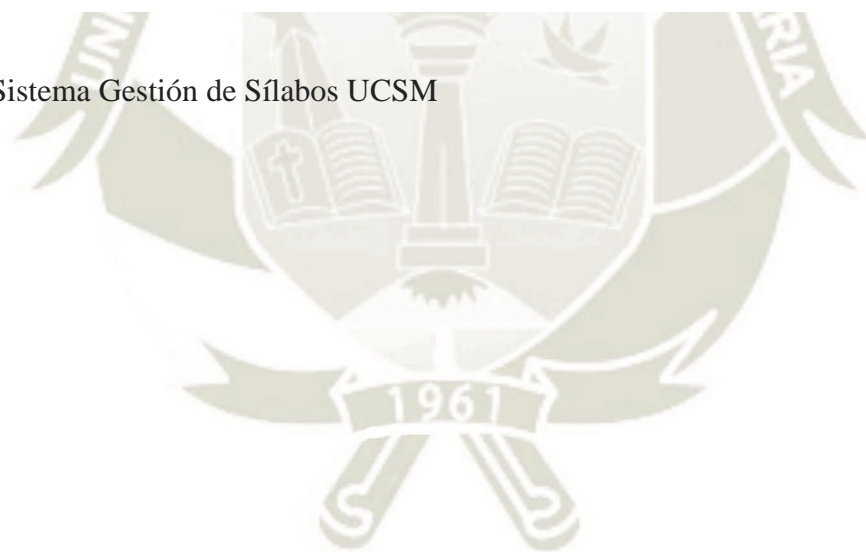


Figura 58 Sílabo de Asignatura - Evaluación de competencias adquiridas, resultados de fase

Resultados de fase

Fase 1

FASE	RESULTADO	DESCRIPCIÓN	SÍNTESIS	PESO TEÓRICO	PESO PRÁCTICO	
1	De conocimiento	Conoce la terminología del proceso de routing y switching, sus funciones y usos		33.3	0	↑ ↓ 🗑️ +
1	De desempeño	Configura adecuadamente los dispositivos de switching, para su gestión en las soluciones del diseño		33.4	0	↑ ↓ 🗑️ +
1	De producto	Propone y evalúa alternativas de conmutación con switches.		0	33.3	↑ ↓ 🗑️ +

Fase 2

Agregar tabla de conocimiento

Agregar tabla de desempeño

Agregar tabla de producto

Fase 3

Agregar tabla de conocimiento

Agregar tabla de desempeño

Agregar tabla de producto

Guardar

Fuente: Sistema Gestión de Sílabos UCSM



Figura 59 Sílabo de Asignatura - Evaluación de competencia adquiridas, Instrumentos de evaluación

Instrumentos de evaluación / Criterios / Estrategias de aprendizaje

Fase 1

Instrumentos de evaluación

DE CONOCIMIENTO			
FASE	RESULTADO	INSTRUMENTO	
1	De conocimiento	Prueba de entrada	↑ ↓ 🗑️
1	De conocimiento	Examen	↑ ↓ 🗑️
1	De conocimiento	Control de lectura	↑ ↓ 🗑️ + Instrumento propio + Instrumento
DE DESEMPEÑO			
FASE	RESULTADO	INSTRUMENTO	
1	De desempeño	Informe de práctica	↑ ↓ 🗑️
1	De desempeño	Informe de trabajo en grupo	↑ ↓ 🗑️ + Instrumento propio + Instrumento
DE PRODUCTO			
FASE	RESULTADO	INSTRUMENTO	
1	De producto	Registro de presentación de trabajos	↑ ↓ 🗑️ + Instrumento propio + Instrumento

Fuente: Sistema Gestión de Sílabos UCSM

Figura 60 Sílabo de Asignatura - Evaluación de competencias adquiridas, Criterios

Criterios

DE CONOCIMIENTO			
FASE	RESULTADO	CRITERIO	
1	De conocimiento	Preguntas objetivos y ensayo	↑ ↓ 🗑️
1	De conocimiento	Preguntas de ensayo	↑ ↓ 🗑️ + Criterio propio + Criterio
DE DESEMPEÑO			
FASE	RESULTADO	CRITERIO	
1	De desempeño	Presentación de la práctica	↑ ↓ 🗑️
1	De desempeño	Presentación grupal	↑ ↓ 🗑️ + Criterio propio + Criterio
DE PRODUCTO			
FASE	RESULTADO	CRITERIO	
1	De producto	Presentación de trabajos en el tiempo establecido	↑ ↓ 🗑️ + Criterio propio + Criterio

Fuente: Sistema Gestión de Sílabos UCSM

Figura 61 Sílabos de Asignatura - Evaluación de competencias adquiridas, estrategias

Estrategias de aprendizaje

DE CONOCIMIENTO			
FASE	RESULTADO	ESTRATEGIA DE APRENDIZAJE	
1	De conocimiento	Clase Magistral	↑ ↓ 🗑️
1	De conocimiento	Lecturas	↑ ↓ 🗑️ + Estrategia propia + Estrategia
DE DESEMPEÑO			
FASE	RESULTADO	ESTRATEGIA DE APRENDIZAJE	
1	De desempeño	Práctica individual y colectiva en laboratorio	↑ ↓ 🗑️
1	De desempeño	Programación de Proyectos	↑ ↓ 🗑️
1	De desempeño	Costo Beneficio	↑ ↓ 🗑️
1	De desempeño	Flujograma de Información	↑ ↓ 🗑️
1	De desempeño	Diccionario de Datos	↑ ↓ 🗑️ + Estrategia propia + Estrategia
DE PRODUCTO			
FASE	RESULTADO	ESTRATEGIA DE APRENDIZAJE	
1	De producto	Dirección y orientación del docente	↑ ↓ 🗑️ + Estrategia propia + Estrategia

Fuente: Sistema Gestión de Sílabos UCSM

g) Bibliografía

En esta sección, se selecciona la bibliografía que sirve como base para la elaboración del sílabo. Cada bibliografía debe agregarse a la fase correspondiente.

Figura 62 Sílabo de Asignatura - Bibliografía

Bibliografía

Fase 1

AUTOR	NOMBRE	EDITORIAL	LUGAR DE PUBLICACIÓN	AÑO DE PUBLICACIÓN	
Kendall Kenneth, Kendall Julie	Análisis y Diseño de Sistemas	Pearson	USA	2014	↑ ↓ 🗑️

Agregar bibliografía

Fase 2

AUTOR	NOMBRE	EDITORIAL	LUGAR DE PUBLICACIÓN	AÑO DE PUBLICACIÓN	
CISCO SYSTEMS	Fundamentos de conmutación, enrutamiento y comunicaciones inalámbricas	CISCO	USA	2020	↑ ↓ 🗑️
ROSAS KARINA LAURA L.	Matemática de Redes	Ed. Universidad Católica de Santa María	Arequipa, Perú	2016	↑ ↓ 🗑️

Agregar bibliografía

Fase 3

AUTOR	NOMBRE	EDITORIAL	LUGAR DE PUBLICACIÓN	AÑO DE PUBLICACIÓN	
No hay datos					

Agregar bibliografía

Guardar

Fuente: Sistema Gestión de Sílabos UCSM

F) Programa Formativo

En la interfaz de programa formativo, el docente visualizará la información ya ingresada previamente en la interfaz de Sílabo de asignatura.

Figura 63 Interfaz Programa Formativo

Fuente: Sistema Gestión de Sílabos UCSM

En esta sección se podrá ingresar el contenido de los proyectos de investigación formativa o responsabilidad social.

Figura 64 Programa Formativo - Matriz

Matriz ▼

Fase

ÁREA	DENOMINACIÓN DE LA ACTIVIDAD	PROPÓSITO	INDICADORES DE EVALUACIÓN	BENEFICIARIOS	RESPONSABLES	CRONOGRAMA	PESO TEÓRICO	PESO PRÁCTICO
Investigación Formativa	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Responsabilidad Social	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	Desarrollar el Diseño de una red LAN	Poner en práctica los	Nivel de aprendizaje de los participantes	Estudiantes	Jefes de Práctica de la asignatura	3 Fase	10.2	10.2

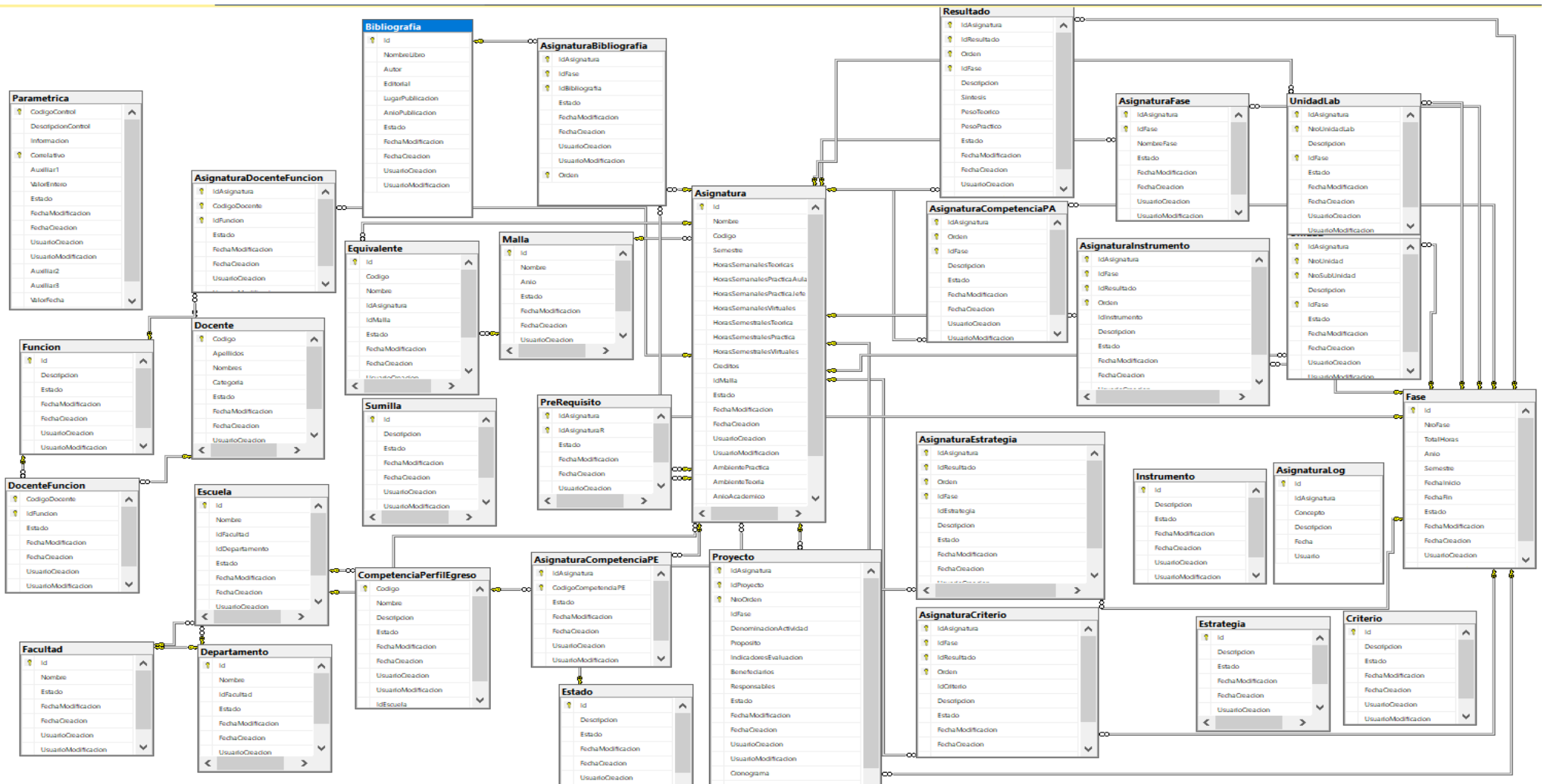
Fuente: Sistema Gestión de Sílabos UCSM

2.3.6 Diseño de Modelo de Datos

De acuerdo con el diseño del Sistema de Gestión de Sílabos elaborado en las entidades descritas en la sección de modelos, se desarrolla la base de datos SilabosDB.

Este modelo relacional de la Base de Datos permite la gestión y elaboración de los sílabos desarrollados en la Universidad Católica de Santa María en las diversas secciones que contiene el sílabo: Plan de Estudio y Programa Formativo. Se aprecia el modelo en la Figura 65.

Figura 65 Modelo relacional de Base de Datos SílabosDB



Fuente: Elaboración propia

A continuación, se detalla el diccionario de datos de la base de datos del sistema de Gestión de Sílabos.

1. Tabla Asignatura

En esta tabla se almacena la información propia de la asignatura migrada al momento de abrirse el semestre a trabajar.

Tabla 26 Diccionario de datos de la tabla «Asignatura»

Campo	Tipo	Nulo	Comentarios
Id	Int (PK)	Not Null	Identificador de la asignatura
Nombre	NVarchar (100)	Not Null	Nombre de la asignatura
Codigo	NVarchar (50)	Not Null	Código de la asignatura
Semestre	Int	Not Null	Semestre en el que se desarrolla la asignatura
HorasSemanalesTeoricas	Real	Not Null	Horas semanales de la asignatura
HorasSemanalesPracticaAula	Real	Not Null	Horas semanales de práctica o laboratorio
HorasSemanalesPracticaJefe	Real	Not Null	Horas semanales de práctica de jefatura
HorasSemanalesVirtuales	Real	Not Null	Horas semanales virtuales de la asignatura
HorasSemestrales	Real	Not Null	Horas Totales Teóricas de la asignatura en el semestre

HorasSemestralesPractica	Real	Not Null	Horas Totales de Practica de la asignatura en el semestre
HorasSemestralesvirtuales	Real	Not Null	Horas virtuales de la asignatura en el semestre
Creditos	Real	Not null	Cantidad de créditos que vale la asignatura
IdMalla	Int (FK)	Not null	Clave foránea de la Malla a la que pertenece la asignatura
Estado	Int	Not Null	Estado en el que se encuentra el registro
FechaModificacion	DateTime2 (7)	Null	Fecha en la que sufre una actualización el registro
FechaCreacion	DateTime2 (7)	Not Null	Fecha en la que se crea el registro
UsuarioCreacion	NVarChar (50)	Not Null	Usuario que creo el registro
UsuarioModificacion	NVarChar (50)	Null	Usuario que modifica el registro
AmbientePractica	NVarChar (Max)	Null	Ambiente donde se desarrolla la práctica de la asignatura
AmbienteTeoria	NVarChar (Max)	Null	Ambiente donde se desarrolla la teoría de la asignatura
AnioAcademico	Int	Not Null	Año en el que se desarrolla la asignatura
IdSumilla	Int	Null	Identificador de la sumilla

IdEstado	Int (FK)	Not Null	Código de estado en el que se encuentra el proceso de la asignatura
IdEscuela	Int (FK)	Not Null	Clave foránea de la escuela en la que se encuentra la asignatura

Fuente: Elaboración propia.

2. Tabla Asignatura Bibliografía

En esta tabla se almacenan la bibliografía que el docente selecciona para la asignatura.

Tabla 27 Diccionario de datos de la tabla «AsignaturaBibliografía»

Campo	Tipo	Nulo	Comentarios
IdAsignatura	Int (PK, FK)	Not Null	Identificador de la asignatura
IdFase	Int (PK, FK)	Not Null	Identificador de la Fase
IdBibliografía	Int (PK, FK)	Not Null	Identificador de bibliografía
Orden	Int (PK)	Not Null	Orden en el que corresponde la bibliografía
Estado	Int	Not Null	Estado en el que se encuentra el registro
FechaModificacion	DateTime2 (7)	Null	Fecha en la que sufre una actualización el registro
FechaCreacion	DateTime2 (7)	Not Null	Fecha en la que se crea el registro

UsuarioCreacion	NVarChar (50)	Not Null	Usuario que creo el registro
UsuarioModificacion	NVarChar (50)	Null	Usuario que modifica el registro

Fuente: Elaboración propia.

3. Tabla Asignatura Competencia Propia Asignatura

En esta tabla se almacenan las competencias propias de la asignatura que el docente ingrese de manera manual.

Tabla 28 Diccionario de datos de la tabla «AsignaturaCompetenciaPA»

Campo	Tipo	Nulo	Comentarios
IdAsignatura	Int (PK, FK)	Not Null	Identificador de la asignatura
Orden	Int (PK)	Not Null	Orden en el que corresponde la competencia
IdFase	Int (PK, FK)	Not Null	Identificador de la Fase
Descripcion	NVarChar (Max)	Not Null	Descripción de la competencia
Estado	Int	Not Null	Estado en el que se encuentra el registro
FechaModificacion	DateTime2 (7)	Null	Fecha en la que sufre una actualización el registro
FechaCreacion	DateTime2 (7)	Not Null	Fecha en la que se crea el registro

UsuarioCreacion	NVarChar (50)	Not Null	Usuario que creo el registro
UsuarioModificacion	NVarChar (50)	Null	Usuario que modifica el registro

Fuente: Elaboración propia.

4. Tabla 3 Asignatura Competencia Perfil Egreso

Tabla en la que se almacenan las competencias propias del perfil de egreso seleccionadas por el docente para incorporar en la asignatura.

Tabla 29 Diccionario de datos de la tabla «AsignaturaCompetenciaPE»

Campo	Tipo	Nulo	Comentarios
IdAsignatura	Int (PK, FK)	Not Null	Identificador de la asignatura
CodigoCompetencia	NVarChar (PK, FK)	Not Null	Código de competencia seleccionada
Estado	Int	Not Null	Estado en el que se encuentra el registro
FechaModificacion	DateTime2 (7)	Null	Fecha en la que sufre una actualización el registro
FechaCreacion	DateTime2 (7)	Not Null	Fecha en la que se crea el registro
UsuarioCreacion	NVarChar (50)	Not Null	Usuario que creo el registro
UsuarioModificacion	NVarChar (50)	Null	Usuario que modifica el registro

Fuente: Elaboración propia.

5. Tabla Asignatura Criterio

Tabla en la que se almacena los criterios seleccionados o ingresados por el docente.

Tabla 30 Diccionario de datos de la tabla «AsignaturaCriterio»

Campo	Tipo	Nulo	Comentarios
IdAsignatura	Int (PK, FK)	Not Null	Identificador de la asignatura
IdFase	Int (PK, FK)	Not Null	Identificador de la fase
IdResultado	Int (PK)	Not Null	Identificador del resultado
Orden	Int (PK)	Not Null	Orden en el que se ingresa el criterio
IdCriterio	Int (FK)	Null	Identificador del Criterio
Descripcion	NVarChar (Max)	Null	Descripción del Criterio
	NVarChar (50)	Null	Usuario que modifica el registro
Estado	Int	Not Null	Estado en el que se encuentra el registro
FechaModificacion	DateTime2 (7)	Null	Fecha en la que sufre una actualización el registro
FechaCreacion	DateTime2 (7)	Not Null	Fecha en la que se crea el registro
UsuarioCreacion	NVarChar (50)	Not Null	Usuario que creo el registro
UsuarioModificacion	NVarChar (50)	Null	Usuario que modifica el registro

Fuente: Elaboración propia.

6. Tabla Asignatura Docente Función

Tabla que almacena la relación entre los docentes y sus funciones con las asignaturas que dictarán en el semestre.

Tabla 31 Diccionario de datos de la tabla «AsignaturaDocenteFuncion»

Campo	Tipo	Nulo	Comentarios
IdAsignatura	Int (PK, FK)	Not Null	Identificador de la asignatura
CodigoDocente	NVarChar (450) (PK, FK)	Not Null	Código del docente
IdFuncion	Int (PK, FK)	Not Null	Identificador de la función
Estado	Int	Not Null	Estado en el que se encuentra el registro
FechaModificacion	DateTime2 (7)	Null	Fecha en la que sufre una actualización el registro
FechaCreacion	DateTime2 (7)	Not Null	Fecha en la que se crea el registro
UsuarioCreacion	NVarChar (50)	Not Null	Usuario que creo el registro
UsuarioModificacion	NVarChar (50)	Null	Usuario que modifica el registro

Fuente: Elaboración propia.

7. Tabla Asignatura Estrategia

Tabla que almacena las diferentes estrategias elegidas por el docente e ingresadas por el mismo, que se consideren oportunas.

Tabla 32 Diccionario de datos de la tabla «AsignaturaEstrategia»

Campo	Tipo	Nulo	Comentarios
IdAsignatura	Int (PK, FK)	Not Null	Identificador de la asignatura
IdFase	Int (PK, FK)	Not Null	Identificador de la fase
IdResultado	Int (PK)	Not Null	Identificador del resultado
Orden	Int (PK)	Not Null	Orden en el que se ingresa la estrategia
IdEstrategia	Int (FK)	Null	Identificador de la Estrategia
Descripcion	NVarChar (Max)	Null	Descripción de la estrategia
Estado	Int	Not Null	Estado en el que se encuentra el registro
FechaModificacion	DateTime2 (7)	Null	Fecha en la que sufre una actualización el registro
FechaCreacion	DateTime2 (7)	Not Null	Fecha en la que se crea el registro
UsuarioCreacion	NVarChar (50)	Not Null	Usuario que creo el registro
UsuarioModificacion	NVarChar (50)	Null	Usuario que modifica el registro

Fuente: Elaboración propia.

8. Tabla Asignatura Fase

Tabla que almacena la información referente al nombre de la fase de cada asignatura.

Tabla 33 Diccionario de datos de la tabla «AsignaturaFase»

Campo	Tipo	Nulo	Comentarios
IdAsignatura	Int (PK, FK)	Not Null	Identificador de la asignatura
IdFase	Int (PK, FK)	Not Null	Identificador de la fase
NombreFase	NVarChar (Max)	Null	Nombre de la Fase
Estado	Int	Not Null	Estado en el que se encuentra el registro
FechaModificacion	DateTime2 (7)	Null	Fecha en la que sufre una actualización el registro
FechaCreacion	DateTime2 (7)	Not Null	Fecha en la que se crea el registro
UsuarioCreacion	NVarChar (50)	Not Null	Usuario que creo el registro
UsuarioModificacion	NVarChar (50)	Null	Usuario que modifica el registro

Fuente: Elaboración propia.

9. Tabla Asignatura Instrumento

Tabla que almacena la información referente a los instrumentos elegidos y redactados por los docentes.

Tabla 34 Diccionario de datos de la tabla «AsignaturaInstrumento»

Campo	Tipo	Nulo	Comentarios
IdAsignatura	Int (PK, FK)	Not Null	Identificador de la asignatura
IdFase	Int (PK, FK)	Not Null	Identificador de la fase
IdResultado	Int (PK)	Not Null	Identificador de Resultado
Orden	Int (PK)	Not Null	Orden en el que se ingresa el instrumento
IdInstrumento	Int (FK)	Null	Identificador del instrumento seleccionado
Descripcion	NVarChar (Max)	Null	Descripción de instrumento redactado
Estado	Int	Not Null	Estado en el que se encuentra el registro
FechaModificacion	DateTime2 (7)	Null	Fecha en la que sufre una actualización el registro
FechaCreacion	DateTime2 (7)	Not Null	Fecha en la que se crea el registro
UsuarioCreacion	NVarChar (50)	Not Null	Usuario que creo el registro
UsuarioModificacion	NVarChar (50)	Null	Usuario que modifica el registro

Fuente: Elaboración propia.

10. Tabla Asignatura Log

En esta tabla se almacenan el Log de los cambios y modificaciones que sufre la asignatura al momento de su desarrollo.

Tabla 35 *Diccionario de datos de la tabla «AsignaturaLog»*

Campo	Tipo	Nulo	Comentarios
Id	Int (PK)	Not Null	Identificador del Log
IdAsignatura	Int	Not Null	Identificador de la asignatura
Concepto	NVarchar(450)	Null	Concepto que sufre modificación
Descripcion	NVarchar(450)	Null	Descripción del cambio
Fecha	Datetime2(7)	Not Null	Fecha en que se registra cambio
Usuario	NVarchar(450)	Null	Usuario que realizó el cambio

Fuente: Elaboración propia.

11. Tabla Bibliografía

En esta tabla se almacenan la bibliografía base para el desarrollo del sílabo, de esta lista se procede a seleccionar.

Tabla 36 *Diccionario de datos de la tabla «Bibliografía»*

Campo	Tipo	Nulo	Comentarios
Id	Int (PK)	Not Null	Identificador de la bibliografía
NombreLibro	NVarchar(max)	Not Null	Nombre del artículo, libro, etc

Autor	NVarchar(max)	Not Null	Autor(es) que realizaron la publicación del texto
Editorial	NVarchar(max)	Not Null	Editorial que realizó la publicación
LugarPublicacion	NVarchar(max)	Not Null	Lugar en el que se encuentra la editorial
AnioPublicacion	NVarchar(max)	NotNull	Año en el que se publicó el texto
Estado	Int	Not Null	Estado en el que se encuentra el registro
FechaModificacion	DateTime2 (7)	Null	Fecha en la que sufre una actualización el registro
FechaCreacion	DateTime2 (7)	Not Null	Fecha en la que se crea el registro
UsuarioCreacion	NVarChar (50)	Not Null	Usuario que creo el registro
UsuarioModificacion	NVarChar (50)	Null	Usuario que modifica el registro

Fuente: Elaboración propia.

12. Tabla Competencia Perfil Egreso

Tabla que almacena las competencias que el departamento académico elaboró para poder utilizar en las diferentes escuelas profesionales.

Tabla 37 Diccionario de datos de la tabla «CompetenciaPerfilEgreso»

Campo	Tipo	Nulo	Comentarios
Codigo	NVarChar (PK) (450)	Not Null	Código de competencia
Nombre	NVarChar (Max)	Not Null	Nombre de la competencia
Descripcion	NVarChar (Max)	Not Null	Descripción de la competencia
IdEscuela	NVarChar (FK) (10)	Not Null	Escuela a la que pertenece la competencia
Estado	Int	Not Null	Estado en el que se encuentra el registro
FechaModificacion	DateTime2 (7)	Null	Fecha en la que sufre una actualización el registro
FechaCreacion	DateTime2 (7)	Not Null	Fecha en la que se crea el registro
UsuarioCreacion	NVarChar (50)	Not Null	Usuario que creo el registro
UsuarioModificacion	NVarChar (50)	Null	Usuario que modifica el registro

Fuente: Elaboración propia.

13. Tabla Criterio

Tabla que almacena los diferentes criterios definidos por la administración o el centro académico y que se encuentran disponibles a elegir en la elaboración del sílabo.

Tabla 38 *Diccionario de datos de la tabla «Criterio»*

Campo	Tipo	Nulo	Comentarios
Id	Int (PK)	Not Null	Identificador del Criterio
Descripcion	NVarChar (Max)	Not Null	Descripción del criterio
Estado	Int	Not Null	Estado en el que se encuentra el registro
FechaModificacion	DateTime2 (7)	Null	Fecha en la que sufre una actualización el registro
FechaCreacion	DateTime2 (7)	Not Null	Fecha en la que se crea el registro
UsuarioCreacion	NVarChar (50)	Not Null	Usuario que creo el registro
UsuarioModificacion	NVarChar (50)	Null	Usuario que modifica el registro

Fuente: Elaboración propia.

14. Tabla Departamento

Tabla que almacena los departamentos o jefaturas que existen en la universidad.

Tabla 39 Diccionario de datos de la tabla «Departamento»

Campo	Tipo	Nulo	Comentarios
Id	Int (PK)	Not Null	Identificador del Departamento
Nombre	NVarChar (100)	Not Null	Nombre del departamento
IdFacultad	Int (FK)	Not Null	Identificador de la Facultad
Estado	Int	Not Null	Estado en el que se encuentra el registro
FechaModificacion	DateTime2 (7)	Null	Fecha en la que sufre una actualización el registro
FechaCreacion	DateTime2 (7)	Not Null	Fecha en la que se crea el registro
UsuarioCreacion	NVarChar (50)	Not Null	Usuario que creo el registro
UsuarioModificacion	NVarChar (50)	Null	Usuario que modifica el registro

Fuente: Elaboración propia.

15. Tabla Docente

Tabla que almacena los departamentos o jefaturas que existen en la universidad.

Tabla 40 Diccionario de datos de la tabla «Docente»

Campo	Tipo	Nulo	Comentarios
Codigo	NVarChar (PK) (Max)	Not Null	Código del docente asignado
Apellidos	NVarChar (50)	Not Null	Apellidos del docente
Nombres	NVarChar (50)	Not Null	Nombres del docente
Categoria	NVarChar (50)	Not Null	Categoría que tiene el docente respecto a la asignatura
Estado	Int	Not Null	Estado en el que se encuentra el registro
FechaModificacion	DateTime2 (7)	Null	Fecha en la que sufre una actualización el registro
FechaCreacion	DateTime2 (7)	Not Null	Fecha en la que se crea el registro
UsuarioCreacion	NVarChar (50)	Not Null	Usuario que creo el registro
UsuarioModificacion	NVarChar (50)	Null	Usuario que modifica el registro

Fuente: Elaboración propia.

16. Tabla Docente Función

Tabla que almacena los roles que tienen los docentes que permiten el ingreso a la aplicación y las opciones que este tiene para navegar.

Tabla 41 Diccionario de datos de la tabla «DocenteFuncion»

Campo	Tipo	Nulo	Comentarios
CodigoDocente	NVarChar (PK, FK) (450)	Not Null	Código del docente asignado
IdFuncion	Int (PK)	Not Null	Identificador de la función
Estado	Int	Not Null	Estado en el que se encuentra el registro
FechaModificacion	DateTime2 (7)	Null	Fecha en la que sufre una actualización el registro
FechaCreacion	DateTime2 (7)	Not Null	Fecha en la que se crea el registro
UsuarioCreacion	NVarChar (50)	Not Null	Usuario que creo el registro
UsuarioModificacion	NVarChar (50)	Null	Usuario que modifica el registro

Fuente: Elaboración propia.

17. Tabla Equivalente

Tabla que almacena los cursos equivalentes de una asignatura y que existieron en mallas anteriores.

Tabla 42 Diccionario de datos de la tabla «Equivalente»

Campo	Tipo	Nulo	Comentarios
Id	Int (PK)	Not Null	Identificador de asignatura equivalente
Codigo	NVarChar (20)	Not Null	Código de asignatura equivalente
Nombre	NVarChar (100)	Not Null	Nombre de la asignatura equivalente
IdAsignatura	Int (FK)	Not Null	Identificador de asignatura vigente en el semestre académico
IdMalla	Int (FK)	Not Null	Identificador de la malla a la que pertenecen
Estado	Int	Not Null	Estado en el que se encuentra el registro
Creditos	Int	Not Null	Créditos del curso equivalente
FechaModificacion	DateTime2 (7)	Null	Fecha en la que sufre una actualización el registro
FechaCreacion	DateTime2 (7)	Not Null	Fecha en la que se crea el registro
UsuarioCreacion	NVarChar (50)	Not Null	Usuario que creo el registro
UsuarioModificacion	NVarChar (50)	Null	Usuario que modifica el registro

Fuente: Elaboración propia.

18. Tabla Escuela

Tabla que almacena las escuelas profesionales existentes en la universidad.

Tabla 43 Diccionario de datos de la tabla «Escuela»

Campo	Tipo	Nulo	Comentarios
Id	NVarChar (PK) (10)	Not Null	Identificador de la escuela
Nombre	NVarChar (100)	Not Null	Nombre de la escuela profesional
IdFacultad	Int (FK)	Not Null	Identificador de la facultad a la que pertenece la escuela
Estado	Int	Not Null	Estado en el que se encuentra el registro
FechaModificacion	DateTime2 (7)	Null	Fecha en la que sufre una actualización el registro
FechaCreacion	DateTime2 (7)	Not Null	Fecha en la que se crea el registro
UsuarioCreacion	NVarChar (50)	Not Null	Usuario que creo el registro
UsuarioModificacion	NVarChar (50)	Null	Usuario que modifica el registro

Fuente: Elaboración propia.

19. Tabla Estado

Tabla que almacena los estados por los que va desarrollándose el sílabo.

Tabla 44 *Diccionario de datos de la tabla «Estado»*

Campo	Tipo	Nulo	Comentarios
Id	Int (PK)	Not Null	Identificador del estado
Descripcion	NVarChar (10)	Not Null	Descripción del Estado
Estado	Int	Not Null	Estado en el que se encuentra el registro
FechaModificacion	DateTime2 (7)	Null	Fecha en la que sufre una actualización el registro
FechaCreacion	DateTime2 (7)	Not Null	Fecha en la que se crea el registro
UsuarioCreacion	NVarChar (50)	Not Null	Usuario que creo el registro
UsuarioModificacion	NVarChar (50)	Null	Usuario que modifica el registro

Fuente: Elaboración propia.

20. Tabla Estrategia

Tabla que almacena las diferentes estrategias definidas por la administración o el centro académico y que se encuentran disponibles a elegir en la elaboración del sílabo.

Tabla 45 Diccionario de datos de la tabla «Estrategia»

Campo	Tipo	Nulo	Comentarios
Id	Int (PK)	Not Null	Identificador de la Estrategia
Descripcion	NVarChar (Max)	Not Null	Descripción de la Estrategia
Estado	Int	Not Null	Estado en el que se encuentra el registro
FechaModificacion	DateTime2 (7)	Null	Fecha en la que sufre una actualización el registro
FechaCreacion	DateTime2 (7)	Not Null	Fecha en la que se crea el registro
UsuarioCreacion	NVarChar (50)	Not Null	Usuario que creo el registro
UsuarioModificacion	NVarChar (50)	Null	Usuario que modifica el registro

Fuente: Elaboración propia.

21. Tabla Facultad

Tabla que almacena las facultades existentes en la universidad.

Tabla 46 Diccionario de datos de la tabla «Facultad»

Campo	Tipo	Nulo	Comentarios
Id	Int (PK)	Not Null	Identificador de la Facultad
Nombre	NVarChar (100)	Not Null	Nombre de la Facultad
Estado	Int	Not Null	Estado en el que se encuentra el registro
FechaModificacion	DateTime2 (7)	Null	Fecha en la que sufre una actualización el registro
FechaCreacion	DateTime2 (7)	Not Null	Fecha en la que se crea el registro
UsuarioCreacion	NVarChar (50)	Not Null	Usuario que creo el registro
UsuarioModificacion	NVarChar (50)	Null	Usuario que modifica el registro

Fuente: Elaboración propia.

22. Tabla Fase

Tabla que las fases correspondientes al año académico, el total de horas que tendrá cada una, el año y semestre en el que se desarrollarán y las fechas de inicio y fin de estas.

Tabla 47 Diccionario de datos de la tabla «Fase»

Campo	Tipo	Nulo	Comentarios
Id	Int (PK)	Not Null	Identificador de la Fase
NroFase	Int	Not Null	Número de la fase
TotalHoras	Int	Not Null	Total Horas que durará la fase
Año	Int	Not Null	Año académico en el que se desarrolla la fase
Semestre	Int	Not Null	Semestre académico en el que se desarrolla la fase
FechaInicio	DateTime2 (7)	Not Null	Fecha de inicio de la fase
FechaFin	DateTime2 (7)	Not Null	Fecha de finalización de la fase
Estado	Int	Not Null	Estado en el que se encuentra el registro
FechaModificacion	DateTime2 (7)	Null	Fecha en la que sufre una actualización el registro
FechaCreacion	DateTime2 (7)	Not Null	Fecha en la que se crea el registro
UsuarioCreacion	NVarChar (50)	Not Null	Usuario que creo el registro
UsuarioModificacion	NVarChar (50)	Null	Usuario que modifica el registro

Fuente: Elaboración propia.

23. Tabla Función

Tabla que almacena las diversas funciones que desempeñarán los usuarios del aplicativo.

Tabla 48 *Diccionario de datos de la tabla «Funcion»*

Campo	Tipo	Nulo	Comentarios
Id	Int (PK)	Not Null	Identificador de la Función
Descripcion	NVarChar (100)	Not Null	Descripción de la función
Estado	Int	Not Null	Estado en el que se encuentra el registro
FechaModificacion	DateTime2 (7)	Null	Fecha en la que sufre una actualización el registro
FechaCreacion	DateTime2 (7)	Not Null	Fecha en la que se crea el registro
UsuarioCreacion	NVarChar (50)	Not Null	Usuario que creo el registro
UsuarioModificacion	NVarChar (50)	Null	Usuario que modifica el registro

Fuente: Elaboración propia.

24. Tabla Instrumento

Tabla que almacena los diferentes instrumentos definidos por la administración o el centro académico y que se encuentran disponibles a elegir en la elaboración del sílabo.

Tabla 49 *Diccionario de datos de la tabla «Instrumento»*

Campo	Tipo	Nulo	Comentarios
Id	Int (PK)	Not Null	Identificador del instrumento
Descripcion	NVarChar (Max)	Not Null	Descripción del instrumento
Estado	Int	Not Null	Estado en el que se encuentra el registro
FechaModificacion	DateTime2 (7)	Null	Fecha en la que sufre una actualización el registro
FechaCreacion	DateTime2 (7)	Not Null	Fecha en la que se crea el registro
UsuarioCreacion	NVarChar (50)	Not Null	Usuario que creo el registro
UsuarioModificacion	NVarChar (50)	Null	Usuario que modifica el registro

Fuente: Elaboración propia.

25. Tabla Malla

Tabla que almacena la malla académica a la que pertenecen las asignaturas.

Tabla 50 Diccionario de datos de la tabla «Malla»

Campo	Tipo	Nulo	Comentarios
Id	Int (PK)	Not Null	Identificador de la Malla académica
Nombre	NVarChar (100)	Not Null	Nombre de la malla académica
Año	Int	Not Null	Año en el que se desarrollo la malla académica
Estado	Int	Not Null	Estado en el que se encuentra el registro
FechaModificacion	DateTime2 (7)	Null	Fecha en la que sufre una actualización el registro
FechaCreacion	DateTime2 (7)	Not Null	Fecha en la que se crea el registro
UsuarioCreacion	NVarChar (50)	Not Null	Usuario que creo el registro
UsuarioModificacion	NVarChar (50)	Null	Usuario que modifica el registro

Fuente: Elaboración propia.

26. Tabla Paramétrica

Tabla que almacena configuración o parametrización de valores requeridos para que el sistema gestión de sílabos funcione.

Tabla 51 Diccionario de datos de la tabla «Parametrica»

Campo	Tipo	Nulo	Comentarios
CodigoControl	Int (PK)	Not Null	Código de Control
DescripcionControl	NVarChar (50)	Not Null	Descripción del control
Información	NVarChar (50)	Not Null	Información que se guardará
Correlativo	Int (PK)	Not Null	Correlativo en caso de existir una lista de valores
ValorEntero	Int	Null	Valor de origen entero que se utilice en alguna función
ValorTexto	NVarChar (Max)	Null	Valor de origen texto que se utilice en alguna funcionalidad
ValorFecha	DateTime2 (7)	Null	Valor de origen fecha que se utilice en alguna funcionalidad
Auxiliar1	Int	Null	Valor auxiliar entero que se necesite en la funcionalidad
Auxiliar2	DateTime2 (7)	Null	Valor auxiliar fecha que se necesite en la funcionalidad
Auxiliar3	NVarChar (Max)	Not Null	Valor Auxiliar texto que se necesite en la funcionalidad
Estado	Int	Not Null	Estado en el que se encuentra el registro

FechaModificacion	DateTime2 (7)	Null	Fecha en la que sufre una actualización el registro
FechaCreacion	DateTime2 (7)	Not Null	Fecha en la que se crea el registro
UsuarioCreacion	NVarChar (50)	Not Null	Usuario que creo el registro
UsuarioModificacion	NVarChar (50)	Null	Usuario que modifica el registro

Fuente: Elaboración propia.

27. Tabla Prerrequisito

Tabla que almacena las asignaturas prerrequisitos que tendrá la asignatura desarrollada en el sílabo.

Tabla 52 Diccionario de datos de la tabla «PreRequisito»

Campo	Tipo	Nulo	Comentarios
IdAsignatura	Int (PK, FK)	Not Null	Identificador de la Asignatura desarrollada
IdAsignaturaR	Int (PK, FK)	Not Null	Identificador de la asignatura Pre Requisito
Estado	Int	Not Null	Estado en el que se encuentra el registro
FechaModificacion	DateTime2 (7)	Null	Fecha en la que sufre una actualización el registro
FechaCreacion	DateTime2 (7)	Not Null	Fecha en la que se crea el registro
UsuarioCreacion	NVarChar (50)	Not Null	Usuario que creo el registro
UsuarioModificacion	NVarChar (50)	Null	Usuario que modifica el registro

Fuente: Elaboración propia.

28. Tabla Proyecto

Tabla que almacena los proyectos a desarrollar en la asignatura y en las diferentes fases de esta.

Tabla 53 Diccionario de datos de la tabla «Proyecto»

Campo	Tipo	Nulo	Comentarios
IdAsignatura	Int (PK, FK)	Not Null	Identificador de la Asignatura
IdProyecto	Int (PK)	Not Null	Identificador del proyecto a realizar (1 – Investigación Formativa, 2 – Responsabilidad Social)
IdFase	Int (PK, FK)	Not Null	Fase al que se le asocia el proyecto
NroOrden	Int (PK)	Not Null	Orden en el que se registrará el proyecto
DenominacionActividad	NVarChar (Max)	Not Null	Nombre de la actividad o descripción
Propósito	NVarChar (Max)	Not Null	Propósito que tendrá la actividad
IndicadoresEvaluacion	NVarChar (Max)	Not Null	Indicadores para evaluar
Beneficiarios	NVarChar (Max)	Not Null	Usuarios Beneficiados
Responsables	NVarChar (Max)	Not Null	Encargados del proyecto
Cronograma	NVarChar (Max)	Not Null	Periodo en el que se desarrollará la actividad
PesoPractico	Float	Not Null	Porcentaje Practico
PesoTeorico	Float	Not Null	Porcentaje Teórico
Estado	Int	Not Null	Estado en el que se encuentra el registro
FechaModificacion	DateTime2 (7)	Null	Fecha en la que sufre una actualización el registro
FechaCreacion	DateTime2 (7)	Not Null	Fecha en la que se crea el registro
UsuarioCreacion	NVarChar (50)	Not Null	Usuario que creo el registro

UsuarioModificacion	NVarChar (50)	Null	Usuario que modifica el registro
---------------------	---------------	------	----------------------------------

Fuente: Elaboración propia.

29. Tabla Resultado

Tabla que almacena los resultados ingresados para las fases de la asignatura, en ella se ingresan las descripciones de estas, las síntesis que aparecerán en el plan de estudio y los porcentajes que corresponderán para teoría y práctica.

Tabla 54 Diccionario de datos de la tabla «Resultado»

Campo	Tipo	Nulo	Comentarios
IdAsignatura	Int (PK, FK)	Not Null	Identificador de la Asignatura
IdFase	Int (PK, FK)	Not Null	Fase al que se le asocia el proyecto
IdResultado	Int (PK)	Not Null	Identificador del Resultado (1 – De Conocimiento, 2 – De desempeño y 3 – De Producto)
Orden	Int (PK)	Not Null	Orden en el que se registrará el Resultado
Descripcion	NVarChar (Max)	Not Null	Descripción del resultado
Síntesis	NVarChar (Max)	Not Null	Síntesis que aparecerá en la sección de evaluación de competencias adquiridas
PesoPractico	Float	Not Null	Porcentaje Practico
PesoTeorico	Float	Not Null	Porcentaje Teórico
Estado	Int	Not Null	Estado en el que se encuentra el registro
FechaModificacion	DateTime2 (7)	Null	Fecha en la que sufre una actualización el registro

FechaCreacion	DateTime2 (7)	Not Null	Fecha en la que se crea el registro
UsuarioCreacion	NVarChar (50)	Not Null	Usuario que creo el registro
UsuarioModificacion	NVarChar (50)	Null	Usuario que modifica el registro

Fuente: Elaboración propia.

30. Tabla Sumilla

Tabla que almacena las sumillas redactadas por el docente.

Tabla 55 Diccionario de datos de la tabla «Sumilla»

Campo	Tipo	Nulo	Comentarios
Id	Int (PK)	Not Null	Identificador de la sumilla
IdSilabo	Int (FK)	Null	Identificador de la Asignatura
Descripcion	NVarChar (Max)	Not Null	Descripción de la sumilla
Estado	Int	Not Null	Estado en el que se encuentra el registro
FechaModificacion	DateTime2 (7)	Null	Fecha en la que sufre una actualización el registro
FechaCreacion	DateTime2 (7)	Not Null	Fecha en la que se crea el registro
UsuarioCreacion	NVarChar (50)	Not Null	Usuario que creo el registro
UsuarioModificacion	NVarChar (50)	Null	Usuario que modifica el registro

Fuente: Elaboración propia.

31. Tabla Unidad

Tabla que almacena las unidades que desarrollará el docente en la asignatura, así como también las sub – unidades o temas de cada unidad.

Tabla 56 *Diccionario de datos de la tabla «Unidad»*

Campo	Tipo	Nulo	Comentarios
IdAsignatura	Int (PK, FK)	Not Null	Identificador de la Asignatura
NroUnidad	Int (PK)	Not Null	Nro de la Unidad a trabajar
NroSubUnidad	Int (PK)	Not Null	Nro de Sub Unidad a trabajar
Descripción	NVarChar (Max)	Null	Descripción que tendrá el registro
IdFase	Int (PK)	Not Null	Fase a la que pertenecerá la unidad o subunidad
Estado	Int	Not Null	Estado en el que se encuentra el registro
FechaModificacion	DateTime2 (7)	Null	Fecha en la que sufre una actualización el registro
FechaCreacion	DateTime2 (7)	Not Null	Fecha en la que se crea el registro
UsuarioCreacion	NVarChar (50)	Not Null	Usuario que creo el registro
UsuarioModificacion	NVarChar (50)	Null	Usuario que modifica el registro

Fuente: Elaboración propia.

32. Tabla Unidad Laboratorio

Tabla que almacena las unidades de laboratorio o práctica de la asignatura, así como su relación en la fase correspondiente.

Tabla 57 Diccionario de datos de la tabla «UnidadLab»

Campo	Tipo	Nulo	Comentarios
IdAsignatura	Int (PK, FK)	Not Null	Identificador de la Asignatura
NroUnidadLab	Int (PK)	Not Null	Nro de la Unidad a trabajar
Descripción	NVarChar (Max)	Null	Descripción que tendrá el registro
IdFase	Int (PK)	Not Null	Fase a la que pertenecerá la unidad o subunidad
Estado	Int	Not Null	Estado en el que se encuentra el registro
FechaModificacion	DateTime2 (7)	Null	Fecha en la que sufre una actualización el registro
FechaCreacion	DateTime2 (7)	Not Null	Fecha en la que se crea el registro
UsuarioCreacion	NVarChar (50)	Not Null	Usuario que creo el registro
UsuarioModificacion	NVarChar (50)	Null	Usuario que modifica el registro

Fuente: Elaboración propia.

2.3.7 Implementación De Funcionalidades Principales

Las funcionalidades principales como ya se mencionó fueron desarrolladas dividiendo la parte de las interfaces o Front-End con los componentes necesarios para el despliegue de las vistas utilizadas por el docente desarrollado principalmente en REACT y el API o Back-End que contiene los servicios que listan o guardan la información manipulada en la vista que se desarrolló principalmente en lenguaje C#. A continuación, se muestra la codificación más importante tanto para Front-End, como para Back-End.

2.3.7.1 Front-End

- **Componente de autenticación por Azure**

Se implementa componente de autenticación para azure, redireccionando a la aplicación en cuanto tenga un acceso satisfactorio y se genera un token de acceso provista por el api Offices 365 (línea 9). Se definen los botones de acceso a la aplicación, botón de login (línea 34) y de logout (línea 47). Según el estado de la sesión del usuario la visualización de los botones de ingreso o salida de la aplicación cuando la aplicación se encuentra bien configurada en la plataforma de Azure (línea 58), en caso de no estar bien configurada, se envía mensaje error de configuración al usuario (línea 62) en la Figura 66.

Figura 66 *Componente para autenticación por Azure*

```

1  import React, { useState } from "react";
2  import AzureAuthenticationContext from "../azure-authentication-context";
3  const ua = window.navigator.userAgent;
4  const msie = ua.indexOf("MSIE ");
5  const msie11 = ua.indexOf("Trident/");
6  const isIE = msie > 0 || msie11 > 0;
7  //-- Componente para AzureAuthentication, implementación de login y logout por api Office 365
8  //-- por botones definidos para la aplicación
9  const AzureAuthenticationButton = ({ onAuthenticated }) => {
10     // Azure client context
11     const authenticationModule = new AzureAuthenticationContext();
12     const [authenticated, setAuthenticated] = useState(false);
13     const [user, setUser] = useState();
14     const logIn = (method) => {
15         const typeName = "loginPopup";
16         const loginType = isIE ? "loginRedirect" : typeName;
17         // Azure Login
18         authenticationModule.login(loginType, returnedAccountInfo);
19     };
20     const logOut = () => {
21         if (user) {
22             onAuthenticated(undefined);
23             // Azure Logout
24             authenticationModule.logout(user);
25         }
26     };
27     const returnedAccountInfo = (user) => {
28         // asignar estado
29         setAuthenticated(user?.name ? true : false);
30         onAuthenticated(user);
31         setUser(user);
32     };
33     //-- Botón Login para ingresar a la aplicación
34     const showLogInButton = () => {
35         return (
36             <button
37                 id="authenticationButton"
38                 className="w-full px-4 py-2 font-bold text-white rounded shadow cursor-pointer m-t-5 bg-green-cato hover:bg-green-cato-dark focus:shadow-outline focus:outline-none"
39                 type="submit"
40                 onClick={() => logIn("loginPopup")}
41             >
42                 Logearse
43             </button>
44         );
45     };
46     //-- Botón Logout para salir de aplicación al usar el evento click
47     const showLogOutButton = () => {
48         return (
49             <div id="authenticationButtonDiv">
50                 <div id="authentication" className="w-full">
51                     <button id="authenticationButton" onClick={() => logOut()}>
52                         Log out
53                     </button>
54                 </div>
55             </div>
56         );
57     };
58     const showButton = () => {
59         return authenticated ? showLogOutButton() : showLogInButton();
60     };
61     //-- Verificar si la autenticación está configurada para la aplicación.
62     return (
63         <div id="authentication" className="w-full">
64             {authenticationModule.isAuthenticationConfigured ? (
65                 showButton()
66             ) : (
67                 <div>Authentication Client ID is not configured.</div>
68             )}
69         </div>
70     );
71 };

```

Fuente: Elaboración propia.

- **Enrutamiento de páginas iniciales.**

En caso de que se tenga al usuario en sesión dentro de la aplicación, se procede direccionar a la ruta raíz de la aplicación (línea 23), como se observa en la Figura 67.

Figura 67 Ingreso y redireccionamiento a vista raíz

```

1 import React, { useState } from "react";
2 import {
3   BrowserRouter as Router,
4   Switch,
5   Route,
6   NavLink,
7 } from "react-router-dom";
8 import { connect } from "react-redux";
9 import Syllabuses from "./pages/Syllabuses";
10 import Login from "./pages/Login";
11 import Courses from "./pages/Courses";
12 import EnableSyllabuses from "./pages/EnableSyllabuses";
13 import LoadSyllabuses from "./pages/LoadSyllabuses";
14 import Spinner from "./components/Spinner";
15 import { logout } from "./store/saveSlice";
16 import { FontAwesomeIcon } from "@fortawesome/react-fontawesome";
17 import logo from "./assets/logo.png";
18 import { faSignOutAlt, faUser } from "@fortawesome/free-solid-svg-icons";
19
20 import classNames from "classnames";
21
22 //--Verificar si existe un usuario en el intento de login. Si existe usuario, redireccionar a vista raíz.
23 function App(props) {
24   const [isMobileMenuOpen, setIsMobileMenuOpen] = useState(false);
25   if (!props.user) {
26     return (
27       <Router>
28         <Switch>
29           <Route exact path="/">
30             <Login />
31           </Route>
32         </Switch>
33       </Router>
34     );
35   }
36

```

Fuente: Elaboración propia.

Se establecen a su vez los componentes que integrarán el menú de la aplicación, pantalla raíz y los apartados de visualizar sílabo, habilitar sílabo, cursos, cargar sílabos (línea 96). El enrutamiento se dará al momento de seleccionar el componente (línea 125) como se aprecia en la Figura 68.

Figura 68 *Navegación de Menú*

```

94 // Creación de la navegación en el menú de la aplicación de sílabos.
95 <NavLink
96   to="/"
97   exact
98   className="block px-6 py-4 border-b text-menu-secondary border-menu-primary hover:text-menu-text-hover"
99   activeClassName="text-menu-text-highlight font-semibold"
100 >
101   Visualizar Sílabo
102 </NavLink>
103 <NavLink
104   to="/habilitar-silabos"
105   exact
106   className="block px-6 py-4 border-b text-menu-secondary border-menu-primary hover:text-menu-text-hover"
107   activeClassName="text-menu-text-highlight font-semibold"
108 >
109   Habilitar Sílabo
110 </NavLink>
111 <NavLink
112   to="/cursos"
113   className="block px-6 py-4 border-b text-menu-secondary border-menu-primary hover:text-menu-text-hover"
114   activeClassName="text-menu-text-highlight font-semibold"
115 >
116   Cursos
117 </NavLink>
118 <NavLink
119   to="/cargar-silabos"
120   className="block px-6 py-4 border-b text-menu-secondary border-menu-primary hover:text-menu-text-hover"
121   activeClassName="text-menu-text-highlight font-semibold"
122 >
123   Cargar Sílabos
124 </NavLink>
125 </nav>
126 </div>
127 
```

”ListarProgramaciónPorFaseDeAprendizaje”, “ListarAsignaturaBibliografia”,
“ListarDocenteFirma”, “ListarProyectosAsignatura”, “ListarAsignaturaBibliografiaAgrupada”
(línea 75), que obtendrán la información básica y migrada de la asignatura, así como también la
información desarrollada por el docente en el transcurso de la elaboración de la asignatura. Todo
esto se observa en la Figura 69.

Figura 69 Método "generatePDF"

```

59  |-- Método que genera el contenido del sílabo
60  export const generatePdf = async (idSyllabus, onDone) => {
61      const [
62          { data: academicIdentification },
63          { data: summary },
64          { data: egressProfileRaw },
65          { data: competencePhases },
66          { data: learningUnitsTheory },
67          { data: learningUnitsLaboratory },
68          { data: resultPhases },
69          { data: instruments },
70          { data: trainingProgramData },
71          { data: bibliographyData },
72          { data: signs },
73          { data: researchActivity },
74          { data: bibliographyDataSummary },
75      ] = await Promise.all([
76          api.post("GestionSilabo/VerIdentificacionAcademica", {
77              idSilabo: idSyllabus,
78          }),
79          api.post("GestionSilabo/BuscarSumillaPorSilabo", {
80              idSilabo: idSyllabus,
81          }),
82          api.post("GestionSilabo/ListarCompetenciasPerfilEgreso", {
83              idSilabo: idSyllabus,
84          }),
85          api.post("GestionSilabo/ListarCompetenciasPropiasAsignatura", {
86              idSilabo: idSyllabus,
87          }),
88          api.post("GestionSilabo/ListarUnidades", {
89              idSilabo: idSyllabus,
90          }),
91          api.post("GestionSilabo/ListarUnidadesLab", {
92              idSilabo: idSyllabus,
93          }),
94          api.post("GestionSilabo/ListarResultados", {
95              idSilabo: idSyllabus,
96          }),
97          api.post("GestionSilabo/ListarAsignaturaInstrumentoAgrupados", {
98              idSilabo: idSyllabus,
99          }),
100         api.post("ProgramaFormativo/ListarProgramacionPorFaseDeAprendizaje", {
101             idSilabo: idSyllabus,
102         }),
103         api.post("GestionSilabo/ListarAsignaturaBibliografia", {
104             idSilabo: idSyllabus,
105         }),
106         api.post("GestionSilabo/ListarDocentesFirma", {
107             idSilabo: idSyllabus,
108         }),
109         api.post("ProgramaFormativo/ListarProyectosAsignatura", {
110             idSilabo: idSyllabus,
111         }),
112         api.post("GestionSilabo/ListarAsignaturaBibliografiaAgrupada", {
113             idSilabo: idSyllabus,
114         }),
115     ]);
116 }

```

Fuente: Elaboración propia.

La definición del documento PDF, se dividirá en dos secciones: Una será para lo correspondiente al Sílabo de la asignatura o plan de estudios, y otra sección corresponde al programa formativo. Si bien su construcción es independiente, la generación debe estar compuesta por ambas como se indica en la Figura 70.

Figura 70 Definición de contenido del documento

```
1258 //-- Definición de las secciones de documento: Sílabo y programa formativo
1259 const docDefinition = {
1260   content: [...sillabus, ...trainingProgram],
1261   styles: {
1262     tableExample: {
1263       margin: [0, 5, 0, 15],
1264     },
1265     tableMedium: {
1266       margin: [0, 5, 0, 15],
1267       fontSize: 9,
1268     },
1269     tableSmall: {
1270       margin: [0, 5, 0, 15],
1271       fontSize: 8.5,
1272     },
1273   },
1274   defaultStyle: {
1275     fontSize: 10,
1276   },
1277 };
1278 const pdfDocGenerator = pdfMake.createPdf(docDefinition);
1279 pdfDocGenerator.getDataUrl((dataUrl) => {
1280   onDone(dataUrl);
1281 });
1282 };
```

Fuente: Elaboración propia.

Se ingresa el contenido de la facultad, escuela profesional de manera dinámica, de manera estática los títulos de plan de estudios, y sílabos de asignatura. Se obtiene y muestra el nombre de la asignatura y su código (línea 262) en la Figura 62. Se aprecia también el diseño de la tabla que alberga los créditos, y horas semestrales y semanales (línea 322), para el diseño se recurre a

la división del espacio en 8 columnas. Internamente se realiza la expansión de columnas y filas para mantener el formato en la Figura 71 y 72.

Figura 71 *Front - Facultad, Escuela, Identificación Académica*

```

260 //-- Datos de títulos con valores dinámicos.
261 {
262   text: [
263     { text: "FACULTAD: ", bold: true },
264     academicIdentification.atributos?.facultadNombre,
265   ],
266   margin: [0, 10, 0, 0],
267 },
268 {
269   text: [
270     { text: "ESCUELA PROFESIONAL: ", bold: true },
271     academicIdentification.atributos?.escuelaNombre,
272   ],
273   margin: [0, 10, 0, 0],
274 },
275 {
276   text: "PLAN DE ESTUDIOS",
277   alignment: "center",
278   bold: true,
279   fontSize: 13,
280   margin: [0, 10, 0, 0],
281 },
282 {
283   text: "SÍLABO DE ASIGNATURA",
284   alignment: "center",
285   bold: true,
286   fontSize: 13,
287   margin: [0, 0, 0, 10],
288 },
289 { text: "1. IDENTIFICACIÓN ACADÉMICA", bold: true },
290 {
291   text: [
292     "1.1.- Nombre de la Asignatura: ",
293     {
294       text: academicIdentification.atributos?.asignaturaNombre,
295       bold: true,
296     },
297   ],
298 },
299 {
300   text: [
301     { text: "Código de la Asignatura: " },
302     {
303       text: academicIdentification.atributos?.codigoAsignatura,
304       bold: true,
305     },
306   ],
307   margin: [26, 0, 0, 0],
308 },
309 {
310   text: [
311     { text: "Semestre Académico en que se desarrolla: " },
312     {
313       text: academicIdentification.atributos?.semestre,
314       bold: true,
315     },
316   ],
317   margin: [26, 0, 0, 0],

```

Fuente: Elaboración propia.

Figura 72 *Front - Tabla para horas semestrales y semanales*

```

322 { //-- Tabla para la cantidad de horas, semanales y semestrales.
323 style: "tableExample",
324 table: {
325   widths: [
326     "auto",
327     "auto",
328     "auto",
329     "auto",
330     "auto",
331     "auto",
332     "auto",
333     "auto",
334   ],
335   body: [
336     [
337       { text: "CRÉDITOS", rowspan: 3, bold: true },
338       { text: "HORAS SEMANALES", colspan: 4, bold: true },
339       "",
340       "",
341       "",
342       { text: "HORAS SEMESTRALES", colspan: 3, bold: true },
343       "",
344       "",
345     ],
346     [
347       "",
348       { text: "Horas Teóricas", rowspan: 2, bold: true },
349       { text: "Horas Prácticas", colspan: 2, bold: true },
350       "",
351       { text: "Horas Virtuales", rowspan: 2, bold: true },
352       { text: "Horas Teóricas", rowspan: 2, bold: true },
353       { text: "Horas Prácticas", rowspan: 2, bold: true },
354       { text: "Horas Virtuales", rowspan: 2, bold: true },
355     ],
356     [
357       "",
358       "",
359       { text: "Práctica de Aula", bold: true },
360       { text: "Jefe de Prácticas", bold: true },
361       "",
362       "",
363       "",
364       "",
365     ],
366     [
367       academicIdentification.atributos?.creditos,
368       academicIdentification.atributos?.horasSemanalesTeoricas,
369       academicIdentification.atributos?.horasSemanalesPracticaAula,
370       academicIdentification.atributos?.horasSemanalesPracticaJefe,
371       academicIdentification.atributos?.horasSemanalesVirtuales,
372       academicIdentification.atributos?.horasSemestralesTeoricas,
373       academicIdentification.atributos?.horasSemestralesPracticas,
374       academicIdentification.atributos?.horasSemestralesVirtuales,
375     ],
376   ],
377 },
378 layout: {
379   fillColor: function (rowIndex, node, columnIndex) {
380     const rowsToFill = new Set([0, 1, 2]);
381     return rowsToFill.has(rowIndex) ? "#DADADA" : null;
382   },
383 },
384 },

```

Fuente: Elaboración propia.

Se desarrolla la inserción de los valores múltiples para los conceptos de cursos equivalentes, cursos prerequisites, lista de competencias que apoyan al perfil de egreso y competencias propias de la asignatura (línea 387) de la Figura 73.

Figura 73 Front - *Equivalencias, prerequisites, competencias propias y que apoyan al perfil*

```

385 //-- Ingreso de códigos y nombres para conceptos con más de un valor en respuesta a los servicios de
386 //-- Equivalencias, prerequisites, competencias del perfil y competencias propias de la asignatura
387 {
388   text: "1.3.- Código, nombre y créditos de Asignaturas Equivalentes:",
389 },
390 ...academicIdentification.equivalentes.map((course) => ({
391   text: `${course.codigo} ${course.nombreEquivalente}`,
392   bold: true,
393   margin: [26, 0, 0, 0],
394 })),
395 {
396   text: "1.4.- Código y nombre de Asignaturas Pre-requisito:",
397 },
398 ...academicIdentification.preRequisitos.map((course) => ({
399   text: `${course.codigo} ${course.nombrePreRequisito}`,
400   bold: true,
401   margin: [26, 0, 0, 0],
402 })),
403 { text: "2. SUMILLA", bold: true, margin: [0, 10, 0, 0] },
404 {
405   text: summary.descripcion,
406 },
407 {
408   text: "3. COMPETENCIAS DEL PERFIL DE EGRESO A LAS QUE CONTRIBUYE ASIGNATURA",
409   bold: true,
410   margin: [0, 10, 0, 0],
411 },
412 ...egressProfile.flatMap((profile) => [
413   {
414     text: `${profile.codigo}. ${profile.nombre}`,
415     decoration: "underline",
416   },
417   {
418     text: profile.descripcion,
419   },
420 ]),
421 {
422   text: "4. COMPETENCIAS PROPIAS DE LA ASIGNATURA",
423   bold: true,
424   margin: [0, 10, 0, 0],
425 },
426 { text: "PRIMERA FASE", bold: true },
427 {
428   ul: competencePhase1.map((element) => element.descripcion),
429 },
430 { text: "SEGUNDA FASE", bold: true },
431 {
432   ul: competencePhase2.map((element) => element.descripcion),
433 },
434 { text: "TERCERA FASE", bold: true },
435 {
436   ul: competencePhase3.map((element) => element.descripcion),
437 },
438 {

```

Fuente: Elaboración propia.

Se ingresa los contenidos básicos por unidades de aprendizaje (línea 440), separando el ingreso para cada fase de manera individual, por otro lado la evaluación de competencias solo es

rellenada con las evidencias e instrumentos (línea 464), por último la bibliografía se mapea con los valores devueltos (línea 501) en la Figura 74.

Figura 74 *Unidades, Evaluación de competencias y Bibliografía*

```

438 | //-- Desarrollo de contenidos basicos por unidad de aprendizaje.
439 | {
440 |     text: "5. CONTENIDOS BÁSICOS POR UNIDADES DE APRENDIZAJE:",
441 |     bold: true,
442 |     margin: [0, 10, 0, 0],
443 | },
444 | {
445 |     text: "CLASES TEÓRICAS:",
446 |     bold: true,
447 | },
448 | ...learningUnitsTheoryPhasePdf(learningUnitsTheoryPhase1, "PRIMERA FASE"),
449 | ...learningUnitsTheoryPhasePdf(learningUnitsTheoryPhase2, "SEGUNDA FASE"),
450 | ...learningUnitsTheoryPhasePdf(learningUnitsTheoryPhase3, "TERCERA FASE"),
451 | { text: "PRIMERA FASE LABORATORIO", bold: true },
452 | {
453 |     ul: learningUnitsLaboratoryPhase1.map((element) => element.descripcion),
454 | },
455 | { text: "SEGUNDA FASE LABORATORIO", bold: true },
456 | {
457 |     ul: learningUnitsLaboratoryPhase2.map((element) => element.descripcion),
458 | },
459 | { text: "TERCERA FASE LABORATORIO", bold: true },
460 | {
461 |     ul: learningUnitsLaboratoryPhase3.map((element) => element.descripcion),
462 | },
463 | //-- Desarrollo de evaluación de competencias adquiridas, para los puntos de evidencias e instrumentos
464 | {
465 |     text: "6. EVALUACIÓN DE COMPETENCIAS ADQUIRIDAS",
466 |     bold: true,
467 |     margin: [0, 10, 0, 0],
468 | },
469 | {
470 |     text: "EVIDENCIAS DE CONOCIMIENTO:",
471 |     bold: true,
472 | },
473 | {
474 |     ul: knowledgeResult.map(
475 |         (element) => element.sintesis || element.descripcion
476 |     ),
477 | },
478 | {
479 |     text: "EVIDENCIAS DE DESEMPEÑO:",
480 |     bold: true,
481 | },
482 | {
483 |     ul: performanceResult.map(
484 |         (element) => element.sintesis || element.descripcion
485 |     ),
486 | },
487 | {
488 |     text: "EVIDENCIAS DE PRODUCTO:",
489 |     bold: true,
490 | },
491 | {
492 |     ul: productResult.map(
493 |         (element) => element.sintesis || element.descripcion
494 |     ),
495 | },
496 | { text: "INSTRUMENTOS DE EVALUACIÓN:", bold: true },
497 | {
498 |     ul: instruments.map((element) => element.descripcion),
499 | },
500 | //-- Desarrollo de bibliografía
501 | {
502 |     text: "7. BIBLIOGRAFÍA",
503 |     bold: true,
504 |     margin: [0, 10, 0, 0],
505 | },
506 | {
507 |     ul: bibliographyDataSummary.map((element) => {
508 |         return `${element.autor}. (${element.anioPublicacion}). ${element.nombreLibro}, ${element.editorial}`;
509 |     }),
510 |     pageBreak: "after",
511 | },
512 | ];

```

Fuente: Elaboración propia.

2.3.7.2 Back-End

- Servicio “VerIndentificacionAcademica”

En la parte inicial del desarrollo de sílabo se tiene la sección de Identificación académica, en la que figuran los datos esenciales de la asignatura obtenidas en la migración de datos a la base del sistema de gestión de sílabos.

El Servicio evalúa el código recibido para determinar si el tipo de dato es correspondiente al que se usa en los identificadores de las asignaturas (línea 52). En caso de no coincidir el tipo de dato, se devuelve mensaje de error (línea 54, línea 63).

Figura 75 Servicio "VerIndentificacionAcademica"

```

37 //-- Retorna el bloque de identificación académica que puede servir para el punto 1 de:
38 //-- Plan de Estudio
39 //-- Programa formativo
40
41 1 referencia
42 public class VerIndentificacionAcademicaParametros
43 {
44     4 referencias
45     public string idSilabo { get; set; }
46 }
47
48 [HttpPost]
49 [Route("VerIndentificacionAcademica")]
50 0 referencias
51 public ActionResult<IdentificacionAcademicaDTO> VerIndentificacionAcademica(VerIndentificacionAcademicaParametros parametro)
52 {
53     //-- Verificar valor del parámetro sílabo, en caso de que existir un valor inadecuado se retorna mensaje de error.
54     if (parametro.idSilabo == null)
55     {
56         return BadRequest("Información errónea.");
57     }
58     if (parametro.idSilabo == "")
59     {
60         parametro.idSilabo = "0";
61     }
62     int idSil;
63     if (!Int32.TryParse(parametro.idSilabo, out idSil))
64     {
65         return BadRequest("Información errónea.");
66     }
67     else
68     {
69         _handler = new GestionSilabos.Handler.Handler(_context);
70
71         //-- Se instancia objeto IdentificaciónAcadémicaDTO que servirá como respuesta a la petición de consulta y contiene
72         //-- la información pertinente de identificación académica.
73         IdentificacionAcademicaDTO identificacion = new IdentificacionAcademicaDTO();
74         identificacion = _handler.ObtenerIdentificacionAcademica(idSil);
75
76         return identificacion;
77     }

```

Fuente: Elaboración propia.

Si supera la validación se llama al método “ObtenerIdentificacionAcademica” de la clase Handler (línea 73). El método lista la información de los atributos de la asignatura en su parte inicial (línea 41) para posteriormente realizar el casteo (línea 81) y asignar el valor a un objeto de atributos de la identificación que se asigna al DTO final.

Figura 76 Método "ObtenerIdentificacionAcademica" atributos

```

38 public IdentificacionAcademicaDTO ObtenerIdentificacionAcademica(int idSil)
39 {
40     //-- Obtener datos esenciales de la asignatura de las entidades relacionadas precargadas.
41     var idenAcadeAtributos = from silabo in _context.AsignaturaDocenteFuncion
42                             join asignatura in _context.Asignatura on silabo.IdAsignatura equals asignatura.Id
43                             join escuela in _context.Escuela on asignatura.IdEscuela equals escuela.Id
44                             join departamento in _context.Departamento on escuela.IdDepartamento equals departamento.Id
45                             join facultad in _context.Facultad on escuela.IdFacultad equals facultad.Id
46                             where asignatura.Id == idSil
47                             select new IdentificacionAcademicaAtributosDTO
48                             {
49                                 FacultadNombre = facultad.Nombre,
50                                 DepartamentoNombre = departamento.Nombre,
51                                 EscuelaNombre = escuela.Nombre,
52                                 AsignaturaNombre = asignatura.Nombre,
53                                CodigoAsignatura = asignatura.Codigo,
54                                 Semestre = (asignatura.Semestre == 1 ? "I" :
55                                             asignatura.Semestre == 2 ? "II" :
56                                             asignatura.Semestre == 3 ? "III" :
57                                             asignatura.Semestre == 4 ? "IV" :
58                                             asignatura.Semestre == 5 ? "V" :
59                                             asignatura.Semestre == 6 ? "VI" :
60                                             asignatura.Semestre == 7 ? "VII" :
61                                             asignatura.Semestre == 8 ? "VIII" :
62                                             asignatura.Semestre == 9 ? "IX" :
63                                             asignatura.Semestre == 10 ? "X" :
64                                             asignatura.Semestre == 11 ? "XI" : "XII"),
65                                 AnioAcademico = asignatura.AnioAcademico,
66                                 Creditos = asignatura.Creditos,
67                                 HorasSemanalesTeoricas = asignatura.HorasSemanalesTeoricas,
68                                 HorasSemanalesPracticaAula = asignatura.HorasSemanalesPracticaAula,
69                                 HorasSemanalesPracticaJefe = asignatura.HorasSemanalesPracticaJefe,
70                                 HorasSemanalesVirtuales = asignatura.HorasSemanalesVirtuales,
71                                 HorasSemestralesTeoricas = asignatura.HorasSemestralesTeorica,
72                                 HorasSemestralesVirtuales = asignatura.HorasSemestralesVirtuales,
73                                 HorasSemestralesPracticas = asignatura.HorasSemestralesPractica,
74                                 AmbienteTeoria = asignatura.AmbienteTeoria,
75                                 AmbientePractica = asignatura.AmbientePractica
76                             };
77
78
79
80     //-- Se castea la información recuperada para convertir de un objeto IQueryable a un objeto IdentificacionAcademicaAtributosDTO
81     IdentificacionAcademicaAtributosDTO idacatri = idenAcadeAtributos.AsQueryable().Cast<IdentificacionAcademicaAtributosDTO>().FirstOrDefault();
82
83

```

Fuente: Elaboración propia.

La siguiente parte del método resuelve los docentes (línea 84) que dictan el curso para la parte teórica y práctica, así como los prerrequisitos que tendrá la asignatura en el semestre actual (línea 98) y finalmente los cursos equivalentes de otras mallas académicas (línea 110).

Por último se asignan los resultados de las variables anónimas al objeto `IdentificacionAcademicaDTO` (línea 122) para tener ordenada la información que se enviará como `.JSON` a la interfaz.

Figura 77 Método "*ObtenerIdentificacionAcademica*" docentes, prerequisitos y equivalentes

```

83  //-- Obtener los docentes que dictan la asignatura en la parte teórica y práctica, junto con demás datos
84  var docentes = _context.AsignaturaDocenteFuncion.Join(_context.Docente, sdf => sdf.CodigoDocente, doc => doc.Codigo, (sdf, doc) => new { sdf, doc })
85  .Join(_context.Funcion, sdf => sdf.sdf.IdFuncion, fun => fun.Id, (doc, fun) => new { doc, fun })
86  .Where(x => x.doc.sdf.IdAsignatura == idSil)
87  .Select(Doc => new DocenteDTO
88  {
89
90      Categoria = Doc.doc.doc.Categoria,
91      Codigo = Doc.doc.doc.Codigo,
92      NombreDocente = (Doc.doc.doc.Apellidos + ", " + Doc.doc.doc.Nombres),
93      Funcion = Doc.fun.Descripcion
94  });
95  });
96
97  //-- Se obtienen los prerequisitos de la asignatura
98  var preRequisito = _context.Asignatura.Join(_context.PreRequisito, asig => asig.Id, preR => preR.IdAsignatura, (asig, preR) => new { asig, preR })
99  .Join(_context.Asignatura, preR => preR.preR.IdAsignaturaR, asig => asig.Id, (preR, asig) => new { preR, asig })
100 .Where(x => x.preR.asig.Id == idSil)
101 .Select(Pre => new PreRequisitoDTO
102 {
103
104     Codigo = Pre.asig.Codigo,
105     NombrePreRequisito = Pre.asig.Nombre
106 });
107 });
108
109 //-- Se obtienen los cursos equivalentes de la asignatura.
110 var equivalente = _context.Asignatura.Join(_context.Equivalente, asig => asig.Id, equi => equi.IdAsignatura, (asig, equi) => new { asig, equi })
111 .Where(x => x.asig.Id == idSil)
112 .Select(Equi => new EquivalenteDTO
113 {
114
115     Codigo = Equi.equi.Codigo,
116     NombreEquivalente = Equi.equi.Nombre,
117     Creditos = Equi.equi.Creditos
118 });
119 });
120
121 //-- Se asignan los resultados al objeto IdentificacionAcademicaDTO para retornar la información en un solo objeto.
122 IdentificacionAcademicaDTO identificacion = new IdentificacionAcademicaDTO
123 {
124     Atributos = idacatri,
125     Docentes = docentes,
126     Equivalentes = equivalente,
127     PreRequisitos = preRequisito
128 };
129 return identificacion;
130 }

```

Fuente: Elaboración propia.

- **Servicio “BuscarSumillaPorSilabo”**

Este servicio permite la búsqueda de la sumilla por el código de una asignatura específica.

El Servicio evalúa el código recibido para determinar si el tipo de dato es correspondiente al que se usa en los identificadores de las asignaturas (línea 93). Se busca el identificador de la

sumilla (línea 110) para determinar si existe o no una sumilla ingresada. El resultado se retorna en el objeto SumillaDTO (línea 121).

Figura 78 Método "BuscarSumillaPorSilabo"

```

79  //-- Servicio que permite buscar la sumilla de un determinado silabo de para el punto 2 de:
80  //-- Plan de Estudio
81  //-- Programa formativo
82  1 referencia
83  public class BuscarSumillaPorSilaboParametro
84  {
85      4 referencias
86      public string idSilabo { get; set; }
87  }
88
89  [HttpPost]
90  [Route("BuscarSumillaPorSilabo")]
91  0 referencias
92  public ActionResult<SumillaDTO> BuscarSumillaPorSilabo(BuscarSumillaPorSilaboParametro parametro)
93  {
94      //-- Verificar valor del parámetro sílabo, en caso de que existir un valor inadecuado se retorna mensaje de error.
95      if (parametro.idSilabo == null) {...}
96      if (parametro.idSilabo == "") {...}
97      int idSil;
98      if (!Int32.TryParse(parametro.idSilabo, out idSil)) {...}
99      else
100     {
101         //-- Obtener Sumilla de asignatura a través del código de asignatura o sílabo.
102         SumillaModel sumilla = new SumillaModel();
103         int? idSumilla = _context.Asignatura.Where(x => x.Id == idSil).Select(s => s.IdSumilla).FirstOrDefault();
104         if (idSumilla == 0 || idSumilla == null) {
105             sumilla.Id = 0;
106             sumilla.Descripcion = "";
107         }
108         else
109         {
110             sumilla = _context.Sumilla.Where(s => s.Id == idSumilla).FirstOrDefault();
111         }
112
113         //-- Mapeo de objeto sumilla hacia Objeto SumillaDTO y retorno del objeto.
114         var sumilladto = _mapper.Map<SumillaDTO>(sumilla);
115         return sumilladto;
116     }
117 }
118
119
120
121
122
123
124

```

Fuente: Elaboración propia.

- **Servicio “GuardarSumillaEnSilabo”**

Este servicio permite guardar la sumilla por el código de una asignatura específica.

El Servicio evalúa el código recibido para determinar si el tipo de dato es correspondiente al que se usa en los identificadores de las asignaturas (línea 140). Superada la validación, se procede a invocar el método GrabarSumillaSilabo() (línea 159) para realizar la inserción.

Figura 79 Método "GuardarSumillaEnSilabo"

```

126  //-- Servicio que permite Guardar la sumilla de un determinado silabo de para el punto 2 del plan de estudio.
127  1 referencia
128  public class GuardarSumillaEnSilaboParametros
129  {
130      2 referencias
131      public SumillaDTO sumilla { get; set; }
132      4 referencias
133      public string idSilabo { get; set; }
134  }
135
136  [HttpPost]
137  [Route("GuardarSumillaEnSilabo")]
138  0 referencias
139  public JsonResult GuardarSumillaEnSilabo(GuardarSumillaEnSilaboParametros parametro)
140  {
141      //-- Verificar valor del parámetro silabo, en caso de que existir un valor inadecuado se retorna mensaje de error.
142      JsonResult json;
143      if (parametro.idSilabo == null)
144      {
145          var resultado = new { estado = "fail", mensaje = "Información errónea." };
146          return json = new JsonResult(resultado);
147      }
148      if (parametro.idSilabo == "")
149      {
150          parametro.idSilabo = "0";
151      }
152      int idSil;
153      if (!Int32.TryParse(parametro.idSilabo, out idSil))
154      {
155          var resultado = new { estado = "fail", mensaje = "Información errónea." };
156          return json = new JsonResult(resultado);
157      }
158      else
159      {
160          //-- En caso de que el código o valor recibida sea correcto, grabar información de sumilla.
161          _handler = new GestionSilabos.Handler.Handler(_context);
162          _handler.GrabarSumillaSilabo(idSil, parametro.sumilla.Id, parametro.sumilla.Descripcion);
163          var resultado = new { estado = "success", mensaje = "Se actualizó correctamente el estado." };
164          return json = new JsonResult(resultado);
165      }
166  }

```

Fuente: Elaboración propia.

Si existe un identificador quiere decir que ya existe una sumilla ingresada, en ese caso se actualiza la descripción (línea 137). Pero en caso de no existir se genera un nuevo objeto (línea 151) con el detalle de la sumilla. Posteriormente se recupera el identificador nuevo generado par

almacenar el valor en la tabla de asignatura (línea 165). Finalmente se almacena en el log la acción de actualización o inserción sobre la sección de sumilla.

Figura 80 Método "GrabarSumillaSilabo"

```

132 public void GrabarSumillaSilabo(int idSil, int idsumilla, string descripcion) {
133
134     String Concepto = "SUMILLA";
135     String Descripcion = "";
136     //-- Si existe un identificador hay sumilla, se procede a actualizar.
137     if (idsumilla > 0)
138     {
139         var sumillaAntigua = _context.Sumilla.Where(s => s.Id == idsumilla).FirstOrDefault();
140         sumillaAntigua.UsuarioModificacion = "User";
141         sumillaAntigua.FechaModificacion = DateTime.Now;
142         sumillaAntigua.Estado = 1;
143         sumillaAntigua.Descripcion = descripcion;
144         _context.Sumilla.Update(sumillaAntigua);
145         _context.SaveChanges();
146         Descripcion = "SUMILLA ACTUALIZADA";
147     }
148     //-- Si no existe un identificador se genera un nuevo objeto para guardar la nueva sumilla.
149     else
150     {
151         SumillaModel sumilla = new SumillaModel
152         {
153             UsuarioCreacion = "User",
154             UsuarioModificacion = "User",
155             FechaCreacion = DateTime.Now,
156             FechaModificacion = DateTime.Now,
157             Estado = 1,
158             Descripcion = descripcion,
159             IdSilabo = idSil
160         };
161         _context.Sumilla.Add(sumilla);
162         _context.SaveChanges();
163
164         //-- Recuperar el identificador de la sumilla para grabar en la tabla de asignatura.
165         int idSumilla = _context.Sumilla.Where(x => x.IdSilabo == idSil).Select(x => x.Id).FirstOrDefault();
166         var silabo = _context.Asignatura.Where(s => s.Id == idSil).FirstOrDefault();
167         silabo.IdSumilla = idSumilla;
168         silabo.UsuarioModificacion = "User";
169         silabo.FechaModificacion = DateTime.Now;
170         _context.SaveChanges();
171         Descripcion = "SUMILLA REGISTRADA";
172     }
173
174     //-- Grabar log sobre las modificaciones de la sumilla
175     GrabarAsignaturaLog(idSil, Concepto, Descripcion, "sumilla");
176
177 }

```

Fuente: Elaboración propia.

- Servicio “ListarCompetenciasPerfilEgreso”

Servicio que permite recuperar las competencias ingresadas por el centro del desarrollo académico para las escuelas profesionales.

El Servicio evalúa el código recibido para determinar si el tipo de dato es correspondiente al que se usa en los identificadores de las asignaturas (línea 177). Obtener el identificador de la escuela (línea 193) para poder extraer las competencias asociadas a la escuela (línea 194).

Con las competencias listadas de la escuela, se contrasta con las asociadas a la asignatura para determinadas cuales se encuentran marcadas o no (línea 207).

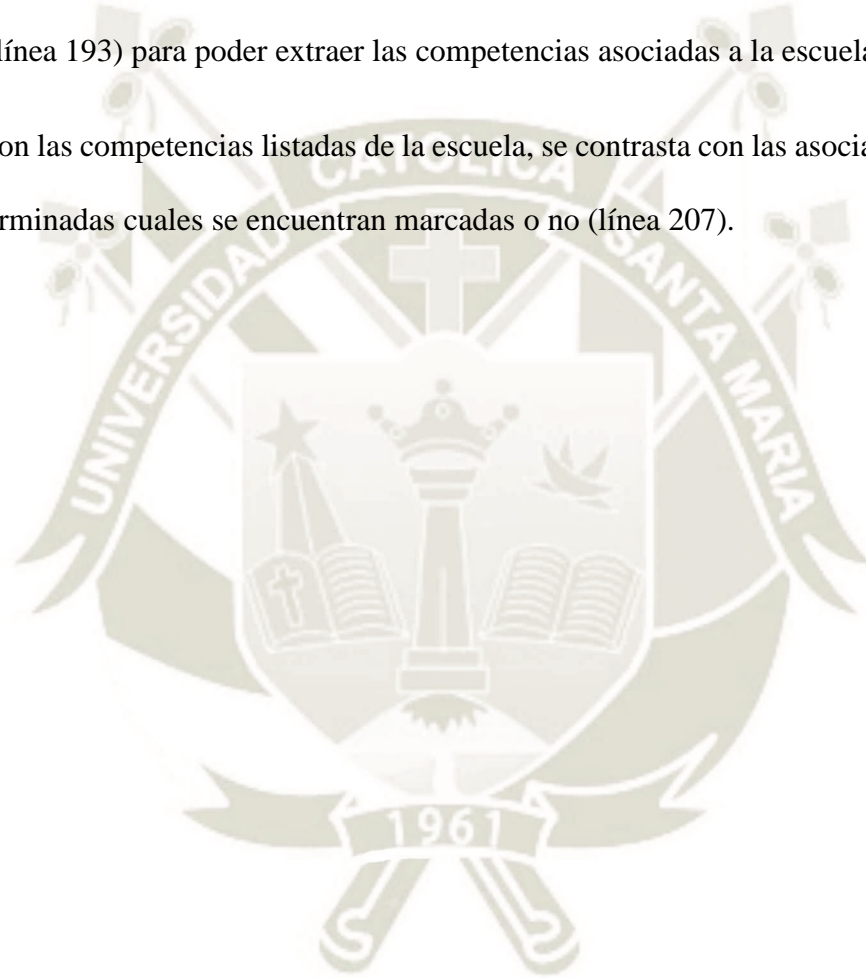


Figura 81 Método "ListarCompetenciasPerfilEgreso"

```

166 //-- Servicio que lista las competencias de perfil de egreso ingresadas para cada escuela profesional
167 1 referencia
168 public class ListarCompetenciasPerfilEgresoParametros
169 {
170     4 referencias
171     public string idSilabo { get; set; }
172 }
173
174 [HttpPost]
175 [Route("ListarCompetenciasPerfilEgreso")]
176 0 referencias
177 public ActionResult<IEnumerable<CompetenciaPerfilEgresoDTO>> ListarCompetenciasPerfilEgreso(ListarCompetenciasPerfilEgresoParametros parametros)
178 {
179     //-- Verificar valor del parámetro sílabo, en caso de que existir un valor inadecuado se retorna mensaje de error.
180     if (parametros.idSilabo == null)
181     {
182         return BadRequest();
183     }
184     if (parametros.idSilabo == "")
185     {
186         parametros.idSilabo = "0";
187     }
188     int idSil;
189     if (!Int32.TryParse(parametros.idSilabo, out idSil))
190     {
191         return BadRequest();
192     }
193     else
194     {
195         //-- Obtener lista de competencias de la escuela profesional que el centro de desarrollo académico estableció.
196         String IdEscuela = _context.Asignatura.Where(x => x.Id == idSil).Select(x => x.IdEscuela).FirstOrDefault();
197         List<CompetenciaPerfilEgresoDTO> competenciaPerfilEgresos = _context.CompetenciaPerfilEgreso.Where(x => x.Estado == 1 && x.IdEscuela == IdEscuela)
198             .Select(y => new CompetenciaPerfilEgresoDTO
199             {
200                 Codigo = y.Codigo,
201                 Descripcion = y.Descripcion,
202                 Nombre = y.Nombre,
203                 UsuarioModificacion = y.UsuarioModificacion,
204                 FechaModificacion = y.FechaModificacion
205             }).ToList();
206
207         //-- Se comprueba si la competencia fue asociada o seleccionada.
208         //-- En caso de ser seleccionada se marca el flag en 'S', en caso de que no, se marca en 'N'
209         var asignaComp = _context.AsignaturaCompetenciaPE.Where(x => x.IdAsignatura == idSil).ToList();
210         List<CompetenciaPerfilEgresoDTO> resultado = new List<CompetenciaPerfilEgresoDTO>();
211         foreach (var item in competenciaPerfilEgresos)
212         {
213             CompetenciaPerfilEgresoDTO Competencia = item;
214             Competencia.Grabo = 'N';
215             for (int i = 0; i < asignaComp.Count(); i++)
216             {
217                 var asco = asignaComp.Select(x => x.CodigoCompetenciaPE).ToArray()[i];
218                 if (String.Equals(asco, Competencia.Codigo))
219                 {
220                     Competencia.Grabo = 'S';
221                     break;
222                 }
223             }
224             resultado.Add(Competencia);
225         }
226         return resultado;
227     }
228 }

```

Fuente: Elaboración propia.

- **Servicio “GuardarAsignaturaCompetenciasPerfilEgreso”**

Servicio que permite guardar las competencias que se asociarán a la asignatura.

El Servicio evalúa el código recibido para determinar si el tipo de dato es correspondiente al que se usa en los identificadores de las asignaturas (línea 242).

Se obtiene el código de la escuela profesional de la asignatura (línea 260), para listar las competencias registradas (línea 262). Los códigos de las competencias registradas se contrastan con la lista de competencias que se intenta asociar a la asignatura (línea 270). Si no se encuentra alguno de los códigos se emite un mensaje de error (línea 289). En caso de que si coincidan todos los códigos, se elimina la lista anterior de códigos para ingresar la lista nueva que indicaría las competencias seleccionadas por el docente (línea 298). Finalmente se registra la acción de guardar las competencias en el log (línea 315).

Figura 82 Método "GuardarAsignaturaCompetenciasPerfilEgreso"

```
228      ///-- Servicio que Guarda las competencias de perfil de egreso seleccionadas para la asignatura a desarrollar.
229      1 referencia
230      public class GuardarAsignaturaCompetenciasPerfilEgresoParametros
231      {
232          5 referencias
233          public string idSilabo { get; set; }
234          2 referencias
235          public List<CompetenciaPerfilEgresoDTO> CompetenciaPE { get; set; }
236      }
237
238      [HttpPost]
239      [Route("GuardarAsignaturaCompetenciasPerfilEgreso")]
240      0 referencias
241      public JsonResult GuardarAsignaturaCompetenciasPerfilEgreso(GuardarAsignaturaCompetenciasPerfilEgresoParametros parametros)
242      {
243          ///-- Verificar valor del parámetro sílabo, en caso de que existir un valor inadecuado se retorna mensaje de error.
244          JsonResult json;
245          if (parametros.idSilabo == null) {...}
246          if (parametros.idSilabo == "") {...}
247          int idSil;
248          if (!Int32.TryParse(parametros.idsilabo, out idsil)) {...}
249          else
250          {
251              ///-- Se comprueba que los códigos de la lista seleccionada coincida con las competencias asignadas a la escuela.
252              String IdEscuela = _context.Asignatura.Where(x => x.Id == idsil).Select(x => x.IdEscuela).FirstOrDefault();
253              List<CompetenciaPerfilEgresoDTO> competenciaPerfilEgresos = new List<CompetenciaPerfilEgresoDTO>();
254              competenciaPerfilEgresos = _context.CompetenciaPerfilEgreso.Where(x => x.Estado == 1 && x.IdEscuela == IdEscuela).Select(y => new CompetenciaPerfilEgresoDTO
255              {
256                  Codigo = y.Codigo,
257                  Description = y.Description,
258                  Nombre = y.Nombre
259              }).ToList();
260
261              char existe = 'N';
262              foreach (var item in parametros.CompetenciaPE)
263              {
264                  for (int i = 0; i < competenciaPerfilEgresos.Count(); i++)
265                  {
266                      existe = 'N';
267                      var Competencia = competenciaPerfilEgresos.Select(x => x.Codigo).ToArray()[i];
268                      if (String.Equals(item.Codigo, Competencia))
269                      {
270                          existe = 'S';
271                          break;
272                      }
273                  }
274              }
275              if (existe == 'N')
276              {
277                  break;
278              }
279
280              ///-- En caso de querer registrar un código de competencia no válido se emite el mensaje de error.
281              if (existe == 'N')
282              {
283                  var resultado = new { estado = "fail", mensaje = "Código de competencia no válido." };
284                  return json = new JsonResult(resultado);
285              }
286          }
287
288          ///-- En caso de querer registrar un código de competencia no válido se emite el mensaje de error.
289          if (existe == 'N')
290          {
291              var resultado = new { estado = "fail", mensaje = "Código de competencia no válido." };
292              return json = new JsonResult(resultado);
293          }
294      }
295  }
```

Fuente: Elaboración propia.

- **Servicio “ListarCompetenciasPropiasAsignatura”**

Servicio que permite guardar las competencias propias ingresadas por el docente que se asociarán a la asignatura.

El Servicio evalúa el código recibido para determinar si el tipo de dato es correspondiente al que se usa en los identificadores de las asignaturas (línea 339). Continúa con la obtención de la configuración sobre si se encuentra actualmente el desarrollo de sílabos sobre un semestre par o impar (línea 357). Con el semestre se obtienen los identificadores de la fase para grabar los conceptos (línea 361). Luego se listan las competencias que se asociaron a la asignatura (línea 369), se determina si la lista contiene o no elementos para enviar un objeto vacío (línea 383) y se realiza el retorno de la información de fases y listas (línea 385).

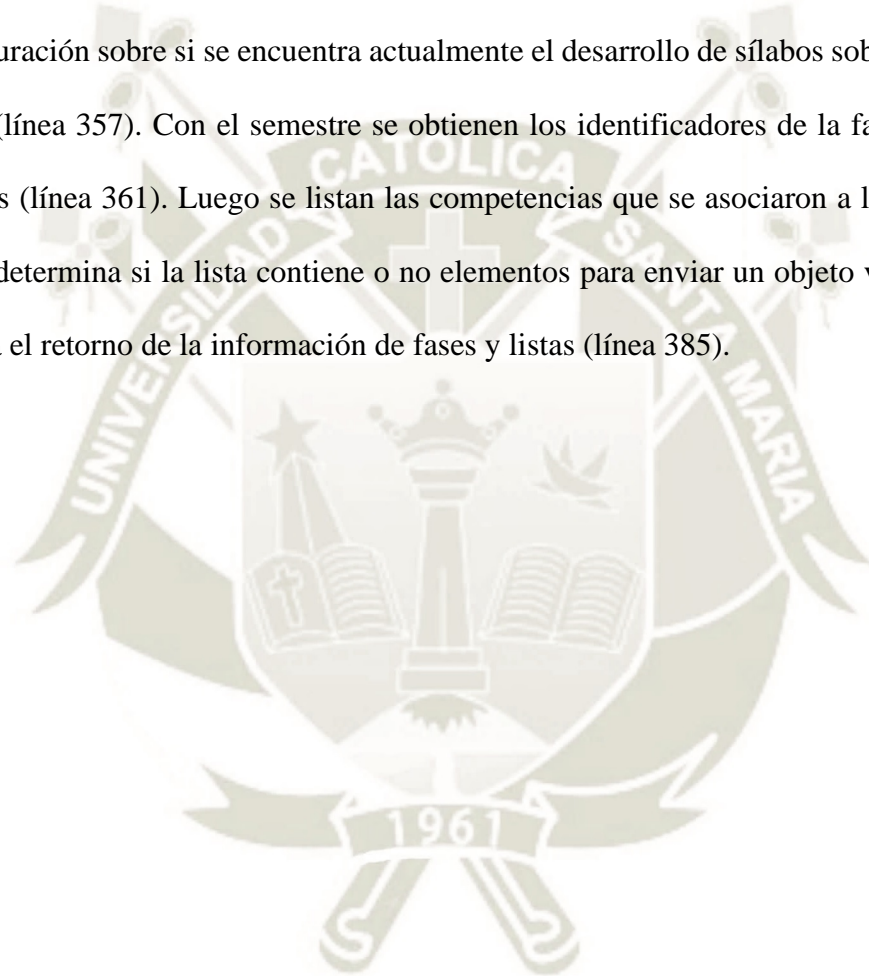


Figura 83 Método "*ListarCompetenciasPropiasAsignatura*"

```

327 //-- Servicio que permite listar las competencias propias de la asignatura y las fases con las que se encuentra trabajando el sistema.
328 1 referencia
329 public class ListarCompetenciasPropiasAsignaturaParametros
330 {
331     4 referencias
332     public string idSilabo { get; set; }
333 }
334 [HttpPost]
335 [Route("ListarCompetenciasPropiasAsignatura")]
336 0 referencias
337 public ActionResult<CompetenciasPAYFasesDTO> ListarCompetenciasPerfilEgreso(ListarCompetenciasPropiasAsignaturaParametros parametros)
338 {
339     //-- Verificar valor del parámetro sílabo, en caso de que existir un valor inadecuado se retorna mensaje de error.
340     if (parametros.idSilabo == null) {...}
341     if (parametros.idSilabo == "") {...}
342     int idSil;
343     if (!Int32.TryParse(parametros.idSilabo, out idSil) {...}
344     else
345     {
346
347         //-- Obtener configuración registrada del semestre par o impar activado
348         int Anio = DateTime.Now.Year;
349         int? Semestre = _context.Parametrica.Where(x => x.CodigoControl == 2 && x.Correlativo == 1)
350             .Select(x => x.ValorEntero)
351             .FirstOrDefault();
352
353         //-- Probablemente se necesiten los ids de las fases para poder grabar correctamente las asignaturas para cada semestre de cada año
354         List<FaseDTO> Fases = _context.Fase.Where(x => x.Semestre == Semestre && x.Anio == Anio)
355             .Select(x => new FaseDTO {
356                 IdFase = x.Id,
357                 NroFase = x.NroFase
358             }).ToList();
359
360         //-- Listar las competencias propias de la asignatura registradas para la asignatura o sílabo a desarrollar.
361         List<AsignaturaCompetenciaPADTO> asigCompPA = _context.AsignaturaCompetenciaPA.Join(_context.Fase, asig => asig.IdFase, fase => fase.Id, (asig, fase) => new { asig, fase })
362             .Where(x => x.asig.IdAsignatura == idSil)
363             .Select(x => new AsignaturaCompetenciaPADTO {
364                 Orden = x.asig.Orden,
365                 IdFase = x.fase.NroFase,
366                 Descripcion = x.asig.Descripcion,
367                 FechaModificacion = x.asig.FechaModificacion,
368                 UsuarioModificacion = x.asig.UsuarioModificacion
369             }).ToList();
370
371         //-- Retorno un objeto o elemento con valores 0 y "" para no retornar nulo.
372         if (asigCompPA.Count() == 0) {
373             asigCompPA.Add(new AsignaturaCompetenciaPADTO { Orden = 0, IdFase = 0, Descripcion = "" });
374         }
375         CompetenciasPAYFasesDTO resultado = new CompetenciasPAYFasesDTO();
376         resultado.AsignaturaCompetenciaPA = asigCompPA;
377         resultado.Fases = Fases;
378         return resultado;
379     }
380 }
381
382
383
384
385
386
387
388
389
390
391

```

Fuente: Elaboración propia.

- **Servicio “GuardarCompetenciasPropiasDeLaAsignatura”**

Servicio que permite almacenar la lista de competencias desarrolladas por el docente que construye el sílabo.

El Servicio evalúa el código recibido para determinar si el tipo de dato es correspondiente al que se usa en los identificadores de las asignaturas (línea 411). Superada la supervisión se procede a llamar al método “GrabarCompetenciasPAPorDTO” (línea 430), en caso de que la

inserción haya sido correcta, se emite mensaje de éxito, si no se devuelve un mensaje de error (línea 430)

Figura 84 Método "GuardarCompetenciasPropiasDeLaAsignatura"

```

395 |    ///- Servicio que permite guardar las competencias referentes o propias de la asignatura
396 |    ///- Son competencias ingresadas manualmente y asociadas a una fase en específico.
397 |    1 referencia
398 |    public class GuardarCompetenciasPropiasDeLaAsignaturaParametros
399 |    {
400 |        4 referencias
401 |        public string idSilabo { get; set; }
402 |
403 |        1 referencia
404 |        public List<CompetenciaPropiaAsignaturaDTO> CompetenciaPA { get; set; }
405 |    }
406 |
407 |    [HttpPost]
408 |    [Route("GuardarCompetenciasPropiasDeLaAsignatura")]
409 |    0 referencias
410 |    public JsonResult GuardarCompetenciasPropiasDeLaAsignatura(GuardarCompetenciasPropiasDeLaAsignaturaParametros parametros)
411 |    {
412 |        ///- Verificar valor del parámetro sílabo, en caso de que existir un valor inadecuado se retorna mensaje de error.
413 |        JsonResult json;
414 |        if (parametros.idSilabo == null)
415 |        {
416 |            var resultado = new { estado = "fail", mensaje = "Información errónea." };
417 |            return json = new JsonResult(resultado);
418 |        }
419 |        if (parametros.idSilabo == "")
420 |        {
421 |            parametros.idSilabo = "0";
422 |        }
423 |        int idSil;
424 |        if (!Int32.TryParse(parametros.idSilabo, out idSil))
425 |        {
426 |            var resultado = new { estado = "fail", mensaje = "Información errónea." };
427 |            return json = new JsonResult(resultado);
428 |        }
429 |        else
430 |        {
431 |            ///- Grabar las competencias enviadas por DTO, se devuelve mensaje de error, en caso de que exista incongruencia o si se registró correctamente.
432 |            _handler = new GestionSilabos.Handler.Handler(_context);
433 |            char grabo = _handler.GrabarCompetenciasPAPorDTO(idSil, parametros.CompetenciaPA);
434 |
435 |            if (grabo == 'S')
436 |            {
437 |                var resultado = new { estado = "success", mensaje = "Se registro correctamente las competencias." };
438 |                return json = new JsonResult(resultado);
439 |            }
440 |            else
441 |            {
442 |                var resultado = new { estado = "fail", mensaje = "La información no es correcta." };
443 |                return json = new JsonResult(resultado);
444 |            }
445 |        }
446 |    }

```

Fuente: Elaboración propia.

El método, obtiene la lista de fases del semestre en cuestión (línea 368). Se eliminan las competencias hasta entonces registradas (línea 370). Posteriormente se registra la lista de competencias asociándolas a la asignatura (línea 375) y en caso de que sea exitosa la inserción se registra la acción en el log (línea 402).

Figura 85 Método "GrabarCompetenciasPAPorDTO"

```

364 public char GrabarCompetenciasPAPorDTO(int idSil, List<CompetenciaPropiaAsignaturaDTO> CompetenciaPA)
365 {
366
367     //-- Obtener lista de fases para distribuir
368     var fases = ObtenerFasesSemestre(idSil);
369     //-- Eliminar competencias asociadas a la asignatura
370     _context.Database.ExecuteSqlRaw("DELETE FROM AsignaturaCompetenciaPA WHERE IdAsignatura = {0}", idSil.ToString());
371     int numfase;
372     int fase;
373     char grabo = 'S';
374     //-- Grabar lista de competencias
375     foreach (var item in CompetenciaPA)
376     {
377         numfase = item.IdFase;
378         fase = IdFasePorNumeroFase(numfase, fases);
379         if (fase != 0)
380         {
381             AsignaturaCompetenciaPAModel AsigCompPA = new AsignaturaCompetenciaPAModel
382             {
383                 UsuarioCreacion = "User",
384                 UsuarioModificacion = "User",
385                 FechaCreacion = DateTime.Now,
386                 FechaModificacion = DateTime.Now,
387                 Estado = 1,
388                 IdAsignatura = idSil,
389                 Orden = item.Orden,
390                 IdFase = fase,
391                 Descripcion = item.Descripcion
392             };
393             _context.AsignaturaCompetenciaPA.Add(AsigCompPA);
394         }
395         else
396         {
397             grabo = 'N';
398         }
399     }
400
401     //-- Si fue exitosa la inserción, se registra en el log la acción.
402     if (grabo == 'S')
403     {
404         _context.SaveChanges();
405         String Concepto = "COMPETENCIASPROPIASDEASIGNATURA";
406         String Descripcion = "COMPETENCIAS PROPIAS DE LA ASIGNATURA REGISTRADAS";
407         GrabarAsignaturaLog(idSil, Concepto, Descripcion, "competenciaspropiasasignatura");
408     }
409     return grabo;
410 }
411

```

Fuente: Elaboración propia.

- **Servicio “ListarAsignaturaFase”**

Servicio que permite listar el nombre de las fases perteneciente a la asignatura desarrollada por el docente.

El Servicio evalúa el código recibido para determinar si el tipo de dato es correspondiente al que se usa en los identificadores de las asignaturas (línea 457). Se obtiene el nombre que debió

ser ingresado por el docente (línea 471), pero en caso de que no exista un resultado, se enviará los objetos vacíos (línea 483).

Figura 86 Método "ListarAsignaturaFase"

```

445  //-- Servicio que permite listar las fases articulares de cada asignatura, específicamente el nombre que se designa a cada fase.
446  1 referencia
447  public class ListarAsignaturaFaseParametros
448  {
449      4 referencias
450      public string idSilabo { get; set; }
451  }
452
453  [HttpPost]
454  [Route("ListarAsignaturaFase")]
455  0 referencias
456  public ActionResult<IEnumerable<AsignaturaFaseDTO>> ListarAsignaturaFase(ListarAsignaturaFaseParametros parametros)
457  {
458      //-- Verificar valor del parámetro sílabo, en caso de que existir un valor inadecuado se retorna mensaje de error.
459      if (parametros.idSilabo == null)
460      {
461          return BadRequest();
462      }
463      if (parametros.idSilabo == "")
464      {
465          parametros.idSilabo = "0";
466      }
467      int idSil;
468      if (!Int32.TryParse(parametros.idSilabo, out idSil))
469      {
470          return BadRequest();
471      }
472      else{
473          //-- Se busca el nombre de la fase. En caso de no haberse registrado aún, los nombres de la fase retornan vacíos.
474          List<AsignaturaFaseDTO> resultado = new List<AsignaturaFaseDTO>();
475          resultado = _context.AsignaturaFase.Join(_context.Fase, asig => asig.IdFase, fase => fase.Id, (asig, fase) => new { asig, fase })
476              .Where(x => x.asig.IdAsignatura == idSil)
477              .Select( x => new AsignaturaFaseDTO {
478                  IdFase = x.fase.NroFase,
479                  NombreFase = x.asig.NombreFase,
480                  UsuarioModificacion = x.asig.UsuarioModificacion,
481                  FechaModificacion = x.asig.FechaModificacion
482              }).ToList();
483
484          if (resultado.Count() == 0){
485              resultado.Add(new AsignaturaFaseDTO { IdFase = 1 , NombreFase = "" });
486              resultado.Add(new AsignaturaFaseDTO { IdFase = 2, NombreFase = "" });
487              resultado.Add(new AsignaturaFaseDTO { IdFase = 3, NombreFase = "" });
488          }
489
490          return resultado;
491      }

```

Fuente: Elaboración propia.

- **Servicio “GuardarAsignaturaFase”**

Servicio que permite guardar el nombre de las fases perteneciente a la asignatura desarrollada por el docente.

El Servicio evalúa el código recibido para determinar si el tipo de dato es correspondiente al que se usa en los identificadores de las asignaturas (línea 506). Superada la evaluación, se

llama el método “GrabarAsignaturaFasePorDTO”, que realiza la inserción de la información (línea 525), si la respuesta es positiva se emite mensaje de éxito; si no, se emite un mensaje de error (línea 527).

Figura 87 Método "GuardarAsignaturaFase"

```

493  //-- Servicio que permite guardar la fase por asignatura, junto con sus nombres de sus fases.
494  1 referencia
495  public class GuardarAsignaturaFaseParametros
496  {
497      4 referencias
498      public string idSilabo { get; set; }
499      1 referencia
500      public List<AsignaturaFaseDTO> AsignaturaFases { get; set; }
501  }
502
503  [HttpPost]
504  [Route("GuardarAsignaturaFase")]
505  0 referencias
506  public JsonResult GuardarAsignaturaFase(GuardarAsignaturaFaseParametros parametros)
507  {
508      //-- Verificar valor del parámetro sílabo, en caso de que existir un valor inadecuado se retorna mensaje de error.
509      JsonResult json;
510      if (parametros.idSilabo == null)
511      {
512          var resultado = new { estado = "fail", mensaje = "Información errónea." };
513          return json = new JsonResult(resultado);
514      }
515      if (parametros.idSilabo == "")
516      {
517          parametros.idSilabo = "0";
518      }
519      int idSil;
520      if (!Int32.TryParse(parametros.idSilabo, out idSil))
521      {
522          var resultado = new { estado = "fail", mensaje = "Información errónea." };
523          return json = new JsonResult(resultado);
524      }
525      else
526      {
527          //-- Se guarda la información, en caso de existir incongruencia se emite mensaje de error, de lo contrario emite mensaje de inserción correcta.
528          _handler = new GestionSilabos.Handler.Handler(_context);
529          char grabo = _handler.GrabarAsignaturaFasePorDTO(idSil, parametros.AsignaturaFases);
530
531          if(grabo == 'S')
532          {
533              var resultado = new { estado = "success", mensaje = "Se actualizó correctamente los títulos de las fases." };
534              return json = new JsonResult(resultado);
535          }
536          else
537          {
538              var resultado = new { estado = "fail", mensaje = "La información no es correcta." };
539              return json = new JsonResult(resultado);
540          }
541      }
542  }
543  }
544  }
545  }
546  }
547  }
548  }
549  }
550  }
551  }
552  }
553  }
554  }
555  }
556  }
557  }
558  }
559  }
560  }
561  }
562  }
563  }
564  }
565  }
566  }
567  }
568  }
569  }
570  }
571  }
572  }
573  }
574  }
575  }
576  }
577  }
578  }
579  }
580  }
581  }
582  }
583  }
584  }
585  }
586  }
587  }
588  }
589  }
590  }
591  }
592  }
593  }
594  }
595  }
596  }
597  }
598  }
599  }
600  }
601  }
602  }
603  }
604  }
605  }
606  }
607  }
608  }
609  }
610  }
611  }
612  }
613  }
614  }
615  }
616  }
617  }
618  }
619  }
620  }
621  }
622  }
623  }
624  }
625  }
626  }
627  }
628  }
629  }
630  }
631  }
632  }
633  }
634  }
635  }
636  }
637  }
638  }
639  }
640  }
641  }
642  }
643  }
644  }
645  }
646  }
647  }
648  }
649  }
650  }
651  }
652  }
653  }
654  }
655  }
656  }
657  }
658  }
659  }
660  }
661  }
662  }
663  }
664  }
665  }
666  }
667  }
668  }
669  }
670  }
671  }
672  }
673  }
674  }
675  }
676  }
677  }
678  }
679  }
680  }
681  }
682  }
683  }
684  }
685  }
686  }
687  }
688  }
689  }
690  }
691  }
692  }
693  }
694  }
695  }
696  }
697  }
698  }
699  }
700  }
701  }
702  }
703  }
704  }
705  }
706  }
707  }
708  }
709  }
710  }
711  }
712  }
713  }
714  }
715  }
716  }
717  }
718  }
719  }
720  }
721  }
722  }
723  }
724  }
725  }
726  }
727  }
728  }
729  }
730  }
731  }
732  }
733  }
734  }
735  }
736  }
737  }
738  }
739  }
740  }
741  }
742  }
743  }
744  }
745  }
746  }
747  }
748  }
749  }
750  }
751  }
752  }
753  }
754  }
755  }
756  }
757  }
758  }
759  }
760  }
761  }
762  }
763  }
764  }
765  }
766  }
767  }
768  }
769  }
770  }
771  }
772  }
773  }
774  }
775  }
776  }
777  }
778  }
779  }
780  }
781  }
782  }
783  }
784  }
785  }
786  }
787  }
788  }
789  }
790  }
791  }
792  }
793  }
794  }
795  }
796  }
797  }
798  }
799  }
800  }
801  }
802  }
803  }
804  }
805  }
806  }
807  }
808  }
809  }
810  }
811  }
812  }
813  }
814  }
815  }
816  }
817  }
818  }
819  }
820  }
821  }
822  }
823  }
824  }
825  }
826  }
827  }
828  }
829  }
830  }
831  }
832  }
833  }
834  }
835  }
836  }
837  }
838  }
839  }
840  }
841  }
842  }
843  }
844  }
845  }
846  }
847  }
848  }
849  }
850  }
851  }
852  }
853  }
854  }
855  }
856  }
857  }
858  }
859  }
860  }
861  }
862  }
863  }
864  }
865  }
866  }
867  }
868  }
869  }
870  }
871  }
872  }
873  }
874  }
875  }
876  }
877  }
878  }
879  }
880  }
881  }
882  }
883  }
884  }
885  }
886  }
887  }
888  }
889  }
890  }
891  }
892  }
893  }
894  }
895  }
896  }
897  }
898  }
899  }
900  }
901  }
902  }
903  }
904  }
905  }
906  }
907  }
908  }
909  }
910  }
911  }
912  }
913  }
914  }
915  }
916  }
917  }
918  }
919  }
920  }
921  }
922  }
923  }
924  }
925  }
926  }
927  }
928  }
929  }
930  }
931  }
932  }
933  }
934  }
935  }
936  }
937  }
938  }
939  }
940  }
941  }
942  }
943  }
944  }
945  }
946  }
947  }
948  }
949  }
950  }
951  }
952  }
953  }
954  }
955  }
956  }
957  }
958  }
959  }
960  }
961  }
962  }
963  }
964  }
965  }
966  }
967  }
968  }
969  }
970  }
971  }
972  }
973  }
974  }
975  }
976  }
977  }
978  }
979  }
980  }
981  }
982  }
983  }
984  }
985  }
986  }
987  }
988  }
989  }
990  }
991  }
992  }
993  }
994  }
995  }
996  }
997  }
998  }
999  }
1000 }
```

Fuente: Elaboración propia.

El método obtiene los códigos de la fase para distribuir la información recibida (línea 183). Se elimina la información registrada hasta este momento (línea 185), se registran los nuevos nombres de las fases como nuevos objetos (línea 190). Acabada la acción si todo fue exitoso, se graba el log con la acción de inserción de títulos de fase (línea 215).

Figura 88 Método "GrabarAsignaturaFasePorDTO"

```

179 public char GrabarAsignaturaFasePorDTO(int idSil, List<AsignaturaFaseDTO> AsignaturaFases)
180 {
181
182     //-- Obtener lista de fases para distribuir
183     var fases = ObtenerFasesSemestre(idSil);
184     //-- Eliminar registros anteriores
185     _context.Database.ExecuteSqlRaw("DELETE FROM AsignaturaFase WHERE IdAsignatura = {0}", idSil.ToString());
186     int numfase;
187     int fase;
188     char grabo = 'S';
189     //-- Grabar nuevos registros con los nuevos nombres de las fases
190     foreach (var item in AsignaturaFases)
191     {
192         numfase = item.IdFase;
193         fase = IdFasePorNumeroFase(numfase, fases);
194         if (fase != 0)
195         {
196             AsignaturaFaseModel asignatura = new AsignaturaFaseModel
197             {
198                 IdAsignatura = idSil,
199                 IdFase = fase,
200                 NombreFase = item.NombreFase,
201                 UsuarioCreacion = "ADMINISTRADOR",
202                 UsuarioModificacion = "ADMINISTRADOR",
203                 FechaCreacion = DateTime.Now,
204                 FechaModificacion = DateTime.Now,
205                 Estado = 1
206             };
207             _context.AsignaturaFase.Add(asignatura);
208         }
209         else
210         {
211             grabo = 'N';
212         }
213     }
214     //-- Grabar en el log la acción de grabar el títulos de fase
215     if (grabo == 'S') {
216         _context.SaveChanges();
217         String Concepto = "TITULOSFASE";
218         String Descripción = "TITULOS DE FASE AGREGADOS";
219         GrabarAsignaturaLog(idSil, Concepto, Descripción, "titulosfase");
220     }
221     return grabo;
222 }

```

Fuente: Elaboración propia.

- **Servicio “ListarUnidades”**

Servicio que permite listar las unidades teóricas perteneciente a la asignatura desarrollada por el docente.

El Servicio evalúa el código recibido para determinar si el tipo de dato es correspondiente al que se usa en los identificadores de las asignaturas (línea 554). Se obtiene las unidades teóricas

ingresadas hasta el momento (línea 570), pero en caso de que no exista un resultado, se enviará los objetos vacíos (línea 582).

Figura 89 Método "ListarUnidades"

```

540  //-- Listar unidades de aprendizaje teóricas registradas para la asignatura.
541  //-- Las unidades a su vez contienen sub-unidades que permiten describir el contenido del curso por temas.
542  //-- Unidades y sub-unidades se asocian a una fase determinada.
543  1 referencia
544  public class ListarUnidadesParametros
545  {
546      4 referencias
547      public string idSilabo { get; set; }
548  }
549
550  [HttpPost]
551  [Route("ListarUnidades")]
552  0 referencias
553  public ActionResult<IEnumerable<UnidadDTO>> ListarUnidadesFase(ListarUnidadesParametros parametros)
554  {
555      //-- Verificar valor del parámetro sílabo, en caso de que existir un valor inadecuado se retorna mensaje de error.
556      if (parametros.idSilabo == null)
557      {
558          return BadRequest();
559      }
560      if (parametros.idSilabo == "")
561      {
562          parametros.idSilabo = "0";
563      }
564      int idSil;
565      if (!Int32.TryParse(parametros.idSilabo, out idSil))
566      {
567          return BadRequest();
568      }
569      else {
570          //-- Listar Unidades teóricas de la asignatura y a la fase a la que pertenecen.
571          List<UnidadDTO> resultado = new List<UnidadDTO>();
572          resultado = _context.Unidad.Join(_context.Fase, uni => uni.IdFase, fase => fase.Id, (uni, fase) => new { uni, fase })
573              .Where(x => x.uni.IdAsignatura == idSil)
574              .Select(x => new UnidadDTO {
575                  NroUnidad = x.uni.NroUnidad,
576                  NroSubUnidad = x.uni.NroSubUnidad,
577                  IdFase = x.fase.NroFase,
578                  Descripcion = x.uni.Descripcion,
579                  UsuarioModificacion = x.uni.UsuarioModificacion,
580                  FechaModificacion = x.uni.FechaModificacion
581              }).ToList();
582
583          //-- En caso de no encontrar unidades se devuelve un objeto vacío.
584          if (resultado.Count() == 0)
585          {
586              resultado.Add(new UnidadDTO { NroUnidad = 0 , NroSubUnidad = 0 , IdFase = 0, Descripcion = "" });
587          }
588          return resultado;
589      }
590  }

```

Fuente: Elaboración propia.

- **Servicio “GuardarUnidades”**

Servicio que permite guardar las unidades teóricas perteneciente a la asignatura desarrollada por el docente.

El Servicio evalúa el código recibido para determinar si el tipo de dato es correspondiente al que se usa en los identificadores de las asignaturas (línea 603). Superada la evaluación, se

llama el método “GrabarUnidadesPorDTO”, que realiza la inserción de la información (línea 623), si la respuesta es positiva se emite mensaje de éxito; si no, se emite un mensaje de error (línea 625).

Figura 90 Método "GuardarUnidades"

```

590 //-- Servicio que permite almacenar las unidades teóricas desarrolladas para la asignatura.
591 1 referencia
592 public class GuardarUnidadesParametros
593 {
594     4 referencias
595     public string idSilabo { get; set; }
596     1 referencia
597     public List<UnidadDTO> Unidades { get; set; }
598 }
599
600 [HttpPost]
601 [Route("GuardarUnidades")]
602 0 referencias
603 public JsonResult GuardarUnidades(GuardarUnidadesParametros parametros)
604 {
605     //-- Verificar valor del parámetro sílabo, en caso de que existir un valor inadecuado se retorna mensaje de error.
606     JsonResult json;
607     if (parametros.idSilabo == null)
608     {
609         var resultado = new { estado = "fail", mensaje = "Información errónea." };
610         return json = new JsonResult(resultado);
611     }
612     if (parametros.idSilabo == "")
613     {
614         parametros.idSilabo = "0";
615     }
616     int idSil;
617     if (!Int32.TryParse(parametros.idSilabo, out idSil))
618     {
619         var resultado = new { estado = "fail", mensaje = "Información errónea." };
620         return json = new JsonResult(resultado);
621     }
622     else
623     {
624         //-- Se guarda las unidades recibidas en el objeto DTO
625         //-- En caso de tener incongruencia retornar mensaje de error, caso contrario retornar mensaje de confirmación de inserción
626         _handler = new GestionSilabos.Handler.Handler(_context);
627         char grabo = _handler.GrabarUnidadesPorDTO(idSil, parametros.Unidades);
628
629         if (grabo == 'S')
630         {
631             var resultado = new { estado = "success", mensaje = "Se actualizó correctamente las unidades teóricas." };
632             return json = new JsonResult(resultado);
633         }
634         else
635         {
636             var resultado = new { estado = "fail", mensaje = "La información no es correcta." };
637             return json = new JsonResult(resultado);
638         }
639     }
640 }

```

Fuente: Elaboración propia.

Se obtiene los códigos o identificadores de las fases para distribuir correctamente la información (línea 272), posteriormente se elimina la información ingresada hasta el momento (línea 274). Eliminada la lista anterior, se inserta la nueva lista de unidades teóricas creando un nuevo objeto por unidad (línea 279). Acabada la inserción, si no existió complicaciones se graba el log con el registro de inserción exitosa (línea 305)

Figura 91 Método "GrabarUnidadesPorDTO"

```

268 public char GrabarUnidadesPorDTO(int idSil, List<UnidadDTO> Unidades)
269 {
270
271     //-- Obtener lista de fases para distribuir
272     var fases = ObtenerFasesSemestre(idSil);
273     //-- Eliminar registros anteriores
274     _context.Database.ExecuteSqlRaw("DELETE FROM Unidad WHERE IdAsignatura = {0}", idSil.ToString());
275     int numfase;
276     int fase;
277     char grabo = 'S';
278     //-- Realizar inserción de las unidades recibidas.
279     foreach (var item in Unidades)
280     {
281         numfase = item.IdFase;
282         fase = IdFasePorNumeroFase(numfase, fases);
283         if (fase != 0)
284         {
285             UnidadModel Unidad = new UnidadModel
286             {
287                 UsuarioCreacion = "User",
288                 UsuarioModificacion = "User",
289                 FechaCreacion = DateTime.Now,
290                 FechaModificacion = DateTime.Now,
291                 Estado = 1,
292                 IdAsignatura = idSil,
293                 NroUnidad = item.NroUnidad,
294                 NroSubUnidad = item.NroSubUnidad,
295                 IdFase = fase,
296                 Descripcion = item.Descripcion
297             };
298             _context.Unidad.Add(Unidad);
299         }
300         else
301         {
302             grabo = 'N';
303         }
304     }
305     //-- grabar en el log la acción de insertar unidades teóricas.
306     if (grabo == 'S')
307     {
308         _context.SaveChanges();
309         String Concepto = "UNIDADESTEORICAS";
310         String Descripcion = "UNIDADES TEÓRICAS REGISTRADAS";
311         GrabarAsignaturaLog(idSil, Concepto, Descripcion, "titulosfase");
312     }
313     return grabo;
314 }

```

Fuente: Elaboración propia.

- **Servicio “ListarUnidadesLab”**

Servicio que permite listar las unidades de laboratorio o de prácticas perteneciente a la asignatura desarrollada por el docente.

El Servicio evalúa el código recibido para determinar si el tipo de dato es correspondiente al que se usa en los identificadores de las asignaturas (línea 554). Se obtiene las unidades teóricas ingresadas hasta el momento (línea 570), pero en caso de que no exista un resultado, se enviará los objetos vacíos (línea 582).

Figura 92 Método "ListarUnidadesLab"

```

638 //-- Servicio que permite listar unidades de laboratorio para a asignatura a desarrollar.
639 1referencia
640 public class ListarUnidadesLabParametros
641 {
642     4referencias
643     public string idSilabo { get; set; }
644 }
645 [HttpPost]
646 [Route("ListarUnidadesLab")]
647 0referencias
648 public ActionResult<IEnumerable<UnidadLabDTO>> ListarUnidadesLabFase(ListarUnidadesLabParametros parametros)
649 {
650     //-- Verificar valor del parámetro sílabo, en caso de que existir un valor inadecuado se retorna mensaje de error.
651     if (parametros.idSilabo == null)
652     {
653         return BadRequest();
654     }
655     if (parametros.idSilabo == "")
656     {
657         parametros.idSilabo = "0";
658     }
659     int idSil;
660     if (!Int32.TryParse(parametros.idSilabo, out idSil))
661     {
662         return BadRequest();
663     }
664     else {
665         //-- Se obtienen los datos de las unidades de laboratorio ingresadas para la asignatura.
666         List<UnidadLabDTO> resultado = new List<UnidadLabDTO>();
667         resultado = _context.UnidadLab.Join(_context.Fase, unilab => unilab.IdFase, fase => fase.Id, (unilab, fase) => new { unilab, fase })
668             .Where(x => x.unilab.IdAsignatura == idSil)
669             .Select(x => new UnidadLabDTO
670                 {
671                     NroUnidadLab = x.unilab.NroUnidadLab,
672                     IdFase = x.fase.NroFase,
673                     Descripcion = x.unilab.Descripcion,
674                     UsuarioModificacion = x.unilab.UsuarioModificacion,
675                     FechaModificacion = x.unilab.FechaModificacion
676                 })
677             .ToList();
678     }
679     //-- En caso de no existir unidades de laboratorio se devuelve una objeto vacío.
680     if (resultado.Count() == 0)
681     {
682         resultado.Add(new UnidadLabDTO { NroUnidadLab = 0, IdFase = 0, Descripcion = "" });
683     }
684     return resultado;
685 }

```

Fuente: Elaboración propia.

- **Servicio “GuardarUnidadesLab”**

Servicio que permite guardar las unidades de laboratorio o prácticas perteneciente a la asignatura desarrollada por el docente.

El Servicio evalúa el código recibido para determinar si el tipo de dato es correspondiente al que se usa en los identificadores de las asignaturas (línea 701). Superada la evaluación, se llama el método “GrabarUnidadesLabPorDTO”, que realiza la inserción de la información (línea 722), si la respuesta es positiva se emite mensaje de éxito; si no, se emite un mensaje de error (línea 724).

Figura 93 "Método "GuardarUnidadesLab"

```

687 //-- Servicio que permite guardar unidades de laboratorio para la asignatura.
688 1 referencia
689 public class GuardarUnidadesLabParametros
690 {
691     4 referencias
692     public string idSilabo { get; set; }
693     1 referencia
694     public List<UnidadLabDTO> UnidadesLab { get; set; }
695 }
696
697 [HttpPost]
698 [Route("GuardarUnidadesLab")]
699 0 referencias
700 public JsonResult GuardarUnidadesLab(GuardarUnidadesLabParametros parametros)
701 {
702     //-- Verificar valor del parámetro sílabo, en caso de que existir un valor inadecuado se retorna mensaje de error.
703     JsonResult json;
704     if (parametros.idSilabo == null)
705     {
706         var resultado = new { estado = "fail", mensaje = "Información errónea." };
707         return json = new JsonResult(resultado);
708     }
709     if (parametros.idSilabo == "")
710     {
711         parametros.idSilabo = "0";
712     }
713     int idSil;
714     if (!Int32.TryParse(parametros.idSilabo, out idSil))
715     {
716         var resultado = new { estado = "fail", mensaje = "Información errónea." };
717         return json = new JsonResult(resultado);
718     }
719     else
720     {
721         //-- Se guardan las unidades listadas en el objeto DTO. En caso de existir incongruencia se emite el mensaje de error.
722         //-- Caso contrario se retorna el mensaje de inserción correcta.
723         _handler = new GestionSilabos.Handler.Handler(_context);
724         char grabo = _handler.GrabarUnidadesLabPorDTO(idSil, parametros.UnidadesLab);
725         if (grabo == 'S')
726         {
727             var resultado = new { estado = "success", mensaje = "Se actualizó correctamente las unidades de laboratorio." };
728             return json = new JsonResult(resultado);
729         }
730         else
731         {
732             var resultado = new { estado = "fail", mensaje = "La información no es correcta." };
733             return json = new JsonResult(resultado);
734         }
735     }
736 }

```

Fuente: Elaboración propia.

Se obtiene los códigos o identificadores de las fases para distribuir correctamente la información (línea 319), posteriormente se elimina la información ingresada hasta el momento

(línea 321). Eliminada la lista anterior, se inserta la nueva lista de unidades de laboratorio creando un nuevo objeto por unidad (línea 326). Acabada la inserción, si no existió complicaciones se graba el log con el registro de inserción exitosa (línea 352).

Figura 94 Método "GrabarUnidadesLabPorDTO"

```

315 public char GrabarUnidadesLabPorDTO(int idSil, List<UnidadLabDTO> UnidadesLab)
316 {
317
318     //-- Obtener lista de fases para distribuir
319     var fases = ObtenerFasesSemestre(idSil);
320     //-- Eliminar registros anteriores
321     _context.Database.ExecuteSqlRaw("DELETE FROM UnidadLab WHERE IdAsignatura = {0}", idSil.ToString());
322     int numfase;
323     int fase;
324     char grabo = 'S';
325     //-- Realizar inserción de las unidades recibidas.
326     foreach (var item in UnidadesLab)
327     {
328         numfase = item.IdFase;
329         fase = IdFasePorNumeroFase(numfase, fases);
330         if (fase != 0)
331         {
332             UnidadLabModel UnidadLab = new UnidadLabModel
333             {
334                 UsuarioCreacion = "User",
335                 UsuarioModificacion = "User",
336                 FechaCreacion = DateTime.Now,
337                 FechaModificacion = DateTime.Now,
338                 Estado = 1,
339                 IdAsignatura = idSil,
340                 NroUnidadLab = item.NroUnidadLab,
341                 IdFase = fase,
342                 Descripcion = item.Descripcion
343             };
344             _context.UnidadLab.Add(UnidadLab);
345         }
346         else
347         {
348             grabo = 'N';
349         }
350     }
351     //-- grabar en el log la acción de insertar unidades de laboratorio.
352     if (grabo == 'S')
353     {
354         _context.SaveChanges();
355         String Concepto = "UNIDADES LAB";
356         String Descripcion = "UNIDADES DE LABORATORIO REGISTRADAS";
357         GrabarAsignaturaLog(idSil, Concepto, Descripcion, "unidadeslab");
358     }
359     return grabo;
360 }
361

```

- Servicio “ListarResultados”

Servicio que permite listar resultados de aprendizaje perteneciente a la asignatura desarrollada por el docente.

El Servicio evalúa el código recibido para determinar si el tipo de dato es correspondiente al que se usa en los identificadores de las asignaturas (línea 749). Se obtiene los resultados de aprendizaje ingresados hasta el momento (línea 765), pero en caso de que no exista un resultado, se enviará los objetos vacíos (línea 783).

Figura 95 Método "ListarResultados"

```

737 //-- Servicio que permite listar los resultados guardados por asignatura.
738 //-- En caso de no tener resultados guardados por primera vez, se devuelve un objeto vacío de la clase ResultadosDTO
1 referencia
739 public class ListarResultadosParametros
740 {
4 referencias
741     public string idSilabo { get; set; }
742 }
743
[HttpPost]
744 [Route("ListarResultados")]
0 referencias
745 public ActionResult<IEnumerable<ResultadosDTO>> ListarResultados(ListarResultadosParametros parametros)
746 {
747     //-- Verificar valor del parámetro sílabo, en caso de que existir un valor inadecuado se retorna mensaje de error.
748     if (parametros.idSilabo == null) {...}
749     if (parametros.idSilabo == "") {...}
750     int idSil;
751     if (!Int32.TryParse(parametros.idSilabo, out idSil)) {...}
752     else
753     {
754         //-- Se listan los resultados y sus datos de cada fase de la asignatura.
755         List<ResultadosDTO> resultado = new List<ResultadosDTO>();
756         resultado = _context.Resultado.Join(_context.Fase, res => res.IdFase, fase => fase.Id, (res, fase) => new { res, fase })
757             .Where(x => x.res.IdAsignatura == idSil)
758             .Select(x => new ResultadosDTO
759             {
760                 IdResultado = x.res.IdResultado,
761                 Orden = x.res.Orden,
762                 IdFase = x.fase.NroFase,
763                 Descripcion = x.res.Descripcion,
764                 Sintesis = x.res.Sintesis,
765                 PesoTeorico = x.res.PesoTeorico,
766                 PesoPractico = x.res.PesoPractico,
767                 UsuarioModificacion = x.res.UsuarioModificacion,
768                 FechaModificacion = x.res.FechaModificacion
769             });
770         resultado.ToList();
771
772         //-- En caso de no contar con resultados todavía, se retorna un objeto vacío.
773         if (resultado.Count() == 0)
774         {
775             resultado.Add(new ResultadosDTO { IdResultado = 0, Orden = 0, IdFase = 0, Descripcion = "", Sintesis = "", PesoTeorico = 0, PesoPractico = 0 });
776         }
777         return resultado;
778     }
779 }

```

Fuente: Elaboración propia.

- **Servicio “GuardarResultados”**

Servicio que permite guardar los resultados de aprendizaje pertenecientes a la asignatura desarrollada por el docente.

El Servicio evalúa el código recibido para determinar si el tipo de dato es correspondiente al que se usa en los identificadores de las asignaturas (línea 810). Superada la evaluación, se llama el método “GrabarResultadosPorDTO”, que realiza la inserción de la información (línea 830), si la respuesta es positiva se emite mensaje de éxito; si no, se emite un mensaje de error (línea 831).

Figura 96 Método "GuardarResultados"

```

792  //-- Servicio que permite guardar los resultados:
793  //-- De conocimiento, De desempeño y De resultado de las asignaturas
794  //-- Estos resultados se graban estos 3 conceptos y en con la posibilidad de un orden establecido para poder facilitar la muestra de la información
795  //-- Se tiene la descripción para mostrar : la programación por fase (Programa formativo) y la síntesis para mostrarse en: La Evaluación de competencias adquiridas (Plan de Estudios)
796  //-- Los pesos se relacionan a los resultados más que a algún elemento en específico.
797  //-- Se tiene relación con los poryectos porque también pueden tener peso académico.
798  1 referencia
799  public class GuardarResultadosParametros
800  {
801      4 referencias
802      public string idSilabo { get; set; }
803      1 referencia
804      public List<ResultadosDTO> Resultados { get; set; }
805  }
806  [HttpPost]
807  [Route("GuardarResultados")]
808  0 referencias
809  public JsonResult GuardarResultados(GuardarResultadosParametros parametros)
810  {
811      //-- Verificar valor del parámetro sílabo, en caso de que existir un valor inadecuado se retorna mensaje de error.
812      JsonResult json;
813      if (parametros.idSilabo == null)[...]
814      if (parametros.idSilabo == "")[...]
815      int idSil;
816      if (!Int32.TryParse(parametros.idSilabo, out idSil)[...])
817      else
818      {
819          //-- Guardar resultados listados en el objeto DTO. En caso de existir incongruencia se emite el mensaje de error.
820          //-- Caso contrario se retorna el mensaje de inserción correcta.
821          _handler = new GestionSilabos.Handler.Handler(_context);
822          char grabo = _handler.GrabarResultadosPorDTO(idSil, parametros.Resultados);
823          if (grabo == 'S')
824          {
825              var resultado = new { estado = "success", mensaje = "Se actualizó correctamente los resultados de aprendizaje." };
826              return json = new JsonResult(resultado);
827          }
828          else
829          {
830              var resultado = new { estado = "fail", mensaje = "La información no es correcta." };
831              return json = new JsonResult(resultado);
832          }
833      }
834  }
835  }
836  }
837  }
838  }
839  }
840  }
841  }
842  }
    
```

Fuente: Elaboración propia.

En el método grabar resultados, se busca primeramente los códigos de fase (línea 426), con ellos se valida si corresponden con lo recibido en el objeto DTO (línea 435). En la revisión si es válido el código, se sumaliza los pesos teóricos y prácticos (línea 455). De haber algún código que no corresponde, se retorna como intento fallido (línea 464).

Figura 97 Método "GrabarResultadosPorDTO" - Parte 1

```

421 public char GrabarResultadosPorDTO(int idSil, List<ResultadosDTO> Resultados)
422 {
423     char grabo = 'S';
424     char existe = 'N';
425     //-- se obtienen los códigos de cada fase
426     var fases = ObtenerFasesSemestre(idSil);
427     int faseI = fases.Select(x => x.Id).ToArray()[0];
428     int faseII = fases.Select(x => x.Id).ToArray()[1];
429     int faseIII = fases.Select(x => x.Id).ToArray()[2];
430
431     double totalResultadosFaseI = 0;
432     double totalResultadosFaseII = 0;
433     double totalResultadosFaseIII = 0;
434     //-- Se veerifica si los códigos de fase corresponden a los que se utilizará en el semestre
435     foreach (var item in Resultados)
436     {
437         existe = 'N';
438         for (int i = 0; i < fases.Count; i++)
439         {
440
441             if (IdFasePorNumeroFase(item.IdFase, fases) == fases.Select(x => x.Id).ToArray()[i])
442             {
443                 existe = 'S';
444                 break;
445             }
446         }
447         if (existe == 'N')
448         {
449             break;
450         }
451     }
452     //-- Debo capturar el total del peso académico por fase antes de guardar para poder analizar si supera el 100%.
453     else
454     {
455         if (IdFasePorNumeroFase(item.IdFase, fases) == faseI)
456             totalResultadosFaseI = totalResultadosFaseI + item.PesoPractico + item.PesoTeorico;
457         if (IdFasePorNumeroFase(item.IdFase, fases) == faseII)
458             totalResultadosFaseII = totalResultadosFaseII + item.PesoPractico + item.PesoTeorico;
459         if (IdFasePorNumeroFase(item.IdFase, fases) == faseIII)
460             totalResultadosFaseIII = totalResultadosFaseIII + item.PesoPractico + item.PesoTeorico;
461     }
462 }
463
464 if (existe == 'N')
465 {
466     return grabo = 'N';
467 }

```

Fuente: Elaboración propia.

En caso de que supere la validación, se procede a consultar los proyectos que pertenecen a la asignatura ya que también contienen un peso académico (línea 470). El total de peso entre resultados y proyectos por fase no debe superar el 100%, si alguna fase supera el límite, se retorna como intento fallido de inserción (línea 473).

Figura 98 Método "GrabarResultadosPorDTO" - Parte 2

```

469 //-- El peso académico también o llevan los proyectos, debido a eso se totaliza resultados y proyectos.
470 var proyecto = _context.Proyecto.Where(x => x.IdAsignatura == idSil).Select(x => new { x.IdFase, x.PesoTeorico, x.PesoPractico }).ToList();
471 foreach (var item in proyecto)
472 {
473     if (IdFasePorNumeroFase(item.IdFase, fases) == faseI)
474         totalResultadosFaseI = totalResultadosFaseI + item.PesoPractico + item.PesoTeorico;
475     if (IdFasePorNumeroFase(item.IdFase, fases) == faseII)
476         totalResultadosFaseII = totalResultadosFaseII + item.PesoPractico + item.PesoTeorico;
477     if (IdFasePorNumeroFase(item.IdFase, fases) == faseIII)
478         totalResultadosFaseIII = totalResultadosFaseIII + item.PesoPractico + item.PesoTeorico;
479 }
480
481 //-- Se debe colocar también el peso académico del proyecto
482
483 if (totalResultadosFaseI > 100 || totalResultadosFaseII > 100 || totalResultadosFaseIII > 100)
484 {
485     return grabo = 'N';
486 }
487

```

Fuente: Elaboración propia.

Posteriormente se elimina la información ingresada hasta el momento (línea 491). Eliminada la lista anterior, se inserta la nueva lista de resultados creando un nuevo objeto por cada resultado (línea 496). Acabada la inserción, si no existió complicaciones se graba el log con el registro de inserción exitosa (línea 526).

Figura 99 Método "GrabarResultadosPorDTO" - Parte 3

```

488     }
489
490     //-- Eliminar lista de resultados anteriormente ingresados
491     _context.Database.ExecuteSqlRaw("DELETE FROM Resultado WHERE IdAsignatura = {0}", idSil.ToString());
492     int numfase;
493     int fase;
494
495     //-- Registrar nueva lista de resultados. Cada resultado es un objeto insertado.
496     foreach (var item in Resultados)
497     {
498         numfase = item.IdFase;
499         fase = IdFasePorNumeroFase(numfase, fases);
500         if (fase != 0)
501         {
502             ResultadosModel resultados = new ResultadosModel
503             {
504                 UsuarioCreacion = "User",
505                 UsuarioModificacion = "User",
506                 FechaCreacion = DateTime.Now,
507                 FechaModificacion = DateTime.Now,
508                 Estado = 1,
509                 IdAsignatura = idSil,
510                 IdResultado = item.IdResultado,
511                 Orden = item.Orden,
512                 IdFase = fase,
513                 Descripcion = item.Descripcion,
514                 Sintesis = item.Sintesis,
515                 PesoTeorico = item.PesoTeorico,
516                 PesoPractico = item.PesoPractico
517             };
518             _context.Resultado.Add(resultados);
519         }
520         else
521         {
522             grabo = 'N';
523         }
524     }
525     //-- Se graba el log con la acción de resultados de aprendizaje.
526     if (grabo == 'S')
527     {
528         _context.SaveChanges();
529         String Concepto = "RESULTADOS";
530         String Descripcion = "RESULTADOS REGISTRADOS";
531         GrabarAsignaturaLog(idSil, Concepto, Descripcion, "resultados");
532     }
533 }
534 return grabo;
535 }

```

Fuente: Elaboración propia.

- **Servicio “ListarAsignaturaEstrategia”**

Servicio que permite listar las estrategias pertenecientes a la asignatura desarrollada por el docente.

El Servicio evalúa el código recibido para determinar si el tipo de dato es correspondiente al que se usa en los identificadores de las asignaturas (línea 909). Se obtienen las estrategias ingresadas hasta el momento (línea 926). A las estrategias que se encuentran enlistadas en la tabla, se le agrega descripción (línea 941), pero en caso de que no exista un resultado, se enviará los objetos vacíos (línea 953).

Figura 100 Método "ListarAsignaturaEstrategia"

```

899 public class ListarAsignaturaEstrategiaParametros
900 {
901     4 referencias
902     public string idSilabo { get; set; }
903 }
904
905 [HttpPost]
906 [Route("ListarAsignaturaEstrategia")]
907 public ActionResult<IEnumerable<AsignaturaEstrategiaDTO>> ListarAsignaturaEstrategia(ListarAsignaturaEstrategiaParametros parametros)
908 {
909     //-- Verificar valor del parámetro sílabo, en caso de que existir un valor inadecuado se retorna mensaje de error.
910     if (parametros.idSilabo == null) {...}
911     if (parametros.idSilabo == "") {...}
912     int idSil;
913     if (!Int32.TryParse(parametros.idSilabo, out idSil)) {...}
914     else
915     {
916         //-- Listar estrategias y sus datos para la asignatura.
917         List<AsignaturaEstrategiaDTO> resultado = new List<AsignaturaEstrategiaDTO>();
918         resultado = _context.AsignaturaEstrategia.Join(_context.Fase, asig => asig.IdFase, fase => fase.Id, (asig, fase) => new { asig, fase })
919             .Where(x => x.asig.IdAsignatura == idSil)
920             .Select(x => new AsignaturaEstrategiaDTO
921                 {
922                     IdResultado = x.asig.IdResultado,
923                     Orden = x.asig.Orden,
924                     IdFase = x.fase.NroFase,
925                     IdEstrategia = x.asig.IdEstrategia,
926                     Descripcion = x.asig.Descripcion,
927                     UsuarioModificacion = x.asig.UsuarioModificacion,
928                     FechaModificacion = x.asig.FechaModificacion
929                 });
930     }
931
932     //-- A las estrategias seleccionadas se les agrega la descripción ya que no se almacena en la misma tabla.
933     var estrategias = _context.Estrategia.Select(x => new { x.Id, x.Descripcion }).ToArray();
934     foreach (var item in resultado)
935     {
936         for (int i = 0; i < estrategias.Count(); i++)
937         {
938             if (item.IdEstrategia == estrategias[i].Id)
939             {
940                 item.Descripcion = estrategias[i].Descripcion;
941                 break;
942             }
943         }
944     }
945
946     //-- En caso de que no existan estrategias, se retorna un objeto vacío.
947     if (resultado.Count() == 0)
948     {
949         resultado.Add(new AsignaturaEstrategiaDTO { IdResultado = 0, Orden = 0, IdEstrategia = 0, IdFase = 0, Descripcion = "" });
950     }
951     return resultado;
952 }
953
954 }
955
956
957
958
959
960

```

Fuente: Elaboración propia.

- Servicio “GuardarAsignaturaEstrategia”

Servicio que permite guardar las estrategias pertenecientes a la asignatura desarrollada por el docente.

El Servicio evalúa el código recibido para determinar si el tipo de dato es correspondiente al que se usa en los identificadores de las asignaturas (línea 976). Superada la evaluación, se llama el método “GrabarEstrategiasPorDTO”, que realiza la inserción de la información (línea 997), si la respuesta es positiva se emite mensaje de éxito; si no, se emite un mensaje de error (línea 999).

Figura 101 Método "GuardarAsignaturaEstrategia"

```

962  //-- Servicio que permite Guardar Estrategias definidas para la asignatura a desarrollar.
963  1 referencia
964  public class GuardarAsignaturaEstrategiaParametros
965  {
966      4 referencias
967      public string idSilabo { get; set; }
968      1 referencia
969      public List<AsignaturaEstrategiaDTO> AsignaturaEstrategias { get; set; }
970  }
971  [HttpPost]
972  [Route("GuardarAsignaturaEstrategia")]
973  0 referencias
974  public JsonResult GuardarAsignaturaEstrategia(GuardarAsignaturaEstrategiaParametros parametros)
975  {
976      //-- Verificar valor del parámetro sílabo, en caso de que existir un valor inadecuado se retorna mensaje de error.
977      JsonResult json;
978      if (parametros.idSilabo == null)
979      {
980          var resultado = new { estado = "fail", mensaje = "Información errónea." };
981          return json = new JsonResult(resultado);
982      }
983      if (parametros.idSilabo == "")
984      {
985          parametros.idSilabo = "0";
986      }
987      int idSil;
988      if (!Int32.TryParse(parametros.idSilabo, out idSil))
989      {
990          var resultado = new { estado = "fail", mensaje = "Información errónea." };
991          return json = new JsonResult(resultado);
992      }
993      else
994      {
995          //-- Guardar Estartegias para la asignatura listados en el objeto DTO. En caso de existir incongruencia se emite el mensaje de error.
996          //-- Caso contrario se retorna el mensaje de inserción correcta.
997          _handler = new GestionSilabos.Handler.Handler(_context);
998          char grabo = _handler.GrabarEstrategiasPorDTO(idSil, parametros.AsignaturaEstrategias);
999
1000         if (grabo == 'S')
1001         {
1002             var resultado = new { estado = "success", mensaje = "Se actualizaron correctamente las estrategias." };
1003             return json = new JsonResult(resultado);
1004         }
1005         else
1006         {
1007             var resultado = new { estado = "fail", mensaje = "La información no es correcta." };
1008             return json = new JsonResult(resultado);
1009         }
1010     }
1011 }

```

Fuente: Elaboración propia.

En el método grabar estrategias, se busca primeramente los códigos de fase (línea 540). Se elimina la lista de estrategias que estaban como preexistentes (línea 542). Se obtienen la lista de estrategias configuradas que se pueden elegir al desarrollar la sección (línea 544) con ellos se valida si corresponden con lo recibido en el objeto DTO (línea 552). En la revisión si son los válidos los códigos, se procede a grabar la información (línea 572). De haber algún código que no corresponde, se retorna como intento fallido (línea 592). Finalmente se graba el log si todo es válido, con la acción de grabar bibliografía (línea 596).

Figura 102 Método "GrabarEstrategiasPorDTO"

```

537 public char GrabarEstrategiasPorDTO(int idSil, List<AsignaturaEstrategiaDTO> AsignaturaEstrategias)
538 {
539     //-- se obtienen los códigos de cada fase
540     var fases = ObtenerFasesSemestre(idSil);
541     //-- Se elimina la lista de estrategias ingresadas previamente
542     _context.Database.ExecuteSqlRaw("DELETE FROM AsignaturaEstrategia WHERE IdAsignatura = {0}", idSil.ToString());
543     //-- se obtienen las estrategias de la lista configurada para seleccionar
544     var estrategia = _context.Estrategia.Select(x => x.Id).ToList();
545     char existeEstrategia;
546     int numfase;
547     int fase;
548     char grabo = 'S';
549     //-- De las estrategias desarrolladas por el docente, se verifica cuales de ellas contienen un Id que coincida con las estrategias preconfiguradas
550     //-- en caso de que si coincida e identificador, se considera para insertar.
551     //-- Las estrategias propias del docente, requieren de una descripción propia del docente.
552     foreach (var item in AsignaturaEstrategias)
553     {
554         numfase = item.Idfase;
555         fase = IdFasePorNumeroFase(numfase, fases);
556
557         existeEstrategia = 'N';
558         for (int i = 0; i < estrategia.Count; i++)
559         {
560             if (item.IdEstrategia == estrategia.ToArray()[i] /*&& String.IsNullOrEmpty(item.Descripcion)*/)
561             {
562                 existeEstrategia = 'S';
563                 break;
564             }
565             if (item.IdEstrategia == 0 && !(String.IsNullOrEmpty(item.Descripcion)))
566             {
567                 existeEstrategia = 'S';
568                 break;
569             }
570         }
571         //-- Si pasa el filtro, se inserta la estrategia
572         if (fase != 0 && existeEstrategia == 'S')
573         {
574             AsignaturaEstrategiaModel AsignaturaEstrategia = new AsignaturaEstrategiaModel
575             {
576                 UsuarioCreacion = "User",
577                 UsuarioModificacion = "User",
578                 FechaCreacion = DateTime.Now,
579                 FechaModificacion = DateTime.Now,
580                 Estado = 1,
581                 IdAsignatura = idSil,
582                 IdResultado = item.IdResultado,
583                 IdEstrategia = item.IdEstrategia,
584                 Orden = item.Orden,
585                 IdFase = fase,
586                 Descripcion = item.Descripcion
587             };
588             _context.AsignaturaEstrategia.Add(AsignaturaEstrategia);
589         }
590         else
591         {
592             grabo = 'N';
593         }
594     }
595     //-- Se graba el log con la acción de estrategias de aprendizaje.
596     if (grabo == 'S')
597     {
598         _context.SaveChanges();
599         String Concepto = "ESTRATEGIAS";
600         String Descripcion = "ESTRATEGIAS REGISTRADAS";
601         GrabarAsignaturaLog(idSil, Concepto, Descripcion, "estrategias");
602     }
603     return grabo;
604 }

```

Fuente: Elaboración propia.

- Servicio “ListarAsignaturaCriterio”

Servicio que permite listar los criterios pertenecientes a la asignatura desarrollada por el docente.

El Servicio evalúa el código recibido para determinar si el tipo de dato es correspondiente al que se usa en los identificadores de las asignaturas (línea 1015). Se obtienen los criterios ingresados hasta el momento (línea 1040). A los criterios que se encuentran enlistadas en la tabla, se le agrega descripción (línea 1054), pero en caso de que no exista un resultado, se enviará los objetos vacíos (línea 1067).

Figura 103 Método "ListarAsignaturaCriterio"

```

1012 //-- Servicio que permite listar los criterios asociados a la asignatura a desarrollar.
1013 1 referencia
1014 public class ListarAsignaturaCriterioParametros
1015 {
1016     4 referencias
1017     public string idSilabo { get; set; }
1018 }
1019
1020 [HttpPost]
1021 [Route("ListarAsignaturaCriterio")]
1022 0 referencias
1023 public ActionResult<IEnumerable<AsignaturaCriterioDTO>> ListarAsignaturaCriterio(ListarAsignaturaCriterioParametros parametros)
1024 {
1025     //-- Verificar valor del parámetro sílabo, en caso de que existir un valor inadecuado se retorna mensaje de error.
1026     if (parametros.idSilabo == null) {...}
1027     if (parametros.idSilabo == "") {...}
1028     int idSil;
1029     if (!Int32.TryParse(parametros.idSilabo, out idSil) {...}
1030     else
1031     {
1032         //-- Listar criterios de la asignatura y sus datos
1033         List<AsignaturaCriterioDTO> resultado = new List<AsignaturaCriterioDTO>();
1034         resultado = _context.AsignaturaCriterio.Join(_context.Fase, asign => asign.IdFase, fase => fase.Id, (asign, fase) => new { asign, fase })
1035             .Where(x => x.asig.IdAsignatura == idSil)
1036             .Select(x => new AsignaturaCriterioDTO
1037             {
1038                 IdResultado = x.asig.IdResultado,
1039                 Orden = x.asig.Orden,
1040                 IdFase = x.fase.NroFase,
1041                 IdCriterio = x.asig.IdCriterio,
1042                 Descripcion = x.asig.Descripcion,
1043                 UsuarioModificacion = x.asig.UsuarioModificacion,
1044                 FechaModificacion = x.asig.FechaModificacion
1045             }).ToList();
1046
1047         //-- A los criterios seleccionados se les agrega la descripción ya que no se almacena en la misma tabla.
1048         var criterios = _context.Criterio.Select(x => new { x.Id, x.Descripcion }).ToArray();
1049         foreach (var item in resultado)
1050         {
1051             for (int i = 0; i < criterios.Count(); i++)
1052             {
1053                 if (item.IdCriterio == criterios[i].Id)
1054                 {
1055                     item.Descripcion = criterios[i].Descripcion;
1056                     break;
1057                 }
1058             }
1059         }
1060
1061         //-- En caso de no existir criterios, se devuelve un objeto vacío.
1062         if (resultado.Count() == 0)
1063         {
1064             resultado.Add(new AsignaturaCriterioDTO { IdResultado = 0, Orden = 0, IdCriterio = 0, IdFase = 0, Descripcion = "" });
1065         }
1066         return resultado;
1067     }
1068 }
1069
1070
1071
1072
1073

```

Fuente: Elaboración propia.

- Servicio “GuardarAsignaturaCriterio”

Servicio que permite guardar criterios pertenecientes a la asignatura desarrollada por el docente.

El Servicio evalúa el código recibido para determinar si el tipo de dato es correspondiente al que se usa en los identificadores de las asignaturas (línea 1089). Superada la evaluación, se llama el método “GrabarCriteriosPorDTO”, que realiza la inserción de la información (línea 1109), si la respuesta es positiva se emite mensaje de éxito; si no, se emite un mensaje de error (línea 1111).

Figura 104 Método “*GuardarAsignaturaCriterio*”

```

1076  //--Servicio que permite guardar los criterios asociados a la asignatura a desarrollar.
1077  1 referencia
1078  public class GuardarAsignaturaCriterioParametros
1079  {
1080      4 referencias
1081      public string idSilabo { get; set; }
1082      1 referencia
1083      public List<AsignaturaCriterioDTO> AsignaturaCriterios { get; set; }
1084  }
1085  [HttpPost]
1086  [Route("GuardarAsignaturaCriterio")]
1087  0 referencias
1088  public JsonResult GuardarAsignaturaCriterio(GuardarAsignaturaCriterioParametros parametros)
1089  {
1090      //-- Verificar valor del parámetro sílabo, en caso de que existir un valor inadecuado se retorna mensaje de error.
1091      JsonResult json;
1092      if (parametros.idSilabo == null)
1093      {
1094          var resultado = new { estado = "fail", mensaje = "Información errónea." };
1095          return json = new JsonResult(resultado);
1096      }
1097      if (parametros.idSilabo == "")
1098      {
1099          parametros.idSilabo = "0";
1100      }
1101      int idSil;
1102      if (!Int32.TryParse(parametros.idSilabo, out idSil))
1103      {
1104          var resultado = new { estado = "fail", mensaje = "Información errónea." };
1105          return json = new JsonResult(resultado);
1106      }
1107      else
1108      {
1109          //--Guardar lista de criterios enviados en objetos DTO. En caso de existir incongruencia se emite el mensaje de error.
1110          //-- Caso contrario se retorna el mensaje de inserción correcta.
1111          _handler = new GestionSilabos.Handler.Handler(_context);
1112          char grabo = _handler.GrabarCriteriosPorDTO(idSil, parametros.AsignaturaCriterios);
1113          if (grabo == 'S')
1114          {
1115              var resultado = new { estado = "success", mensaje = "Se actualizaron correctamente los criterios." };
1116              return json = new JsonResult(resultado);
1117          }
1118          else
1119          {
1120              var resultado = new { estado = "fail", mensaje = "La información no es correcta." };
1121              return json = new JsonResult(resultado);
1122          }
1123      }
1124  }

```

Fuente: Elaboración propia.

En el método grabar criterios, se busca primeramente los códigos de fase (línea 616). Se elimina la lista de criterios que estaban como preexistentes (línea 613). Se obtienen la lista de criterios configurados que se pueden elegir al desarrollar la sección (línea 615) con ellos se valida si corresponden con lo recibido en el objeto DTO (línea 623). En la revisión si son los válidos los códigos, se procede a grabar la información (línea 643). De haber algún código que no corresponde, se retorna como intento fallido (línea 663). Finalmente se graba el log si todo es válido, con la acción de grabar bibliografía (línea 667).



Figura 105 Método "GrabarCriteriosPorDTO"

```

607 public char GrabarCriteriosPorDTO(int idSil, List<AsignaturaCriterioDTO> AsignaturaCriterios)
608 {
609     // --se obtienen los códigos de cada fase
610     var fases = ObtenerFasesSemestre(idSil);
611     //-- Se elimina la lista de criterios ingresados previamente
612     _context.Database.ExecuteSqlCommand("DELETE FROM AsignaturaCriterio WHERE IdAsignatura = {0}", idSil.ToString());
613     //-- se obtienen los criterios de la lista configurada para seleccionar
614     var criterio = _context.Criterio.Select(x => x.Id).ToList();
615     char existeCriterio;
616     int numfase;
617     int fase;
618     char grabo = 'S';
619     //-- De los criterios desarrolladas por el docente, se verifica cuales de ellas contienen un Id que coincida con los criterios preconfigurados
620     //-- en caso de que si coincide e identificador, se considera para insertar.
621     //-- Los criterios propios del docente, requieren de una descripción propia del docente.
622     foreach (var item in AsignaturaCriterios)
623     {
624         numfase = item.IdFase;
625         fase = IdFasePorNumeroFase(numfase, fases);
626
627         existeCriterio = 'N';
628         for (int i = 0; i < criterio.Count; i++)
629         {
630             if (item.IdCriterio == criterio.ToArray()[i] /*&& String.IsNullOrEmpty(item.Descripcion)*/)
631             {
632                 existeCriterio = 'S';
633                 break;
634             }
635             if (item.IdCriterio == 0 && !(String.IsNullOrEmpty(item.Descripcion)))
636             {
637                 existeCriterio = 'S';
638                 break;
639             }
640         }
641         //-- Si pasa el filtro, se inserta el criterio
642         if (fase != 0 && existeCriterio == 'S')
643         {
644             AsignaturaCriterioModel AsignaturaCriterio = new AsignaturaCriterioModel
645             {
646                 UsuarioCreacion = "User",
647                 UsuarioModificacion = "User",
648                 FechaCreacion = DateTime.Now,
649                 FechaModificacion = DateTime.Now,
650                 Estado = 1,
651                 IdAsignatura = idSil,
652                 IdResultado = item.IdResultado,
653                 IdCriterio = item.IdCriterio,
654                 Orden = item.Orden,
655                 IdFase = fase,
656                 Descripcion = item.Descripcion
657             };
658             _context.AsignaturaCriterio.Add(AsignaturaCriterio);
659         }
660         else
661         {
662             grabo = 'N';
663         }
664     }
665     //-- Se graba el log con la acción de criterios de aprendizaje.
666     if (grabo == 'S')
667     {
668         _context.SaveChanges();
669         String Concepto = "CRITERIOS";
670         String Descripcion = "CRITERIOS REGISTRADOS";
671         GrabarAsignaturaLog(idSil, Concepto, Descripcion, "criterios");
672     }
673     return grabo;
674 }
675

```

Fuente: Elaboración propia.

- Servicio “ListarAsignaturaInstrumento”

Servicio que permite listar los instrumentos pertenecientes a la asignatura desarrollada por el docente.

El Servicio evalúa el código recibido para determinar si el tipo de dato es correspondiente al que se usa en los identificadores de las asignaturas (línea 1035). Se obtienen los instrumentos ingresados hasta el momento (línea 1152). A los instrumentos que se encuentran enlistados en la tabla, se le agrega descripción (línea 1167), pero en caso de que no exista un resultado, se enviará los objetos vacíos (línea 1180).

Figura 106 Método "ListarAsignaturaInstrumento"

```

1130 [HttpPost]
1131 [Route("ListarAsignaturaInstrumento")]
1132 public ActionResult<IEnumerable<AsignaturaInstrumentoDTO>> ListarAsignaturaInstrumento(ListarAsignaturaInstrumentoParametros parametros)
1133 {
1134     //-- Verificar valor del parámetro sílabo, en caso de que existir un valor inadecuado se retorna mensaje de error.
1135     if (parametros.idSilabo == null) {...}
1139     if (parametros.idSilabo == "") {...}
1143     int idSil;
1144     if (!Int32.TryParse(parametros.idSilabo, out idSil)) {...}
1148     else
1149     {
1150         //-- Listar instrumentos relacionados a la asignatura de cada fase ingresada.
1151         List<AsignaturaInstrumentoDTO> resultado = new List<AsignaturaInstrumentoDTO>();
1152         resultado = _context.AsignaturaInstrumento.Join(_context.Fase, asig => asig.IdFase, fase => fase.Id, (asig, fase) => new { asig, fase })
1153             .Where(x => x.asig.IdAsignatura == idSil)
1154             .Select(x => new AsignaturaInstrumentoDTO
1155                 {
1156                     IdResultado = x.asig.IdResultado,
1157                     Orden = x.asig.Orden,
1158                     IdFase = x.fase.NroFase,
1159                     IdInstrumento = x.asig.IdInstrumento,
1160                     Descripcion = x.asig.Descripcion,
1161                     UsuarioModificacion = x.asig.UsuarioModificacion,
1162                     FechaModificacion = x.asig.FechaModificacion
1163                 });
1164             .ToList();
1165
1166         //-- A los instrumentos seleccionados se les agrega la descripción ya que no se almacena en la misma tabla.
1167         var instrumentos = _context.Instrumento.Select(x => new { x.Id, x.Descripcion }).ToArray();
1168         foreach (var item in resultado)
1169         {
1170             for (int i = 0; i < instrumentos.Count(); i++)
1171             {
1172                 if (item.IdInstrumento == instrumentos[i].Id)
1173                 {
1174                     item.Descripcion = instrumentos[i].Descripcion;
1175                     break;
1176                 }
1177             }
1178         }
1179         //-- De no existir instrumentos, retornar un objeto vacío.
1180         if (resultado.Count() == 0)
1181         {
1182             resultado.Add(new AsignaturaInstrumentoDTO { IdResultado = 0, Orden = 0, IdInstrumento = 0, IdFase = 0, Descripcion = "" });
1183         }
1184         return resultado;
1185     }
1186 }

```

Fuente: Elaboración propia.

- Servicio “GuardarAsignaturaInstrumento”

Servicio que permite guardar instrumentos pertenecientes a la asignatura desarrollada por el docente.

El Servicio evalúa el código recibido para determinar si el tipo de dato es correspondiente al que se usa en los identificadores de las asignaturas (línea 1264). Superada la evaluación, se llama el método “GrabarInstrumentosPorDTO”, que realiza la inserción de la información (línea 1284), si la respuesta es positiva se emite mensaje de éxito; si no, se emite un mensaje de error (línea 1286).

Figura 107 Método "GuardarAsignaturaInstrumento"

```

1251 1251 //-- Servicio que permite guardar la lista de instrumentos asociados a la asignatura.
1252 1252 1 referencia
1253 1253 public class GuardarAsignaturaInstrumentoParametros
1254 {
1255     4 referencias
1256     public string idSilabo { get; set; }
1257     1 referencia
1258     public List<AsignaturaInstrumentoDTO> AsignaturaInstrumentos { get; set; }
1259 }
1260
1261 [HttpPost]
1262 [Route("GuardarAsignaturaInstrumento")]
1263 0 referencias
1264 public JsonResult GuardarAsignaturaInstrumento(GuardarAsignaturaInstrumentoParametros parametros)
1265 {
1266     //-- Verificar valor del parámetro sílabo, en caso de que existir un valor inadecuado se retorna mensaje de error.
1267     JsonResult json;
1268     if (parametros.idSilabo == null)
1269     {
1270         var resultado = new { estado = "fail", mensaje = "Información errónea." };
1271         return json = new JsonResult(resultado);
1272     }
1273     if (parametros.idSilabo == "")
1274     {
1275         parametros.idSilabo = "0";
1276     }
1277     int idSil;
1278     if (!Int32.TryParse(parametros.idSilabo, out idSil))
1279     {
1280         var resultado = new { estado = "fail", mensaje = "Información errónea." };
1281         return json = new JsonResult(resultado);
1282     }
1283     else
1284     {
1285         //-- Guardar instrumentos listados en los objetos DTO asociados a la asignatura. En caso de existir incongruencia se emite el mensaje de error.
1286         //-- Caso contrario se retorna el mensaje de inserción correcta.
1287         _handler = new GestionSilabos.Handler.Handler(_context);
1288         char grabo = _handler.GrabarInstrumentosPorDTO(idSil, parametros.AsignaturaInstrumentos);
1289
1290         if (grabo == 'S')
1291         {
1292             var resultado = new { estado = "success", mensaje = "Se actualizaron correctamente los instrumentos." };
1293             return json = new JsonResult(resultado);
1294         }
1295         else
1296         {
1297             var resultado = new { estado = "fail", mensaje = "La información no es correcta." };
1298             return json = new JsonResult(resultado);
1299         }
1300     }
1301 }

```

Fuente: Elaboración propia.

En el método grabar instrumentos, se busca primeramente los códigos de fase (línea 681). Se elimina la lista de instrumentos que estaban como preexistentes (línea 683). Se obtienen la lista de instrumentos configurados que se pueden elegir al desarrollar la sección (línea 685) con ellos se valida si corresponden con lo recibido en el objeto DTO (línea 693). En la revisión si son los válidos los códigos, se procede a grabar la información (línea 713). De haber algún código que no corresponde, se retorna como intento fallido (línea 733). Finalmente se graba el log si todo es válido, con la acción de grabar bibliografía (línea 737).

Figura 108 Método "GrabarInstrumentosPorDTO"

```

677 public char GrabarInstrumentosPorDTO(int idSil, List<AsignaturaInstrumentoDTO> AsignaturaInstrumentos)
678 {
679     // --se obtienen los códigos de cada fase
680     var fases = ObtenerFasesSemestre(idSil);
681     // -- Se elimina la lista de instrumentos ingresados previamente
682     _context.Database.ExecuteSqlRaw("DELETE FROM AsignaturaInstrumento WHERE IdAsignatura = {0}", idSil.ToString());
683     // -- se obtienen los instrumentos de la lista configurada para seleccionar
684     var instrumento = _context.Instrumento.Select(x => x.Id).ToList();
685     char existeInstrumento;
686     int numfase;
687     int fase;
688     char grabo = 'S';
689     // -- De los instrumentos desarrolladas por el docente, se verifica cuales de ellas contienen un Id que coincida con los instrumentos preconfigurados
690     // -- en caso de que si coincide e identificador, se considera para insertar.
691     // -- Los instrumentos propios del docente, requieren de una descripción propia del docente.
692     foreach (var item in AsignaturaInstrumentos)
693     {
694         numfase = item.IdFase;
695         fase = IdFasePorNumeroFase(numfase, fases);
696         existeInstrumento = 'N';
697         for (int i = 0; i < instrumento.Count; i++)
698         {
699             if (item.IdInstrumento == instrumento.ToArray()[i] /*&& String.IsNullOrEmpty(item.Descripcion)*/)
700             {
701                 existeInstrumento = 'S';
702                 break;
703             }
704             if (item.IdInstrumento == 0 && !(String.IsNullOrEmpty(item.Descripcion)))
705             {
706                 existeInstrumento = 'S';
707                 break;
708             }
709         }
710     }
711     // -- Si pasa el filtro, se inserta el instrumento
712     if (fase != 0 && existeInstrumento == 'S')
713     {
714         AsignaturaInstrumentoModel AsignaturaInstrumento = new AsignaturaInstrumentoModel
715         {
716             UsuarioCreacion = "User",
717             UsuarioModificacion = "User",
718             FechaCreacion = DateTime.Now,
719             FechaModificacion = DateTime.Now,
720             Estado = 1,
721             IdAsignatura = idSil,
722             IdResultado = item.IdResultado,
723             IdInstrumento = item.IdInstrumento,
724             Orden = item.Orden,
725             IdFase = fase,
726             Descripcion = item.Descripcion
727         };
728         _context.AsignaturaInstrumento.Add(AsignaturaInstrumento);
729     }
730     else
731     {
732         grabo = 'N';
733     }
734     // -- Se graba el log con la acción de instrumentos de aprendizaje.
735     if (grabo == 'S')
736     {
737         _context.SaveChanges();
738         String Concepto = "INSTRUMENTOS";
739         String Descripcion = "INSTRUMENTOS REGISTRADOS";
740         GrabarAsignaturaLog(idSil, Concepto, Descripcion, "instrumentos");
741     }
742     return grabo;
743 }
744 }
745

```

Fuente: Elaboración propia.

- Servicio “ListarAsignaturaBibliografia”

Servicio que permite listar la bibliografía perteneciente a la asignatura desarrollada por el docente.

El Servicio evalúa el código recibido para determinar si el tipo de dato es correspondiente al que se usa en los identificadores de las asignaturas (línea 1405). Se obtiene la bibliografía ingresada hasta el momento (línea 1422). En caso de que no exista un resultado, se enviará los objetos vacíos (línea 1441).

Figura 109 Método "ListarAsignaturaBibliografía"

```

1394 //-- Servicio que permite listar la bibliografía asociada a la asignatura por cada fase.
1395 1 referencia
1396 public class ListarAsignaturaBibliografiaParametros
1397 {
1398     4 referencias
1399     public string idSilabo { get; set; }
1400 }
1401 [HttpPost]
1402 [Route("ListarAsignaturaBibliografia")]
1403 0 referencias
1404 public ActionResult<IEnumerable<AsignaturaBibliografiaDTO>> ListarAsignaturaBibliografia(ListarAsignaturaBibliografiaParametros parametros)
1405 {
1406     //-- Verificar valor del parámetro silabo, en caso de que existir un valor inadecuado se retorna mensaje de error.
1407     if (parametros.idSilabo == null)
1408     {
1409         return BadRequest();
1410     }
1411     if (parametros.idSilabo == "")
1412     {
1413         parametros.idSilabo = "0";
1414     }
1415     int idSil;
1416     if (!Int32.TryParse(parametros.idSilabo, out idSil))
1417     {
1418         return BadRequest();
1419     }
1420     else
1421     {
1422         //-- Búsqueda de las bibliografías que se asocian a la bibliografía, rescatando los datos propios de cada una para mostrar en la interfaz.
1423         List<AsignaturaBibliografiaDTO> resultado = new List<AsignaturaBibliografiaDTO>();
1424         resultado = _context.AsignaturaBibliografia.Join(_context.Bibliografia, asig => asig.IdBibliografia, bibl => bibl.Id, (asig, bibl) => new { asig, bibl })
1425             .Join(_context.Fase, asig => asig.asig.IdFase, fase => fase.Id, (asig, fase) => new { asig, fase })
1426             .Where(x => x.asig.asig.IdAsignatura == idSil)
1427             .Select(Bibl => new AsignaturaBibliografiaDTO
1428             {
1429                 IdBibliografia = Bibl.asig.asig.IdBibliografia,
1430                 IdFase = Bibl.fase.NroFase,
1431                 Orden = Bibl.asig.asig.Orden,
1432                 Autor = Bibl.asig.bibl.Autor,
1433                 Editorial = Bibl.asig.bibl.Editorial,
1434                 NombreLibro = Bibl.asig.bibl.NombreLibro,
1435                 LugarPublicacion = Bibl.asig.bibl.LugarPublicacion,
1436                 AnioPublicacion = Bibl.asig.bibl.AnioPublicacion,
1437                 UsuarioModificacion = Bibl.asig.asig.UsuarioModificacion,
1438                 FechaModificacion = Bibl.asig.asig.FechaModificacion
1439             }).ToList();
1440         //-- De no encontrar bibliografía, se retorna un objeto vacío.
1441         if (resultado.Count() == 0)
1442         {
1443             resultado.Add(new AsignaturaBibliografiaDTO
1444             { IdBibliografia = 0, IdFase = 0, Orden = 0, Autor = "", Editorial = "", NombreLibro = "", LugarPublicacion = "", AnioPublicacion = "", UsuarioModificacion = "", FechaModificacion = null });
1445         }
1446         return resultado;
1447     }
1448 }

```

Fuente: Elaboración propia.

- Servicio “GuardarAsignaturaBibliografia”

Servicio que permite guardar la bibliografía de la asignatura desarrollada por el docente.

El Servicio evalúa el código recibido para determinar si el tipo de dato es correspondiente al que se usa en los identificadores de las asignaturas (línea 1359). Superada la evaluación, se llama el método “GrabarBibliografiaPorDTO”, que realiza la inserción de la información (línea 1379), si la respuesta es positiva se emite mensaje de éxito; si no, se emite un mensaje de error (línea 1381).

Figura 110 Método "GuardarAsignaturaBibliografia"

```

1344 1344 //--- Servicio que permite guardar la bibliografía asociada a la asignatura
1345 1345 1 referencia
1346 1346 public class GuardarAsignaturaBibliografiaParametros
1347 1347 {
1348 1348     4 referencias
1349 1349     public string idSilabo { get; set; }
1350 1350     1 referencia
1351 1351     public List<AsignaturaBibliografiaDTO> AsignaturaBibliografia { get; set; }
1352 1352 }
1353 1353 [HttpPost]
1354 1354 [Route("GuardarAsignaturaBibliografia")]
1355 1355 0 referencias
1356 1356 public JsonResult GuardarAsignaturaBibliografia(GuardarAsignaturaBibliografiaParametros parametros)
1357 1357 {
1358 1358     //--- Verificar valor del parámetro silabo, en caso de que existir un valor inadecuado se retorna mensaje de error.
1359 1359     JsonResult json;
1360 1360     if (parametros.idSilabo == null)
1361 1361     {
1362 1362         var resultado = new { estado = "fail", mensaje = "Información errónea." };
1363 1363         return json = new JsonResult(resultado);
1364 1364     }
1365 1365     if (parametros.idSilabo == "")
1366 1366     {
1367 1367         parametros.idSilabo = "0";
1368 1368     }
1369 1369     int idSil;
1370 1370     if (!Int32.TryParse(parametros.idSilabo, out idSil))
1371 1371     {
1372 1372         var resultado = new { estado = "fail", mensaje = "Información errónea." };
1373 1373         return json = new JsonResult(resultado);
1374 1374     }
1375 1375     else
1376 1376     {
1377 1377         //--- Guardar bibliografía listada en los objetos DTO asociados a la asignatura. En caso de existir incongruencia se emite el mensaje de error.
1378 1378         //--- Caso contrario se retorna el mensaje de inserción correcta.
1379 1379         _handler = new GestionSilabos.Handler.Handler(_context);
1380 1380         char grabo = _handler.GrabarBibliografiaPorDTO(idSil, parametros.AsignaturaBibliografia);
1381 1381         if (grabo == 'S')
1382 1382         {
1383 1383             var resultado = new { estado = "success", mensaje = "Se actualizó correctamente la bibliografía." };
1384 1384             return json = new JsonResult(resultado);
1385 1385         }
1386 1386         else
1387 1387         {
1388 1388             var resultado = new { estado = "fail", mensaje = "La información no es correcta." };
1389 1389             return json = new JsonResult(resultado);
1390 1390         }
1391 1391     }
1392 1392 }

```

Fuente: Elaboración propia.

En el método grabar bibliografía, se busca primeramente los códigos de fase (línea 752). Se elimina la lista de bibliografía que estaba como preexiste (línea 754). Se obtienen la lista de elementos de bibliografía configurados que se pueden elegir al desarrollar la sección (línea 756) con ellos se valida si corresponden con lo recibido en el objeto DTO (línea 763). En la revisión si son los válidos los códigos, se procede a grabar la información (línea 778). De haber algún código que no corresponde, se retorna como intento fallido (línea 796). Finalmente se graba el log si todo es válido, con la acción de grabar bibliografía (línea 800).

Figura 111 Método "GrabarAsignaturaBibliografiaDTO"

```

749 public char GrabarBibliografiaPorDTO(int idSil, List<AsignaturaBibliografiaDTO> AsignaturaBibliografias)
750 {
751     // --se obtienen los códigos de cada fase
752     var fases = ObtenerFasesSemestre(idSil);
753     //-- Se elimina la lista de la bibliografía ingresada previamente
754     _context.Database.ExecuteSqlRaw("DELETE FROM AsignaturaBibliografia WHERE IdAsignatura = {0}", idSil.ToString());
755     //-- se obtiene la bibliografía de la lista configurada para seleccionar
756     var bibliografia = _context.Bibliografia.Select(x => x.Id).ToList();
757     char existeBibliografia;
758     int numfase;
759     int fase;
760     char grabo = 'S';
761     //-- De la bibliografía seleccionada por el docente, se verifica cuales de ellas contienen un Id que coincida con la bibliografía preconfigurada
762     //-- en caso de que si coincide e identificador, se considera para insertar.
763     foreach (var item in AsignaturaBibliografias)
764     {
765         numfase = item.IdFase;
766         fase = IdFasePorNumeroFase(numfase, fases);
767
768         existeBibliografia = 'N';
769         for (int i = 0; i < bibliografia.Count; i++)
770         {
771             if (item.IdBibliografia == bibliografia.ToArray()[i] /*&& String.IsNullOrEmpty(item.Descripcion)*/)
772             {
773                 existeBibliografia = 'S';
774                 break;
775             }
776         }
777         //-- Si pasa el filtro, se inserta la bibliografía
778         if (fase != 0 && existeBibliografia == 'S')
779         {
780             AsignaturaBibliografiaModel AsignaturaBibliografia = new AsignaturaBibliografiaModel
781             {
782                 UsuarioCreacion = "User",
783                 UsuarioModificacion = "User",
784                 FechaCreacion = DateTime.Now,
785                 FechaModificacion = DateTime.Now,
786                 Estado = 1,
787                 IdAsignatura = idSil,
788                 Orden = item.Orden,
789                 IdFase = fase,
790                 IdBibliografia = item.IdBibliografia
791             };
792             _context.AsignaturaBibliografia.Add(AsignaturaBibliografia);
793         }
794         else
795         {
796             grabo = 'N';
797         }
798     }
799     //-- Se graba el log con la acción de bibliografía.
800     if (grabo == 'S')
801     {
802         _context.SaveChanges();
803         String Concepto = "BIBLIOGRAFIA";
804         String Descripcion = "BIBLIOGRAFÍA INGRESADA";
805         GrabarAsignaturaLog(idSil, Concepto, Descripcion, "bibliografia");
806     }
807     return grabo;
808 }

```

Fuente: Elaboración propia.

- Servicio “ListarProyectosAsignatura”

Servicio que permite listar los proyectos pertenecientes a la asignatura desarrollada por el docente.

El Servicio evalúa el código recibido para determinar si el tipo de dato es correspondiente al que se usa en los identificadores de las asignaturas (línea 37). Se obtienen los proyectos ingresados hasta el momento (línea 55).

Figura 112 Método "ListarProyectosAsignatura"

```

27 public class ListarProyectosAsignaturaParametro
28 {
29     4 referencias
30     public string idSilabo { get; set; }
31 }
32
33 [HttpPost]
34 [Route("ListarProyectosAsignatura")]
35 public ActionResult<IEnumerable<ProyectoDTO>> ListarProyectosAsignatura(ListarProyectosAsignaturaParametro parametro)
36 {
37     //-- Verificar valor del parámetro sílabo, en caso de que existir un valor inadecuado se retorna mensaje de error.
38     if (parametro.idSilabo == null)
39     {
40         return BadRequest("Información errónea.");
41     }
42     if (parametro.idSilabo == "")
43     {
44         parametro.idSilabo = "0";
45     }
46     int idSil;
47     if (!Int32.TryParse(parametro.idSilabo, out idSil))
48     {
49         return BadRequest("Información errónea.");
50     }
51     else
52     {
53         //-- Obtener proyectos y datos asociados a la asignatura para que sean mostrados en la interfaz
54         List<ProyectoDTO> proyectos = new List<ProyectoDTO>();
55         proyectos = _context.Proyecto.Join(_context.Fase, asig => asig.IdFase, fase => fase.Id, (asig, fase) => new { asig, fase })
56             .Where(x => x.asig.IdAsignatura == idSil)
57             .Select(p => new ProyectoDTO {
58                 IdProyecto = p.asig.IdProyecto,
59                 NroOrden = p.asig.NroOrden,
60                 IdFase = p.fase.NroFase,
61                 Beneficiarios = p.asig.Beneficiarios,
62                 DenominacionActividad = p.asig.DenominacionActividad,
63                 IndicadoresEvaluacion = p.asig.IndicadoresEvaluacion,
64                 Proposito = p.asig.Proposito,
65                 Responsables = p.asig.Responsables,
66                 Cronograma = p.asig.Cronograma,
67                 PesoTeorico = p.asig.PesoTeorico,
68                 PesoPractico = p.asig.PesoPractico,
69                 UsuarioModificacion = p.asig.UsuarioModificacion,
70                 FechaModificacion = p.asig.FechaModificacion
71             }).ToList();
72
73         return proyectos;
74     }
75 }
76

```

Fuente: Elaboración propia.

- Servicio “GuardarProyectosAsignatura”

Servicio que permite guardar los proyectos de la asignatura desarrollada por el docente.

El Servicio evalúa el código recibido para determinar si el tipo de dato es correspondiente al que se usa en los identificadores de las asignaturas (línea 91). Superada la evaluación, se llama el método “GrabarProyectosPorDTO”, que realiza la inserción de la información (línea 113), si la respuesta es positiva se emite mensaje de éxito; si no, se emite un mensaje de error (línea 115).

Figura 113 Método “GuardarProyectosAsignatura”

```

78  //-- Servicio que permite guardar los proyectos diseñados para la asignatura.
79  1 referencia
80  public class GuardarProyectosAsignaturaParametros
81  {
82      4 referencias
83      public string idSilabo { get; set; }
84      2 referencias
85      public List<ProyectoDTO> proyectos { get; set; }
86  }
87
88  [HttpPost]
89  [Route("GuardarProyectosAsignatura")]
90  0 referencias
91  public JsonResult GuardarProyectosAsignatura(GuardarProyectosAsignaturaParametros parametros)
92  {
93      //-- Verificar valor del parámetro sílabo, en caso de que existir un valor inadecuado se retorna mensaje de error.
94      JsonResult json;
95      if (parametros.idSilabo == null)
96      {
97          var resultado = new { estado = "fail", mensaje = "Información errónea." };
98          return json = new JsonResult(resultado);
99      }
100     if (parametros.idSilabo == "")
101     {
102         parametros.idSilabo = "0";
103     }
104     int idsil;
105     if (!Int32.TryParse(parametros.idSilabo, out idsil))
106     {
107         var resultado = new { estado = "fail", mensaje = "Información errónea." };
108         return json = new JsonResult(resultado);
109     }
110     else
111     {
112         //-- En caso de contar con proyectos, grabar la información. En caso de existir incongruencia se emite el mensaje de error.
113         //-- Caso contrario se retorna el mensaje de inserción correcta.
114         if (parametros.proyectos.Count() > 0)
115         {
116             _handler = new GestionSilabos.Handler.Handler(_context);
117             char grabo = _handler.GrabarProyectosPorDTO(idsil, parametros.proyectos);
118
119             if (grabo == 'S')
120             {
121                 var resultado = new { estado = "success", mensaje = "Se actualizaron correctamente los proyectos." };
122                 return json = new JsonResult(resultado);
123             }
124             else
125             {
126                 var resultado = new { estado = "fail", mensaje = "La información no es correcta." };
127                 return json = new JsonResult(resultado);
128             }
129         }
130         else
131         {
132             var resultado = new { estado = "fail", mensaje = "No se actualizó ningún registro." };
133             return json = new JsonResult(resultado);
134         }
135     }
136 }

```

Fuente: Elaboración propia.

En el método grabar proyectos, se busca primeramente los códigos de fase (línea 815). Se verifican los códigos de la fase comparándolos con los recibidos por el objeto DTO para identificar que haya equivalencia (línea 827). A la par se graba el peso académico del proyecto de modo que se tenga un acumulado de los proyectos por fase (línea 847). Si la revisión contiene un elemento que no superó la validación, entonces se determina como grabación fallida (línea 856).

Figura 114 Método "GrabarProyectosPorDTO" – Parte 1

```

810 public char GrabarProyectosPorDTO(int idSil, List<ProyectoDTO> Proyectos)
811 {
812     // --se obtienen los códigos de cada fase
813     char grabo = 'S';
814     char existe = 'N';
815     //-- se obtiene los identificadores de la fase que se está trabajando del semestre
816     var fases = ObtenerFasesSemestre(idSil);
817     int faseI = fases.Select(x => x.Id).ToArray()[0];
818     int faseII = fases.Select(x => x.Id).ToArray()[1];
819     int faseIII = fases.Select(x => x.Id).ToArray()[2];
820
821     double totalResultadosFaseI = 0;
822     double totalResultadosFaseII = 0;
823     double totalResultadosFaseIII = 0;
824     //-- De los proyectos seleccionada por el docente, se verifica cuales de ellas contienen un Id de fase correspondiente
825     //-- a las fases que se están trabajando en el semestre
826     //-- en caso de que si coincida el identificador, se considera para insertar.
827     foreach (var item in Proyectos)
828     {
829         existe = 'N';
830         for (int i = 0; i < fases.Count; i++)
831         {
832             if (IdFasePorNumeroFase(item.IdFase, fases) == fases.Select(x => x.Id).ToArray()[i])
833             {
834                 existe = 'S';
835                 break;
836             }
837         }
838         if (existe == 'N')
839         {
840             break;
841         }
842         //-- Debo capturar el total del peso académico por fase antes de guardar para poder analizar si supera el 100%.
843         else
844         {
845             if (IdFasePorNumeroFase(item.IdFase, fases) == faseI)
846                 totalResultadosFaseI = totalResultadosFaseI + item.PesoPractico + item.PesoTeorico;
847             if (IdFasePorNumeroFase(item.IdFase, fases) == faseII)
848                 totalResultadosFaseII = totalResultadosFaseII + item.PesoPractico + item.PesoTeorico;
849             if (IdFasePorNumeroFase(item.IdFase, fases) == faseIII)
850                 totalResultadosFaseIII = totalResultadosFaseIII + item.PesoPractico + item.PesoTeorico;
851         }
852     }
853     //-- Si no es correcto alguno de los identificadores de fase, entonces se retorna como fallido
854     if (existe == 'N')
855     {
856         return grabo = 'N';
857     }
858 }
859

```

Fuente: Elaboración propia.

En caso de que todos los identificadores sean correctos, se totaliza el peso de los resultados al peso de los proyectos (línea 862). Se valida que el peso de cada fase no supere el 100% (línea 873). Si el total por fase es correcto, entonces se procede a eliminar la lista anterior de proyectos (línea 881). Luego se graban los nuevos proyectos recibidos en el DTO (línea 885). En caso de que sea exitoso se graba el log (línea 918).

Figura 115 Método "GrabarProyectosPorDTO"

```

860 //-- El peso académico también o llevan los proyectos, debido a eso se totaliza resultados y proyectos.
861 var resultados = _context.Resultado.Where(x => x.IdAsignatura == idSil).Select(x => new { x.IdFase, x.PesoTeorico, x.PesoPractico }).ToList();
862 foreach (var item in resultados)
863 {
864     if (IdFasePorNumeroFase(item.IdFase, fases) == faseI)
865         totalResultadosFaseI = totalResultadosFaseI + item.PesoPractico + item.PesoTeorico;
866     if (IdFasePorNumeroFase(item.IdFase, fases) == faseII)
867         totalResultadosFaseII = totalResultadosFaseII + item.PesoPractico + item.PesoTeorico;
868     if (IdFasePorNumeroFase(item.IdFase, fases) == faseIII)
869         totalResultadosFaseIII = totalResultadosFaseIII + item.PesoPractico + item.PesoTeorico;
870 }
871
872 //-- Se debe colocar también el peso académico del proyecto
873 if (totalResultadosFaseI > 100 || totalResultadosFaseII > 100 || totalResultadosFaseIII > 100)
874 {
875     return grabo = 'N';
876 }
877
878 else
879 {
880     //-- Eliminar lista de proyectos que fueron ingresados previamente
881     _context.Database.ExecuteSqlRaw("DELETE FROM Proyecto WHERE IdAsignatura = {0}", idSil.ToString());
882     int numFase;
883     int fase;
884
885     //-- Grabar los proyectos
886     foreach (var item in Proyectos)
887     {
888         numFase = item.IdFase;
889         fase = IdFasePorNumeroFase(numFase, fases);
890         if (fase != 0)
891         {
892             ProyectoModel proyecto = new ProyectoModel();
893             proyecto.UsuarioCreacion = "User";
894             proyecto.UsuarioModificacion = "User";
895             proyecto.FechaCreacion = DateTime.Now;
896             proyecto.FechaModificacion = DateTime.Now;
897             proyecto.Estado = 1;
898             proyecto.IdAsignatura = idSil;
899             proyecto.IdProyecto = item.IdProyecto;
900             proyecto.NroOrden = item.NroOrden;
901             proyecto.IdFase = fase;
902             proyecto.DenominacionActividad = item.DenominacionActividad;
903             proyecto.Proposito = item.Proposito;
904             proyecto.IndicadoresEvaluacion = item.IndicadoresEvaluacion;
905             proyecto.Beneficiarios = item.Beneficiarios;
906             proyecto.Responsables = item.Responsables;
907             proyecto.Cronograma = item.Cronograma;
908             proyecto.PesoPractico = item.PesoPractico;
909             proyecto.PesoTeorico = item.PesoTeorico;
910             _context.Proyecto.Add(proyecto);
911         }
912         else
913         {
914             grabo = 'N';
915         }
916     }
917
918     //-- Grabar log con la acción de proyectos
919     if (grabo == 'S')
920     {
921         _context.SaveChanges();
922         String Concepto = "PROYECTOS";
923         String Descripcion = "PROYECTOS INGRESADOS";
924         GrabarAsignaturaLog(idSil, Concepto, Descripcion, "proyectos");
925     }
926     return grabo;
927 }

```

Fuente: Elaboración propia.

- Servicio “TranscribirAsignatura”

Servicio que permite transcribir el contenido de un sílabo desarrollado el año anterior al actual.

Se verifica el tipo de dato de los códigos identificadores de los sílabos participantes (línea 1750). En caso de superar la validación se llama al método “TranscribirSilabo” para realizar la transcripción (línea 1770).

Figura 116 Método "TranscribirAsignatura"

```

1737 //-- Servicio que realiza la transcripción de la asignatura de un periodo anterior al periodo actual.
1738 1 referencia
1739 public class TranscribirAsignaturaParametros
1740 {
1741     4 referencias
1742     public string idSilabo { get; set; }
1743     4 referencias
1744     public string idSilaboAnt { get; set; }
1745 }
1746
1747 [HttpPost]
1748 [Route("TranscribirAsignatura")]
1749 0 referencias
1750 public JsonResult TranscribirAsignatura(TranscribirAsignaturaParametros parametros)
1751 {
1752     //-- Verificar valor del parámetro sílabo, en caso de que existir un valor inadecuado se retorna mensaje de error.
1753     JsonResult json;
1754     if (parametros.idSilabo == null || parametros.idSilaboAnt == null)
1755     {
1756         var resultado = new { estado = "fail", mensaje = "Información errónea." };
1757         return json = new JsonResult(resultado);
1758     }
1759     if (parametros.idSilabo == "" || parametros.idSilaboAnt == "")
1760     {
1761         parametros.idSilabo = "0";
1762         parametros.idSilaboAnt = "0";
1763     }
1764     int idSil;
1765     int idSilAnt;
1766     if (!Int32.TryParse(parametros.idSilabo, out idSil) || !Int32.TryParse(parametros.idSilaboAnt, out idSilAnt))
1767     {
1768         var resultado = new { estado = "fail", mensaje = "Información errónea." };
1769         return json = new JsonResult(resultado);
1770     }else
1771     {
1772         //-- Realizar la transcripción.
1773         _handler = new GestionSilabos.Handler.Handler(_context);
1774         _handler.TranscribirSilabo(idSil, idSilAnt);
1775
1776         var resultado = new { estado = "success", mensaje = "Se completó correctamente la transcripción." };
1777         return json = new JsonResult(resultado);
1778     }
1779 }

```

Fuente: Elaboración propia.

El método obtiene el contenido desarrollado del sílabo anterior para los conceptos de sumilla, competencias propias de asignatura, competencias que apoyan al perfil de egreso, unidades teóricas y de laboratorio, resultados de aprendizajes, criterios, estrategias, instrumentos, bibliografía, proyectos (línea 935), posteriormente se elimina el contenido para el sílabo actual a desarrollar (línea 949). Para finalizar los resultados obtenidos, se insertan al sílabo actual (línea 963).

Figura 117 Método "TranscribirSilabo"

```

931 //-- Método para realizar la transcripción de un sílabo anterior al nuevo sílabo a desarrollar
932 //información
933 public void TranscribirSilabo(int idSil, int idSilAnt)
934 {
935     //-- Obtener contenido del sílabo anterior de los conceptos:
936     var sumillaAnt = _context.Sumilla.Where(s => s.IdSilabo == idSilAnt).FirstOrDefault();
937     var asicompPaAnt = _context.AsignaturaCompetenciaPE.Where(x => x.IdAsignatura == idSilAnt).ToList();
938     var asigCompPaAnt = _context.AsignaturaCompetenciaPA.Where(x => x.IdAsignatura == idSilAnt).ToList();
939     var asiFaseAnt = _context.AsignaturaFase.Where(x => x.IdAsignatura == idSilAnt).ToList();
940     var unidadAnt = _context.Unidad.Where(x => x.IdAsignatura == idSilAnt).ToList();
941     var unidadLabAnt = _context.UnidadLab.Where(x => x.IdAsignatura == idSilAnt).ToList();
942     var resultadosAnt = _context.Resultado.Where(x => x.IdAsignatura == idSilAnt).ToList();
943     var asiEstAnt = _context.AsignaturaEstrategia.Where(x => x.IdAsignatura == idSilAnt).ToList();
944     var asiCriAnt = _context.AsignaturaCriterio.Where(x => x.IdAsignatura == idSilAnt).ToList();
945     var asiInsAnt = _context.AsignaturaInstrumento.Where(x => x.IdAsignatura == idSilAnt).ToList();
946     var asiBibAnt = _context.AsignaturaBibliografia.Where(x => x.IdAsignatura == idSilAnt).ToList();
947     var ProyectoAnt = _context.Proyecto.Where(x => x.IdAsignatura == idSilAnt).ToList();
948
949     //-- Se elimina la información que se encuentre del sílabo a desarrollar en caso hubiera
950     _context.Database.ExecuteSqlRaw("DELETE FROM AsignaturaFase WHERE IdAsignatura = {0}", idSil.ToString());
951     _context.Database.ExecuteSqlRaw("DELETE FROM AsignaturaCompetenciaPE WHERE IdAsignatura = {0}", idSil.ToString());
952     _context.Database.ExecuteSqlRaw("DELETE FROM AsignaturaCompetenciaPA WHERE IdAsignatura = {0}", idSil.ToString());
953     _context.Database.ExecuteSqlRaw("DELETE FROM Unidad WHERE IdAsignatura = {0}", idSil.ToString());
954     _context.Database.ExecuteSqlRaw("DELETE FROM UnidadLab WHERE IdAsignatura = {0}", idSil.ToString());
955     _context.Database.ExecuteSqlRaw("DELETE FROM Resultado WHERE IdAsignatura = {0}", idSil.ToString());
956     _context.Database.ExecuteSqlRaw("DELETE FROM AsignaturaEstrategia WHERE IdAsignatura = {0}", idSil.ToString());
957     _context.Database.ExecuteSqlRaw("DELETE FROM AsignaturaCriterio WHERE IdAsignatura = {0}", idSil.ToString());
958     _context.Database.ExecuteSqlRaw("DELETE FROM AsignaturaInstrumento WHERE IdAsignatura = {0}", idSil.ToString());
959     _context.Database.ExecuteSqlRaw("DELETE FROM Proyecto WHERE IdAsignatura = {0}", idSil.ToString());
960     _context.Database.ExecuteSqlRaw("DELETE FROM AsignaturaBibliografia WHERE IdAsignatura = {0}", idSil.ToString());
961
962     //-- Se agrega el nuevo contenido del sílabo anterior a los conceptos del nuevo sílabo
963     //-- Sumilla
964     int? idSumilla = _context.Asignatura.Where(x => x.Id == idSil).Select(s => s.IdSumilla).FirstOrDefault();
965     if (idSumilla == 0 || idSumilla == null) { idSumilla = 0; }
966     GrabarSumillaSilabo(idSil, (int?)idSumilla, sumillaAnt.Descripcion);
967
968     //-- AsignaturaFase
969     List<FaseModel> fasesAnt = ObtenerFasesSemestre(idSilAnt);
970     List<FaseModel> fases = ObtenerFasesSemestre(idSil);
971     GrabarAsignaturaFaseTranscribir(idSil, fases, fasesAnt, asiFaseAnt);
972
973     //-- Competencias por Perfil de Egreso
974     GrabarCompetenciasPETranscribir(idSil, asicompPaAnt);
975
976     //-- Unidades teóricas
977     GrabarCompetenciasPATranscribir(idSil, fases, fasesAnt, asigCompPaAnt);
978
979     //-- Unidades teóricas
980     GrabarUnidadesTranscribir(idSil, fases, fasesAnt, unidadAnt);
981
982     //-- Unidades de laboratorio
983     GrabarUnidadesLabTranscribir(idSil, fases, fasesAnt, unidadLabAnt);
984
985     //-- Resultados
986     GrabarResultadosTranscribir(idSil, fases, fasesAnt, resultadosAnt);
987
988     //-- Estrategias
989     GrabarEstrategiasTranscribir(idSil, fases, fasesAnt, asiEstAnt);
990
991     //-- Criterios
992     GrabarCriteriosTranscribir(idSil, fases, fasesAnt, asiCriAnt);
993
994     //-- Instrumentos
995     GrabarInstrumentosTranscribir(idSil, fases, fasesAnt, asiInsAnt);
996
997     //-- Bibliografía
998     GrabarBibliografiaTranscribir(idSil, fases, fasesAnt, asiBibAnt);
999
1000     //-- Proyectos
1001     GrabarProyectosTranscribir(idSil, fases, fasesAnt, ProyectoAnt);
1002 }

```

Fuente: Elaboración propia.

2.4 Pruebas de Ejecución

La aplicación Sistema de Gestión de sílabo, se encuentra alojada en un ambiente para las pruebas de software orientadas a tratamiento de datos y emisión de información (Tabla 58).

Tabla 58 *Desarrollo de pruebas de aplicación*

PRUEBAS DE EJECUCIÓN DE APLICACIÓN	
Pruebas funcionales	Sí
Pruebas de usabilidad	Sí
Pruebas de accesibilidad	Sí
Pruebas de seguridad	Sí
Pruebas de Carga	Sí

Fuente: Elaboración propia.

2.4.1 Pruebas funcionales

Para las pruebas funcionales del proyecto enfocados al prototipo de Sistema de Gestión de Sílabos se utilizará la técnica de caja negra que consiste en el enfoque de la funcionalidad del componente del sistema. Esto significa, que no se cuenta con acceso al código fuente, si no se basa en la funcionalidad por los parámetros de entrada y salida intentando encontrar errores en la estructura de datos, inicialización y finalización de rendimiento, además de interfaces de entrada y salida.

Para ello se presentan los siguientes casos de prueba:

Tabla 59 Caso de Prueba CPGS01

CPGS01 – Inicio de desarrollo de Sílabo
<p>Descripción</p> <p>El usuario docente ingresa con su usuario y contraseña al sistema, y luego de ingresar se dirige a la opción de Cursos, para luego elegir la asignatura de la que desarrollará el sílabo</p>
<p>Prerrequisitos</p> <p>Usuario docente ingresa a aplicación Utilizar navegador web como Google Chrome</p>
<p>Pasos</p> <ol style="list-style-type: none"> 1) Ingresar a la dirección de la aplicación 2) En la interfaz de login ingresar credenciales 3) Seleccionar la pestaña de curso e ingresar a la interfaz de plan de estudio
<p>Resultado Esperado</p> <p>El cambio de estado del sílabo debe ser a abierto y la aplicación debe emitir un mensaje informativo sobre la apertura del sílabo.</p>
<p>Resultado Obtenido</p> <p>Caso de prueba de ejecutó correctamente obteniendo el resultado esperado</p>

Fuente: Elaboración propia.

Tabla 60 Caso de Prueba CPGS02

CPGS02 – Ingreso de Sumilla
<p>Descripción</p> <p>El usuario docente ingresa con su usuario y contraseña al sistema, y luego de ingresar se dirige a la opción de Cursos, para luego elegir la asignatura de la que desarrollará el sílabo. Posteriormente accede a la opción de Sumilla e ingresa el contenido</p>
<p>Prerrequisitos</p> <p>Usuario docente ingresa a aplicación Utilizar navegador web como Google Chrome</p>
<p>Pasos</p> <ol style="list-style-type: none"> 1) Ingresar a la dirección de la aplicación 2) En la interfaz de login ingresar credenciales 3) Seleccionar la pestaña de curso e ingresar a la interfaz de plan de estudio 4) Elige la opción sumilla e ingresa contenido 5) Seleccionar el botón guardar
<p>Resultado Esperado</p> <p>La sumilla es ingresada correctamente y se recibe mensaje de confirmación de la inserción</p>
<p>Resultado Obtenido</p> <p>Caso de prueba de ejecutó correctamente obteniendo el resultado esperado</p>

Fuente: Elaboración propia.

Tabla 61 *Caso de Prueba CPGS03*

CPGS03 – Selección de competencias que apoyan al perfil de egreso
<p>Descripción</p> <p>El usuario docente ingresa con su usuario y contraseña al sistema, y luego de ingresar se dirige a la opción de Cursos, para luego elegir la asignatura de la que desarrollará el sílabo. Posteriormente accede a la opción de Competencias de la asignatura que apoyan al perfil de egreso y selecciona competencias</p>
<p>Prerrequisitos</p> <p>Usuario docente ingresa a aplicación Utilizar navegador web como Google Chrome</p>
<p>Pasos</p> <ol style="list-style-type: none"> 1) Ingresar a la dirección de la aplicación 2) En la interfaz de login ingresar credenciales 3) Seleccionar la pestaña de curso e ingresar a la interfaz de plan de estudio 4) Elige la opción de competencias de la asignatura que apoyan al perfil de egreso 5) Elige las competencias que están asociadas a su asignatura 6) Seleccionar el botón guardar
<p>Resultado Esperado</p> <p>Las competencias son ingresadas correctamente y se recibe mensaje de confirmación de la inserción</p>
<p>Resultado Obtenido</p> <p>Caso de prueba de ejecutó correctamente obteniendo el resultado esperado</p>

Fuente: Elaboración propia.

Tabla 62 *Caso de Prueba CPGS04*

CPGS04 – Ingreso de Competencias propias de la asignatura
<p>Descripción</p> <p>El usuario docente ingresa con su usuario y contraseña al sistema, y luego de ingresar se dirige a la opción de Cursos, para luego elegir la asignatura de la que desarrollará el sílabo. Posteriormente accede a la opción de Competencias propias de la asignatura e ingresa contenido</p>
<p>Prerrequisitos</p> <p>Usuario docente ingresa a aplicación Utilizar navegador web como Google Chrome</p>
<p>Pasos</p> <ol style="list-style-type: none"> 1) Ingresar a la dirección de la aplicación 2) En la interfaz de login ingresar credenciales 3) Seleccionar la pestaña de curso e ingresar a la interfaz de plan de estudio 4) Elige la opción de competencias propias de la asignatura 5) Ingresa las competencias para cada fase de la asignatura 6) Seleccionar el botón guardar por fase
<p>Resultado Esperado</p> <p>Las competencias son ingresadas correctamente y se recibe mensaje de confirmación de la inserción</p>
<p>Resultado Obtenido</p> <p>Caso de prueba de ejecutó correctamente obteniendo el resultado esperado</p>

Fuente: Elaboración propia.

Tabla 63 *Caso de Prueba CPGS05*

CPGS05 – Reordenar competencias propias de asignatura y actualizar
<p>Descripción</p> <p>El usuario docente ingresa con su usuario y contraseña al sistema, y luego de ingresar se dirige a la opción de Cursos, para luego elegir la asignatura de la que desarrollará el sílabo. Posteriormente accede a la opción de Competencias propias de la asignatura y reordenar contenido</p>
<p>Prerrequisitos</p> <p>Usuario docente ingresa a aplicación Utilizar navegador web como Google Chrome</p>
<p>Pasos</p> <ol style="list-style-type: none"> 1) Ingresar a la dirección de la aplicación 2) En la interfaz de login ingresar credenciales 3) Seleccionar la pestaña de curso e ingresar a la interfaz de plan de estudio 4) Elige la opción de competencias propias de la asignatura 5) Utilizar la opción de reordenamiento de competencias por fase 6) Seleccionar el botón guardar por fase
<p>Resultado Esperado</p> <p>Las competencias son reordenadas a criterio del docente y son actualizadas correctamente y se recibe mensaje de confirmación de la actualización</p>
<p>Resultado Obtenido</p> <p>Caso de prueba de ejecutó correctamente obteniendo el resultado esperado</p>

Fuente: Elaboración propia.

Tabla 64 *Caso de Prueba CPGS06*

CPGS06 – Eliminación de competencia propia de asignatura
<p>Descripción</p> <p>El usuario docente ingresa con su usuario y contraseña al sistema, y luego de ingresar se dirige a la opción de Cursos, para luego elegir la asignatura de la que desarrollará el sílabo. Posteriormente accede a la opción de Competencias propias de la asignatura y elimina competencia</p>
<p>Prerrequisitos</p> <p>Usuario docente ingresa a aplicación Utilizar navegador web como Google Chrome</p>
<p>Pasos</p> <ol style="list-style-type: none"> 1) Ingresar a la dirección de la aplicación 2) En la interfaz de login ingresar credenciales 3) Seleccionar la pestaña de curso e ingresar a la interfaz de plan de estudio 4) Elige la opción de competencias propias de la asignatura 5) Eliminar competencia por fase 6) Seleccionar el botón guardar por fase
<p>Resultado Esperado</p> <p>La competencia propia de asignatura es eliminada de la fase y recibe mensaje de confirmación de la eliminación</p>
<p>Resultado Obtenido</p> <p>Caso de prueba de ejecutó correctamente obteniendo el resultado esperado</p>

Fuente: Elaboración propia.

Tabla 65 *Caso de Prueba CPGS07*

CPGS07 – Ingreso de Contenido por unidades de aprendizaje
Descripción
El usuario docente ingresa con su usuario y contraseña al sistema, y luego de ingresar se dirige a la opción de Cursos, para luego elegir la asignatura de la que desarrollará el sílabo. Posteriormente accede a la opción de Contenidos básicos por unidad de aprendizaje e ingresa contenido teórico y de laboratorio para cada fase
Prerrequisitos
Usuario docente ingresa a aplicación Utilizar navegador web como Google Chrome
Pasos
1) Ingresar a la dirección de la aplicación 2) En la interfaz de login ingresar credenciales 3) Seleccionar la pestaña de curso e ingresar a la interfaz de plan de estudio 4) Elige la opción de Contenidos básicos por unidades de aprendizaje 5) Ingresa contenido para unidades y subunidades teóricas y unidades de laboratorio por fase 6) Seleccionar el botón guardar por fase
Resultado Esperado
Unidades y subunidades teóricas son ingresadas correctamente, así como las unidades de laboratorio. Se recibe mensaje de confirmación de la inserción
Resultado Obtenido
Caso de prueba de ejecutó correctamente obteniendo el resultado esperado

Fuente: Elaboración propia.

Tabla 66 *Caso de Prueba CPGS08*

CPGS08 – Reordenar Contenido por unidades de aprendizaje
<p>Descripción</p> <p>El usuario docente ingresa con su usuario y contraseña al sistema, y luego de ingresar se dirige a la opción de Cursos, para luego elegir la asignatura de la que desarrollará el sílabo. Posteriormente accede a la opción de Contenidos básicos por unidad de aprendizaje y reordena contenido teórico y de laboratorio para cada fase</p>
<p>Prerrequisitos</p> <p>Usuario docente ingresa a aplicación Utilizar navegador web como Google Chrome</p>
<p>Pasos</p> <ol style="list-style-type: none"> 1) Ingresar a la dirección de la aplicación 2) En la interfaz de login ingresar credenciales 3) Seleccionar la pestaña de curso e ingresar a la interfaz de plan de estudio 4) Elige la opción de Contenidos básicos por unidades de aprendizaje 5) Utilizar la opción de reordenamiento de contenido para unidades y subunidades teóricas y unidades de laboratorio por fase 6) Seleccionar el botón guardar por fase
<p>Resultado Esperado</p> <p>Unidades y subunidades teóricas son ordenadas correctamente, así como las unidades de laboratorio. Se recibe mensaje de confirmación de la actualización</p>
<p>Resultado Obtenido</p> <p>Caso de prueba de ejecutó correctamente obteniendo el resultado esperado</p>

Fuente: Elaboración propia.

Tabla 67 *Caso de Prueba CPGS09*

CPGS09 – Eliminar Contenido por unidades de aprendizaje
<p>Descripción</p> <p>El usuario docente ingresa con su usuario y contraseña al sistema, y luego de ingresar se dirige a la opción de Cursos, para luego elegir la asignatura de la que desarrollará el sílabo. Posteriormente accede a la opción de Contenidos básicos por unidad de aprendizaje y elimina contenido teórico y de laboratorio para cada fase</p>
<p>Prerrequisitos</p> <p>Usuario docente ingresa a aplicación Utilizar navegador web como Google Chrome</p>
<p>Pasos</p> <ol style="list-style-type: none"> 1) Ingresar a la dirección de la aplicación 2) En la interfaz de login ingresar credenciales 3) Seleccionar la pestaña de curso e ingresar a la interfaz de plan de estudio 4) Elige la opción de Contenidos básicos por unidades de aprendizaje 5) Eliminar contenido para unidades y subunidades teóricas y unidades de laboratorio por fase 6) Seleccionar el botón guardar por fase
<p>Resultado Esperado</p> <p>Unidades y subunidades teóricas son eliminadas correctamente, así como las unidades de laboratorio. Se recibe mensaje de confirmación de la eliminación</p>
<p>Resultado Obtenido</p> <p>Caso de prueba de ejecutó correctamente obteniendo el resultado esperado</p>

Fuente: Elaboración propia.

Tabla 68 *Caso de Prueba CPGS10*

CPGS10 – Ingresar títulos por fase
<p>Descripción</p> <p>El usuario docente ingresa con su usuario y contraseña al sistema, y luego de ingresar se dirige a la opción de Cursos, para luego elegir la asignatura de la que desarrollará el sílabo. Posteriormente accede a la opción de Evaluación de competencias adquiridas e ingresa títulos para cada fase</p>
<p>Prerrequisitos</p> <p>Usuario docente ingresa a aplicación Utilizar navegador web como Google Chrome</p>
<p>Pasos</p> <ol style="list-style-type: none"> 1) Ingresar a la dirección de la aplicación 2) En la interfaz de login ingresar credenciales 3) Seleccionar la pestaña de curso e ingresar a la interfaz de plan de estudio 4) Elige la opción de Evaluación de competencias adquiridas 5) Ingresar títulos para cada fase 6) Seleccionar el botón guardar
<p>Resultado Esperado</p> <p>Los títulos de las fases de aprendizaje son ingresados correctamente. Se recibe mensaje de confirmación.</p>
<p>Resultado Obtenido</p> <p>Caso de prueba de ejecutó correctamente obteniendo el resultado esperado</p>

Fuente: Elaboración propia.

Tabla 69 *Caso de Prueba CPGS11*

CPGS11 – Ingresar Resultados de Fase	
Descripción	El usuario docente ingresa con su usuario y contraseña al sistema, y luego de ingresar se dirige a la opción de Cursos, para luego elegir la asignatura de la que desarrollará el sílabo. Posteriormente accede a la opción de Evaluación de competencias adquiridas e ingresa resultados de fase para resultados de conocimiento, de desempeño y de producto
Prerrequisitos	Usuario docente ingresa a aplicación Utilizar navegador web como Google Chrome
Pasos	<ol style="list-style-type: none"> 1) Ingresar a la dirección de la aplicación 2) En la interfaz de login ingresar credenciales 3) Seleccionar la pestaña de curso e ingresar a la interfaz de plan de estudio 4) Elige la opción de Evaluación de competencias adquiridas 5) Ingresar resultados de conocimiento, desempeño y producto 6) Seleccionar el botón guardar por fase
Resultado Esperado	Los resultados de conocimiento, desempeño y producto son ingresados correctamente. Se recibe mensaje de confirmación.
Resultado Obtenido	Caso de prueba de ejecutó correctamente obteniendo el resultado esperado

Fuente: Elaboración propia.

Tabla 70 *Caso de Prueba CPGS12*

CPGS12 – Reordenar Resultados de Fase
<p>Descripción</p> <p>El usuario docente ingresa con su usuario y contraseña al sistema, y luego de ingresar se dirige a la opción de Cursos, para luego elegir la asignatura de la que desarrollará el sílabo. Posteriormente accede a la opción de Evaluación de competencias adquiridas y reordenar los resultados de fase para resultados de conocimiento, de desempeño y de producto</p>
<p>Prerrequisitos</p> <p>Usuario docente ingresa a aplicación Utilizar navegador web como Google Chrome</p>
<p>Pasos</p> <ol style="list-style-type: none"> 1) Ingresar a la dirección de la aplicación 2) En la interfaz de login ingresar credenciales 3) Seleccionar la pestaña de curso e ingresar a la interfaz de plan de estudio 4) Elige la opción de Evaluación de competencias adquiridas 5) Utilizar la opción de reordenamiento de resultados de conocimiento, desempeño y producto 6) Seleccionar el botón guardar por fase
<p>Resultado Esperado</p> <p>Los resultados de conocimiento, desempeño y producto son reordenados correctamente. Se recibe mensaje de confirmación.</p>
<p>Resultado Obtenido</p> <p>Caso de prueba de ejecutó correctamente obteniendo el resultado esperado</p>

Fuente: Elaboración propia.

Tabla 71 *Caso de Prueba CPGS13*

CPGS13 – Eliminar Resultados de Fase
<p>Descripción</p> <p>El usuario docente ingresa con su usuario y contraseña al sistema, y luego de ingresar se dirige a la opción de Cursos, para luego elegir la asignatura de la que desarrollará el sílabo. Posteriormente accede a la opción de Evaluación de competencias adquiridas y eliminar resultados de fase para resultados de conocimiento, de desempeño y de producto</p>
<p>Prerrequisitos</p> <p>Usuario docente ingresa a aplicación Utilizar navegador web como Google Chrome</p>
<p>Pasos</p> <ol style="list-style-type: none"> 1) Ingresar a la dirección de la aplicación 2) En la interfaz de login ingresar credenciales 3) Seleccionar la pestaña de curso e ingresar a la interfaz de plan de estudio 4) Elige la opción de Evaluación de competencias adquiridas 5) Eliminar resultados de conocimiento, desempeño y producto 6) Seleccionar el botón guardar por fase
<p>Resultado Esperado</p> <p>Los resultados de conocimiento, desempeño y producto son eliminados correctamente. Se recibe mensaje de confirmación.</p>
<p>Resultado Obtenido</p> <p>Caso de prueba de ejecutó correctamente obteniendo el resultado esperado</p>

Fuente: Elaboración propia.

Tabla 72 *Caso de Prueba CPGS14*

CPGS14 – Ingresar Criterios, Instrumentos y Estrategias
<p>Descripción</p> <p>El usuario docente ingresa con su usuario y contraseña al sistema, y luego de ingresar se dirige a la opción de Cursos, para luego elegir la asignatura de la que desarrollará el sílabo. Posteriormente accede a la opción de Evaluación de competencias adquiridas e insertar criterios, instrumentos y estrategias por cada tipo de resultado (conocimiento, desempeño y producto)</p>
<p>Prerrequisitos</p> <p>Usuario docente ingresa a aplicación Utilizar navegador web como Google Chrome</p>
<p>Pasos</p> <ol style="list-style-type: none"> 1) Ingresar a la dirección de la aplicación 2) En la interfaz de login ingresar credenciales 3) Seleccionar la pestaña de curso e ingresar a la interfaz de plan de estudio 4) Elige la opción de Evaluación de competencias adquiridas 5) Ingresar criterios, instrumentos y estrategias por tipos de resultado 6) Seleccionar el botón guardar por fase
<p>Resultado Esperado</p> <p>Los criterios, instrumentos y estrategias son ingresados correctamente. Se recibe mensaje de confirmación.</p>
<p>Resultado Obtenido</p> <p>Caso de prueba de ejecutó correctamente obteniendo el resultado esperado</p>

Fuente: Elaboración propia.

Tabla 73 *Caso de Prueba CPGS15*

CPGS15 – Reordenar Criterios, Instrumentos y Estrategias
<p>Descripción</p> <p>El usuario docente ingresa con su usuario y contraseña al sistema, y luego de ingresar se dirige a la opción de Cursos, para luego elegir la asignatura de la que desarrollará el sílabo. Posteriormente accede a la opción de Evaluación de competencias adquiridas y reordenar criterios, instrumentos y estrategias por cada tipo de resultado (conocimiento, desempeño y producto)</p>
<p>Prerrequisitos</p> <p>Usuario docente ingresa a aplicación Utilizar navegador web como Google Chrome</p>
<p>Pasos</p> <ol style="list-style-type: none"> 1) Ingresar a la dirección de la aplicación 2) En la interfaz de login ingresar credenciales 3) Seleccionar la pestaña de curso e ingresar a la interfaz de plan de estudio 4) Elige la opción de Evaluación de competencias adquiridas 5) Utilizar la opción de reordenación para criterios, instrumentos y estrategias por tipos de resultado 6) Seleccionar el botón guardar por fase
<p>Resultado Esperado</p> <p>Los criterios, instrumentos y estrategias son reordenados correctamente. Se recibe mensaje de confirmación.</p>
<p>Resultado Obtenido</p> <p>Caso de prueba de ejecutó correctamente obteniendo el resultado esperado</p>

Fuente: Elaboración propia.

Tabla 74 *Caso de Prueba CPGS16*

CPGS16 – Eliminar Criterios, Instrumentos y Estrategias
<p>Descripción</p> <p>El usuario docente ingresa con su usuario y contraseña al sistema, y luego de ingresar se dirige a la opción de Cursos, para luego elegir la asignatura de la que desarrollará el sílabo. Posteriormente accede a la opción de Evaluación de competencias adquiridas y eliminar criterios, instrumentos y estrategias por cada tipo de resultado (conocimiento, desempeño y producto)</p>
<p>Prerrequisitos</p> <p>Usuario docente ingresa a aplicación Utilizar navegador web como Google Chrome</p>
<p>Pasos</p> <ol style="list-style-type: none"> 1) Ingresar a la dirección de la aplicación 2) En la interfaz de login ingresar credenciales 3) Seleccionar la pestaña de curso e ingresar a la interfaz de plan de estudio 4) Elige la opción de Evaluación de competencias adquiridas 5) Eliminar criterios, instrumentos y estrategias por tipos de resultado 6) Seleccionar el botón guardar por fase
<p>Resultado Esperado</p> <p>Los criterios, instrumentos y estrategias son eliminados correctamente. Se recibe mensaje de confirmación.</p>
<p>Resultado Obtenido</p> <p>Caso de prueba de ejecutó correctamente obteniendo el resultado esperado</p>

Fuente: Elaboración propia.

Tabla 75 *Caso de Prueba CPGS17*

CPGS17 – Ingresar Bibliografía
<p>Descripción</p> <p>El usuario docente ingresa con su usuario y contraseña al sistema, y luego de ingresar se dirige a la opción de Cursos, para luego elegir la asignatura de la que desarrollará el sílabo. Posteriormente accede a la opción de bibliografía para seleccionar la que corresponderá a cada fase</p>
<p>Prerrequisitos</p> <p>Usuario docente ingresa a aplicación Utilizar navegador web como Google Chrome</p>
<p>Pasos</p> <ol style="list-style-type: none"> 1) Ingresar a la dirección de la aplicación 2) En la interfaz de login ingresar credenciales 3) Seleccionar la pestaña de curso e ingresar a la interfaz de plan de estudio 4) Elige la opción de Bibliografía 5) Seleccionar Agregar Bibliografía 6) Utilizar filtros para buscar bibliografía 7) Seleccionar cada bibliografía para la fase 8) Seleccionar el botón guardar
<p>Resultado Esperado</p> <p>La bibliografía es ingresada correctamente. Se recibe mensaje de confirmación.</p>
<p>Resultado Obtenido</p> <p>Caso de prueba de ejecutó correctamente obteniendo el resultado esperado</p>

Fuente: Elaboración propia.

Tabla 76 *Caso de Prueba CPGS18*

CPGS18 – Reordenar Bibliografía
<p>Descripción</p> <p>El usuario docente ingresa con su usuario y contraseña al sistema, y luego de ingresar se dirige a la opción de Cursos, para luego elegir la asignatura de la que desarrollará el sílabo. Posteriormente accede a la opción de bibliografía para reordenar por fase</p>
<p>Prerrequisitos</p> <p>Usuario docente ingresa a aplicación Utilizar navegador web como Google Chrome</p>
<p>Pasos</p> <ol style="list-style-type: none"> 1) Ingresar a la dirección de la aplicación 2) En la interfaz de login ingresar credenciales 3) Seleccionar la pestaña de curso e ingresar a la interfaz de plan de estudio 4) Elige la opción de Bibliografía 5) Utilizar la opción de reordenamiento de bibliografía 6) Seleccionar el botón guardar
<p>Resultado Esperado</p> <p>La bibliografía es reordenada y actualizada correctamente. Se recibe mensaje de confirmación.</p>
<p>Resultado Obtenido</p> <p>Caso de prueba de ejecutó correctamente obteniendo el resultado esperado</p>

Fuente: Elaboración propia.

Tabla 77 *Caso de Prueba CPGS19*

CPGS19 – Eliminar Bibliografía
<p>Descripción</p> <p>El usuario docente ingresa con su usuario y contraseña al sistema, y luego de ingresar se dirige a la opción de Cursos, para luego elegir la asignatura de la que desarrollará el sílabo. Posteriormente accede a la opción de bibliografía para eliminar bibliografía por fase</p>
<p>Prerrequisitos</p> <p>Usuario docente ingresa a aplicación Utilizar navegador web como Google Chrome</p>
<p>Pasos</p> <ol style="list-style-type: none"> 1) Ingresar a la dirección de la aplicación 2) En la interfaz de login ingresar credenciales 3) Seleccionar la pestaña de curso e ingresar a la interfaz de plan de estudio 4) Elige la opción de Bibliografía 5) Eliminar bibliografía 6) Seleccionar el botón guardar
<p>Resultado Esperado</p> <p>La bibliografía es eliminada correctamente. Se recibe mensaje de confirmación.</p>
<p>Resultado Obtenido</p> <p>Caso de prueba de ejecutó correctamente obteniendo el resultado esperado</p>

Fuente: Elaboración propia.

Tabla 78 *Caso de Prueba CPGS20*

CPGS20 – Ingresar Proyectos
<p>Descripción</p> <p>El usuario docente ingresa con su usuario y contraseña al sistema, y luego de ingresar se dirige a la opción de Cursos, para luego elegir la asignatura de la que desarrollará el sílabo. Posteriormente accede a programa formativo, en la cual seleccionará la opción de matriz</p>
<p>Prerrequisitos</p> <p>Usuario docente ingresa a aplicación Utilizar navegador web como Google Chrome</p>
<p>Pasos</p> <ol style="list-style-type: none"> 1) Ingresar a la dirección de la aplicación 2) En la interfaz de login ingresar credenciales 3) Seleccionar la pestaña de curso e ingresar a la interfaz de programa formativo 4) Elige la opción de Matriz 5) Registrar proyecto por cada fase 6) Seleccionar el botón guardar
<p>Resultado Esperado</p> <p>La bibliografía es eliminada correctamente. Se recibe mensaje de confirmación.</p>
<p>Resultado Obtenido</p> <p>Caso de prueba de ejecutó correctamente obteniendo el resultado esperado</p>

Fuente: Elaboración propia.

Tabla 79 *Caso de Prueba CPGS21*

CPGS21 – Transcribir Sílabo
<p>Descripción</p> <p>El usuario docente ingresa con su usuario y contraseña al sistema, y luego de ingresar se dirige a la opción de Cursos, para luego elegir la asignatura de la que desarrollará el sílabo. Posteriormente accede a curso y realiza la transcripción</p>
<p>Prerrequisitos</p> <p>Usuario docente ingresa a aplicación Utilizar navegador web como Google Chrome</p>
<p>Pasos</p> <ol style="list-style-type: none"> 1) Ingresar a la dirección de la aplicación 2) En la interfaz de login ingresar credenciales 3) Seleccionar la pestaña de curso 4) Elige la opción Transcribir sílabo 5) Seleccionar la opción Reemplazar
<p>Resultado Esperado</p> <p>El contenido registrado en el sílabo anterior más reciente debe copiarse a la asignatura que se está desarrollando</p>
<p>Resultado Obtenido</p> <p>Caso de prueba de ejecutó correctamente obteniendo el resultado esperado</p>

Fuente: Elaboración propia.

Tabla 80 *Caso de Prueba CPGS22*

CPGS22 – Visualizar Sílabo
<p>Descripción</p> <p>El usuario docente ingresa con su usuario y contraseña al sistema, y luego de ingresar se dirige a la opción de Cursos, para luego elegir la asignatura de la que desarrollará el sílabo. Posteriormente accede a la opción de visualizar sílabo</p>
<p>Prerrequisitos</p> <p>Usuario docente ingresa a aplicación Utilizar navegador web como Google Chrome</p>
<p>Pasos</p> <ol style="list-style-type: none"> 1) Ingresar a la dirección de la aplicación 2) En la interfaz de login ingresar credenciales 3) Seleccionar la pestaña de curso 4) Elige la opción visualizar sílabo
<p>Resultado Esperado</p> <p>Observar que el contenido ingresado sea el correcto según el desarrollo realizado</p>
<p>Resultado Obtenido</p> <p>Caso de prueba de ejecutó correctamente obteniendo el resultado esperado</p>

Fuente: Elaboración propia.

Tabla 81 *Caso de Prueba CPGS23*

CPGS23 – Aprobar Sílabo
<p>Descripción</p> <p>El usuario docente ingresa con su usuario y contraseña al sistema, y luego de ingresar se dirige a la opción de Cursos, para luego elegir la asignatura de la que desarrollará el sílabo. Posteriormente terminado el desarrollo, realiza la aprobación del sílabo como culminación de este</p>
<p>Prerrequisitos</p> <p>Usuario docente ingresa a aplicación Utilizar navegador web como Google Chrome</p>
<p>Pasos</p> <ol style="list-style-type: none"> 1) Ingresar a la dirección de la aplicación 2) En la interfaz de login ingresar credenciales 3) Seleccionar la pestaña de curso 4) Realiza el desarrollo del sílabo 5) Seleccionar la opción Aprobar Sílabo
<p>Resultado Esperado</p> <p>El sílabo es aprobado y las opciones de edición de cada sección de desarrollo no pueden editarse</p>
<p>Resultado Obtenido</p> <p>Caso de prueba de ejecutó correctamente obteniendo el resultado esperado</p>

Fuente: Elaboración propia.

Tabla 82 *Caso de Prueba CPGS24*

CPGS24 – Editar Sílabo
<p>Descripción</p> <p>El usuario docente ingresa con su usuario y contraseña al sistema, y luego de ingresar se dirige a la opción de Cursos, para luego elegir la asignatura de la que desarrollará el sílabo. Se rehabilita la edición de desarrollo del sílabo.</p>
<p>Prerrequisitos</p> <p>Usuario docente ingresa a aplicación Utilizar navegador web como Google Chrome</p>
<p>Pasos</p> <ol style="list-style-type: none"> 1) Ingresar a la dirección de la aplicación 2) En la interfaz de login ingresar credenciales 3) Seleccionar la pestaña de curso 4) Seleccionar Editar Sílabo
<p>Resultado Esperado</p> <p>El sílabo regresa al estado de desarrollo y se habilitan los campos para poder realizar modificaciones</p>
<p>Resultado Obtenido</p> <p>Caso de prueba de ejecutó correctamente obteniendo el resultado esperado</p>

Fuente: Elaboración propia.

Tabla 83 *Caso de Prueba CPGS25*

CPGS25 – Buscar sílabo y visualizar
<p>Descripción</p> <p>El usuario docente ingresa con su usuario y contraseña al sistema, y luego de ingresar se dirige a la opción de Visualizar Sílabo, realizar la búsqueda del sílabo por medio de filtros y visualizar el sílabo en formato PDF</p>
<p>Prerrequisitos</p> <p>Usuario docente ingresa a aplicación Utilizar navegador web como Google Chrome</p>
<p>Pasos</p> <ol style="list-style-type: none"> 1) Ingresar a la dirección de la aplicación 2) En la interfaz de login ingresar credenciales 3) Seleccionar la pestaña de Visualizar Sílabo 4) Realizar búsqueda por filtros 5) Visualizar sílabo por enlace en ir
<p>Resultado Esperado</p> <p>Observar que el contenido ingresado sea el correcto según el desarrollo realizado</p>
<p>Resultado Obtenido</p> <p>Caso de prueba de ejecutó correctamente obteniendo el resultado esperado</p>

Fuente: Elaboración propia.

Tabla 84 *Caso de Prueba CPGS26*

CPGS26 – Buscar sílabo y Habilitar
<p>Descripción</p> <p>El usuario docente ingresa con su usuario y contraseña al sistema, y luego de ingresar se dirige a la opción de Habilitar Sílabo, realizar la búsqueda del sílabo por medio de filtros y se habilita sílabo eligiendo opción.</p>
<p>Prerrequisitos</p> <p>Usuario docente ingresa a aplicación Utilizar navegador web como Google Chrome</p>
<p>Pasos</p> <ol style="list-style-type: none"> 1) Ingresar a la dirección de la aplicación 2) En la interfaz de login ingresar credenciales 3) Seleccionar la pestaña de Habilitar Sílabo 4) Realizar búsqueda por filtros 4) Elige la opción en la columna Acciones 5) Seleccionar estado de sílabo
<p>Resultado Esperado</p> <p>Observar que el sílabo actualice a nuevo estado.</p>
<p>Resultado Obtenido</p> <p>Caso de prueba de ejecutó correctamente obteniendo el resultado esperado</p>

Fuente: Elaboración propia.

Conclusión: De los veintiséis casos de prueba realizados en la modalidad de caja negra, se puede concluir que no figuraron errores de funcionamiento de software. Tales resultados pueden argumentar que el Sistema de Gestión de Sílabo funciona correctamente en las opciones que presenta.

2.4.2 Pruebas de usabilidad

En las pruebas de usabilidad se conoce el grado de satisfacción del usuario frente al Sistema de Gestión de Sílabos. Para ello cuando se realizaron las pruebas junto a una encuesta mediante la herramienta Google Forms que consta doce preguntas. Los encuestados fueron 10 usuarios profesionales de educación superior. La encuesta se puede encontrar en el apéndice B junto a las respuestas de los encuestados.

A continuación, se procede a analizar las respuestas de las preguntas que fueron formuladas en la encuesta:

Pregunta 1: ¿Existe relación entre el Sistema Gestión de Sílabos y los colores e imágenes de la Universidad Católica de Santa María?

La tabla 85 y la figura 118 nos muestran los resultados obtenidos para la pregunta 1 de la encuesta, de la que se puede afirmar que el 60% de los encuestados ve una alta relación en los colores e imagen de la Universidad Católica de Santa María, 30% opina que tiene una considerable relación, 10% opina que tiene una mediana relación y 0% indica que hay baja o poca relación. Lo que demuestra que el usuario percibe una coherencia de presentación e imagen de la interfaz.

Tabla 85 Pruebas de usabilidad - Respuestas Pregunta 1

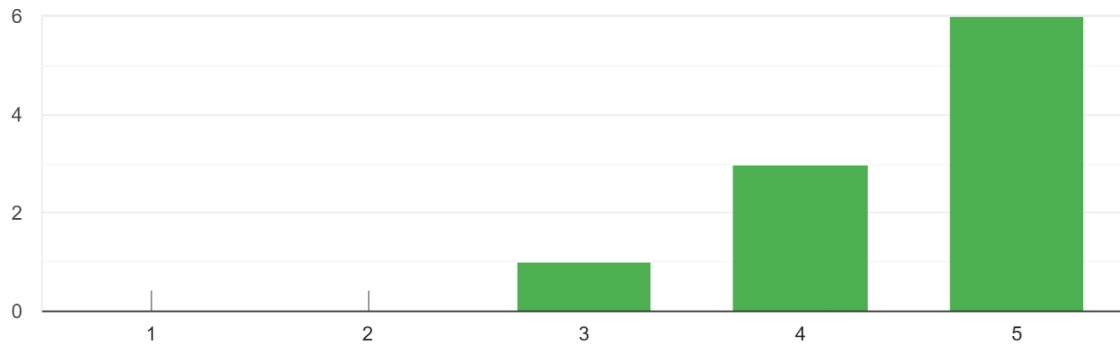
Opciones	Respuestas	
1 – Poca Relación	0	0%
2	0	0%
3	1	10%
4	3	30%
5 – Alta Relación	6	60%

Fuente: Elaboración propia.

Figura 118 Pruebas de usabilidad - Gráfico pregunta 1

¿Existe relación entre el Sistema Gestión de Sílabos y los colores e imágenes de la Universidad Católica de Santa María?

10 respuestas



Fuente: Elaboración propia con Google Forms.

Pregunta 2: ¿Cuál sería el público objetivo de la aplicación?

Como se observa en la tabla 87 y la figura 119, los encuestados indican en un 100% que el público objetivo de la aplicación son los docentes, un 80% que está orientado a los jefes de departamento, un 60% que es una herramienta útil para el centro de desarrollo académico y un 0% percibe a otros usuarios como finales.

Lo que indica que los docentes se perciben a sí mismos como los usuarios principales en la utilización de la aplicación.

Tabla 86 Pruebas de usabilidad - Respuestas Pregunta 2

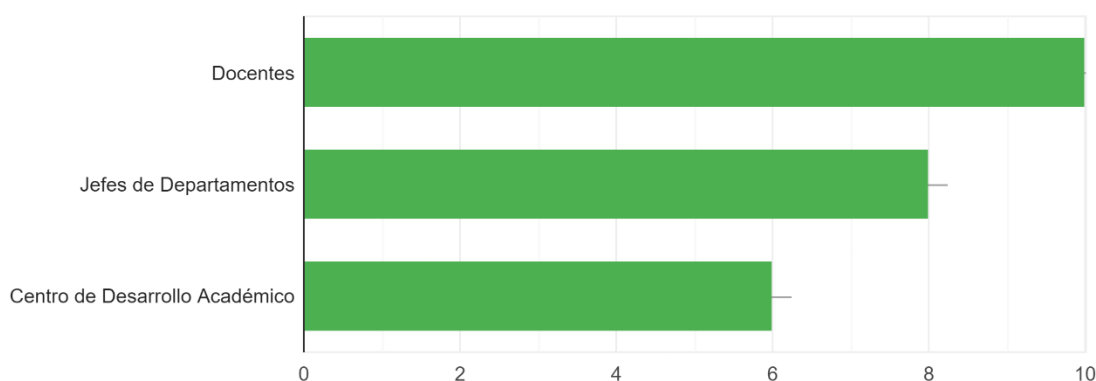
Opciones	Respuestas	
Docentes	10	100%
Jefes de Departamentos	8	80%
Centro de Desarrollo Académico	6	60%
Otra	0	0%

Fuente: Elaboración propia.

Figura 119 Pruebas de usabilidad - Gráfico pregunta 2

¿Cuál sería el público objetivo de la aplicación?

10 respuestas



Fuente: Elaboración propia con Google Forms.

Pregunta 3: ¿Qué apreciación tiene de la interfaz del Sistema Gestión de Sílabos?

En la tabla 88 y la figura 120, se aprecia que los encuestados indican en un 70% que la aplicación es amigable, 30% que la aplicación es muy amigable y 0% indica que es complejo. Esto demuestra que la aplicación es intuitiva y fácil de utilizar para los usuarios.

Tabla 87 Pruebas de usabilidad - Respuestas Pregunta 3

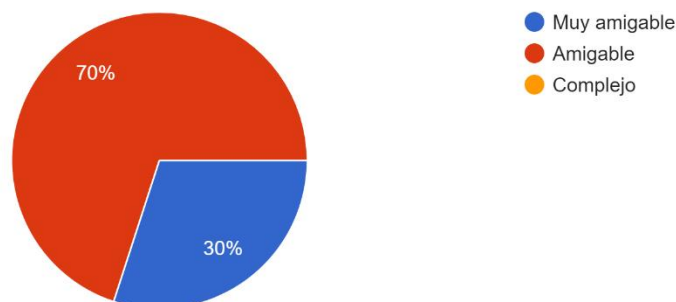
Opciones	Respuestas	
Muy amigable	3	30%
Amigable	7	70%
Complejo	0	0%

Fuente: Elaboración propia.

Figura 120 Pruebas de usabilidad - Gráfico pregunta 3

¿Qué apreciación tiene de la interfaz del Sistema Gestión de Silabos?

10 respuestas



Fuente: Elaboración propia con Google Forms.

Pregunta 4: ¿Requiere asesoría para entender el funcionamiento de la aplicación?

En la tabla 89 y figura 121, se observa que el 40% de los encuestados requirió mediana asesoría, 30% indicó que necesitó poca asesoría, 20% precisó buena asesoría y 10% requirió una baja asesoría. Esto reafirma en líneas generales que el aplicativo es entendible e intuitivo.

Tabla 88 Pruebas de usabilidad - Respuestas Pregunta 4

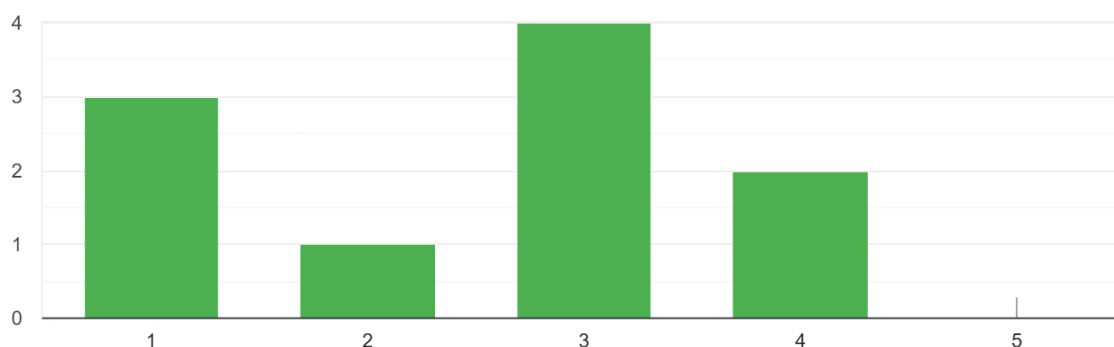
Opciones	Respuestas
1 – Poca asesoría	3 30%
2	1 10%
3	4 40%
4	2 20%
5 – Alta asesoría	0 0

Fuente: Elaboración propia.

Figura 121 Pruebas de usabilidad - Gráfico pregunta 4

¿Requiere asesoría para entender el funcionamiento de la aplicación?

10 respuestas



Fuente: Elaboración propia con Google Forms.

Pregunta 5: ¿Cuál considera el elemento de mayor impacto en la aplicación?

En la tabla 90 y en la figura 122, se aprecia que el 40% de los encuestados aprecia al plan de estudios como el elemento de mayor impacto, el 20% para la opción de Visualización de sílabo, 20% para la opción de Habilitación de sílabos y 20% para la sección de programa

formativo. Lo que indica que el enfoque mayoritario lo tiene el plan de estudios donde se desarrolla cada elemento del sílabo.

Tabla 89 Pruebas de usabilidad - Respuestas Pregunta 5

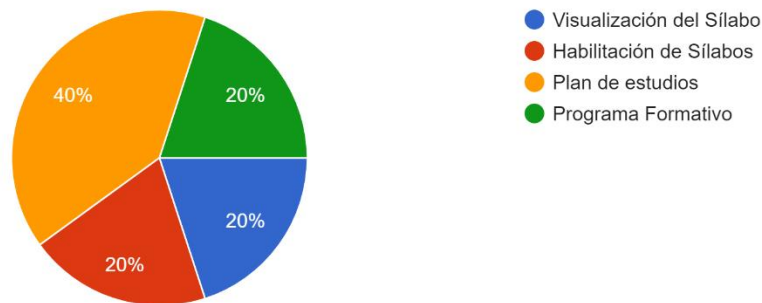
Opciones	Respuestas
Visualización del sílabo	2 20%
Habilitación de Sílabos	2 20%
Plan de estudios	4 40%
Programa Formativo	2 20%

Fuente: Elaboración propia.

Figura 122 Pruebas de usabilidad - Gráfico pregunta 5

¿Cuál considera el elemento de mayor impacto en la aplicación ?

10 respuestas



Fuente: Elaboración propia con Google Forms.

Pregunta 6: En el uso de la aplicación ¿Puede identificar estos elementos de navegación?

Se puede afirmar por la tabla 91 y la figura 123, que el elemento de navegación más perceptible son los filtros para búsqueda de sílabos en un 90%, seguido de la barra de navegación

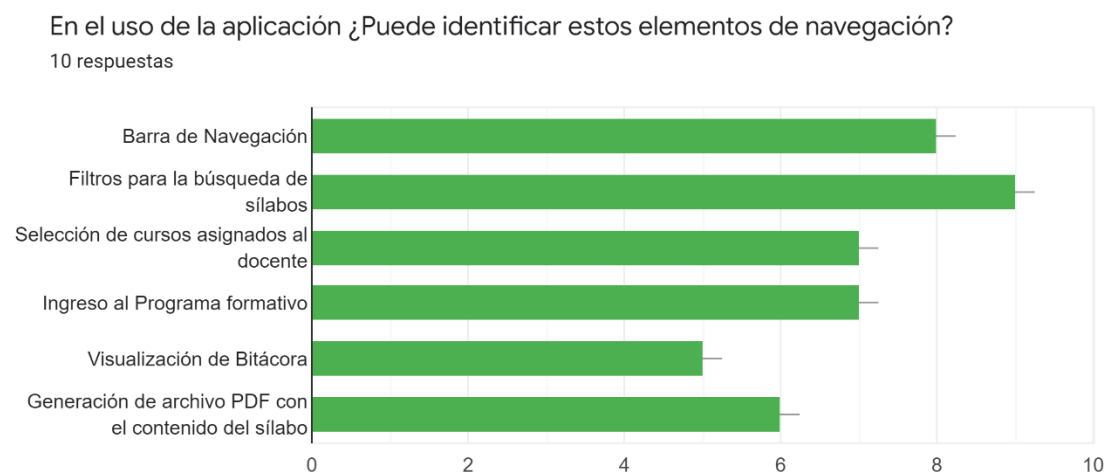
con 80%, los siguientes elementos son la selección de cursos asignados al docente y el ingreso al programa formativo con 70%, el 60% indica a la generación de archivos PDF como el siguiente elemento, el 50% la visualización de la bitácora y 0% otro elemento de navegación.

Tabla 90 Pruebas de usabilidad - Respuestas Pregunta 6

Opciones	Respuestas	
Barra de Navegación	8	80%
Filtros para la búsqueda de sílabos	9	90%
Selección de cursos asignados al docente	7	70%
Ingreso al Programa formativo	7	70%
Visualización de Bitácora	5	50%
Generación de archivo PDF con el contenido del sílabo	6	60%
Otra	0	0%

Fuente: Elaboración propia.

Figura 123 Pruebas de usabilidad - Gráfico pregunta 6



Fuente: Elaboración propia con Google Forms.

Pregunta 7: ¿Las herramientas le ayudan a desarrollar el sílabo correctamente?

Por medio de la tabla 92 y de la figura 124, se percibe que el 50% aprecia a la herramienta como útil para desarrollar el sílabo correctamente, 30% como medianamente útil, 20% como altamente útil y 0% como de baja y poca utilidad. Lo que indica que el sistema es considerado como una herramienta buena para el desarrollo de sílabos.

Tabla 91 Pruebas de usabilidad - Respuestas Pregunta 7

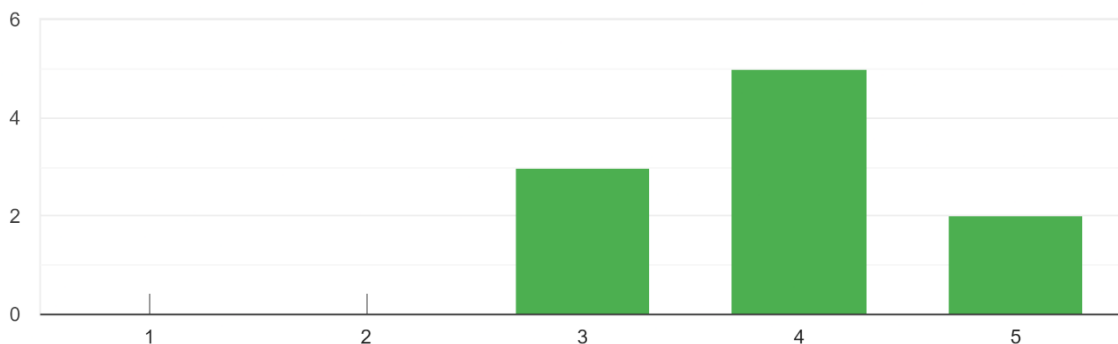
Opciones	Respuestas	Porcentaje
1 – Poca utilidad	0	0%
2	0	0%
3	3	30%
4	5	50%
5 – Alta utilidad	2	20%

Fuente: Elaboración propia.

Figura 124 Pruebas de usabilidad - Gráfico pregunta 7

¿Las herramientas le ayudan a desarrollar el sílabo correctamente?

10 respuestas



Fuente: Elaboración propia con Google Forms.

Pregunta 8: ¿La información encontrada en el Sistema de Gestión de Sílabos es de ayuda y utilidad?

Por medio de la tabla 93 y la figura 125, observamos que los encuestados opinan en un 30% que la información encontrada en el sistema es altamente útil, 30% indica que es útil, 30% menciona que es medianamente útil, 10% aprecia a la información como no tan útil y 0% cree que es nada útil. Esto demuestra que el sistema presenta información de ayuda al usuario.

Tabla 92 Pruebas de usabilidad - Respuestas Pregunta 8

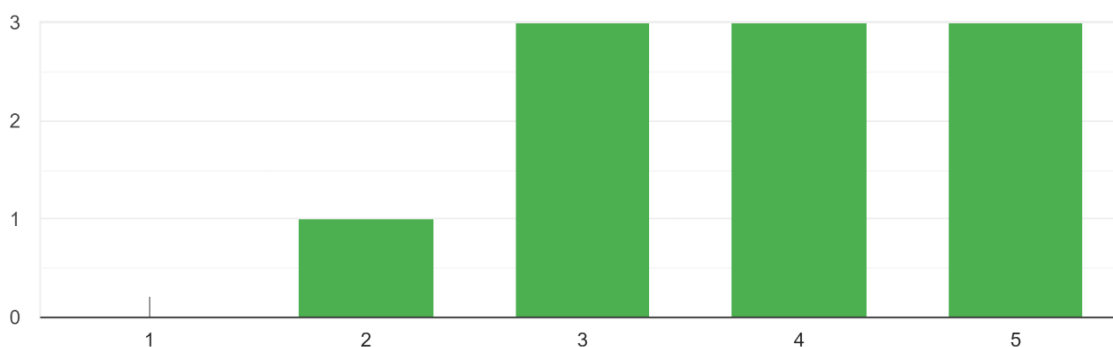
Opciones	Respuestas	Porcentaje
1 – Poca utilidad	0	0%
2	1	10%
3	3	30%
4	3	30%
5 – Alta utilidad	3	30%

Fuente: Elaboración propia.

Figura 125 Pruebas de usabilidad - Gráfico pregunta 8

¿La información encontrada en el Sistema de Gestión de Sílabos es de ayuda y utilidad?

10 respuestas



Fuente: Elaboración propia con Google Forms.

Pregunta 9: ¿Las secciones de desarrollo pueden ser ubicadas fácilmente?

A través de la tabla 94 y la Figura 126, el 50% de los encuestados indica que las secciones de desarrollo son relativamente rápidas de encontrar, 40% indica que su ubicación es rápida de buscar, el 10% cree que es algo dificultosa su búsqueda, y el 0% cree que es medianamente y muy dificultosa la búsqueda de las secciones de desarrollo. Esto demuestra que las secciones pueden ser ubicadas fácilmente por el usuario.

Tabla 93 Pruebas de usabilidad - Respuestas Pregunta 9

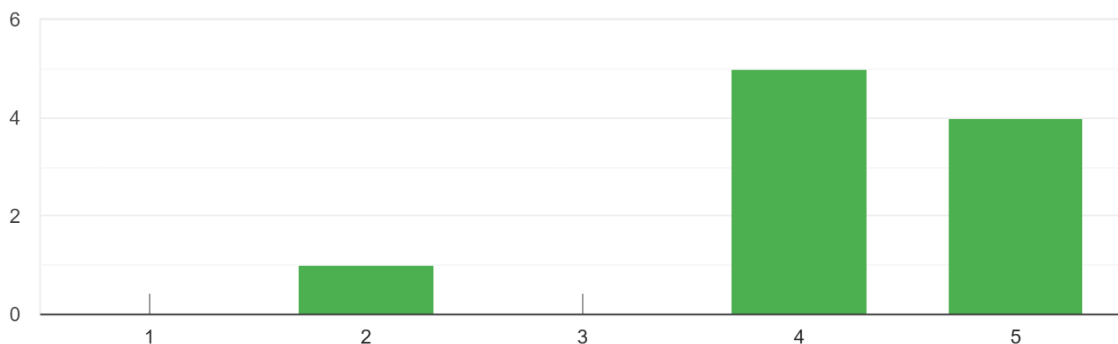
Opciones	Respuestas	Porcentaje
1 – Ubicación dificultosa	0	0%
2	1	10%
3	0	0%
4	5	50%
5 – Ubicación rápida	4	40%

Fuente: Elaboración propia.

Figura 126 Pruebas de usabilidad - Gráfico pregunta 9

¿Las secciones de desarrollo pueden ser ubicadas fácilmente?

10 respuestas



Fuente: Elaboración propia con Google Forms.

Pregunta 10: ¿El contenido mostrado y las opciones implementadas permiten que el desarrollo del sílabo sea ágil y mejor?

La tabla 95 y la figura 127 muestran que el 70% de los encuestados cree que las opciones implementadas permiten el desarrollo del sílabo de manera ágil y mejor de manera buena, 10% indica que es las opciones otorgan gran agilidad de desarrollo, el 10% opina que otorgan una agilidad relativamente buena, el 10% indica que es no muy ágil el desarrollo y el 0% ve que hay una mínima agilidad. Con esto se indica que el contenido y las opciones permiten un desarrollo ágil de los sílabos.

Tabla 94 Pruebas de usabilidad - Respuestas Pregunta 10

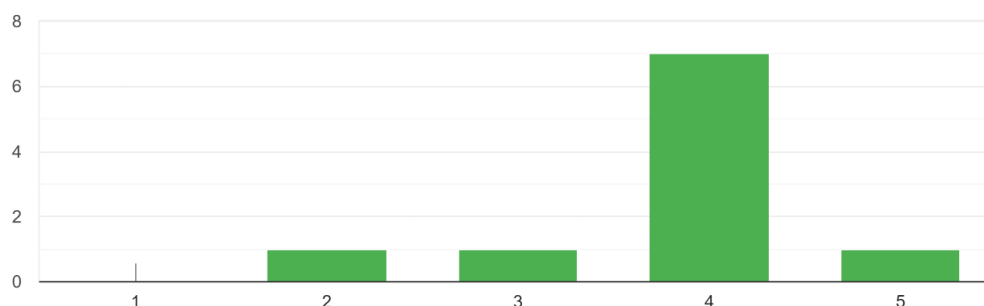
Opciones	Respuestas	
1 – Mínima agilidad de desarrollo	0	0%
2	1	10%
3	1	10%
4	7	70%
5 – Gran agilidad de desarrollo	1	10%

Fuente: Elaboración propia.

Figura 127 Pruebas de usabilidad - Gráfico pregunta 10

¿El contenido mostrado y las opciones implementadas permiten que el desarrollo del sílabo sea ágil y mejor?

10 respuestas



Fuente: Elaboración propia con Google Forms.

Pregunta 11: ¿Recomendaría el prototipo como una solución para el desarrollo de sílabos?

Se puede afirmar mediante la tabla 96 y la figura 128, que el 40% recomendaría el prototipo como una solución al desarrollo de sílabos, el 30% indicaría una alta recomendación, el 20% una mediana recomendación, el 10% poca recomendación y el 0% una baja recomendación. El prototipo es recomendable como solución de desarrollos de sílabos

Tabla 95 Pruebas de usabilidad - Respuestas Pregunta 11

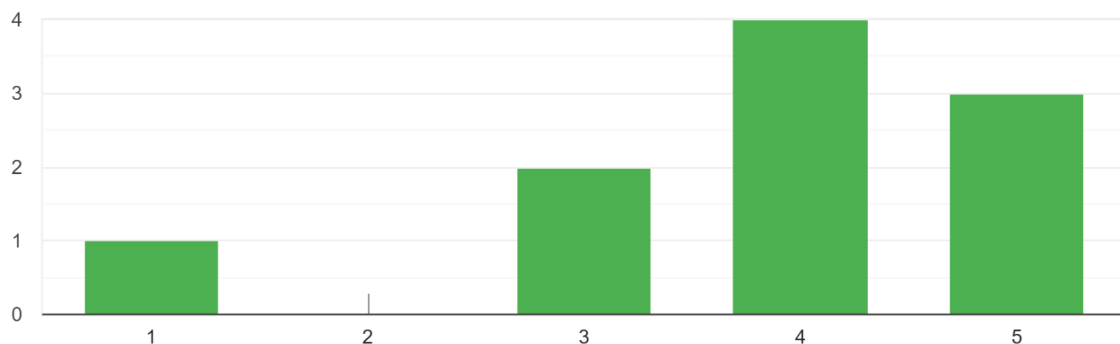
Opciones	Respuestas	Porcentaje
1 – Poca recomendación	1	10%
2	0	0%
3	2	20%
4	4	40%
5 – Alta recomendación	3	30%

Fuente: Elaboración propia.

Figura 128 Pruebas de usabilidad - Gráfico pregunta 11

¿Recomendaría el prototipo como una solución para el desarrollo de sílabos?

10 respuestas



Fuente: Elaboración propia con Google Forms.

Pregunta 12: ¿Cómo valora su experiencia utilizando la aplicación de Gestión de Sílabos?

Se aprecia a través de la tabla 97 y la figura 129 que el 60% de los encuestados indican que tuvieron una buena experiencia con la aplicación, el 20% una experiencia relativamente buena, el 10% una experiencia satisfactoria y el 10% una experiencia no grata. Esto indica que la utilización del sistema es altamente valorada.

Tabla 96 Pruebas de usabilidad - Respuestas Pregunta 12

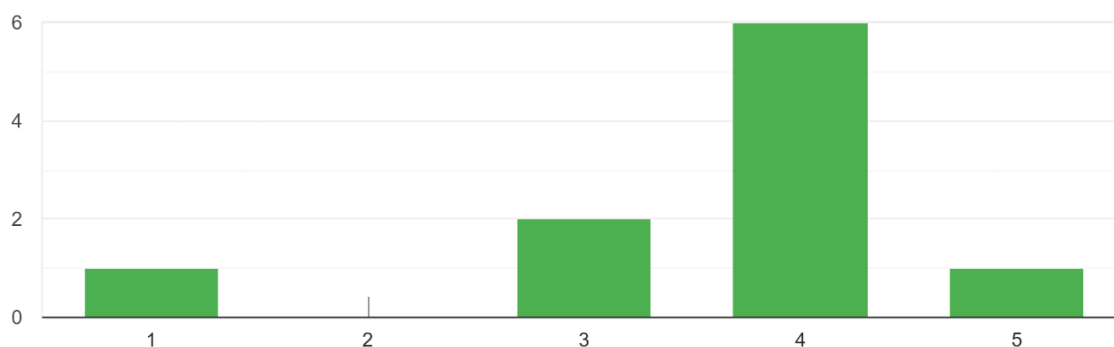
Opciones	Respuestas	Porcentaje
1 – Nada satisfecho	1	10%
2	0	0%
3	2	20%
4	6	60%
5 – Muy Satisfecho	1	10%

Fuente: Elaboración propia.

Figura 129 Pruebas de usabilidad - Gráfico pregunta 12

¿Cómo valora su experiencia utilizando la aplicación de Gestión de Sílabos?

10 respuestas



Fuente: Elaboración propia con Google Forms.

Los resultados que se pueden apreciar de la encuesta realizada indican las siguientes conclusiones:

- **Identidad Institucional:** La mayoría de los usuarios consideran que hay una relación alta entre los colores e imagen institucional de la Universidad Católica de Santa María y el Sistema Gestión de Sílabos, además que perciben correctamente los usuarios finales a los que apunta la aplicación.
- **Contenido de la aplicación:** Los encuestados en su totalidad consideran que la aplicación es amigable y requieren de mediana o poca asesoría para utilizar la aplicación. También identifican correctamente a la sección principal del sistema donde se realiza el desarrollo del sílabo
- **Navegación:** Los encuestados identifican correctamente los elementos de navegación, así como aprecian que es una herramienta que puede ayudarles a desarrollar el sílabo correctamente.
- **Búsqueda de la aplicación:** La mayoría de los encuestados encuentra a la información como útil y de ayuda, además que las opciones son fácilmente ubicadas.
- **Utilidad de la aplicación:** Los encuestados indican en su mayoría que las opciones y contenido permite el que el desarrollo sea ágil. También que sería un sistema recomendable para utilizar y tuvieron una experiencia buena al momento de utilizar la aplicación.

2.4.3 Pruebas de accesibilidad

Las pruebas de accesibilidad determinan la facilidad de acceso al contenido de las aplicaciones para personas con discapacidades. Las pautas de la WCAG (Content Accessibility Guidelines) indican sobre la evaluación del código fuente de la interfaz web para estar alineado a este estándar internacional.

Para esta prueba se utilizó la herramienta WAN que examina las propiedades de WCAG:

Perceptible: Capacidad de captar la información de la aplicación

Operable: Capacidad de interactuar con la aplicación

Robusto: Capacidad de ingresar a la aplicación indiferentemente de los avances futuros

Comprensible: Capacidad de entender la información de la aplicación

Figura 130 Análisis perceptible del sistema usando la herramienta TAW

Perceptible

La información y los componentes de la interfaz de usuario deben ser presentados a los usuarios de modo que puedan percibirlos.

Pauta	Nivel	Resultado	Problemas	Advertencias	No verificados
1.1-Textos alternativos			0	0	0
1.1.1 - Contenido no textual ?	A	na			
1.2-Medios basados en el tiempo			0	0	0
1.2.1 - Sólo audio y solo video (grabaciones) ?	A	na			
1.2.2 - Subtítulos (pregrabados) ?	A	na			
1.2.3 - Audiodescripción o Medio Alternativo (Pregrabado) ?	A	na			
1.2.4 - Subtítulos (en directo) ?	AA	na			
1.2.5 - Descripción auditiva (Pregrabada) ?	AA	na			
1.2.6 - Lenguaje de signos ?	AAA	na			
1.2.7 - Audiodescripción Extendida (Pregrabada) ?	AAA	na			
1.2.8 - Alternativa textual completa ?	AAA	na			
1.2.9 - Sólo audio (en directo) ?	AAA	na			
1.3-Adaptable			0	1	1
1.3.1 - Información y relaciones ?	A	!		1	
1.3.2 - Secuencia con significado ?	A	na			
1.3.3 - Características sensoriales ?	A	?			1
1.4-Distinguible			0	0	1
1.4.1 - Uso del color ?	A	?			1
1.4.2 - Control del audio ?	A	na			
1.4.3 - Contraste (Mínimo) ?	A	?			1
1.4.4 - Redimensionamiento del texto ?	AA	na			
1.4.5 - Imágenes de texto ?	AA	?			1
1.4.6 - Contraste (Mejorado) ?	AAA	?			1
1.4.7 - Sonido de fondo bajo o ausente ?	AAA	na			
1.4.8 - Presentación visual ?	AAA	?			1
1.4.9 - Imágenes de texto (sin excepciones) ?	AAA	?			1

Fuente: Elaboración propia con herramienta TAW.

Figura 131 Análisis operable del sistema usando la herramienta TAW

Operable

Los componentes de la interfaz de usuario y la navegación deben ser operables.

Pauta	Nivel	Resultado	Problemas	Advertencias	No verificados
2.1- Accesible mediante el teclado			0	0	1
2.1.1 - Teclado ?	A	?			1
2.1.2 - Sin bloqueos de teclado ?	A	?			1
2.1.3 - Teclado (sin excepciones) ?	AAA	na			
2.2- Tiempo suficiente			0	0	1
2.2.1 - Tiempo ajustable ?	A	?			1
2.2.2 - Pausar, detener, ocultar ?	A	?			1
2.2.3 - Sin límite temporal ?	AAA	?			1
2.2.4 - Interrupciones ?	AAA	na			
2.2.5 - Nueva autenticación ?	AAA	?			1
2.3- Provocar ataques			0	0	1
2.3.1 - Umbral de tres destellos o menos ?	A	?			1
2.3.2 - Tres destellos ?	AAA	?			1
2.4- Navegable			0	2	5
2.4.1 - Evitar bloques ?	A	?			1
2.4.2 - Páginas tituladas ?	A	!		1	
2.4.3 - Orden del foco ?	A	?			1
2.4.4 - Propósito de los enlaces (en contexto) ?	A	na			
2.4.5 - Múltiples vías ?	AA	?			1
2.4.6 - Encabezados y etiquetas ?	AA	!		1	
2.4.7 - Foco visible ?	AA	?			1
2.4.8 - Ubicación ?	AAA	?			1
2.4.9 - Propósito de los enlaces (Sólo enlaces) ?	AAA	na			
2.4.10 - Encabezados de sección ?	AAA	✓			

Fuente: Elaboración propia con herramienta TAW.

Figura 132 Análisis de la comprensibilidad del sistema usando la herramienta TAW

Comprensible

La información y el manejo de la interfaz de usuario debe ser comprensible.



Pauta	Nivel	Resultado	Problemas	Advertencias	No verificados
3.1-Legible			1	0	6
3.1.1 - Idioma de la página ?	A	✗	1		1
3.1.2 - Idioma de las partes ?	AA	?			1
3.1.3 - Palabras inusuales ?	AAA	?			1
3.1.4 - Abreviaturas ?	AAA	?			1
3.1.5 - Nivel de lectura ?	AAA	?			1
3.1.6 - Pronunciación ?	AAA	?			1
3.2-Predecible			0	0	1
3.2.1 - Al recibir el foco ?	A	?			1
3.2.2 - Al introducir datos ?	A	?			1
3.2.3 - Navegación consistente ?	AA	?			1
3.2.4 - Identificación consistente ?	AA	?			1
3.2.5 - Cambios bajo petición ?	AAA	na			
3.3-Introducción de datos asistida			0	0	0
3.3.1 - Identificación de errores ?	A	na			
3.3.2 - Etiquetas o instrucciones ?	A	na			
3.3.3 - Sugerencias ante errores ?	AA	na			
3.3.4 - Prevención de errores (legales, financieros, datos) ?	AA	na			
3.3.5 - Ayuda ?	AAA	na			
3.3.6 - Prevención de errores (todos) ?	AAA	na			





Fuente: Elaboración propia con herramienta TAW.

Figura 133 Análisis de la robustez del sistema usando la herramienta TAW

Robusto

El contenido debe ser suficientemente robusto como para ser interpretado de forma fiable por una amplia variedad de agentes de usuario, incluyendo las ayudas técnicas.

Pauta	Nivel	Resultado	Problemas	Advertencias	No verificados
4.1-Compatible			0	0	1
4.1.1 - Procesamiento 	A	✓			
4.1.2 - Nombre, función, valor 	A	?			1

 No se han encontrado problemas
  Existen problemas
  Requiere revisión manual
  Imposible realizar comprobación automática
 n/a no aplicable

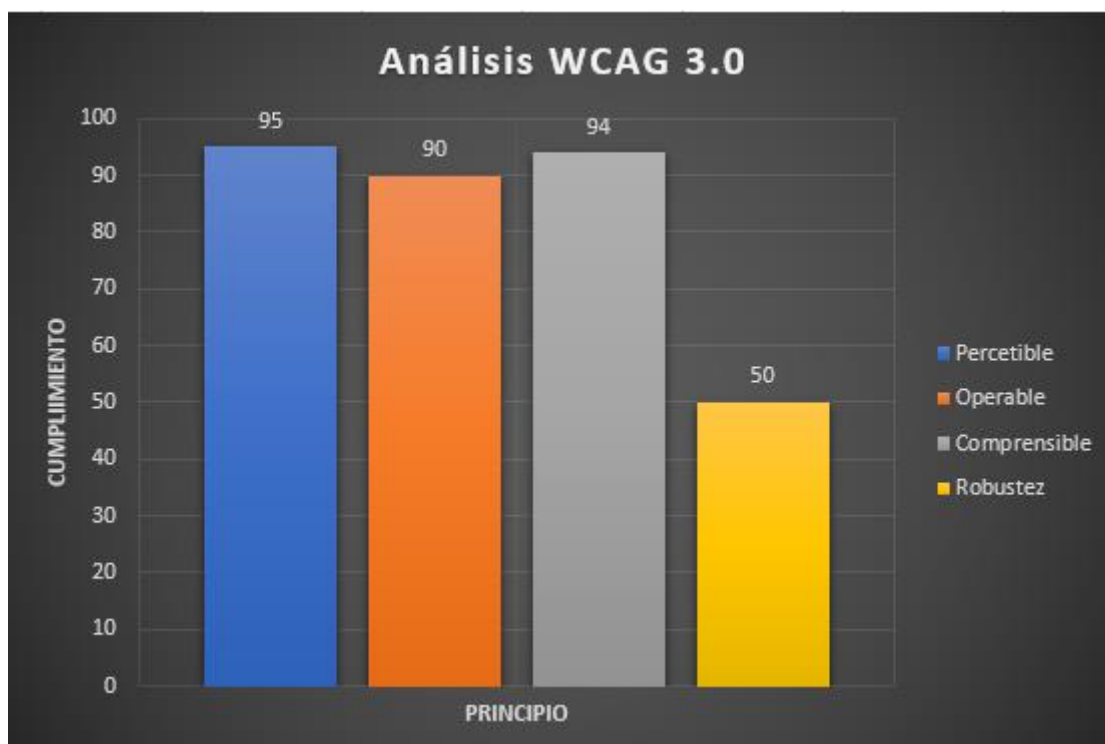
Fuente: Elaboración propia con herramienta TAW.

Conclusión:

Después de realizar las pruebas de accesibilidad y obtener los resultados para el sistema de Gestión de Sílabos, se puede decir que se cumple con las recomendaciones de la guía WCAG 3.0.

En la Figura 134 para los resultados calculados en las mediciones de cumplimiento de accesibilidad, se muestra que la aplicación presenta una alta aceptación del 95% en el principio “Perceptible”, en el principio “Comprensible” de un 94%, un alto cumplimiento de un 90% del principio “Operable” y un mediano cumplimiento del 50% en el principio “Robusto”. Podemos concluir que el sistema de Gestión de Sílabos es accesible según la herramienta TAW.

Figura 134 Gráfico de análisis de resultados



Fuente: Elaboración propia.



2.4.4 Pruebas de seguridad

El acceso a la aplicación está condicionado al correo institucional que se utiliza en la Universidad Católica de Santa María.

Figura 135 Prueba de Seguridad - Ingreso

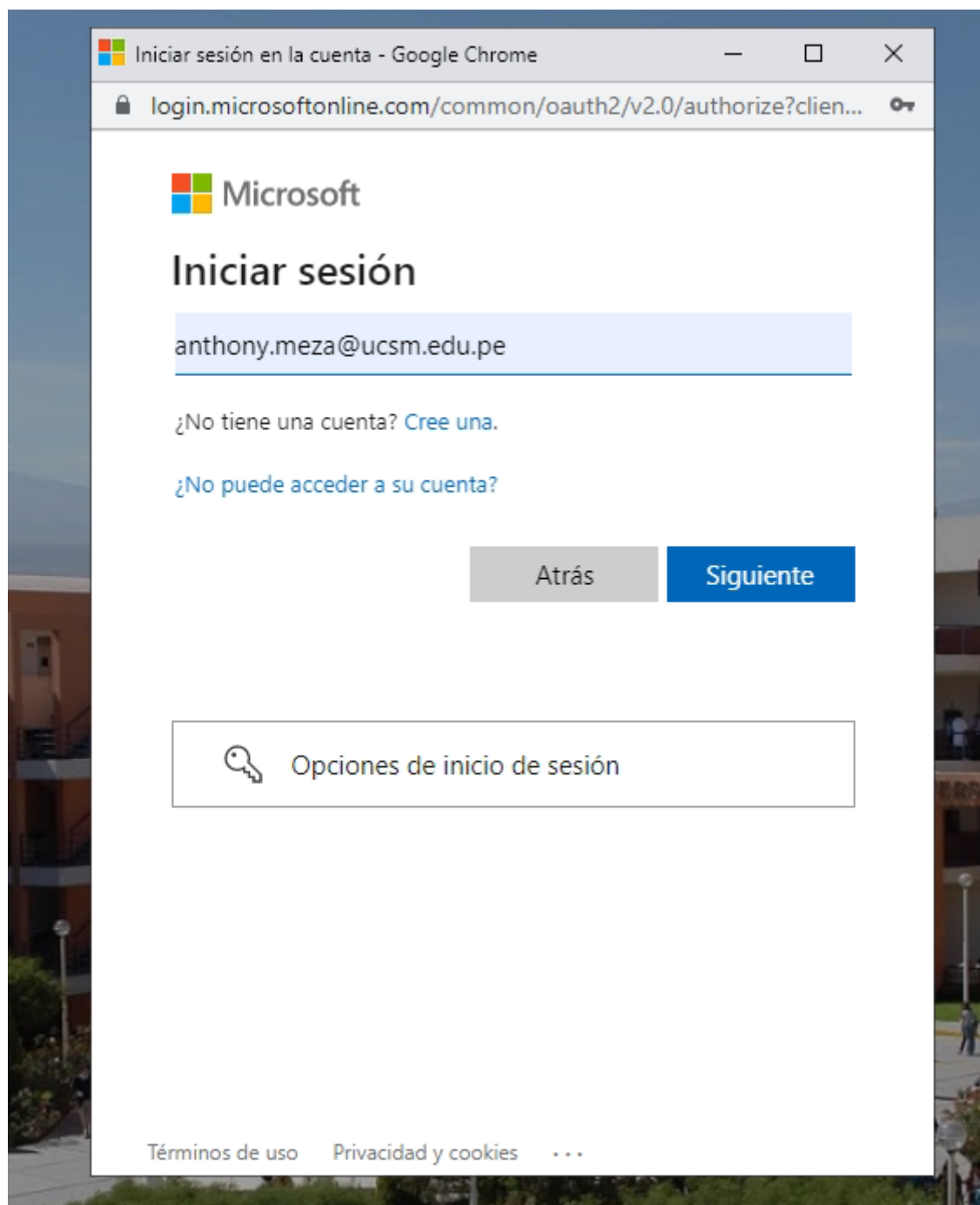


Fuente: Elaboración propia.



Al colocar el correo institucional, ingreso es permitido, debido a que está asociado al dominio de la institución como se muestra en la Figura 136.

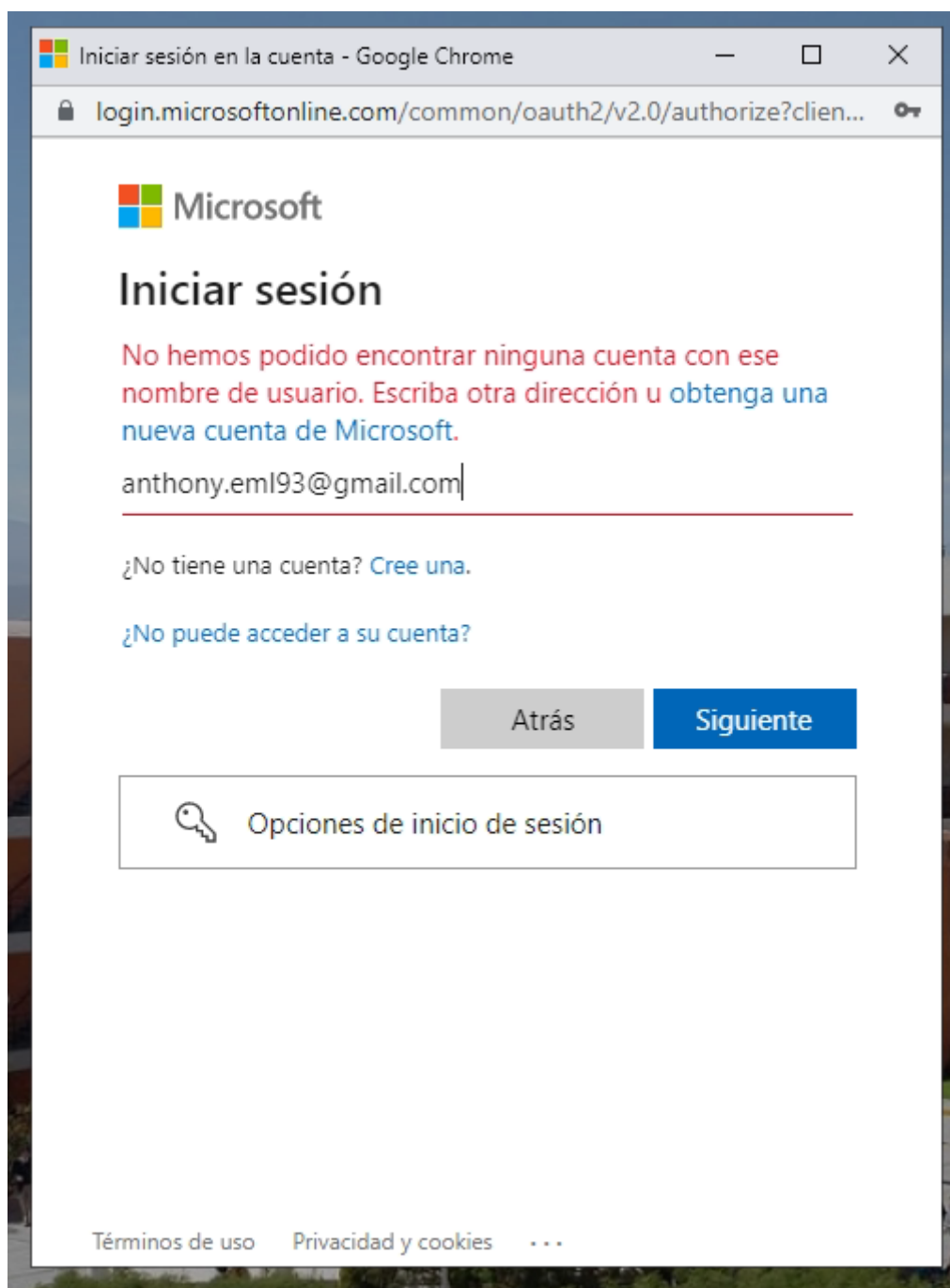
Figura 136 Prueba de Seguridad - Ingreso de correo institucional



Fuente: Elaboración propia. API Office 365.

Pero al intentar utilizar un correo que no se encuentre asociado al dominio de la institución, se emitirá el mensaje de error mostrado en la Figura 137.

Figura 137 Prueba de Seguridad - Error en acceso



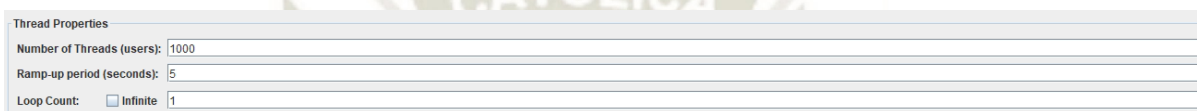
Fuente: Elaboración propia API Office 365.

Se concluye que las pruebas de seguridad para el acceso al aplicativo se refuerzan por medio de la utilización del API Office 365.

2.4.5 Pruebas de Carga

Para la prueba de carga se utilizará la herramienta Apache JMeter, simulando la interacción de los usuarios con la aplicación. Se hará el enfoque a las funcionalidades más concurridas para poder sacar las conclusiones respecto al rendimiento del sistema.

Figura 138 *Propiedades de Hilo en herramienta JMeter*



Thread Properties	
Number of Threads (users):	1000
Ramp-up period (seconds):	5
Loop Count: <input type="checkbox"/> Infinite	1

Fuente: Elaboración propia con herramienta JMeter.

Las funcionalidades que pasarán esta prueba de estrés serán: Visualizar Sílabos, Ver Identificación Académica, Listar cursos de docentes y Ver Programa Formativo.

Visualizar Sílabo

La Figura 139, evidencia el resultado de las mil muestras que simulan a los usuarios interactuando con la aplicación y utilizando la funcionalidad de Visualizar Sílabo, donde la respuesta que se obtiene es el tiempo de inicio, el nombre del hilo, el tiempo de demora, el estado y la latencia de cada hilo es mostrada.

Figura 139 Resultados de prueba de estrés para Visualizar Sílabo

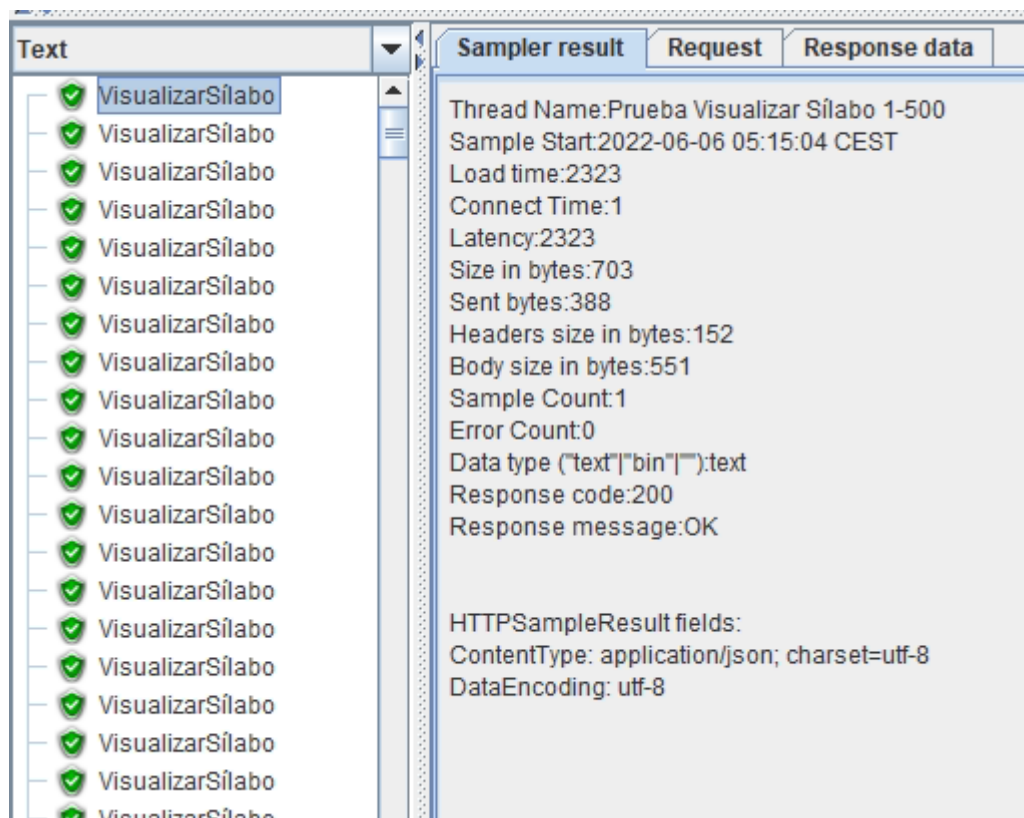
Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
1	05:15:02.239	Prueba Visualizar Sílabo 1-1	VisualizarSílabo	2071	✔	703	388	2071	1
2	05:15:02.243	Prueba Visualizar Sílabo 1-2	VisualizarSílabo	2102	✔	703	388	2102	0
3	05:15:02.248	Prueba Visualizar Sílabo 1-3	VisualizarSílabo	2106	✔	703	388	2106	0
4	05:15:02.254	Prueba Visualizar Sílabo 1-4	VisualizarSílabo	2116	✔	703	388	2116	0
5	05:15:02.258	Prueba Visualizar Sílabo 1-5	VisualizarSílabo	2113	✔	703	388	2113	0
6	05:15:02.262	Prueba Visualizar Sílabo 1-6	VisualizarSílabo	2110	✔	703	388	2110	1
7	05:15:02.267	Prueba Visualizar Sílabo 1-7	VisualizarSílabo	2106	✔	703	388	2106	1
8	05:15:02.272	Prueba Visualizar Sílabo 1-8	VisualizarSílabo	2138	✔	703	388	2138	1
9	05:15:02.278	Prueba Visualizar Sílabo 1-9	VisualizarSílabo	2133	✔	703	388	2133	0
10	05:15:02.284	Prueba Visualizar Sílabo 1-10	VisualizarSílabo	2128	✔	703	388	2128	0
11	05:15:02.287	Prueba Visualizar Sílabo 1-11	VisualizarSílabo	2126	✔	703	388	2126	0
12	05:15:02.293	Prueba Visualizar Sílabo 1-12	VisualizarSílabo	2121	✔	703	388	2121	0
13	05:15:02.297	Prueba Visualizar Sílabo 1-13	VisualizarSílabo	2117	✔	703	388	2117	1
14	05:15:02.302	Prueba Visualizar Sílabo 1-14	VisualizarSílabo	2142	✔	703	388	2142	0
15	05:15:02.306	Prueba Visualizar Sílabo 1-15	VisualizarSílabo	2139	✔	703	388	2139	1
16	05:15:02.313	Prueba Visualizar Sílabo 1-16	VisualizarSílabo	2133	✔	703	388	2133	1
17	05:15:02.317	Prueba Visualizar Sílabo 1-17	VisualizarSílabo	2131	✔	703	388	2131	0
18	05:15:02.322	Prueba Visualizar Sílabo 1-18	VisualizarSílabo	2127	✔	703	388	2127	0
19	05:15:02.329	Prueba Visualizar Sílabo 1-19	VisualizarSílabo	2121	✔	703	388	2121	0
20	05:15:02.333	Prueba Visualizar Sílabo 1-20	VisualizarSílabo	2139	✔	703	388	2139	0
21	05:15:02.338	Prueba Visualizar Sílabo 1-21	VisualizarSílabo	2138	✔	703	388	2138	0
22	05:15:02.342	Prueba Visualizar Sílabo 1-22	VisualizarSílabo	2220	✔	703	388	2220	1
23	05:15:02.347	Prueba Visualizar Sílabo 1-23	VisualizarSílabo	2243	✔	703	388	2243	1
24	05:15:02.352	Prueba Visualizar Sílabo 1-24	VisualizarSílabo	2240	✔	703	388	2240	1
25	05:15:02.360	Prueba Visualizar Sílabo 1-25	VisualizarSílabo	2233	✔	703	388	2233	1
26	05:15:02.363	Prueba Visualizar Sílabo 1-26	VisualizarSílabo	2236	✔	703	388	2236	0
27	05:15:02.368	Prueba Visualizar Sílabo 1-27	VisualizarSílabo	2232	✔	703	388	2232	1
28	05:15:02.373	Prueba Visualizar Sílabo 1-28	VisualizarSílabo	2228	✔	703	388	2228	0
29	05:15:02.377	Prueba Visualizar Sílabo 1-29	VisualizarSílabo	2228	✔	703	388	2228	1
30	05:15:02.382	Prueba Visualizar Sílabo 1-30	VisualizarSílabo	2227	✔	703	388	2227	1
31	05:15:02.387	Prueba Visualizar Sílabo 1-31	VisualizarSílabo	2223	✔	703	388	2223	0
32	05:15:02.394	Prueba Visualizar Sílabo 1-32	VisualizarSílabo	2217	✔	703	388	2217	0
33	05:15:02.399	Prueba Visualizar Sílabo 1-33	VisualizarSílabo	2220	✔	703	388	2220	1
34	05:15:02.464	Prueba Visualizar Sílabo 1-34	VisualizarSílabo	2156	✔	703	388	2156	1
35	05:15:02.466	Prueba Visualizar Sílabo 1-35	VisualizarSílabo	2155	✔	703	388	2155	0

Scroll automatically?
 Child samples?
 No of Samples 1000
 Latest Sample 783
 Average 1874
 Deviation 395

Fuente: Elaboración propia con herramienta JMeter.

Se aprecia en la Figura 140 el resultado individual de un hilo ejecutado para la visualización de sílabos, donde se observa el resultado de: en la primera petición el conteo de error (Error Count) es 0, el código de respuesta (Response Code) es de 200 cuyo significado es de exitoso, el tiempo de carga (Load Time) es de 2323 milisegundos, la latencia (Latency) es de 2323 y tiene un tamaño (Headers size in bytes) de 152 en bytes.

Figura 140 Resultado de hilo para funcionalidad Visualizar Sílabo



Fuente: Elaboración propia con herramienta JMeter.

La Figura 141 muestra el resumen del resultado de la simulación donde el promedio mínimo y máximo están en milisegundos, dando una desviación estándar de 395.10 y 0% en error.

Figura 141 Tabla resumen de resultados para Visualizar Sílabo

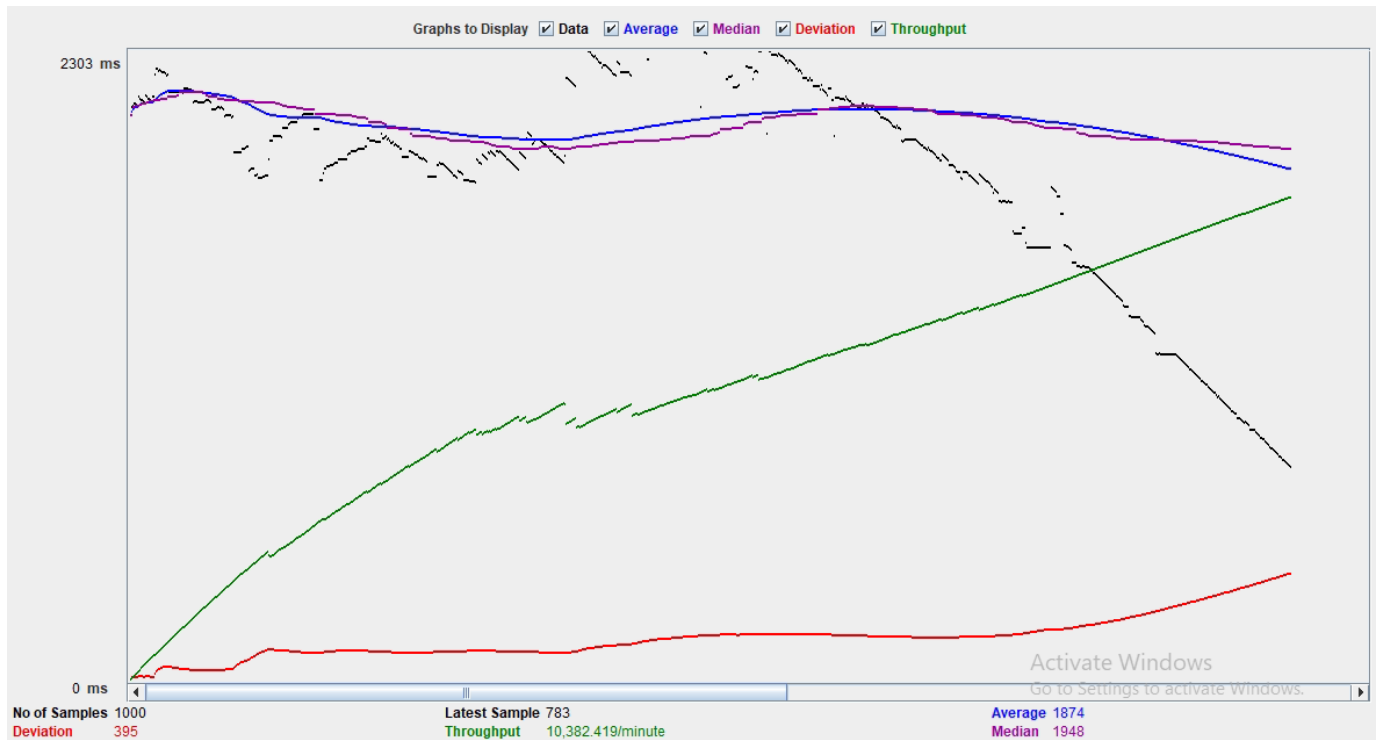
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
VisualizarSílabo	1000	1874	783	2419	395.10	0.00%	173.0/sec	118.80	65.57	703.0
TOTAL	1000	1874	783	2419	395.10	0.00%	173.0/sec	118.80	65.57	703.0

Fuente: Elaboración propia con herramienta JMeter.

Por último, la Figura 142 indica el resumen gráfico en tiempo, mostrando la cantidad de tiempo que estuvo el sistema en ejecución y en la proporción en que los datos fueron llegando.

Los colores del gráfico contienen la leyenda debajo de la misma gráfica; obteniendo un intervalo de comunicación constante con el servidor (color azul).

Figura 142 Resumen gráfico de resultados de Visualizar Sílabos



Fuente: Elaboración propia con herramienta JMeter.

Identificación Académica

La Figura 143, evidencia el resultado de las mil muestras que simulan a los usuarios interactuando con la aplicación y utilizando la funcionalidad de Identificación Académica, donde la respuesta que se obtiene es el tiempo de inicio, el nombre del hilo, el tiempo de demora, el estado y la latencia de cada hilo es mostrada.

Figura 143 Resultados de prueba de estrés para Ident. Académ.

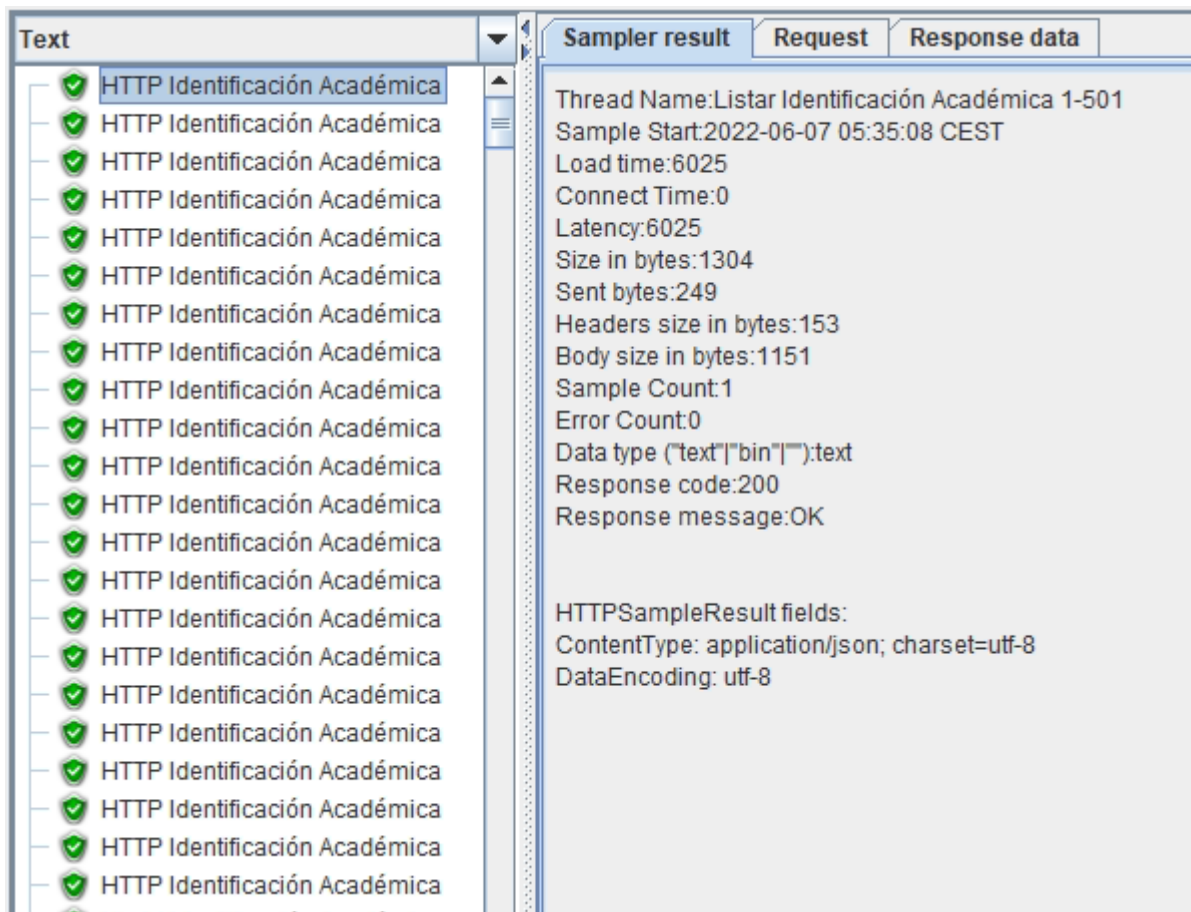
Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time...
1	05:36:18.576	Listar Identificación Académica 1-1	HTTP Identificación Académica	4366	✓	1304	249	4366	1
2	05:36:18.579	Listar Identificación Académica 1-2	HTTP Identificación Académica	4438	✓	1304	249	4438	1
3	05:36:18.584	Listar Identificación Académica 1-3	HTTP Identificación Académica	4466	✓	1304	249	4466	1
4	05:36:18.589	Listar Identificación Académica 1-4	HTTP Identificación Académica	4477	✓	1304	249	4477	0
5	05:36:18.594	Listar Identificación Académica 1-5	HTTP Identificación Académica	4490	✓	1304	249	4490	0
6	05:36:18.599	Listar Identificación Académica 1-6	HTTP Identificación Académica	4489	✓	1304	249	4489	0
7	05:36:18.605	Listar Identificación Académica 1-7	HTTP Identificación Académica	4501	✓	1304	249	4501	0
8	05:36:18.609	Listar Identificación Académica 1-8	HTTP Identificación Académica	4511	✓	1304	249	4511	0
9	05:36:18.614	Listar Identificación Académica 1-9	HTTP Identificación Académica	4533	✓	1304	249	4533	0
10	05:36:18.618	Listar Identificación Académica 1-10	HTTP Identificación Académica	4544	✓	1304	249	4544	0
11	05:36:18.623	Listar Identificación Académica 1-11	HTTP Identificación Académica	4555	✓	1304	249	4555	0
12	05:36:18.630	Listar Identificación Académica 1-12	HTTP Identificación Académica	4553	✓	1304	249	4553	0
13	05:36:18.633	Listar Identificación Académica 1-13	HTTP Identificación Académica	4577	✓	1304	249	4577	0
14	05:36:18.638	Listar Identificación Académica 1-14	HTTP Identificación Académica	4588	✓	1304	249	4588	0
15	05:36:18.646	Listar Identificación Académica 1-15	HTTP Identificación Académica	4600	✓	1304	249	4600	0
16	05:36:18.649	Listar Identificación Académica 1-16	HTTP Identificación Académica	4603	✓	1304	249	4603	0
17	05:36:18.654	Listar Identificación Académica 1-17	HTTP Identificación Académica	4613	✓	1304	249	4613	0
18	05:36:18.660	Listar Identificación Académica 1-18	HTTP Identificación Académica	4624	✓	1304	249	4624	0
19	05:36:18.664	Listar Identificación Académica 1-19	HTTP Identificación Académica	4674	✓	1304	249	4674	0
20	05:36:18.669	Listar Identificación Académica 1-20	HTTP Identificación Académica	4681	✓	1304	249	4681	0
21	05:36:18.676	Listar Identificación Académica 1-21	HTTP Identificación Académica	4682	✓	1304	249	4682	0
22	05:36:18.679	Listar Identificación Académica 1-22	HTTP Identificación Académica	4680	✓	1304	249	4680	0
23	05:36:18.683	Listar Identificación Académica 1-23	HTTP Identificación Académica	4703	✓	1304	249	4703	0
24	05:36:18.689	Listar Identificación Académica 1-24	HTTP Identificación Académica	4698	✓	1304	249	4698	0
25	05:36:18.693	Listar Identificación Académica 1-25	HTTP Identificación Académica	4709	✓	1304	249	4709	1
26	05:36:18.699	Listar Identificación Académica 1-26	HTTP Identificación Académica	4708	✓	1304	249	4708	0
27	05:36:18.705	Listar Identificación Académica 1-27	HTTP Identificación Académica	4703	✓	1304	249	4703	0
28	05:36:18.710	Listar Identificación Académica 1-28	HTTP Identificación Académica	4733	✓	1304	249	4733	0
29	05:36:18.714	Listar Identificación Académica 1-29	HTTP Identificación Académica	4753	✓	1304	249	4753	0
30	05:36:18.719	Listar Identificación Académica 1-30	HTTP Identificación Académica	4763	✓	1304	249	4763	0
31	05:36:18.726	Listar Identificación Académica 1-31	HTTP Identificación Académica	4761	✓	1304	249	4761	0
32	05:36:18.729	Listar Identificación Académica 1-32	HTTP Identificación Académica	4769	✓	1304	249	4769	0
33	05:36:18.734	Listar Identificación Académica 1-33	HTTP Identificación Académica	4773	✓	1304	249	4773	0
34	05:36:18.740	Listar Identificación Académica 1-34	HTTP Identificación Académica	4781	✓	1304	249	4781	0
35	05:36:18.744	Listar Identificación Académica 1-35	HTTP Identificación Académica	4778	✓	1304	249	4778	0

Scroll automatically?
 Child samples?
 No of Samples 1000
 Latest Sample 3577
 Average 5077
 Deviation 826

Fuente: Elaboración propia con herramienta JMeter.

Se aprecia en la Figura 144 el resultado individual de un hilo ejecutado para la Identificación Académica, donde se observa el resultado de: en la primera petición el conteo de error (Error Count) es 0, el código de respuesta (Response Code) es de 200 cuyo significado es de exitoso, el tiempo de carga (Load Time) es de 6025 milisegundos, la latencia (Latency) es de 6025 y tiene un tamaño (Headers size in bytes) de 153 en bytes.

Figura 144 Resultado de hilo para funcionalidad Ident. Académ.



Fuente: Elaboración propia con herramienta JMeter.

La Figura 145 muestra el resumen del resultado de la simulación donde el promedio mínimo y máximo están en milisegundos, dando una desviación estándar de 826.24 y 0% en error.

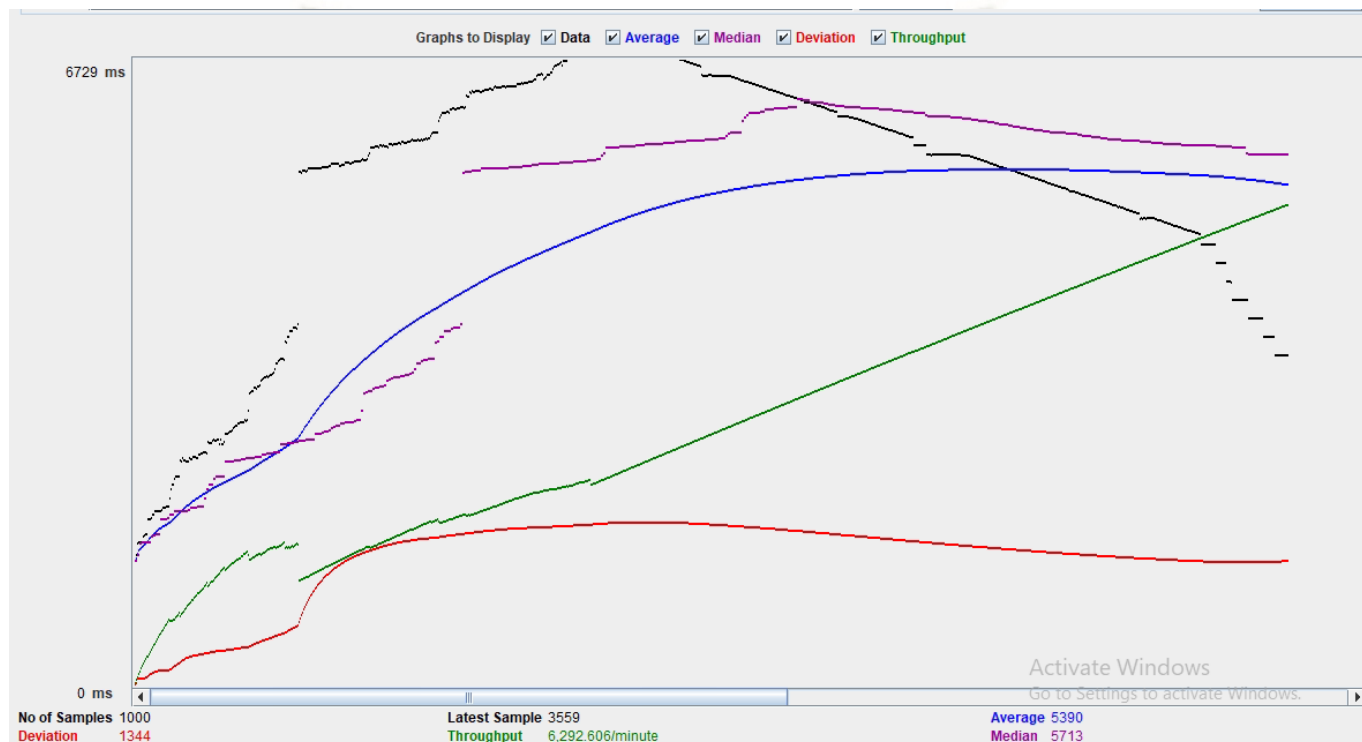
Figura 145 Tabla resumen de resultados para Ident. Académ.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Identificación...	1000	5077	3577	6481	826.24	0.00%	116.6/sec	148.54	28.36	1304.0
TOTAL	1000	5077	3577	6481	826.24	0.00%	116.6/sec	148.54	28.36	1304.0

Fuente: Elaboración propia con herramienta JMeter.

Por último, la Figura 146 indica el resumen gráfico en tiempo, mostrando la cantidad de tiempo que estuvo el sistema en ejecución y en la proporción en que los datos fueron llegando. Los colores del gráfico contienen la leyenda debajo de la misma gráfica; obteniendo un intervalo de comunicación constante con el servidor (color azul).

Figura 146 Resumen gráfico de resultados de Ident. Académ.



Fuente: Elaboración propia con herramienta JMeter.

Listar Cursos Docente

La Figura 147, evidencia el resultado de las mil muestras que simulan a los usuarios interactuando con la aplicación y utilizando la funcionalidad de Listar los cursos de los docentes, donde la respuesta que se obtiene es el tiempo de inicio, el nombre del hilo, el tiempo de demora, el estado y la latencia de cada hilo es mostrada.

Figura 147 Propiedades de Hilo en herramienta JMeter Cursos Docente

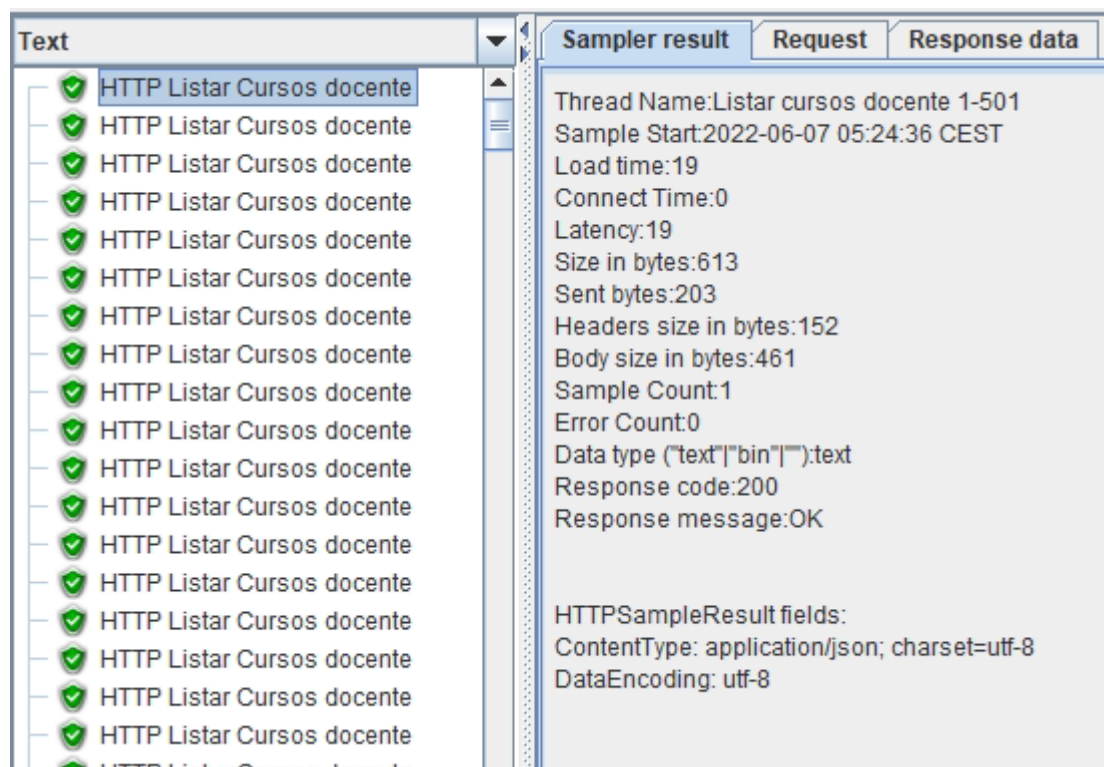
Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	05:24:34.200	Listar cursos docente 1-1	HTTP Listar Cursos docente	463	✓	613	203	463	1
2	05:24:34.202	Listar cursos docente 1-2	HTTP Listar Cursos docente	539	✓	613	203	539	1
3	05:24:34.208	Listar cursos docente 1-3	HTTP Listar Cursos docente	538	✓	613	203	538	1
4	05:24:34.215	Listar cursos docente 1-4	HTTP Listar Cursos docente	534	✓	613	203	534	0
5	05:24:34.218	Listar cursos docente 1-5	HTTP Listar Cursos docente	534	✓	613	203	534	0
6	05:24:34.224	Listar cursos docente 1-6	HTTP Listar Cursos docente	529	✓	613	203	529	0
7	05:24:34.228	Listar cursos docente 1-7	HTTP Listar Cursos docente	527	✓	613	203	527	0
8	05:24:34.232	Listar cursos docente 1-8	HTTP Listar Cursos docente	548	✓	613	203	548	0
9	05:24:34.237	Listar cursos docente 1-9	HTTP Listar Cursos docente	546	✓	613	203	546	0
10	05:24:34.243	Listar cursos docente 1-10	HTTP Listar Cursos docente	543	✓	613	203	543	0
11	05:24:34.253	Listar cursos docente 1-12	HTTP Listar Cursos docente	535	✓	613	203	535	0
12	05:24:34.247	Listar cursos docente 1-11	HTTP Listar Cursos docente	543	✓	613	203	543	0
13	05:24:34.258	Listar cursos docente 1-13	HTTP Listar Cursos docente	534	✓	613	203	534	0
14	05:24:34.263	Listar cursos docente 1-14	HTTP Listar Cursos docente	530	✓	613	203	530	0
15	05:24:34.267	Listar cursos docente 1-15	HTTP Listar Cursos docente	539	✓	613	203	539	0
16	05:24:34.274	Listar cursos docente 1-16	HTTP Listar Cursos docente	535	✓	613	203	535	0
17	05:24:34.277	Listar cursos docente 1-17	HTTP Listar Cursos docente	533	✓	613	203	533	0
18	05:24:34.283	Listar cursos docente 1-18	HTTP Listar Cursos docente	529	✓	613	203	529	0
19	05:24:34.290	Listar cursos docente 1-19	HTTP Listar Cursos docente	530	✓	613	203	530	0
20	05:24:34.301	Listar cursos docente 1-21	HTTP Listar Cursos docente	557	✓	613	203	557	0
21	05:24:34.293	Listar cursos docente 1-20	HTTP Listar Cursos docente	566	✓	613	203	566	0
22	05:24:34.303	Listar cursos docente 1-22	HTTP Listar Cursos docente	557	✓	613	203	557	0
23	05:24:34.308	Listar cursos docente 1-23	HTTP Listar Cursos docente	553	✓	613	203	553	1
24	05:24:34.312	Listar cursos docente 1-24	HTTP Listar Cursos docente	549	✓	613	203	549	0
25	05:24:34.317	Listar cursos docente 1-25	HTTP Listar Cursos docente	546	✓	613	203	546	0
26	05:24:34.323	Listar cursos docente 1-26	HTTP Listar Cursos docente	541	✓	613	203	541	0
27	05:24:34.328	Listar cursos docente 1-27	HTTP Listar Cursos docente	537	✓	613	203	537	0
28	05:24:34.333	Listar cursos docente 1-28	HTTP Listar Cursos docente	533	✓	613	203	533	0
29	05:24:34.343	Listar cursos docente 1-30	HTTP Listar Cursos docente	524	✓	613	203	524	0
30	05:24:34.338	Listar cursos docente 1-29	HTTP Listar Cursos docente	529	✓	613	203	529	1
31	05:24:34.348	Listar cursos docente 1-31	HTTP Listar Cursos docente	520	✓	613	203	520	0
32	05:24:34.354	Listar cursos docente 1-32	HTTP Listar Cursos docente	533	✓	613	203	533	0
33	05:24:34.358	Listar cursos docente 1-33	HTTP Listar Cursos docente	530	✓	613	203	530	0
34	05:24:34.363	Listar cursos docente 1-34	HTTP Listar Cursos docente	525	✓	613	203	525	0
35	05:24:34.369	Listar cursos docente 1-35	HTTP Listar Cursos docente	522	✓	613	203	522	1

Scroll automatically?
 Child samples?
 No of Samples 1000
 Latest Sample 209
 Average 250
 Deviation 189

Fuente: Elaboración propia con herramienta JMeter.

Se aprecia en la Figura 148 el resultado individual de un hilo ejecutado para listar los cursos de los docentes, donde se observa el resultado de: en la primera petición el conteo de error (Error Count) es 0, el código de respuesta (Response Code) es de 200 cuyo significado es de exitoso, el tiempo de carga (Load Time) es de 19 milisegundos, la latencia (Latency) es de 613 y tiene un tamaño (Headers size in bytes) de 152 en bytes.

Figura 148 Resultado de hilo para funcionalidad Cursos Docente



Fuente: Elaboración propia con herramienta JMeter.

La Figura 149 muestra el resumen del resultado de la simulación donde el promedio mínimo y máximo están en milisegundos, dando una desviación estándar de 189.67 y 0% en error.

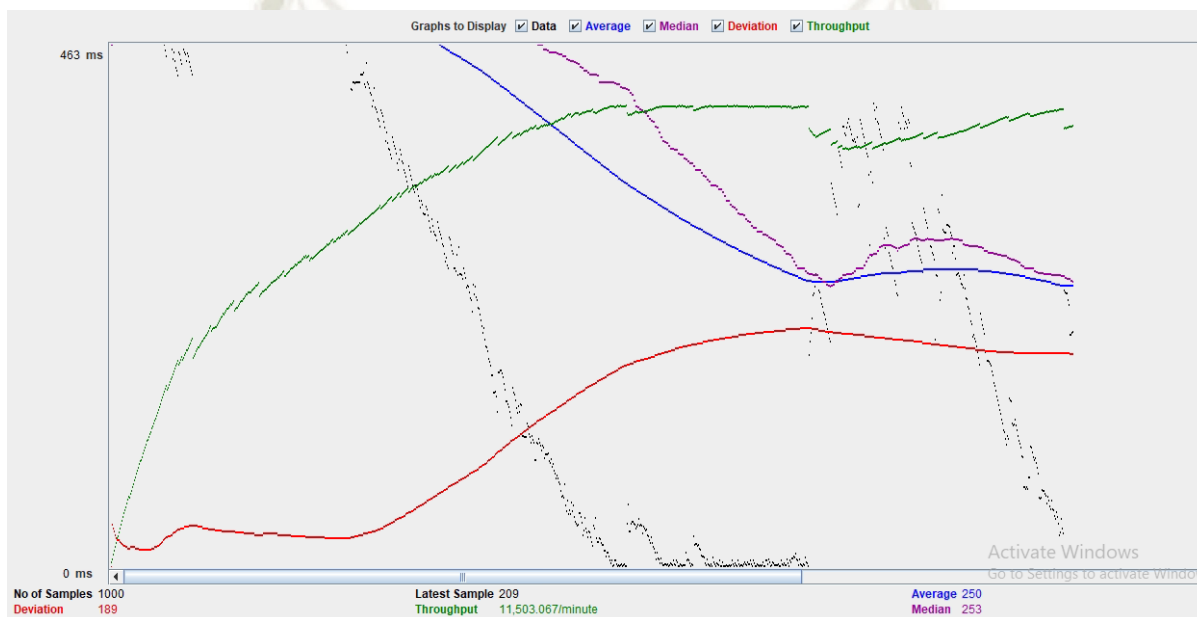
Figura 149 Tabla resumen de resultados para Cursos Docente

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Listar Curso...	1000	250	1	566	189.67	0.00%	191.7/sec	114.77	38.01	613.0
TOTAL	1000	250	1	566	189.67	0.00%	191.7/sec	114.77	38.01	613.0

Fuente: Elaboración propia.

Por último, la Figura 150 indica el resumen gráfico en tiempo, mostrando la cantidad de tiempo que estuvo el sistema en ejecución y en la proporción en que los datos fueron llegando. Los colores del gráfico contienen la leyenda debajo de la misma gráfica; obteniendo un intervalo de comunicación constante con el servidor (color azul).

Figura 150 Resumen gráfico de resultados de Cursos Docente



Fuente: Elaboración propia con herramienta JMeter.

Programa Formativo

La Figura 151, evidencia el resultado de las mil muestras que simulan a los usuarios interactuando con la aplicación y utilizando la funcionalidad de ver el Programa Formativo de la asignatura, donde la respuesta que se obtiene es el tiempo de inicio, el nombre del hilo, el tiempo de demora, el estado y la latencia de cada hilo es mostrada.

Figura 151 Propiedades de Hilo en herramienta JMeter Prog. Formativo

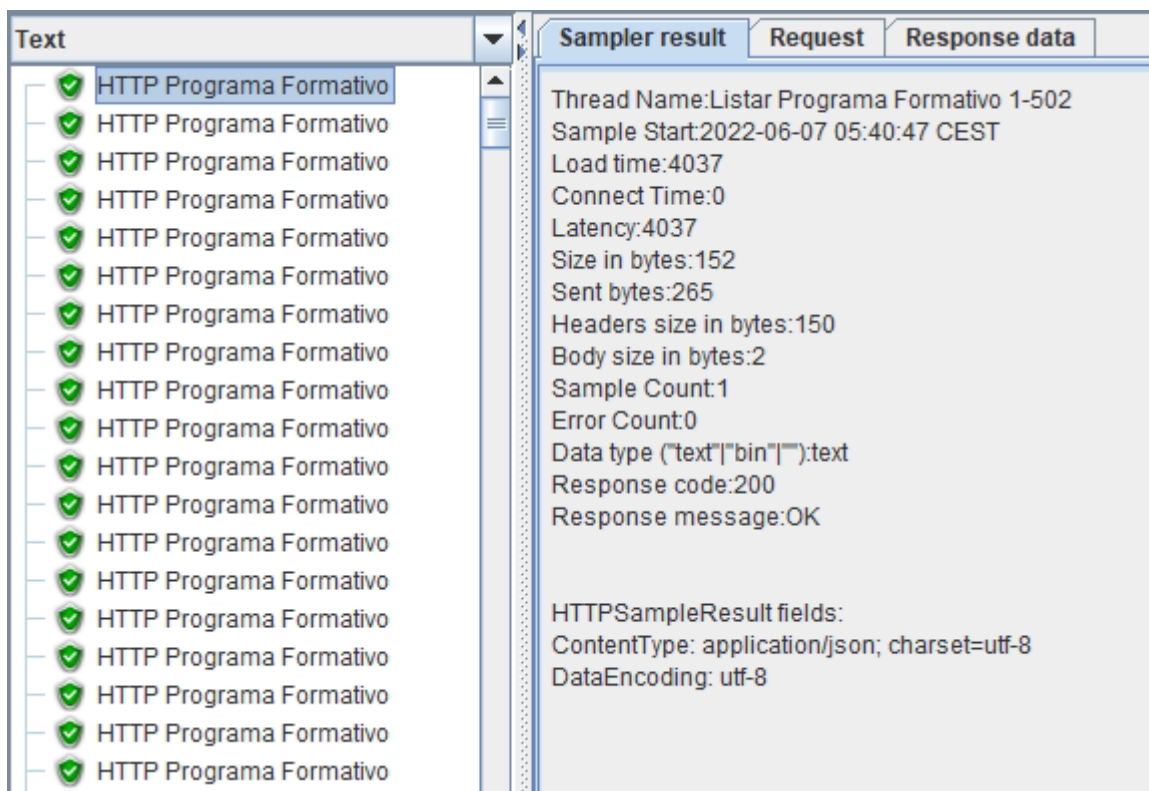
Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
1	05:40:45.183	Listar Programa Formativo 1-1	HTTP Programa Formativo	6128	✓	152	265	6128	182
2	05:40:45.362	Listar Programa Formativo 1-37	HTTP Programa Formativo	5967	✓	152	265	5967	3
3	05:40:45.353	Listar Programa Formativo 1-35	HTTP Programa Formativo	6014	✓	152	265	6014	13
4	05:40:45.358	Listar Programa Formativo 1-36	HTTP Programa Formativo	6014	✓	152	265	6014	8
5	05:40:45.348	Listar Programa Formativo 1-34	HTTP Programa Formativo	6029	✓	152	265	6029	18
6	05:40:45.341	Listar Programa Formativo 1-33	HTTP Programa Formativo	6053	✓	152	265	6053	26
7	05:40:45.339	Listar Programa Formativo 1-32	HTTP Programa Formativo	6059	✓	152	265	6059	28
8	05:40:45.332	Listar Programa Formativo 1-31	HTTP Programa Formativo	6072	✓	152	265	6072	36
9	05:40:45.327	Listar Programa Formativo 1-30	HTTP Programa Formativo	6081	✓	152	265	6081	41
10	05:40:45.324	Listar Programa Formativo 1-29	HTTP Programa Formativo	6088	✓	152	265	6088	44
11	05:40:45.318	Listar Programa Formativo 1-28	HTTP Programa Formativo	6098	✓	152	265	6098	50
12	05:40:45.311	Listar Programa Formativo 1-27	HTTP Programa Formativo	6111	✓	152	265	6111	57
13	05:40:45.308	Listar Programa Formativo 1-26	HTTP Programa Formativo	6118	✓	152	265	6118	60
14	05:40:45.304	Listar Programa Formativo 1-25	HTTP Programa Formativo	6128	✓	152	265	6128	64
15	05:40:45.296	Listar Programa Formativo 1-24	HTTP Programa Formativo	6141	✓	152	265	6141	73
16	05:40:45.291	Listar Programa Formativo 1-23	HTTP Programa Formativo	6149	✓	152	265	6149	78
17	05:40:45.288	Listar Programa Formativo 1-22	HTTP Programa Formativo	6155	✓	152	265	6155	81
18	05:40:45.284	Listar Programa Formativo 1-21	HTTP Programa Formativo	6165	✓	152	265	6165	85
19	05:40:45.277	Listar Programa Formativo 1-20	HTTP Programa Formativo	6203	✓	152	265	6203	92
20	05:40:45.272	Listar Programa Formativo 1-19	HTTP Programa Formativo	6209	✓	152	265	6209	98
21	05:40:45.267	Listar Programa Formativo 1-18	HTTP Programa Formativo	6221	✓	152	265	6221	104
22	05:40:45.263	Listar Programa Formativo 1-17	HTTP Programa Formativo	6226	✓	152	265	6226	108
23	05:40:45.257	Listar Programa Formativo 1-16	HTTP Programa Formativo	6233	✓	152	265	6233	115
24	05:40:45.252	Listar Programa Formativo 1-15	HTTP Programa Formativo	6239	✓	152	265	6239	120
25	05:40:45.246	Listar Programa Formativo 1-14	HTTP Programa Formativo	6246	✓	152	265	6246	126
26	05:40:45.242	Listar Programa Formativo 1-13	HTTP Programa Formativo	6250	✓	152	265	6250	130
27	05:40:45.237	Listar Programa Formativo 1-12	HTTP Programa Formativo	6256	✓	152	265	6256	135
28	05:40:45.233	Listar Programa Formativo 1-11	HTTP Programa Formativo	6261	✓	152	265	6261	140
29	05:40:45.227	Listar Programa Formativo 1-10	HTTP Programa Formativo	6267	✓	152	265	6267	146
30	05:40:45.221	Listar Programa Formativo 1-9	HTTP Programa Formativo	6275	✓	152	265	6275	152
31	05:40:45.217	Listar Programa Formativo 1-8	HTTP Programa Formativo	6279	✓	152	265	6279	156
32	05:40:45.212	Listar Programa Formativo 1-7	HTTP Programa Formativo	6285	✓	152	265	6285	162
33	05:40:45.209	Listar Programa Formativo 1-6	HTTP Programa Formativo	6289	✓	152	265	6289	165
34	05:40:45.201	Listar Programa Formativo 1-5	HTTP Programa Formativo	6298	✓	152	265	6298	173
35	05:40:45.197	Listar Programa Formativo 1-4	HTTP Programa Formativo	6303	✓	152	265	6303	177

Scroll automatically?
 Child samples?
 No of Samples 1000
 Latest Sample 2265
 Average 3879
 Deviation 917

Fuente: Elaboración propia con herramienta JMeter.

Se aprecia en la Figura 152 el resultado individual de un hilo ejecutado ver el programa formativo, donde se observa el resultado de: en la primera petición el conteo de error (Error Count) es 0, el código de respuesta (Response Code) es de 200 cuyo significado es de exitoso, el tiempo de carga (Load Time) es de 4037 milisegundos, la latencia (Latency) es de 4037 y tiene un tamaño (Headers size in bytes) de 150 en bytes.

Figura 152 Resultado de hilo para funcionalidad Programa Formativo



Fuente: Elaboración propia con herramienta JMeter.

La Figura 153 muestra el resumen del resultado de la simulación donde el promedio mínimo y máximo están en milisegundos, dando una desviación estándar de 917.88 y 0% en error.

Figura 153 Tabla resumen de resultados para Programa Formativo

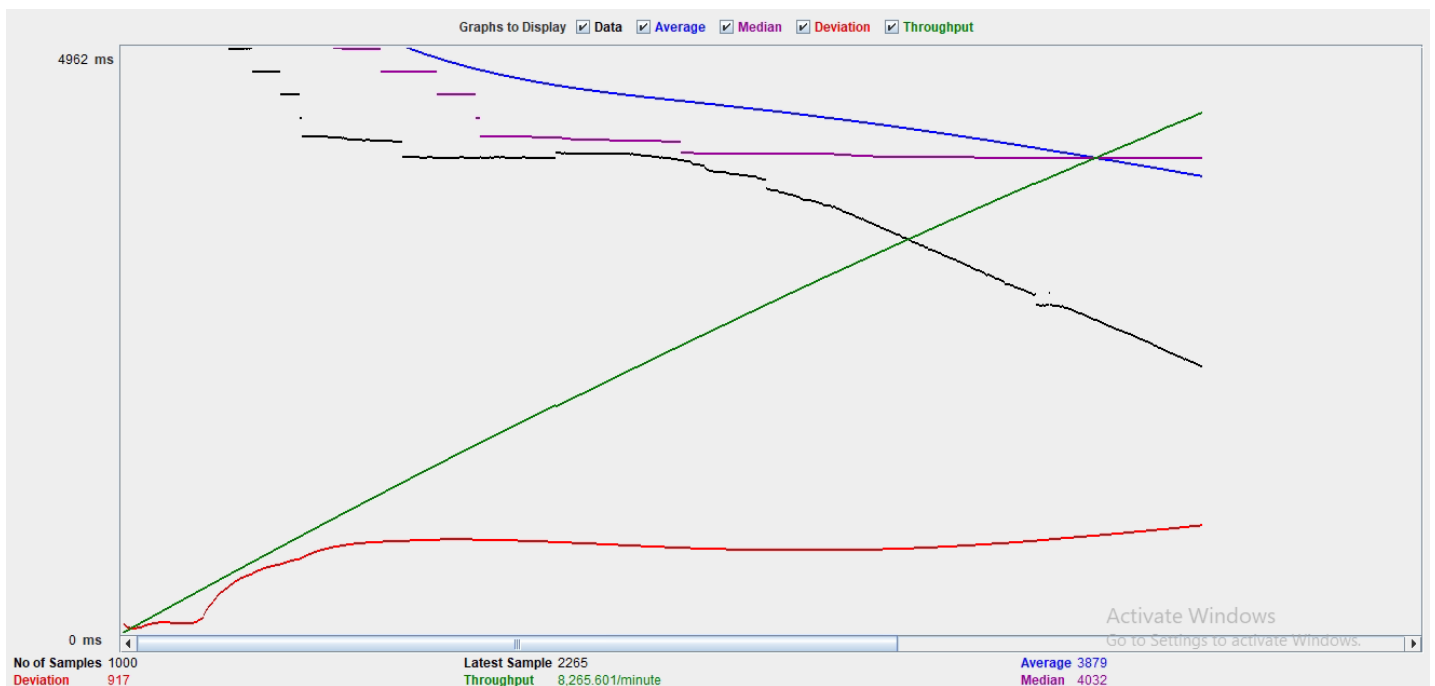
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Programa Fo...	1000	3879	2265	6315	917.88	0.00%	137.8/sec	20.45	35.65	152.0
TOTAL	1000	3879	2265	6315	917.88	0.00%	137.8/sec	20.45	35.65	152.0

Fuente: Elaboración propia con herramienta JMeter.

Por último, la Figura 154 indica el resumen gráfico en tiempo, mostrando la cantidad de tiempo que estuvo el sistema en ejecución y en la proporción en que los datos fueron llegando.

Los colores del gráfico contienen la leyenda debajo de la misma gráfica; obteniendo un intervalo de comunicación constante con el servidor (color azul).

Figura 154 Resumen gráfico de resultados de Programa Formativo



Fuente: Elaboración propia con herramienta JMeter.

Conclusiones:

Las pruebas de estrés que se aplicaron al Sistema Gestión de Sílabos a través de la herramienta JMeter fueron pasadas. Las pruebas por medio de la herramienta consistieron en la ejecución con una cantidad de mil instancias o usuarios, en un periodo de carga de 5 segundos. Se crearon los hilos o instancias para las funcionalidades de visualización de sílabo, identificación académica, listar cursos de docentes y visualización de programa formativo, obteniendo un resultado de 0% de errores y evidenciando que el sistema funciona en óptimas condiciones.

2.5 Manuales de Usuario

El manual de usuario explica la utilización del sistema en cada apartado para los diversos roles que existen en el sistema.

El manual se encuentra en el Apéndice A: Manual de Usuario



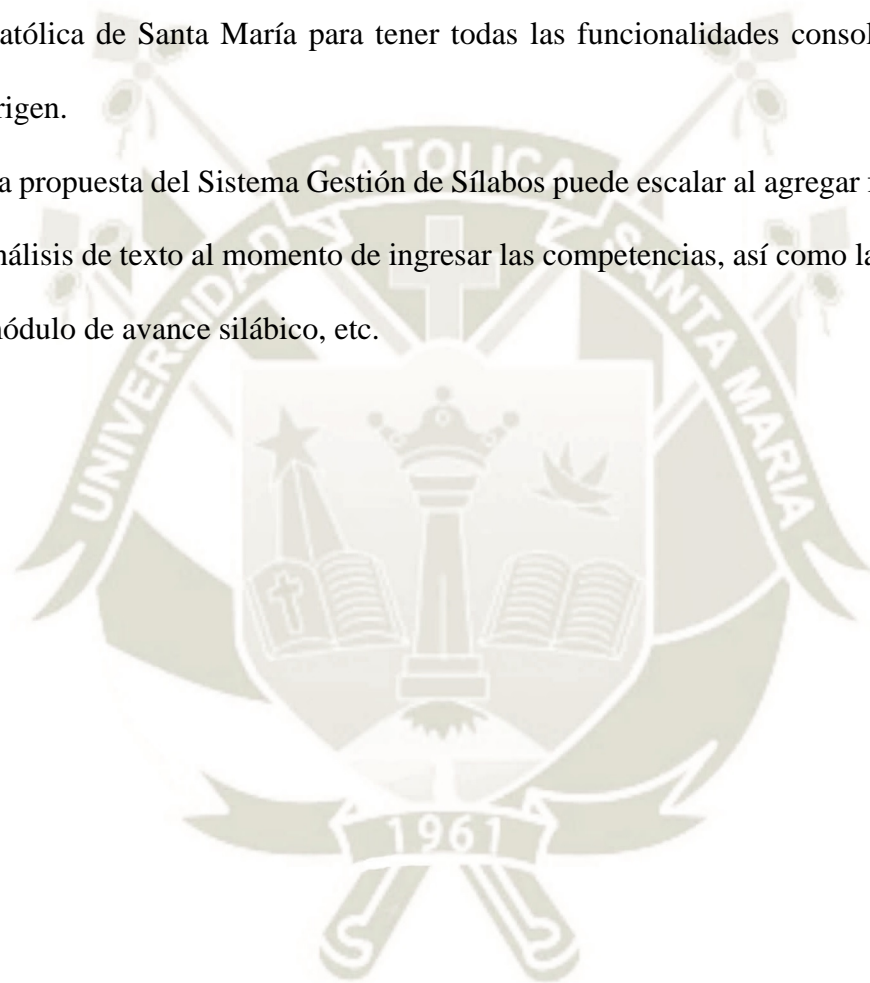
CONCLUSIONES

1. Se ha desarrollado con éxito el prototipo de Sistema de Gestión de Sílabos. Este aplicativo permite el desarrollo y gestión en las diversas etapas de creación del sílabo y monitoreo hasta su culminación.
2. La propuesta del sistema de gestión de sílabos tiene un diseño que permite de manera fácil elaborar y gestionar los sílabos por medio del aplicativo web. Su utilización se vuelve fácil de reconocer debido a que está basado en la plantilla desarrollada por el centro académico de la Universidad Católica de Santa María.
3. La propuesta del sistema para la Gestión de sílabos fue implementada con tecnologías libres para la reducción del costo del desarrollo. Se utilizaron las tecnologías para la parte de iterfaz visual como REACT, TailwindCSS, PDFMake; y como framework para la programación del API a .Net Core. Estas tecnologías tienen el respaldo de una comunidad y las convierten en opciones potentes para el desarrollo.
4. El sistema piloto fue alojado en el servidor de desarrollo de la Universidad Católica de Santa María, fue utilizado por los catedráticos y validada su funcionalidad, pudiendo apreciar como este cumple como solución para la gestión de sílabos
5. Se mejora la experiencia del docente en el uso y gestión de sílabos académicos en la Universidad Católica de Santa María mediante el uso de servicios que se basan en tecnologías web e internet enfocados en el desarrollo de sílabos.

RECOMENDACIONES

Las recomendaciones que surgen después del desarrollo de este trabajo son:

1. Se sugiere integrar el sistema gestión de sílabos con el ERP ya existente en la Universidad Católica de Santa María para tener todas las funcionalidades consolidadas en un solo origen.
2. La propuesta del Sistema Gestión de Sílabos puede escalar al agregar funcionalidades de análisis de texto al momento de ingresar las competencias, así como la integración de un módulo de avance silábico, etc.



Referencias Bibliográficas

Bhargava, H. K., Power, D. J., & Sun, D. (2007). Progress in Web-based decision support technologies. *Decis. Support Syst.*

Chelvarayan, A., Jayothisa, C., Hashim, H. & Nizat, K. (2011). Syllabus Management System for Academics Practicing Knowledge Management. *Journal of Organizational Knowledge Management*

Dimakis, N., Polymenakos, L., & Soldatos, J. (2006). Enhancing learning experiences through context-aware collaborative services: Software architecture and prototype system. in *Proceedings - Fourth IEEE International Workshop on Wireless, Mobile and Ubiquitous Technology in Education, WMUTE 2006.*

Force, D. C. & Zhang, J. (2016). Knowledge discovery from within: An examination of records management and electronic records management syllabi. *Rec. Manag. J.*

Gluga, R., Kay, J., & Lever, T. (2012). Foundations for Modeling University Curricula in Terms of Multiple Learning Goal Sets. *IEEE Transactions on Learning Technologies*, 6, 25 - 37. Obtenido de <https://ieeexplore.ieee.org/document/6261335>

Gerbé, O., Raynauld, J., & Camarero, R. (2010). Learning outcomes in a model-based approach to curriculum design. *5th International Conference on Internet and Web Applications and Services, ICIW 2010*, pp. 550–553.

Júnior, A. A. C., Misra, S., & Soares, M. S. (2019). ArchCaMO - A Maturity Model for Software Architecture Description Based on ISO/IEC/IEEE 42010:2011. *Lecture Notes in*

Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics).

Kawaba, T., Tsuchiya, T., & Koyanagi, K. (2012). Research on inter-domain collaborative syllabus services with cross-retrieval - IEEE Conference Publication. Obtenido de <https://ieeexplore.ieee.org/abstract/document/6268513>.

Martines, H. (2013). El desarrollo de competencias en la formación universitaria – Guía de diseño de la instrucción del docente de la UCSM. *Centro de Desarrollo Académico*

Molontay, R., Horváth, N., Bergmann, J., Szekrényes, D. & Szabó, M. (2020). Characterizing curriculum prerequisite networks by a student flow approach. *IEEE Transactions on Learning Technologies*.

Morales, G. R. & Benedí, J. P. (2017). Towards a reference software architecture for improving the accessibility and usability of open course ware. *ACM International Conference Proceeding Series*.

Ota, S. & Mima, Hideki (2011). Machine Learning-based Syllabus Classification toward Automatic Organization of Issue-oriented Interdisciplinary Curricula. *Pacific Association for Computational Linguistics (PACLING 2011)*

Yucel, R. (2012). Designing Knowledge Management Syllabus: How Business Administration Students Learn at Summer School? *PROCEEDINGS OF THE 9TH INTERNATIONAL CONFERENCE ON INTELLECTUAL CAPITAL, KNOWLEDGE MANAGEMENT & ORGANISATIONAL LEARNING*.

Vidoni, M. & Vecchietti, A. (2016). Towards a reference architecture for advanced planning systems. *ICEIS 2016 - Proceedings of the 18th International Conference on Enterprise Information Systems*.

APÉNDICE A: MANUAL DE USUARIO

MANUAL DE USUARIO

SISTEMA DE GESTIÓN DE SÍLABOS

VERSIÓN 2.0



ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

FACULTAD DE CIENCIAS E INGENIERÍAS

FÍSICAS Y FORMALES

Junio del 2022

AREQUIPA – PERÚ

1. RESUMEN DEL PROYECTO

1.1. Objetivos

Optimizar la gestión de sílabos en la UCSM.

2. 1.2 Métodos

Se efectuará un estudio documental de los sílabos de la Escuela de Ingeniería de Sistemas. Para la etapa de diseño, desarrollo e implementación del sistema nos apoyaremos en conceptos relacionados a la metodología SCRUM, para establecernos metas, agilizar el trabajo y registrar la documentación adecuada.

3. PRINCIPALES FUNCIONALIDADES

3.1. Roles de la aplicación

La aplicación tiene los siguientes roles:

- **Jefe de departamento:** Es el encargado de realizar las operaciones con los sílabos que los docentes han elaborado.
- **Administrador:** Es el encargado de realizar la carga de información proveniente del sistema de biblioteca, académico y de personal y otros procesos propios del perfil.
- **Docente:** Es el encargado de la elaboración del sílabo según la carga académica asignada a cada uno de los docentes.

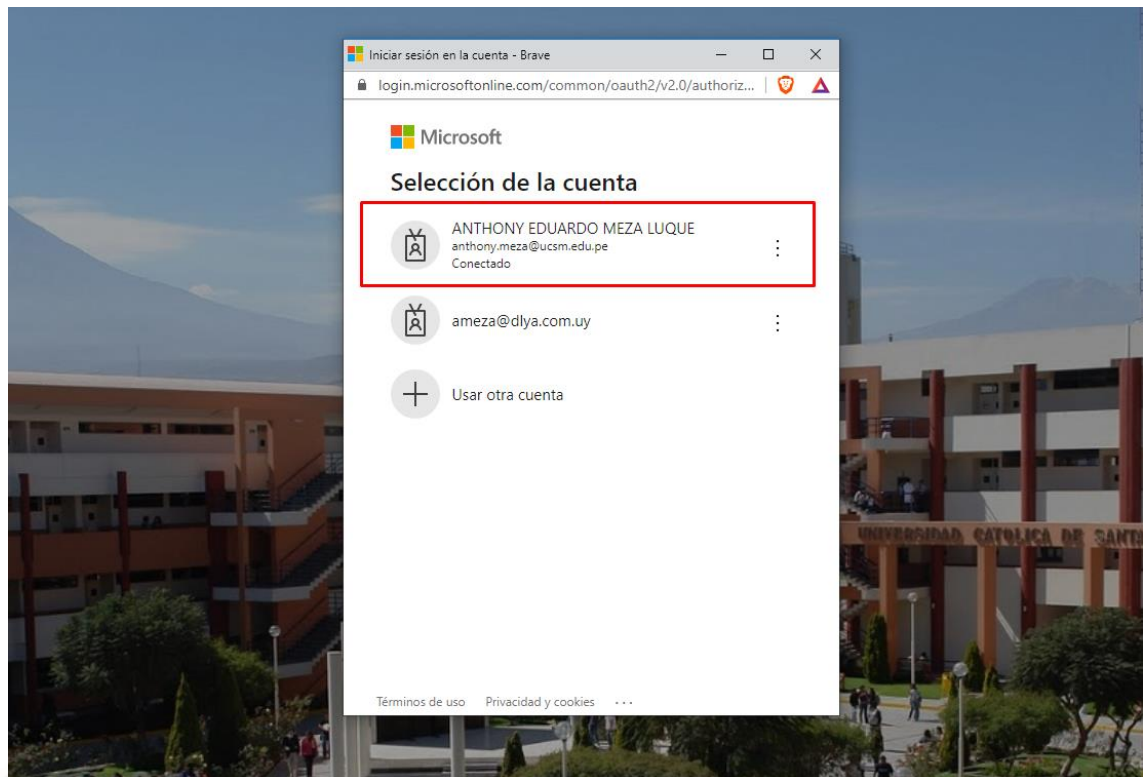
3.2. HU0101 Login de la aplicación

Para la autenticación a la aplicación se hará uso de las API que proporciona Office 365. En la siguiente pantalla se muestra el Login que utilizarán los diferentes usuarios para ingresar a la aplicación. El inicio de sesión será por medio del correo institucional. Ingresando al dominio correspondiente, apreciamos el siguiente acceso.



Ingreso a sistema e inicio de sesión

Todos los usuarios con los roles jefe de departamento, administrador y docente acceden a la aplicación a través de la pantalla Login.

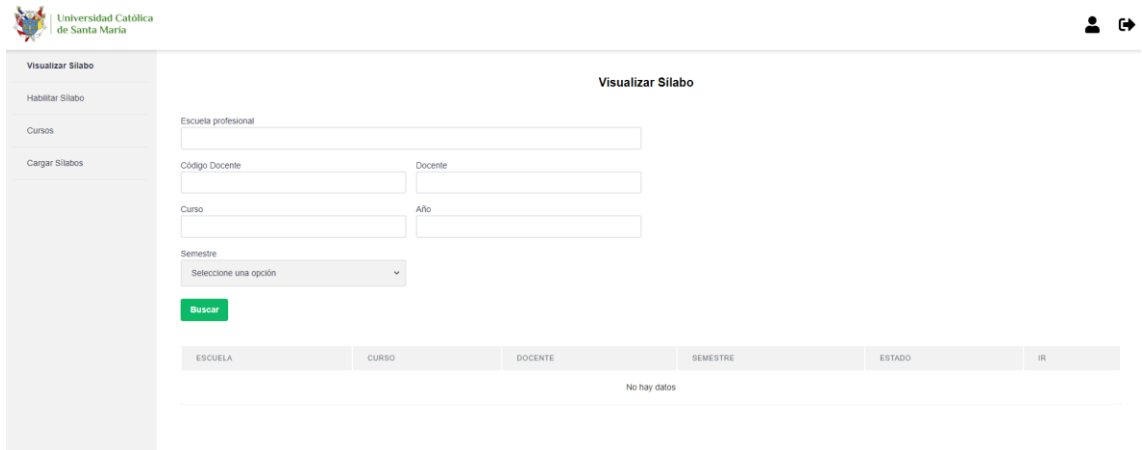


Selección de correo institucional e ingreso

3.3. HU0102 Página principal de la aplicación

Los roles que acceden a la pantalla principal son: jefe de departamento, administrador y docente.

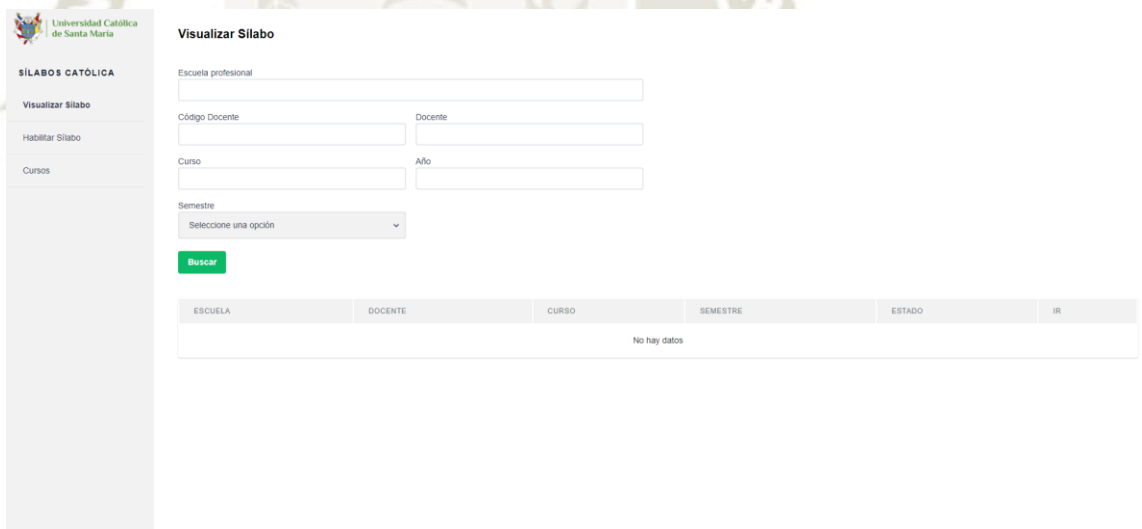
En la siguiente imagen se muestra la localización de los elementos básicos de la aplicación, en el lado izquierdo superior se encontrará el logo de la universidad y en la parte inferior el menú. En el lado derecho superior la imagen del usuario.



Pantalla de ingreso

3.4. HU0401 Pantalla Visualizar Sílabos

El rol que accede a la pantalla de visualización de sílabos es el rol jefe de departamento.



Visualizar sílabos

La pantalla de inicio del jefe de departamento tendrá la posibilidad buscar por escuela profesional, código de docente, nombre del docente, curso, año de la asignatura y el semestre

en el que se dicta. En la parte inferior visualizará los cursos de acuerdo con la búsqueda, podrá ver el estado de los sílabos y podrá visualizarlos.

No se mostrarán datos a menos que se ingresen como mínimo el año en que se requiera buscar la información.

Los estados de los sílabos son: Abierto, En proceso, Aprobado, Concluido, Cerrado.

Visualizar Sílabo

Escuela profesional

Código Docente Docente

Curso Año

Semestre
Selecione una opción

ESCUELA	CURSO	DOCENTE	SEMESTRE	ESTADO	IR
INGENIERIA DE SISTEMAS	ANÁLISIS Y DISEÑO DE SISTEMAS	JORGE MARTINEZ MUÑOZ	V	EN PROCESO	Ir
INGENIERIA DE SISTEMAS	ANÁLISIS Y DISEÑO DE SISTEMAS	JOSÉ ALFREDO SULLA TORRES	V	EN PROCESO	Ir
INGENIERIA DE SISTEMAS	ANÁLISIS Y DISEÑO DE SISTEMAS	KARINA ROSAS	V	EN PROCESO	Ir
INGENIERIA DE SISTEMAS	CÁLCULO DIFERENCIAL	FREDDY QUISPE	III	ABIERTO	Ir

Búsqueda de sílabos

Links

1.- HU0402 Visualizar Sílabos PDF

El resultado emite la información de la escuela en la que se encuentra la asignatura, el nombre del curso, los docentes que se encuentran desarrollando el sílabo, el semestre académico, además del estado de desarrollo y un vínculo para poder revisar el sílabo en formato PDF.

3.5. HU0402 Pantalla Visualizar Sílabos PDF

Los roles que acceden a la pantalla principal son: jefe de departamento, administrador.

Se mostrará en una ventana emergente, en formato PDF los sílabos de los docentes sin importar el estado actual.

Visualizar Sílabo

Habilitar Sílabo

Cursos

Cargar Sílabos

Visualizar Sílabo

UNIVERSIDAD CATÓLICA DE SANTA MARÍA
AREQUIPA- PERÚ

FACULTAD: FACULTAD DE CIENCIAS FÍSICAS Y FORMALES
ESCUELA PROFESIONAL: INGENIERÍA DE SISTEMAS

PLAN DE ESTUDIOS
SÍLABO DE ASIGNATURA

1. IDENTIFICACIÓN ACADÉMICA

1.1.- Nombre de la Asignatura: ANÁLISIS Y DISEÑO DE SISTEMAS
Código de la Asignatura: 7105037
Semestre Académico en que se desarrolla: V

1.2.- Peso Académico de la Asignatura.

CRÉDITOS	HORAS SEMANALES				HORAS SEMESTRALES		
	Horas Teóricas	Horas Prácticas	Jefe de Prácticas	Horas Virtuales	Horas Teóricas	Horas Prácticas	Horas Virtuales
4	2	2	2	0	36	72	0

1.3.- Código, nombre y créditos de Asignaturas Equivalentes:
7106037 Análisis y Diseño de Sistemas
7107188 Laboratorio de Análisis y Diseño de Sistemas

1.4.- Código y nombre de Asignaturas Pre-requisito:

INGENIERIA DE SISTEMAS COMPUTACIÓN EN RED II KARINA ROSAS V APROBADO Ir

INGENIERIA DE SISTEMAS DESARROLLO I MARIA GRACIA AYALA I ABIERTO Ir

Sílabo en formato PDF

3.6. HU0501 Pantalla Habilitar Silabus

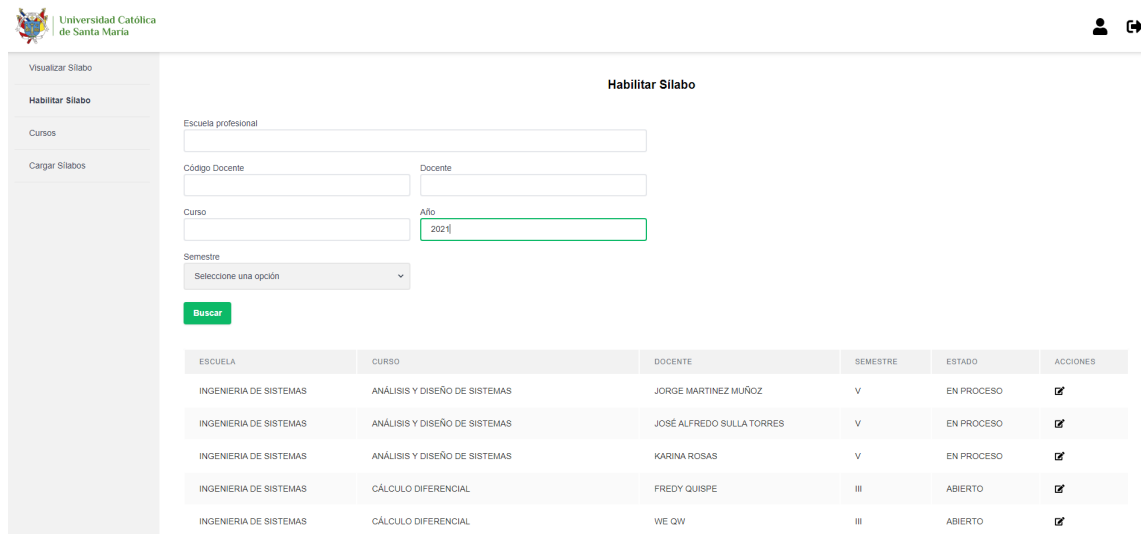
El jefe de departamento tendrá la opción de cambiar el estado de los sílabos que terminan su periodo de desarrollo.

Los estados manejarán la siguiente dinámica:

- **Abierto:** Estado del Silabo después de la carga correspondiente al semestre académico.
- **En proceso:** Silabo iniciado en su desarrollo y en modificación por el(los) docente(s).
- **Aprobado:** Silabo ha sido concluido por parte del docente y envía solicitud de revisión al jefe de departamento para que pueda concluirlo.
- **Cerrado:** Silabo ha finalizado su revisión y es dada la conformidad por parte del jefe de departamento. El silabo no permite modificaciones. Tras una fecha límite todos los sílabos pasaran al estado cerrado.

Link:

1.- HU0502 Habilitar Sílabos



ESCUELA	CURSO	DOCENTE	SEMESTRE	ESTADO	ACCIONES
INGENIERIA DE SISTEMAS	ANÁLISIS Y DISEÑO DE SISTEMAS	JORGE MARTINEZ MUÑOZ	V	EN PROCESO	🗑️
INGENIERIA DE SISTEMAS	ANÁLISIS Y DISEÑO DE SISTEMAS	JOSÉ ALFREDO SULLA TORRES	V	EN PROCESO	🗑️
INGENIERIA DE SISTEMAS	ANÁLISIS Y DISEÑO DE SISTEMAS	KARINA ROSAS	V	EN PROCESO	🗑️
INGENIERIA DE SISTEMAS	CÁLCULO DIFERENCIAL	FREDY QUISPE	III	ABIERTO	🗑️
INGENIERIA DE SISTEMAS	CÁLCULO DIFERENCIAL	WE QW	III	ABIERTO	🗑️

Búsqueda de sílabos pantalla Habilitar Sílabos

3.7. HU0502 Pantalla Habilitar Sílabos

El Jefe de Departamento es el encargado de supervisar los sílabos para que el docente pueda trabajar con ellos.

Se tendrá un botón en el apartado de acciones, para realizar el cambio de estado en caso de que el Jefe de Departamento lo considere oportuno. Se desplegará una ventana emergente con los estados.

ESCUELA	CURSO	DOCENTE	SEMESTRE	ESTADO	ACCIONES
INGENIERIA DE SISTEMAS	ANÁLISIS Y DISEÑO DE SISTEMAS	JORGE MARTINEZ MUÑOZ	V	APROBADO	
INGENIERIA DE SISTEMAS	ANÁLISIS Y DISEÑO DE SISTEMAS	JOSÉ ALFREDO SULLA TORRES	V	APROBADO	
INGENIERIA DE SISTEMAS	ANÁLISIS Y DISEÑO DE SISTEMAS	KARINA ROSAS	V	APROBADO	
INGENIERIA DE SISTEMAS	CÁLCULO DIFERENCIAL	ÁLVARO MOSCOSO	III	ABIERTO	
INGENIERIA DE SISTEMAS	CÁLCULO DIFERENCIAL	FREDY QUISPE	III	ABIERTO	
INGENIERIA DE SISTEMAS	CÁLCULO INTEGRAL	FREDY QUISPE	III	ABIERTO	
INGENIERIA DE SISTEMAS	COMPUTACIÓN EN RED II	KARINA ROSAS	V	APROBADO	
INGENIERIA DE SISTEMAS	DESARROLLO I	MARIA GRACIA AYALA	I	ABIERTO	
INGENIERIA DE SISTEMAS	DESARROLLO II	APAZA CORRALES	II	ABIERTO	
INGENIERIA DE SISTEMAS	DIBUJO ARQUITECTÓNICO	JOSE SULLA	II	EN PROCESO	
INGENIERIA DE SISTEMAS	DISEÑO DE SOFTWARE A BAJO NIVEL	JOSE SULLA	V	EN PROCESO	
INGENIERIA DE SISTEMAS	DISEÑO DE SOFTWARE A BAJO NIVEL	KARINA ROSAS	V	EN PROCESO	
INGENIERIA DE SISTEMAS	ESTRUCTURAS DISCRETAS	FREDY QUISPE	II	ABIERTO	
INGENIERIA DE SISTEMAS	ESTRUCTURAS DISCRETAS	GUILLERMO CALDERON	II	ABIERTO	
INGENIERIA DE SISTEMAS	INTRODUCCION A LA INGENIERIA DE SISTEMAS	GUILLERMO CALDERON	I	ABIERTO	



Actualización de estado – Pantalla Habilitar Sílabos

3.8. HU0503 Pantalla Habilitar sílabos

El rol Jefe de Departamento seleccionará el sílabo a actualizar. En caso de que al momento de revisar el sílabo encuentre algún punto para corregir, podrá devolver al estado “En Proceso”, que permitirá al docente subsanar los pendientes.

ESCUELA	CURSO	DOCENTE	SEMESTRE	ESTADO	ACCIONES
INGENIERIA DE SISTEMAS	ANÁLISIS Y DISEÑO DE SISTEMAS	JORGE MARTINEZ MUÑOZ	V	APROBADO	
INGENIERIA DE SISTEMAS	ANÁLISIS Y DISEÑO DE SISTEMAS	JOSÉ ALFREDO SULLA TORRES	V	APROBADO	
INGENIERIA DE SISTEMAS	ANÁLISIS Y DISEÑO DE SISTEMAS	KARINA ROSAS	V	APROBADO	
INGENIERIA DE SISTEMAS	CÁLCULO DIFERENCIAL	ÁLVARO MOSCOSO	III	ABIERTO	
INGENIERIA DE SISTEMAS	CÁLCULO DIFERENCIAL	FREDY QUISPE	III	ABIERTO	
INGENIERIA DE SISTEMAS	CÁLCULO INTEGRAL	FREDY QUISPE	III	ABIERTO	
INGENIERIA DE SISTEMAS	COMPUTACIÓN EN RED II	KARINA ROSAS	V	APROBADO	
INGENIERIA DE SISTEMAS	DESARROLLO I	MARIA GRACIA AYALA	I	ABIERTO	
INGENIERIA DE SISTEMAS	DESARROLLO II	APAZA CORRALES	II	ABIERTO	
INGENIERIA DE SISTEMAS	DIBUJO ARQUITECTÓNICO	JOSE SULLA	II	EN PROCESO	
INGENIERIA DE SISTEMAS	DISEÑO DE SOFTWARE A BAJO NIVEL	JOSE SULLA	V	EN PROCESO	
INGENIERIA DE SISTEMAS	DISEÑO DE SOFTWARE A BAJO NIVEL	KARINA ROSAS	V	EN PROCESO	
INGENIERIA DE SISTEMAS	ESTRUCTURAS DISCRETAS	FREDY QUISPE	II	ABIERTO	
INGENIERIA DE SISTEMAS	ESTRUCTURAS DISCRETAS	GUILLERMO CALDERON	II	ABIERTO	
INGENIERIA DE SISTEMAS	INTRODUCCION A LA INGENIERIA DE SISTEMAS	GUILLERMO CALDERON	I	ABIERTO	

Modificación de estado ✕

APROBADO ▼

Cancelar Guardar



Actualización de estado – Pantalla Habilitar Sílabos

Si el sílabo se encuentra correctamente desarrollado, entonces el Jefe de Departamento puede cerrar el sílabo en señal de conformidad.

ESCUELA	CURSO	DOCENTE	SEMESTRE	ESTADO	ACCIONES
INGENIERIA DE SISTEMAS	ANÁLISIS Y DISEÑO DE SISTEMAS	JORGE MARTINEZ MUÑOZ	V	APROBADO	✍
INGENIERIA DE SISTEMAS	ANÁLISIS Y DISEÑO DE SISTEMAS	JOSÉ ALFREDO SULLA TORRES	V	APROBADO	✍
INGENIERIA DE SISTEMAS	ANÁLISIS Y DISEÑO DE SISTEMAS	KARINA ROSAS	V	APROBADO	✍
INGENIERIA DE SISTEMAS	CÁLCULO DIFERENCIAL	ÁLVARO MOSCOSO	III	ABIERTO	✍
INGENIERIA DE SISTEMAS	CÁLCULO DIFERENCIAL	FREDY QUISPE	III	ABIERTO	✍
INGENIERIA DE SISTEMAS	CÁLCULO INTEGRAL	FREDY QUISPE	III	ABIERTO	✍
INGENIERIA DE SISTEMAS			V	APROBADO	✍
INGENIERIA DE SISTEMAS			I	ABIERTO	✍
INGENIERIA DE SISTEMAS			II	ABIERTO	✍
INGENIERIA DE SISTEMAS			II	EN PROCESO	✍
INGENIERIA DE SISTEMAS	DISEÑO DE SOFTWARE A BAJO NIVEL	JOSE SULLA	V	EN PROCESO	✍
INGENIERIA DE SISTEMAS	DISEÑO DE SOFTWARE A BAJO NIVEL	KARINA ROSAS	V	EN PROCESO	✍
INGENIERIA DE SISTEMAS	ESTRUCTURAS DISCRETAS	FREDY QUISPE	II	ABIERTO	✍
INGENIERIA DE SISTEMAS	ESTRUCTURAS DISCRETAS	GUILLERMO CALDERON	II	ABIERTO	✍
INGENIERIA DE SISTEMAS	INTRODUCCION A LA INGENIERIA DE SISTEMAS	GUILLERMO CALDERON	I	ABIERTO	✍

Actualización de estado – Pantalla Habilitar Sílabos

Después de seleccionar el estado, se procede a “Guardar” los cambios.

ESCUELA	CURSO	DOCENTE	SEMESTRE	ESTADO	ACCIONES
INGENIERIA DE SISTEMAS	ANÁLISIS Y DISEÑO DE SISTEMAS	JORGE MARTINEZ MUÑOZ	V	APROBADO	✍
INGENIERIA DE SISTEMAS	ANÁLISIS Y DISEÑO DE SISTEMAS	JOSÉ ALFREDO SULLA TORRES	V	APROBADO	✍
INGENIERIA DE SISTEMAS	ANÁLISIS Y DISEÑO DE SISTEMAS	KARINA ROSAS	V	APROBADO	✍
INGENIERIA DE SISTEMAS	CÁLCULO DIFERENCIAL	ÁLVARO MOSCOSO	III	ABIERTO	✍
INGENIERIA DE SISTEMAS	CÁLCULO DIFERENCIAL	FREDY QUISPE	III	ABIERTO	✍
INGENIERIA DE SISTEMAS	CÁLCULO INTEGRAL	FREDY QUISPE	III	ABIERTO	✍
INGENIERIA DE SISTEMAS			V	APROBADO	✍
INGENIERIA DE SISTEMAS			I	ABIERTO	✍
INGENIERIA DE SISTEMAS			II	ABIERTO	✍
INGENIERIA DE SISTEMAS			II	EN PROCESO	✍
INGENIERIA DE SISTEMAS	DISEÑO DE SOFTWARE A BAJO NIVEL	JOSE SULLA	V	EN PROCESO	✍
INGENIERIA DE SISTEMAS	DISEÑO DE SOFTWARE A BAJO NIVEL	KARINA ROSAS	V	EN PROCESO	✍
INGENIERIA DE SISTEMAS	ESTRUCTURAS DISCRETAS	FREDY QUISPE	II	ABIERTO	✍
INGENIERIA DE SISTEMAS	ESTRUCTURAS DISCRETAS	GUILLERMO CALDERON	II	ABIERTO	✍
INGENIERIA DE SISTEMAS	INTRODUCCION A LA INGENIERIA DE SISTEMAS	GUILLERMO CALDERON	I	ABIERTO	✍

Actualización de estado – Pantalla Habilitar Sílabos

Visualizar el cambio de estado, y la confirmación por un mensaje emergente.

Semestre

ESCUELA	CURSO	DOCENTE	SEMESTRE	ESTADO	ACCIONES
INGENIERIA DE SISTEMAS	ANÁLISIS Y DISEÑO DE SISTEMAS	JORGE MARTINEZ MUÑOZ	V	CERRADO	<input type="checkbox"/>
INGENIERIA DE SISTEMAS	ANÁLISIS Y DISEÑO DE SISTEMAS	JOSÉ ALFREDO SULLA TORRES	V	CERRADO	<input type="checkbox"/>
INGENIERIA DE SISTEMAS	ANÁLISIS Y DISEÑO DE SISTEMAS	KARINA ROSAS	V	CERRADO	<input type="checkbox"/>
INGENIERIA DE SISTEMAS	CÁLCULO DIFERENCIAL	ÁLVARO MOSCOSO	III	ABIERTO	<input type="checkbox"/>
INGENIERIA DE SISTEMAS	CÁLCULO DIFERENCIAL	FREDY QUISPE	III	ABIERTO	<input type="checkbox"/>
INGENIERIA DE SISTEMAS	CÁLCULO INTEGRAL	FREDY QUISPE	III	ABIERTO	<input type="checkbox"/>
INGENIERIA DE SISTEMAS	COMPUTACIÓN EN RED II	KARINA ROSAS	V	APROBADO	<input type="checkbox"/>
INGENIERIA DE SISTEMAS	DESARROLLO I	MARIA GRACIA AYALA	I	ABIERTO	<input type="checkbox"/>
INGENIERIA DE SISTEMAS	DESARROLLO II	APAZA CORRALES	II	ABIERTO	<input type="checkbox"/>
INGENIERIA DE SISTEMAS	DIBUJO ARQUITECTÓNICO	JOSE SULLA	II	EN PROCESO	<input type="checkbox"/>
INGENIERIA DE SISTEMAS	DISEÑO DE SOFTWARE A BAJO NIVEL	JOSE SULLA	V	EN PROCESO	<input type="checkbox"/>
INGENIERIA DE SISTEMAS	DISEÑO DE SOFTWARE A BAJO NIVEL	KARINA ROSAS	V	EN PROCESO	<input type="checkbox"/>
INGENIERIA DE SISTEMAS	ESTRUCTURAS DISCRETAS	FREDY QUISPE	II	ABIERTO	<input type="checkbox"/>

Actualización de estado – Pantalla Habilitar Sílabos

3.9. HU0201 Pantalla Carga de Archivos

El rol Administrador es el responsable de realizar la carga de archivos.

El sistema al iniciar el semestre debe importar la siguiente información: Plan de estudios, Carga Académica, Docentes y Bibliografía de forma automática, y la siguiente pantalla permitirá visualizar esta importación.

También permitirá visualizar esta importación y/o realizar de nuevo esta importación.

Se debe adaptar los procesos a la entidad de estudios superiores para obtener la información.

CARGA	ESTADO	
Plan de Estudios	●	⊕ ⊖
Carga Académica	●	⊕ ⊖
Docentes	●	⊕ ⊖

Carga de información para los sílabos

3.10. HU0601 Pantalla Seleccionar Curso

El rol Docente tiene acceso a la pantalla principal para seleccionar el sílabo con el que trabajará.

Sera la pantalla principal que se mostrará al ingresar al sistema, después de seleccionar la asignatura, puede iniciar su desarrollo.

CURSO	ESCUELA	SEMESTRE	TOTAL HORAS	EDITAR
DISEÑO DE SOFTWARE A BAJO NIVEL	INGENIERIA DE SISTEMAS	V	120	✍
COMPUTACIÓN EN RED II	INGENIERIA DE SISTEMAS	V	108	✍
ANÁLISIS Y DISEÑO DE SISTEMAS	INGENIERIA DE SISTEMAS	V	108	✍
TECNOLOGÍAS MÓVILES	INGENIERIA DE SISTEMAS	VII	108	✍

Asignaturas asignadas al docente

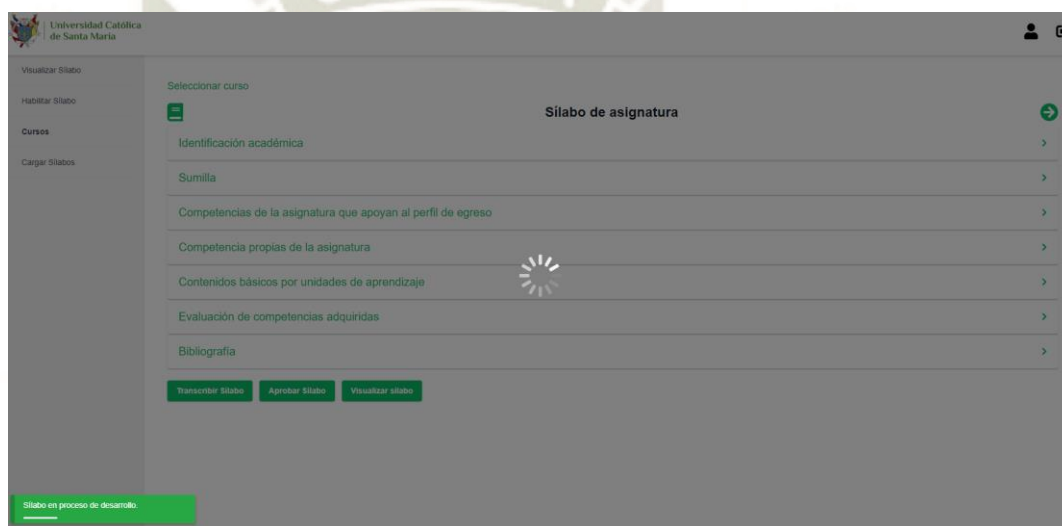
3.11. HU0601 Pantalla Silabo del curso

El rol Docente tiene acceso a la pantalla para la elaboración del sílabo.

En el área de trabajo, Se puede observar la esquina superior la opción de seleccionar cursos a continuación se encuentra un libro que representa la bitácora del sílabo, en el medio un título informativo del curso a desarrollar el sílabo, y en el lado derecho una flecha que representara la opción de ir al programa formativo.

En el medio del área de trabajo se mostrará el silabo dividido por desglosables.

En la parte inferior siempre se mostrará las opciones de transcribir silabo anterior, aprobar silabo y visualizar silabo.



Gestión del sílabo – Carga de información

Después de cargar la información, se habilitan las opciones que dispone la asignatura en el momento de su desarrollo.



Gestión del sílabo – Desarrollo

Transcripción de sílabo

Al presionar el botón “Transcribir Sílabo”, se busca el Sílabo anterior más cercano, el cual se mostrará en la siguiente pantalla en formato PDF y que reemplazará el contenido correspondiente, salvo los datos elementales que se encuentran en información académica, debido a que eso es inalterable.



Gestión del sílabo - Transcripción

La ventana emergente, mostrará el sílabo anterior. Si el o los docentes consideran adecuado utilizar la información anterior como base para el nuevo sílabo, se puede proceder con “Reemplazar”. Si no se considera adecuada, se puede “Cancelar” la acción.

UNIVERSIDAD CATÓLICA DE SANTA MARÍA

PROGRAMA FORMATIVO DE ASIGNATURA

I. IDENTIFICACIÓN ACADÉMICA

- Facultad: FACULTAD DE CIENCIAS FÍSICAS Y FORMALES
- Departamento Académico: JEFATURA DE LA FACULTAD DE CIENCIAS FÍSICAS Y FORMALES
- Nombre de la Asignatura: ANÁLISIS Y DISEÑO DE SISTEMAS
Código: 7155027
- ESCUELA PROFESIONAL, donde se desarrolla la asignatura: INGENIERÍA DE SISTEMAS
- Docente(s) y /o Jefe(s) de Práctica(s)

Código	Apellidos y Nombres	Función	Categoría
1	ROSAS, KARINA	JEFE DE PRÁCTICA	PRINCIPAL
1635	SULLA TORRES, JOSÉ ALFREDO	DOCENTE	PRINCIPAL

6. Ubicación y Peso Académico de la Asignatura

AÑO ACADÉMICO	SEMESTRE	CRÉDITOS	HORAS SEMANALES				HORAS SEMESTRALES		
			Horas Teóricas	Horas Prácticas de Aula	Horas Prácticas de Laboratorio	Horas Visuales	Horas Teóricas	Horas Prácticas	Horas Visuales
2020	V	4	2	2	2	0	36	72	0

7. Ambiente donde se realiza el aprendizaje

Cancelar Reemplazar

Gestión del sílabo - Reemplazar

Al culminar la transcripción se podrá visualizar la información registrada en el sílabo a desarrollar.

UNIVERSIDAD CATÓLICA DE SANTA MARÍA

Selecionar curso

Sílabo de asignatura

- Identificación académica
- Sumaria
- Competencias de la asignatura que apoyan al perfil de egreso
- Competencia propias de la asignatura
- Contenidos básicos por unidades de aprendizaje
- Evaluación de competencias adquiridas
- Bibliografía

Transcribir sílabo Aprobar sílabo Visualizar sílabo

Se completó correctamente la transcripción

Gestión del sílabo – Transcripción completada

Seleccionar “Visualizar Silabo”, para observar el contenido.

Visualizar Silabo

Habilitar Silabo

Cursos

Cargar Silabos

icd79g9MSkPLRMWDSJLJOAyzluMMEZpIeTqESRGicQooXhCHiWGiBQA... 1 / 9 | 100% +

FACULTAD: FACULTAD DE CIENCIAS FÍSICAS Y FORMALES
ESCUELA PROFESIONAL: INGENIERIA DE SISTEMAS
**PLAN DE ESTUDIOS
SILABO DE ASIGNATURA**

1. IDENTIFICACIÓN ACADÉMICA
1.1.- Nombre de la Asignatura: ANÁLISIS Y DISEÑO DE SISTEMAS
Código de la Asignatura: 7105037
Semestre Académico en que se desarrolla: V
1.2.- Peso Académico de la Asignatura:

CRÉDITOS	HORAS SEMANALES			HORAS SEMESTRALES			
	Horas Teóricas	Horas Prácticas Práctica de Aula	Jefe de Prácticas	Horas Virtuales	Horas Teóricas	Horas Prácticas	Horas Virtuales
4	2	2	2	0	36	72	0

1.3.- Código, nombre y créditos de Asignaturas Equivalentes:
7106037 Análisis y Diseño de Sistemas
7107188 Laboratorio de Análisis y Diseño de Sistemas
1.4.- Código y nombre de Asignaturas Pre-requisito:

2. SUMILLA
La asignatura de Análisis y Diseño de Sistemas pertenece al área de Formación Básica Tecnológica; es teórico-práctica. Su propósito es el de proporcionar al estudiante los conocimientos para conceptualizar, comprender, discutir, analizar y manejar adecuadamente las técnicas, procedimientos, herramientas, metodologías para el análisis y diseño de sistemas de información en las instituciones; adquiriendo las habilidades y la formación de criterio propios para plantear soluciones a través del análisis y diseño; valorando su importancia para las instituciones.

3. COMPETENCIAS DEL PERFIL DE EGRESO A LAS QUE CONTRIBUYE ASIGNATURA
001. CONOCIMIENTOS DE INGENIERÍA
Capacidad de aplicar conocimientos de matemáticas, ciencias e ingeniería en la solución de problemas complejos de ingeniería.
003. COMUNICACIÓN

Gestión del sílabo – Visualizar Sílabo

Aprobar Sílabo

Al “Aprobar Sílabo”, el docente solicita que se haga la revisión de la asignatura. Está solicitud entiende que ha desarrollado e ingresado el contenido pertinente para el curso.

Visualizar Silabo

Habilitar Silabo

Cursos

Cargar Silabos

Seleccionar curso

Silabo de asignatura

- Identificación académica
- Sumilla
- Competencias de la asignatura que apoyan al perfil de egreso
- Competencia propias de la asignatura
- Contenidos básicos por unidades de aprendizaje
- Evaluación de competencias adquiridas
- Bibliografía

Transcribir Silabo | **Aprobar Silabo** | Visualizar silabo

Gestión del sílabo – Aprobar Sílabo

Si aprobamos el sílabo, el sílabo cambiará de estado; por lo cual cambiará las opciones en la parte inferior a Editar y visualizar sílabo.

Gestión del sílabo – Aprobar Sílabo



Bitácora

Al presionar el botón bitácora nos mostrar el siguiente modal en la cual muestra las acciones realizadas por los docentes.

Gestión del sílabo – Bitácora

Se permite la búsqueda por docente y concepto que se modificó.

USUARIO	FECHA	ACCIÓN	COMENTARIO
SULLA TORRES, JOSÉ ALFREDO	10/01/2022	PROYECTOS	PROYECTOS INGRESADOS POR TRANSCRIPCIÓN
SULLA TORRES, JOSÉ ALFREDO	10/01/2022	BIBLIOGRAFÍA	BIBLIOGRAFÍA INGRESADA POR TRANSCRIPCIÓN
SULLA TORRES, JOSÉ ALFREDO	10/01/2022	INSTRUMENTOS	INTRUMENTOS REGISTRADOS POR TRANSCRIPCIÓN
SULLA TORRES, JOSÉ ALFREDO	10/01/2022	CRITERIOS	CRITERIOS REGISTRADOS POR TRANSCRIPCIÓN
SULLA TORRES, JOSÉ ALFREDO	10/01/2022	ESTRATEGIAS	ESTRATEGIAS REGISTRADAS POR TRANSCRIPCIÓN
SULLA TORRES, JOSÉ ALFREDO	10/01/2022	RESULTADOS	RESULTADOS REGISTRADOS POR TRANSCRIPCIÓN
SULLA TORRES, JOSÉ ALFREDO	10/01/2022	UNIDADES LAB	UNIDADES DE LABORATORIO REGISTRADAS POR TRANSCRIPCIÓN
SULLA TORRES, JOSÉ ALFREDO	10/01/2022	UNIDADES TEÓRICAS	UNIDADES TEÓRICAS REGISTRADAS POR TRANSCRIPCIÓN
SULLA TORRES, JOSÉ ALFREDO	10/01/2022	COMPETENCIAS PROPIAS DE LA ASIGNATURA	COMPETENCIAS PROPIAS DE LA ASIGNATURA REGISTRADAS POR TRANSCRIPCIÓN
SULLA TORRES, JOSÉ ALFREDO	10/01/2022	COMPETENCIAS POR PERFIL DE EGRESO	COMPETENCIAS POR PERFIL DE EGRESO REGISTRADAS POR TRANSCRIPCIÓN
SULLA TORRES, JOSÉ ALFREDO	10/01/2022	TÍTULOS DE FASE	TÍTULOS DE FASE AGREGADOS POR TRANSCRIPCIÓN
SULLA TORRES, JOSÉ ALFREDO	10/01/2022	SUMILLA	SUMILLA ACTUALIZADA
SULLA TORRES, JOSÉ ALFREDO	08/01/2022	PROYECTOS	PROYECTOS INGRESADOS

Gestión del sílabo – Bitácora

Identificación académica

La división de identificación académica será solo informativa, toda la información será importada de la base de datos.



Identificación académica

Facultad FACULTAD DE CIENCIAS FÍSICAS Y FORMALES
Departamento JEFATURA DE LA FACULTAD DE CIENCIAS FÍSICAS Y FORMALES
Nombre de la asignatura ANÁLISIS Y DISEÑO DE SISTEMAS
Código 7105037
Escuela profesional INGENIERIA DE SISTEMAS

Ubicación y peso académico de la asignatura

AÑO ACADÉMICO	SEMESTRE	CRÉDITOS	HORAS TEÓRICAS	HORAS SEMANALES			HORAS SEMESTRALES		
				PRÁCTICA DE AULA	JEFE DE PRÁCTICAS	HORAS VIRTUALES	HORAS TEÓRICAS	HORAS PRÁCTICAS	HORAS VIRTUALES
2021	V	4	2	2	2	0	36	72	0

Ambiente donde se realiza el aprendizaje
Teoría MS Teams
Práctica MS Teams

Identificación académica

Sumilla

La división de sumilla es editable por lo cual se podrá tener el área de ingreso de sumilla por parte del docente. Al final el ingreso se procede a guardar la información.

Sumilla

Competencias de la asignatura que apoyan al perfil de egreso

Al mostrar la división de competencias deberá traer las competencias por defecto que están destinadas a la escuela profesional. Estas competencias se encuentran definidas por el Centro de Desarrollo Académico y establecidas para el uso en las diferentes asignaturas de la universidad.

Competencias que apoyan al perfil de egreso

Competencias propias de la asignatura

Competencias ingresadas por el docente, definidas para complementar lo desarrollado en la asignatura.

Cada fase tendrá una sección para poder ingresar las competencias que se formulen al momento de seleccionar “Agregar competencia”, se creará un registro donde se podrá redactar la competencia. Al terminar, se guarda la competencia elaborada.

En caso de tener más de una competencia por fase, se puede definir el orden utilizando las flechas del lado derecho, para ascender o descender la competencia.

Al momento de guardar las competencias por fase, se ingresará la información en el orden que se definió.

Si se desea eliminar el registro, se tiene el botón de eliminar.

Competencia propias de la asignatura

Fase 1

COMPETENCIA

Efectúa, explica y desarrolla los conceptos sobre la administración del desarrollo de sistemas de información, así como del análisis de sistemas a partir en primer lugar del análisis estructurado.

↑ ↓ 🗑️

Agregar competencia **Guardar**

Fase 2

COMPETENCIA

Establece, explica y exhibe las metodologías para el análisis de sistemas orientado a objetos y por prototipos como alternativas viables.

↑ ↓ 🗑️

Agregar competencia **Guardar**

Fase 3

COMPETENCIA

Diseña, elabora, propone el diseño desde el punto de vista de entradas, salidas y otros del módulo o sistema de información para su implementación.

↑ ↓ 🗑️

Agregar competencia **Guardar**

Competencias propias de la asignatura

Contenidos básicos por unidades de aprendizaje

Se ingresa en esta sección el contenido de temas a tratar en la fase. Estará dividido en dos grandes secciones: Unidades clases teóricas y Unidades Laboratorio.

A su vez, contarán con el ingreso de la información por fase. Esto permitirá al docente tener ordenada las unidades teóricas y de laboratorio.

La numeración de las unidades será de manera automática, según la posición de los registros. Esta numeración es referencial para tener un orden en el contenido ingresado.

Las unidades clases teóricas, corresponden a los temas a tratar por los docentes en la asignatura en la parte teórica.

Al “Agregar Unidad”, se creará una sección para ingresar el tema general y albergar subunidades en que se dividirá el tema. El botón con el símbolo “+”, permite ingresar un registro dentro de la unidad agregada o “SubUnidades”. Así será de manera incremental la inserción de contenidos o subunidades.

Las subunidades pueden reordenarse con las fechas en la parte derecha, que permitirán ascender o descender el concepto. También se cuenta con la posibilidad de eliminar el registro.

SILABOS CATÓLICA

Visualizar Silabo

Habilitar Silabo

Cursos

Identificación académica

Sumilla

Competencias de la asignatura que apoyan al perfil de egreso

Competencia propias de la asignatura

Contenidos básicos por unidades de aprendizaje

Unidades clases teóricas

Fase 1

FASE	UNIDAD	SUB-UNIDAD	DESCRIPCIÓN	
1	1	0	SWITCHING	↑ ↓ 🗑
1	1	1	Configuración básica de dispositivos	↑ ↓ 🗑
1	1	2	Conceptos de conmutación	↑ ↓ 🗑
1	1	3	VLAN	↑ ↓ 🗑
1	1	4	Enrutamiento entre VLAN	↑ ↓ 🗑
1	1	5	STP	↑ ↓ 🗑
1	1	6	EtherChannel	↑ ↓ 🗑 +

Agregar unidad

Unidades clases teóricas Fase 1

Para las demás fases se procede de igual manera la inserción de las unidades. Al terminar de elaborar las unidades, se selecciona “Guardar” la información.

FASE	UNIDAD	SUB-UNIDAD	DESCRIPCIÓN	
2	2	0	DHCP, FHRP y SEGURIDAD LAN	↑ ↓ 🗑
2	2	1	DHCPv4	↑ ↓ 🗑
2	2	2	Conceptos SLAAC y DHCPv6	↑ ↓ 🗑
2	2	3	Conceptos de FHRP	↑ ↓ 🗑
2	2	4	Conceptos de seguridad de LAN	↑ ↓ 🗑
2	2	5	Configuración de seguridad del switch	↑ ↓ 🗑 +

Agregar unidad

FASE	UNIDAD	SUB-UNIDAD	DESCRIPCIÓN	
3	3	0	WLAN Y ENRUTAMIENTO	↑ ↓ 🗑
3	3	1	Conceptos de WLAN	↑ ↓ 🗑
3	3	2	Configuración de WLAN	↑ ↓ 🗑
3	3	3	Conceptos de enrutamiento	↑ ↓ 🗑
3	3	4	Rutas IP estáticas	↑ ↓ 🗑
3	3	5	Resolución de problemas de rutas estáticas y predeterminadas	↑ ↓ 🗑 +

Agregar unidad

Guardar

Unidades clases teóricas Fase 2 y 3

Las Unidades Laboratorio, tienen un tratamiento similar, solo que en esta sección existe solo el concepto de unidad, más no de subunidad.

Se colocan en este apartado lo correspondiente a elementos de prácticas, laboratorios, etc. que se tratarán en la asignatura para cada fase. Lo ingresado se mostrará en el programa formativo después de las unidades teóricas elaboradas.

Unidades laboratorio

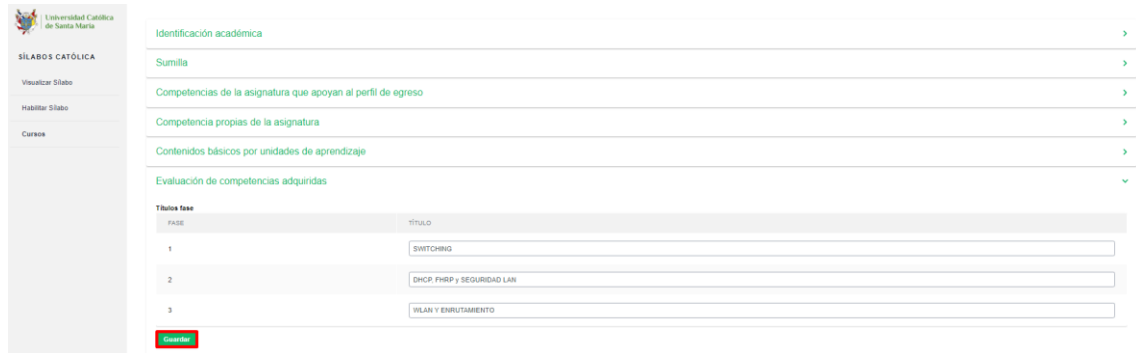
Fase 1			
FASE	UNIDAD	DESCRIPCIÓN	
1	1	Práctica 1: Repaso de configuración de routers y switches.	↑ ↓ 🗑
1	2	Práctica 2: Configuración de VLAN, enrutamiento inter Vlan.	↑ ↓ 🗑
1	3	Práctica 3: Configuración STP y Etherchannel.	↑ ↓ 🗑
1	4	Práctica 4: Habilidades de configuración.	↑ ↓ 🗑
Agregar unidad			
Fase 2			
FASE	UNIDAD	DESCRIPCIÓN	
2	5	Práctica 4: Configuración servidor y clientes DHCP con IPv4 e IPv6.	↑ ↓
2	6	Práctica 5: Configuración de FHRP, seguridad del switch.	↑ ↓
2	7	Práctica 6: Habilidades de configuración.	↑ ↓
Agregar unidad			
Fase 3			
FASE	UNIDAD	DESCRIPCIÓN	
3	8	Práctica 7: Configuración redes WLAN.	↑ ↓
3	9	Práctica 8: Configuración de enrutamiento IP estático.	↑ ↓
3	10	Práctica 9: Revisión caso de estudio.	↑ ↓
Agregar unidad			
Guardar			

Unidades laboratorio

Evaluación de Competencias Adquiridas

En la división de evaluación se editará toda la información de programa formativo como: Títulos de la fase, contenido de los resultados de aprendizaje, criterios de evaluación, estrategias de aprendizaje-enseñanza e instrumentos de evaluación.

Cada fase tendrá la posibilidad de ingresar un título que figurará en el programa formativo.



FASE	TÍTULO
1	SWITCHING
2	DHCP, FHRP y SEGURIDAD LAN
3	VLAN Y ENRUTAMIENTO

Títulos de las fases

Los resultados de cada fase también se ingresan de acuerdo a si se trata de: conocimiento, desempeño y producto.

De igual manera se puede ingresar “+” más de un registro por resultado en fase; en ese caso, se puede ordenar la visualización, así como eliminar el registro.

Los resultados, en la columna de descripción se coloca lo correspondiente al texto mostrado en el programa formativo. Es obligatorio ingresarlo.

En la columna síntesis, se ingresará lo que se visualizará en la sección de evaluación de competencias adquiridas del plan de estudios del sílabo cuando se exporte en formato PDF.

Las síntesis no son obligatorias y se concatenarán las que sean ingresadas.

El peso teórico y peso práctico indicarán el porcentaje que representan sobre la asignatura.

Resultados de fase

Fase 1

FASE	RESULTADO	DESCRIPCIÓN	SÍNTESIS	PESO TEÓRICO	PESO PRÁCTICO	
1	De conocimiento	Conoce la terminología del proceso de routing y switching, sus funciones y usos		33.3	0	↑ ↓ 🗑️ +
1	De desempeño	Configura adecuadamente los dispositivos de switching, para su gestión en las soluciones del diseño de redes empresariales.		33.4	0	↑ ↓ 🗑️ +
1	De producto	Propone y evalúa alternativas de conmutación con switches.		0	33.3	↑ ↓ 🗑️ +

Fase 2

[Agregar tabla de conocimiento](#) [Agregar tabla de desempeño](#) [Agregar tabla de producto](#)

Fase 3

[Agregar tabla de conocimiento](#) [Agregar tabla de desempeño](#) [Agregar tabla de producto](#)

[Guardar](#)

Resultados de fase

Biblioteca

La división de bibliografía mostrará los libros que utilizará el curso y estén en la base de datos de la UCSM.

Se procede a “Agregar bibliografía” por cada fase del semestre académico.

Bibliografía

Fase 1

AUTOR	NOMBRE	EDITORIAL	LUGAR DE PUBLICACIÓN	AÑO DE PUBLICACIÓN	
No hay datos					

[Agregar bibliografía](#)

Fase 2

AUTOR	NOMBRE	EDITORIAL	LUGAR DE PUBLICACIÓN	AÑO DE PUBLICACIÓN	
CISCO SYSTEMS	Fundamentos de conmutación, enrutamiento y comunicaciones inalámbricas	CISCO	USA	2020	↑ ↓ 🗑️
ROSAS KARINA LAURA L.	Matemática de Redes	Ed. Universidad Católica de Santa María	Arequipa, Perú	2016	↑ ↓ 🗑️

[Agregar bibliografía](#)

Fase 3

AUTOR	NOMBRE	EDITORIAL	LUGAR DE PUBLICACIÓN	AÑO DE PUBLICACIÓN	
No hay datos					

[Agregar bibliografía](#)

[Guardar](#)

Bibliografía del Sílabo por Fase

Tenemos las opciones o filtros para poder realizar la búsqueda de la bibliografía.

Búsqueda de Bibliografía

Seleccionar bibliografía y “Añadir”.

	AUTOR	NOMBRE	EDITORIAL	LUGAR DE PUBLICACIÓN
<input checked="" type="checkbox"/>	CISCO SYSTEMS	Fundamentos de conmutación, enrutamiento y comunicaciones inalámbricas	CISCO	USA
<input checked="" type="checkbox"/>	ROSAS KARINA LAURAL	Matemática de Redes	Ed. Universidad Católica de Santa María	Arequipa,
<input type="checkbox"/>	ARIGANELLO ERNESTO	"Redes Cisco guía de estudio para la certificación CCNA Routing and Switching	Edición 4	Colombia
<input type="checkbox"/>	Prueba	Fundamentos prueba	Editorial Prueba	Lima
<input type="checkbox"/>	Alan Dennis, Barbara Wixom, Roberta Roth	Systems Analysis and Design	Wiley	USA
<input type="checkbox"/>	Kendall Kenneth, Kendall Julie	Análisis y Diseño de Sistemas	Pearson	USA
<input type="checkbox"/>	Báez Pérez Carmen Inés-Suárez Zsarabamda, Martha Isabel	Proceso de desarrollo de Software Versión 1.0 Basado en la Articulación de RUP y CCMi priorizando su calidad	Pearson	USA
<input type="checkbox"/>	RosenBlatt	System Analysis and Design	Cengage	USA

Seleccionar Bibliografía

Después de agregar la bibliografía a cada una de las fases, “Guardar” la información.

Bibliografía

Fase 1

AUTOR	NOMBRE	EDITORIAL	LUGAR DE PUBLICACIÓN	AÑO DE PUBLICACIÓN	
CISCO SYSTEMS	Fundamentos de conmutación, enrutamiento y comunicaciones inalámbricas	CISCO	USA	2020	↑ ↓ 🗑️
ROSAS KARINA LAURAL	Matemática de Redes	Ed. Universidad Católica de Santa María	Arequipa, Perú	2016	↑ ↓ 🗑️

Agregar bibliografía

Fase 2

AUTOR	NOMBRE	EDITORIAL	LUGAR DE PUBLICACIÓN	AÑO DE PUBLICACIÓN	
CISCO SYSTEMS	Fundamentos de conmutación, enrutamiento y comunicaciones inalámbricas	CISCO	USA	2020	↑ ↓ 🗑️
ROSAS KARINA LAURAL	Matemática de Redes	Ed. Universidad Católica de Santa María	Arequipa, Perú	2016	↑ ↓ 🗑️

Agregar bibliografía

Fase 3

AUTOR	NOMBRE	EDITORIAL	LUGAR DE PUBLICACIÓN	AÑO DE PUBLICACIÓN	
No hay datos					

Agregar bibliografía

Guardar

Guardar Bibliografía

3.12. HU0619 Pantalla de Programa Formativo

El rol Docente es el responsable de la elaboración del programa formativo.

Programa Formativo

Las divisiones del programa formativo son de lectura a excepción de matriz en donde se ingresan los proyectos.



Programa Formativo - Divisiones

La información desarrollada en el plan de estudios se podrá apreciar más no modificar en este apartado.

Identificación Académica

La identificación académica, mostrará además de lo visto en el plan de estudio, la lista de los docentes y ambientes donde se desarrollará la asignatura.

Programa formativo

Identificación académica

Facultad: FACULTAD DE CIENCIAS FÍSICAS Y FORMALES
 Departamento: JEFATURA DE LA FACULTAD DE CIENCIAS FÍSICAS Y FORMALES
 Nombre de la asignatura: ANÁLISIS Y DISEÑO DE SISTEMAS
 Código: 7105037
 Escuela profesional: INGENIERIA DE SISTEMAS

Docentes

CÓDIGO	APELLIDOS Y NOMBRES	FUNCIÓN	CATEGORÍA
1	ROSAS, KARINA	JEFE DE PRÁCTICA	PRINCIPAL
1425	MARTINEZ MUÑOZ, JORGE	JEFE DE PRÁCTICA	INVITADO
1635	SULLATORRES, JOSÉ ALFREDO	DOCENTE	PRINCIPAL

Ubicación y peso académico de la asignatura

AÑO ACADÉMICO	SEMESTRE	CRÉDITOS	HORAS SEMANALES				HORAS SEMESTRALES		
			HORAS TEÓRICAS	HORAS PRÁCTICAS PRÁCTICA DE AULA	JEFE DE PRÁCTICAS	HORAS VIRTUALES	HORAS TEÓRICAS	HORAS PRÁCTICAS	HORAS VIRTUALES
2021	V	4	2	2	2	0	36	72	0

Ambiente donde se realiza el aprendizaje

Teoría: MS Teams
 Práctica: MS Teams

Programa Formativo – Identificación Académica

Sumilla

La sumilla se aprecia de modo informativo. No se puede realizar una modificación en esta sección.

Programa formativo

Sumilla

La asignatura de Análisis y Diseño de Sistemas pertenece al área de Formación Básica Tecnológica; es teórico - práctica. Su propósito es el de proporcionar al estudiante los conocimientos para conceptualizar, comprender, discutir, analizar y manejar adecuadamente las técnicas, procedimientos, herramientas, metodologías para el análisis y diseño de sistemas de información en las instituciones; adquiriendo las habilidades y la formación de criterios propios para plantear soluciones a través del análisis y diseño, valorando su importancia para las instituciones.

Competencias de la asignatura que apoyan al perfil de egreso

Competencia propias de la asignatura

Programación por fase de aprendizaje

Matriz

Aprobar Sílabo Visualizar sílabo

Programa Formativo - Sumilla

Competencias de la asignatura que apoyan al perfil de egreso

Solo las competencias que apoyan al perfil de egreso que se seleccionaron se mostrarán.

The screenshot shows the 'Programa formativo' section of a course syllabus. The 'Competencias de la asignatura que apoyan al perfil de egreso' section is highlighted with a red box. It lists three competencies:

- Capacidad de aplicar conocimientos de matemáticas, ciencias e ingeniería en la solución de problemas complejos de ingeniería.
- La capacidad de comunicarse eficazmente, mediante la comprensión y redacción de informes eficaces y documentación de diseño, la realización de exposiciones eficaces, y la transmisión y recepción de instrucciones claras.
- La capacidad de desenvolverse eficazmente como individuo, como miembro o líder en diversos equipos, y en entornos multidisciplinarios.

Programa Formativo – Competencias que apoyan al perfil de egreso

Competencias propias de la asignatura

Las competencias propias de la asignatura que se registraron para cada fase también aparecerán listadas.

The screenshot shows the 'Programa formativo' section of a course syllabus. The 'Competencia propias de la asignatura' section is highlighted with a red box. It lists three phases:

- Fase 1**
 - Efectúa, explica y desarrolla los conceptos sobre la administración del desarrollo de sistemas de información, así como del análisis de sistemas a partir en primer lugar del análisis estructurado.
- Fase 2**
 - Establece, explica y exhibe las metodologías para el análisis de sistemas orientado a objetos y por prototipos como alternativas viables.
- Fase 3**
 - Diseña, elabora, propone el diseño desde el punto de vista de entradas, salidas y otros del módulo o sistema de información para su implementación.

Programa Formativo – Competencias propias de la asignatura

Programación por fase de aprendizaje

Se puede observar la programación por fase de aprendizaje estructurada en la matriz que comprende la lista de cada concepto ingresado en el plan de estudio de manera detallada.

Se puede cambiar por medio de la elección entre fases.

Programación por fase de aprendizaje

FASE	TÍTULO DE FASE	TOTAL DE HORAS DE FASE	CRONOGRAMA DE LA FASE	DESDE	HASTA																																			
1	Administración del Desarrollo de sistemas de información	36		29 / 03 / 2021	06 / 05 / 2021																																			
COMPETENCIA • Explica, aplica y desarrolla los conceptos sobre la administración del desarrollo de sistemas de información, así como del análisis de sistemas a partir en primer lugar del análisis estructurado.																																								
UNIDADES DE COMPETENCIA <table border="1"> <thead> <tr> <th>TEMAS DE LA FASE</th> <th>ESTRATEGIAS DE APRENDIZAJE-ENSEÑANZA</th> <th colspan="3">EVALUACIÓN DE LOS APRENDIZAJES</th> </tr> <tr> <th></th> <th></th> <th>CRITERIOS</th> <th>INSTRUMENTOS DE EVALUACIÓN</th> <th>PORCENTAJE % TEORÍA PRÁCTICA</th> </tr> </thead> <tbody> <tr> <td> DE CONOCIMIENTO • Explica los conceptos sobre la administración del desarrollo de sistemas de información y el análisis estructurado </td> <td> 1ra. Unidad: Introducción al Análisis y Diseño de Sistemas. 1.1 Objetivos, problemas y problemática del análisis, responsabilidades, relación analista-usuario 1.2 Ciclo de desarrollo, organización, funciones </td> <td> • Clase Magistral • Lecturas </td> <td> • Preguntas objetivas y ensayo • Preguntas de ensayo </td> <td> • Prueba de entrada • Examen • Control de lectura 40 0 </td> </tr> <tr> <td> DE DESEMPEÑO • Obtiene los fundamentos del análisis estructurado </td> <td> 1.3 Ciclo de vida del SW, modelos 1.4 Metodologías de desarrollo de SW, características, clasificación 1.5 Análisis de Costos y Beneficios del proyecto 1.6 Preparación de la Propuesta de Sistemas 2da Unidad: Gestión de Proyectos 2.1 Identificación de Proyectos </td> <td> • Práctica individual y colectiva en laboratorio • Programación de Proyectos • Costo Beneficio • Flujograma de Información • Diccionario de Datos </td> <td> • Presentación de la práctica • Presentación grupal </td> <td> • Informe de práctica • Informe de trabajo en grupo 0 30 </td> </tr> <tr> <td> DE PRODUCTO • Valora el análisis como base del desarrollo de sistemas de información </td> <td> 2.2 Análisis de Factibilidad 2.3 Selección de Proyectos 2.4 Herramientas de gestión de proyectos 2.5 Creación y gestión del plan de trabajo </td> <td> • Dirección y orientación del docente </td> <td> • Presentación de trabajos en el tiempo establecido </td> <td> • Registro de presentación de trabajos 30 0 </td> </tr> <tr> <td> ACTIVIDAD DE INVESTIGACIÓN FORMATIVA Y/O RESPONSABILIDAD SOCIAL </td> <td> Análisis y Diseño de un Módulo o Subistema de Información </td> <td></td> <td></td> <td>0 0</td> </tr> <tr> <td>TOTAL</td> <td></td> <td></td> <td></td> <td>100</td> </tr> </tbody> </table>						TEMAS DE LA FASE	ESTRATEGIAS DE APRENDIZAJE-ENSEÑANZA	EVALUACIÓN DE LOS APRENDIZAJES					CRITERIOS	INSTRUMENTOS DE EVALUACIÓN	PORCENTAJE % TEORÍA PRÁCTICA	DE CONOCIMIENTO • Explica los conceptos sobre la administración del desarrollo de sistemas de información y el análisis estructurado	1ra. Unidad: Introducción al Análisis y Diseño de Sistemas. 1.1 Objetivos, problemas y problemática del análisis, responsabilidades, relación analista-usuario 1.2 Ciclo de desarrollo, organización, funciones	• Clase Magistral • Lecturas	• Preguntas objetivas y ensayo • Preguntas de ensayo	• Prueba de entrada • Examen • Control de lectura 40 0	DE DESEMPEÑO • Obtiene los fundamentos del análisis estructurado	1.3 Ciclo de vida del SW, modelos 1.4 Metodologías de desarrollo de SW, características, clasificación 1.5 Análisis de Costos y Beneficios del proyecto 1.6 Preparación de la Propuesta de Sistemas 2da Unidad: Gestión de Proyectos 2.1 Identificación de Proyectos	• Práctica individual y colectiva en laboratorio • Programación de Proyectos • Costo Beneficio • Flujograma de Información • Diccionario de Datos	• Presentación de la práctica • Presentación grupal	• Informe de práctica • Informe de trabajo en grupo 0 30	DE PRODUCTO • Valora el análisis como base del desarrollo de sistemas de información	2.2 Análisis de Factibilidad 2.3 Selección de Proyectos 2.4 Herramientas de gestión de proyectos 2.5 Creación y gestión del plan de trabajo	• Dirección y orientación del docente	• Presentación de trabajos en el tiempo establecido	• Registro de presentación de trabajos 30 0	ACTIVIDAD DE INVESTIGACIÓN FORMATIVA Y/O RESPONSABILIDAD SOCIAL	Análisis y Diseño de un Módulo o Subistema de Información			0 0	TOTAL				100
TEMAS DE LA FASE	ESTRATEGIAS DE APRENDIZAJE-ENSEÑANZA	EVALUACIÓN DE LOS APRENDIZAJES																																						
		CRITERIOS	INSTRUMENTOS DE EVALUACIÓN	PORCENTAJE % TEORÍA PRÁCTICA																																				
DE CONOCIMIENTO • Explica los conceptos sobre la administración del desarrollo de sistemas de información y el análisis estructurado	1ra. Unidad: Introducción al Análisis y Diseño de Sistemas. 1.1 Objetivos, problemas y problemática del análisis, responsabilidades, relación analista-usuario 1.2 Ciclo de desarrollo, organización, funciones	• Clase Magistral • Lecturas	• Preguntas objetivas y ensayo • Preguntas de ensayo	• Prueba de entrada • Examen • Control de lectura 40 0																																				
DE DESEMPEÑO • Obtiene los fundamentos del análisis estructurado	1.3 Ciclo de vida del SW, modelos 1.4 Metodologías de desarrollo de SW, características, clasificación 1.5 Análisis de Costos y Beneficios del proyecto 1.6 Preparación de la Propuesta de Sistemas 2da Unidad: Gestión de Proyectos 2.1 Identificación de Proyectos	• Práctica individual y colectiva en laboratorio • Programación de Proyectos • Costo Beneficio • Flujograma de Información • Diccionario de Datos	• Presentación de la práctica • Presentación grupal	• Informe de práctica • Informe de trabajo en grupo 0 30																																				
DE PRODUCTO • Valora el análisis como base del desarrollo de sistemas de información	2.2 Análisis de Factibilidad 2.3 Selección de Proyectos 2.4 Herramientas de gestión de proyectos 2.5 Creación y gestión del plan de trabajo	• Dirección y orientación del docente	• Presentación de trabajos en el tiempo establecido	• Registro de presentación de trabajos 30 0																																				
ACTIVIDAD DE INVESTIGACIÓN FORMATIVA Y/O RESPONSABILIDAD SOCIAL	Análisis y Diseño de un Módulo o Subistema de Información			0 0																																				
TOTAL				100																																				

Programa Formativo – Programación por fase de aprendizaje

Matriz de proyectos

Los proyectos de investigación formativa y responsabilidad social son ingresados en la sección de Matriz. Esta comprende cada elemento que servirá como detalle de la actividad. Además, con la posibilidad de destinar su desarrollo por la fase correspondiente.

Se asigna el peso en caso de que aplique, teórico y práctico. Si no se considera con alguna ponderación para la materia, ingresar los valores en cero.

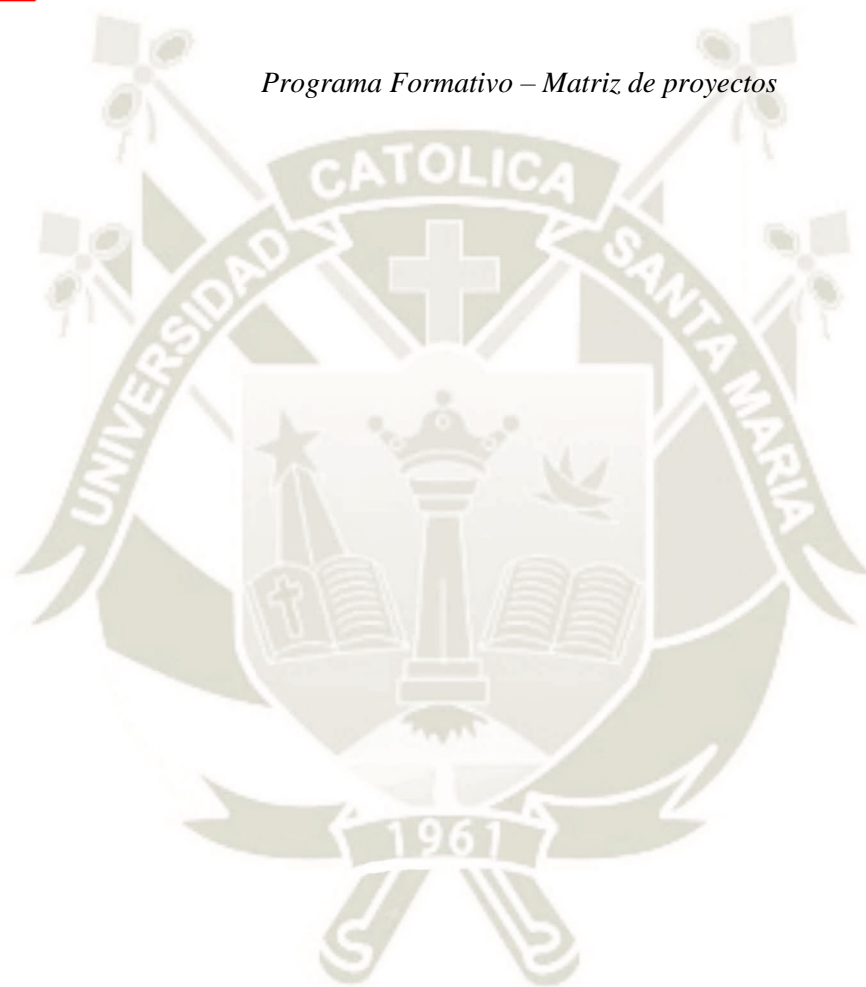
Culminada la inserción, se procede a “Guardar” los proyectos.

Matriz
Fase
1

ÁREA	DENOMINACIÓN DE LA ACTIVIDAD	PROPÓSITO	INDICADORES DE EVALUACIÓN	BENEFICIARIOS	RESPONSABLES	CRONOGRAMA	PESO TEÓRICO	PESO PRÁCTICO
Investigación Formativa	Análisis y Diseño de un Módulo o Subistema de Información	Análisis y diseñar un módulo/subistema	Presentación del informe	Estudiantes del V semestre	Docente de la asignatura	Primera fase: identificación	0	0
Responsabilidad Social								

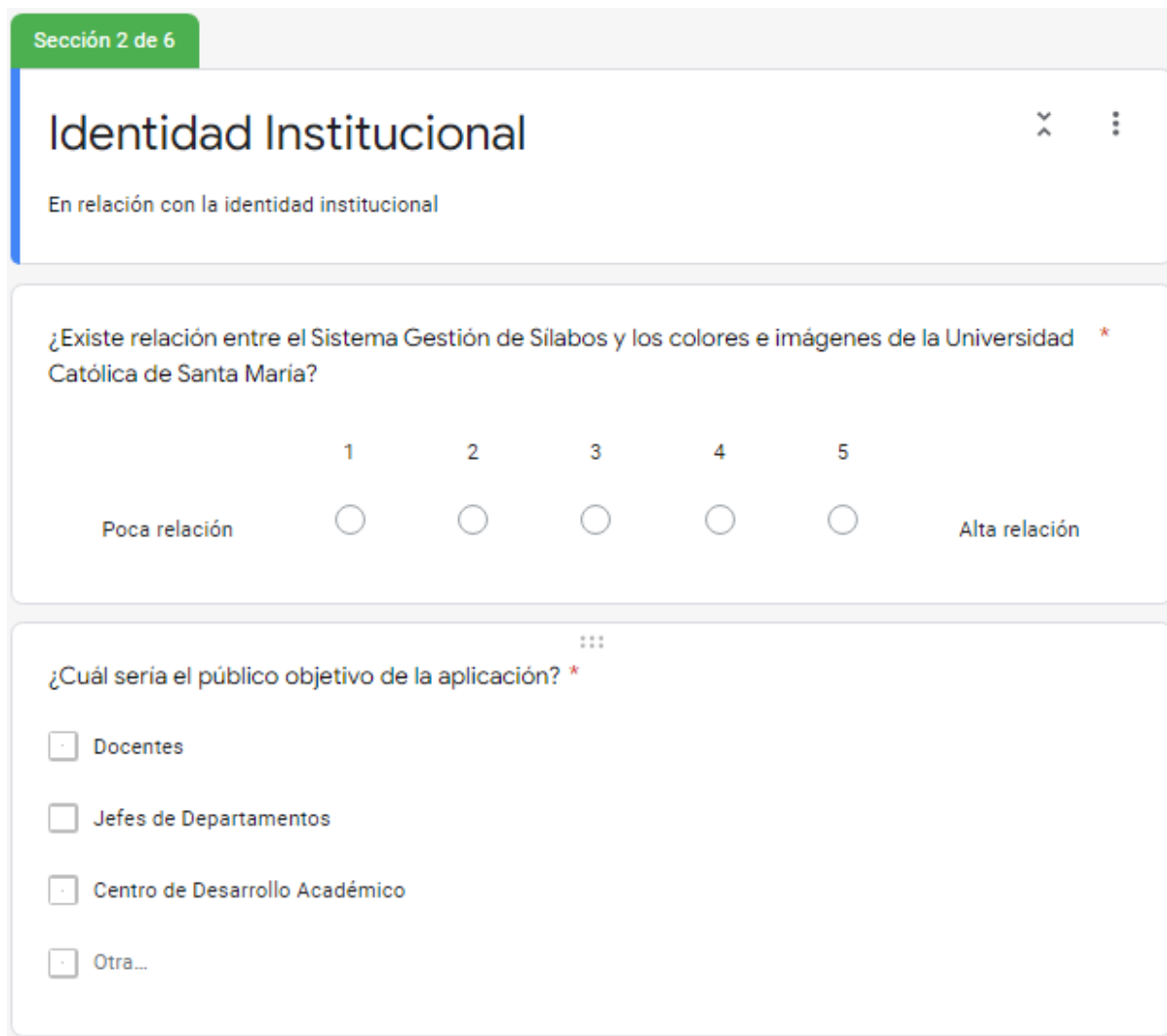
Guardar

Programa Formativo – Matriz de proyectos



APÉNDICE B: ENCUESTAS CON USUARIOS

El formato de encuesta utilizado fue desarrollado en la herramienta de Google Forms y es el siguiente:



Sección 2 de 6

Identidad Institucional

En relación con la identidad institucional

¿Existe relación entre el Sistema Gestión de Silabos y los colores e imágenes de la Universidad Católica de Santa María? *

Poca relación 1 2 3 4 5 Alta relación

¿Cuál sería el público objetivo de la aplicación? *

- Docentes
- Jefes de Departamentos
- Centro de Desarrollo Académico
- Otra...

Sección 3 de 6

Contenido de la aplicación



En relación al diseño y vista de la aplicación

¿Qué apreciación tiene de la interfaz del Sistema Gestión de Sílabos? *

- Muy amigable
- Amigable
- Complejo



¿Requiere asesoría para entender el funcionamiento de la aplicación? *

- | | | | | | | |
|---------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|---------------|
| | 1 | 2 | 3 | 4 | 5 | |
| Poca asesoría | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Alta asesoría |

¿Cuál considera el elemento de mayor impacto en la aplicación? *

- Visualización del Sílabo
- Habilitación de Sílabos
- Plan de estudios
- Programa Formativo

Sección 4 de 6

Navegación



En relación a la navegación en la aplicación



En el uso de la aplicación ¿Puede identificar estos elementos de navegación? *

- Barra de Navegación
- Filtros para la búsqueda de sílabos
- Selección de cursos asignados al docente
- Ingreso al Programa formativo
- Visualización de Bitácora
- Generación de archivo PDF con el contenido del sílabo
- Otra...

¿Las herramientas le ayudan a desarrollar el sílabo correctamente? *

	1	2	3	4	5	
Poca utilidad	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Alta utilidad



Sección 5 de 6

Búsqueda de aplicación



En relación a la búsqueda de información en la aplicación

¿La información encontrada en el Sistema de Gestión de Silabos es de ayuda y utilidad? *

Poca utilidad 1 2 3 4 5 Alta utilidad



¿Las secciones de desarrollo pueden ser ubicadas fácilmente? *

Ubicación dificultosa 1 2 3 4 5 Ubicación rápida



Sección 6 de 6

Utilidad de la aplicación



En relación a la utilidad de la aplicación para los usuarios

¿El contenido mostrado y las opciones implementadas permiten que el desarrollo del silabo sea *
ágil y mejor?

1 2 3 4 5

Mínima agilidad de desarrollo Gran agilidad de desarrollo

¿Recomendaría el prototipo como una solución para el desarrollo de silabos? *

1 2 3 4 5

Poca recomendación Alta recomendación

¿Cómo valora su experiencia utilizando la aplicación de Gestión de Silabos? *

1 2 3 4 5

Nada satisfecho Muy Satisfecho



Las respuestas a la encuesta son:

Encuesta #1	
Correo de Docente	dlazo@ucsm.edu.pe
1) ¿Existe relación entre el Sistema Gestión de Sílabos y los colores e imágenes de la Universidad Católica de Santa María?	4
2) ¿Cuál sería el público objetivo de la aplicación?	Docentes
3) ¿Qué apreciación tiene de la interfaz del Sistema Gestión de Sílabos?	Muy amigable
4) ¿Requiere asesoría para entender el funcionamiento de la aplicación?	1
5) ¿Cuál considera el elemento de mayor impacto en la aplicación?	Plan de estudios
6) En el uso de la aplicación ¿Puede identificar estos elementos de navegación?	<ul style="list-style-type: none"> • Barra de Navegación • Filtros para la búsqueda de sílabos • Selección de cursos asignados al docente • Ingreso al Programa formativo • Visualización de Bitácora • Generación de archivo PDF con el contenido del sílabo
7) ¿Las herramientas le ayudan a desarrollar el sílabo correctamente?	4
8) ¿La información encontrada en el Sistema de Gestión de Sílabos es de ayuda y utilidad?	5
9) ¿Las secciones de desarrollo pueden ser ubicadas fácilmente?	4
10) ¿El contenido mostrado y las opciones implementadas permiten que el desarrollo del sílabo sea ágil y mejor?	4
11) ¿Recomendaría el prototipo como una solución para el desarrollo de sílabos?	4
12) ¿Cómo valora su experiencia utilizando la aplicación de Gestión de Sílabos?	4

Encuesta #2

Correo de Docente	jesquicha@ucsm.edu.pe	
1) ¿Existe relación entre el Sistema Gestión de Sílabos y los colores e imágenes de la Universidad Católica de Santa María?		4
2) ¿Cuál sería el público objetivo de la aplicación?	<ul style="list-style-type: none"> • Docentes • Jefes de Departamentos • Centro de Desarrollo Académico 	
3) ¿Qué apreciación tiene de la interfaz del Sistema Gestión de Sílabos?		Amigable
4) ¿Requiere asesoría para entender el funcionamiento de la aplicación?		4
5) ¿Cuál considera el elemento de mayor impacto en la aplicación?		Visualización del Sílabo
6) En el uso de la aplicación ¿Puede identificar estos elementos de navegación?	<ul style="list-style-type: none"> • Barra de Navegación • Filtros para la búsqueda de sílabos 	
7) ¿Las herramientas le ayudan a desarrollar el sílabo correctamente?		4
8) ¿La información encontrada en el Sistema de Gestión de Sílabos es de ayuda y utilidad?		4
9) ¿Las secciones de desarrollo pueden ser ubicadas fácilmente?		4
10) ¿El contenido mostrado y las opciones implementadas permiten que el desarrollo del sílabo sea ágil y mejor?		4
11) ¿Recomendaría el prototipo como una solución para el desarrollo de sílabos?		4
12) ¿Cómo valora su experiencia utilizando la aplicación de Gestión de Sílabos?		4

Encuesta #3

Correo de Docente		afernandez@ucsm.edu.pe
1)	¿Existe relación entre el Sistema Gestión de Sílabos y los colores e imágenes de la Universidad Católica de Santa María?	5
2)	¿Cuál sería el público objetivo de la aplicación?	<ul style="list-style-type: none"> • Docentes • Jefes de Departamentos • Centro de Desarrollo Académico
3)	¿Qué apreciación tiene de la interfaz del Sistema Gestión de Sílabos?	Muy amigable
4)	¿Requiere asesoría para entender el funcionamiento de la aplicación?	2
5)	¿Cuál considera el elemento de mayor impacto en la aplicación?	Plan de estudios
6)	En el uso de la aplicación ¿Puede identificar estos elementos de navegación?	<ul style="list-style-type: none"> • Barra de Navegación • Filtros para la búsqueda de sílabos • Selección de cursos asignados al docente • Ingreso al Programa formativo • Visualización de Bitácora • Generación de archivo PDF con el contenido del sílabo
7)	¿Las herramientas le ayudan a desarrollar el sílabo correctamente?	5
8)	¿La información encontrada en el Sistema de Gestión de Sílabos es de ayuda y utilidad?	5
9)	¿Las secciones de desarrollo pueden ser ubicadas fácilmente?	5
10)	¿El contenido mostrado y las opciones implementadas permiten que el desarrollo del sílabo sea ágil y mejor?	4
11)	¿Recomendaría el prototipo como una solución para el desarrollo de sílabos?	5
12)	¿Cómo valora su experiencia utilizando la aplicación de Gestión de Sílabos?	5

Encuesta #4

Correo de Docente	karina.r.paredes@gmail.com	
1) ¿Existe relación entre el Sistema Gestión de Sílabos y los colores e imágenes de la Universidad Católica de Santa María?		5
2) ¿Cuál sería el público objetivo de la aplicación?	<ul style="list-style-type: none"> • Docentes • Jefes de Departamentos 	
3) ¿Qué apreciación tiene de la interfaz del Sistema Gestión de Sílabos?		Amigable
4) ¿Requiere asesoría para entender el funcionamiento de la aplicación?		1
5) ¿Cuál considera el elemento de mayor impacto en la aplicación?		Programa Formativo
6) En el uso de la aplicación ¿Puede identificar estos elementos de navegación?	<ul style="list-style-type: none"> • Barra de Navegación • Filtros para la búsqueda de sílabos • Selección de cursos asignados al docente • Ingreso al Programa formativo • Visualización de Bitácora 	
7) ¿Las herramientas le ayudan a desarrollar el sílabo correctamente?		5
8) ¿La información encontrada en el Sistema de Gestión de Sílabos es de ayuda y utilidad?		5
9) ¿Las secciones de desarrollo pueden ser ubicadas fácilmente?		5
10) ¿El contenido mostrado y las opciones implementadas permiten que el desarrollo del sílabo sea ágil y mejor?		4
11) ¿Recomendaría el prototipo como una solución para el desarrollo de sílabos?		5
12) ¿Cómo valora su experiencia utilizando la aplicación de Gestión de Sílabos?		4

Encuesta #5

Correo de Docente	janguelo@ucsm.edu.pe	
1) ¿Existe relación entre el Sistema Gestión de Sílabos y los colores e imágenes de la Universidad Católica de Santa María?		3
2) ¿Cuál sería el público objetivo de la aplicación?	<ul style="list-style-type: none"> • Docentes • Jefes de Departamentos • Centro de Desarrollo Académico 	
3) ¿Qué apreciación tiene de la interfaz del Sistema Gestión de Sílabos?		Amigable
4) ¿Requiere asesoría para entender el funcionamiento de la aplicación?		3
5) ¿Cuál considera el elemento de mayor impacto en la aplicación?		Habilitación de Sílabos
6) En el uso de la aplicación ¿Puede identificar estos elementos de navegación?	<ul style="list-style-type: none"> • Barra de Navegación • Filtros para la búsqueda de sílabos • Selección de cursos asignados al docente • Visualización de Bitácora 	
7) ¿Las herramientas le ayudan a desarrollar el sílabo correctamente?		3
8) ¿La información encontrada en el Sistema de Gestión de Sílabos es de ayuda y utilidad?		2
9) ¿Las secciones de desarrollo pueden ser ubicadas fácilmente?		2
10) ¿El contenido mostrado y las opciones implementadas permiten que el desarrollo del sílabo sea ágil y mejor?		2
11) ¿Recomendaría el prototipo como una solución para el desarrollo de sílabos?		1
12) ¿Cómo valora su experiencia utilizando la aplicación de Gestión de Sílabos?		1

Encuesta #6

Correo de Docente	amontesinos@ucsm.edu.pe	
1) ¿Existe relación entre el Sistema Gestión de Sílabos y los colores e imágenes de la Universidad Católica de Santa María?	5	
2) ¿Cuál sería el público objetivo de la aplicación?		<ul style="list-style-type: none"> • Docentes • Jefes de Departamentos • Centro de Desarrollo Académico
3) ¿Qué apreciación tiene de la interfaz del Sistema Gestión de Sílabos?	Amigable	
4) ¿Requiere asesoría para entender el funcionamiento de la aplicación?	3	
5) ¿Cuál considera el elemento de mayor impacto en la aplicación?	Plan de estudios	
6) En el uso de la aplicación ¿Puede identificar estos elementos de navegación?		<ul style="list-style-type: none"> • Barra de Navegación • Ingreso al Programa formativo • Visualización de Bitácora
7) ¿Las herramientas le ayudan a desarrollar el sílabo correctamente?	3	
8) ¿La información encontrada en el Sistema de Gestión de Sílabos es de ayuda y utilidad?	3	
9) ¿Las secciones de desarrollo pueden ser ubicadas fácilmente?	4	
10) ¿El contenido mostrado y las opciones implementadas permiten que el desarrollo del sílabo sea ágil y mejor?	3	
11) ¿Recomendaría el prototipo como una solución para el desarrollo de sílabos?	3	
12) ¿Cómo valora su experiencia utilizando la aplicación de Gestión de Sílabos?	3	

Encuesta #7

Correo de Docente	msantillana@ucsm.edu.pe	
1) ¿Existe relación entre el Sistema Gestión de Sílabos y los colores e imágenes de la Universidad Católica de Santa María?	5	
2) ¿Cuál sería el público objetivo de la aplicación?		<ul style="list-style-type: none"> • Docentes • Jefes de Departamentos • Centro de Desarrollo Académico
3) ¿Qué apreciación tiene de la interfaz del Sistema Gestión de Sílabos?	Amigable	
4) ¿Requiere asesoría para entender el funcionamiento de la aplicación?	3	
5) ¿Cuál considera el elemento de mayor impacto en la aplicación?	Programa Formativo	
6) En el uso de la aplicación ¿Puede identificar estos elementos de navegación?		<ul style="list-style-type: none"> • Filtros para la búsqueda de sílabos • Selección de cursos asignados al docente • Ingreso al Programa formativo • Generación de archivo PDF con el contenido del sílabo
7) ¿Las herramientas le ayudan a desarrollar el sílabo correctamente?	4	
8) ¿La información encontrada en el Sistema de Gestión de Sílabos es de ayuda y utilidad?	4	
9) ¿Las secciones de desarrollo pueden ser ubicadas fácilmente?	4	
10) ¿El contenido mostrado y las opciones implementadas permiten que el desarrollo del sílabo sea ágil y mejor?	4	
11) ¿Recomendaría el prototipo como una solución para el desarrollo de sílabos?	4	
12) ¿Cómo valora su experiencia utilizando la aplicación de Gestión de Sílabos?	4	

Encuesta #8

Correo de Docente		crevilla@ucsm.edu.pe
1)	¿Existe relación entre el Sistema Gestión de Sílabos y los colores e imágenes de la Universidad Católica de Santa María?	5
2)	¿Cuál sería el público objetivo de la aplicación?	<ul style="list-style-type: none"> • Docentes • Jefes de Departamentos
3)	¿Qué apreciación tiene de la interfaz del Sistema Gestión de Sílabos?	Muy amigable
4)	¿Requiere asesoría para entender el funcionamiento de la aplicación?	4
5)	¿Cuál considera el elemento de mayor impacto en la aplicación?	Plan de estudios
6)	En el uso de la aplicación ¿Puede identificar estos elementos de navegación?	<ul style="list-style-type: none"> • Barra de Navegación • Filtros para la búsqueda de sílabos • Selección de cursos asignados al docente • Ingreso al Programa formativo • Generación de archivo PDF con el contenido del sílabo
7)	¿Las herramientas le ayudan a desarrollar el sílabo correctamente?	4
8)	¿La información encontrada en el Sistema de Gestión de Sílabos es de ayuda y utilidad?	4
9)	¿Las secciones de desarrollo pueden ser ubicadas fácilmente?	5
10)	¿El contenido mostrado y las opciones implementadas permiten que el desarrollo del sílabo sea ágil y mejor?	5
11)	¿Recomendaría el prototipo como una solución para el desarrollo de sílabos?	5
12)	¿Cómo valora su experiencia utilizando la aplicación de Gestión de Sílabos?	4

Encuesta #9

Correo de Docente	oflores@ucsm.edu.pe	
1) ¿Existe relación entre el Sistema Gestión de Sílabos y los colores e imágenes de la Universidad Católica de Santa María?	5	
2) ¿Cuál sería el público objetivo de la aplicación?	<ul style="list-style-type: none"> • Docentes • Centro de Desarrollo Académico 	
3) ¿Qué apreciación tiene de la interfaz del Sistema Gestión de Sílabos?	Amigable	
4) ¿Requiere asesoría para entender el funcionamiento de la aplicación?	1	
5) ¿Cuál considera el elemento de mayor impacto en la aplicación?	Habilitación de Sílabos	
6) En el uso de la aplicación ¿Puede identificar estos elementos de navegación?	<ul style="list-style-type: none"> • Filtros para la búsqueda de sílabos • Selección de cursos asignados al docente • Generación de archivo PDF con el contenido del sílabo 	
7) ¿Las herramientas le ayudan a desarrollar el sílabo correctamente?	4	
8) ¿La información encontrada en el Sistema de Gestión de Sílabos es de ayuda y utilidad?	3	
9) ¿Las secciones de desarrollo pueden ser ubicadas fácilmente?	5	
10) ¿El contenido mostrado y las opciones implementadas permiten que el desarrollo del sílabo sea ágil y mejor?	4	
11) ¿Recomendaría el prototipo como una solución para el desarrollo de sílabos?	4	
12) ¿Cómo valora su experiencia utilizando la aplicación de Gestión de Sílabos?	4	

Encuesta #10

Correo de Docente	carlocorrales@gmail.com	
1) ¿Existe relación entre el Sistema Gestión de Sílabos y los colores e imágenes de la Universidad Católica de Santa María?		4
2) ¿Cuál sería el público objetivo de la aplicación?	<ul style="list-style-type: none"> • Docentes • Jefes de Departamentos 	
3) ¿Qué apreciación tiene de la interfaz del Sistema Gestión de Sílabos?		Amigable
4) ¿Requiere asesoría para entender el funcionamiento de la aplicación?		3
5) ¿Cuál considera el elemento de mayor impacto en la aplicación?		Visualización del Sílabo
6) En el uso de la aplicación ¿Puede identificar estos elementos de navegación?	<ul style="list-style-type: none"> • Barra de Navegación • Filtros para la búsqueda de sílabos • Ingreso al Programa formativo • Generación de archivo PDF con el contenido del sílabo 	
7) ¿Las herramientas le ayudan a desarrollar el sílabo correctamente?		3
8) ¿La información encontrada en el Sistema de Gestión de Sílabos es de ayuda y utilidad?		3
9) ¿Las secciones de desarrollo pueden ser ubicadas fácilmente?		4
10) ¿El contenido mostrado y las opciones implementadas permiten que el desarrollo del sílabo sea ágil y mejor?		4
11) ¿Recomendaría el prototipo como una solución para el desarrollo de sílabos?		3
12) ¿Cómo valora su experiencia utilizando la aplicación de Gestión de Sílabos?		3