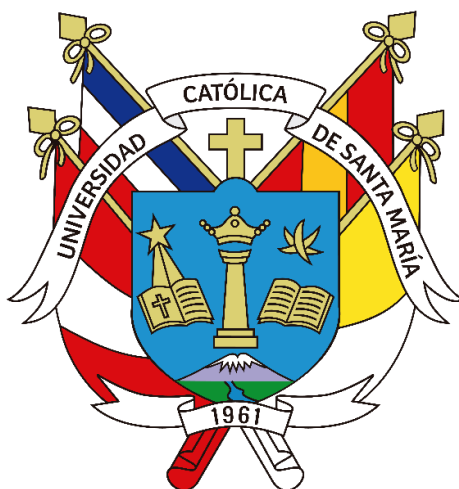


Universidad Católica de Santa María

**Facultad de Ciencias e Ingenierías Físicas y
Formales**

Escuela Profesional de Ingeniería de Sistemas



**DESARROLLO DE UN MARCO DE TRABAJO PARA EL
INTERCAMBIO DE INFORMACIÓN ENTRE UNA ENTIDAD
EDUCATIVA Y FINANCIERA UTILIZANDO SOA Y KANBAN**

Tesis presentada por el bachiller:

Salas Rodríguez, Antony Alonso

para optar al Título profesional de
Ingeniero de Sistemas

Asesor:

Mg. Montesinos Murillo, Ángel
Felipe

Arequipa - Perú

2022

UCSM-ERP

UNIVERSIDAD CATÓLICA DE SANTA MARÍA

INGENIERIA DE SISTEMAS

TITULACIÓN CON TESIS

DICTAMEN APROBACIÓN DE BORRADOR

Arequipa, 13 de Julio del 2022

Dictamen: 005189-C-EPIS-2022

Visto el borrador del expediente 005189, presentado por:

2016602871 - SALAS RODRIGUEZ ANTONY ALONSO

Titulado:

**DESARROLLO DE UN MARCO DE TRABAJO PARA EL INTERCAMBIO DE INFORMACIÓN ENTRE
UNA ENTIDAD EDUCATIVA Y FINANCIERA UTILIZANDO SOA Y KANBAN**

Nuestro dictamen es:

APROBADO

**1221 - PAREDES MARCHENA FERNANDO GERMAN
DICTAMINADOR**

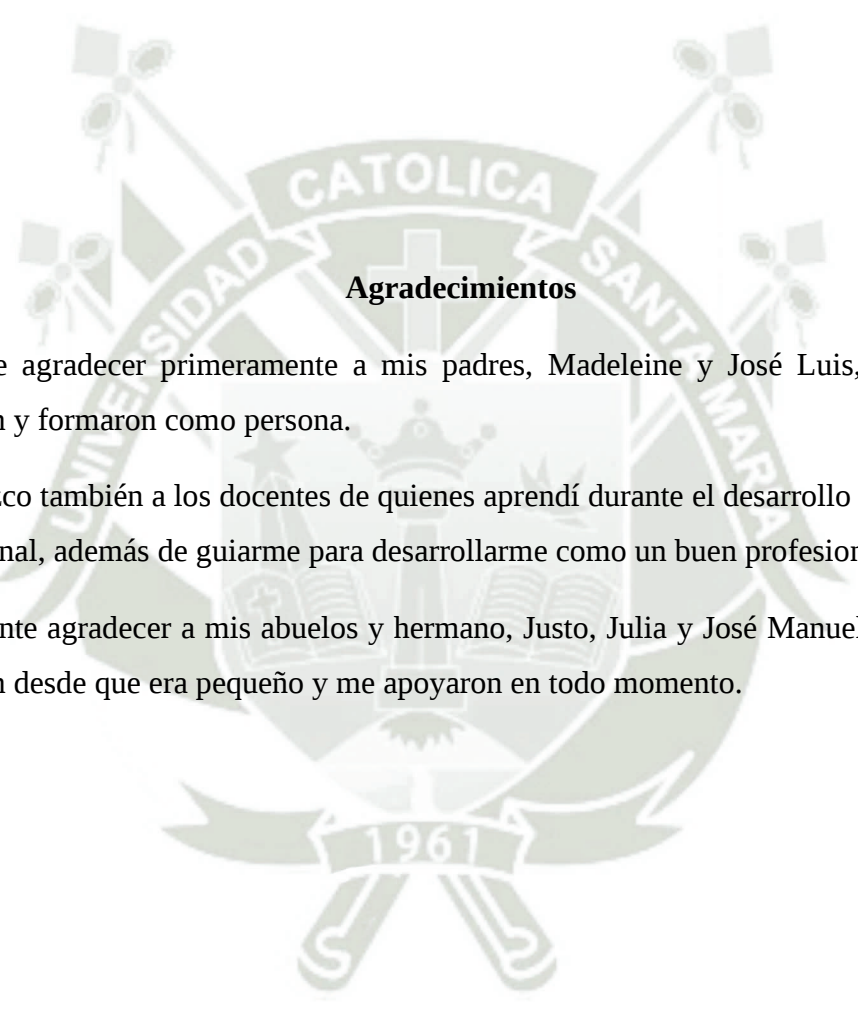


**1561 - GUEVARA PUENTE DE LA VEGA KARIM
DICTAMINADOR**



**2820 - ESQUICHA TEJADA JOSE DAVID
DICTAMINADOR**





Agradecimientos

Debo de agradecer primeramente a mis padres, Madeleine y José Luis, quienes me cuidaron y formaron como persona.

Agradezco también a los docentes de quienes aprendí durante el desarrollo de mi carrera profesional, además de guiarme para desarrollarme como un buen profesional.

Finalmente agradecer a mis abuelos y hermano, Justo, Julia y José Manuel, quienes me cuidaron desde que era pequeño y me apoyaron en todo momento.

RESUMEN

En la actualidad el intercambio de datos interinstitucional está avanzando cada vez más debido a la aparición de nuevas técnicas y formatos lo cual ha dado lugar a que casi la mayor parte de instituciones cuente con una o más conexiones con diferentes entidades permitiéndoles acceder a sus bases de datos y obtener y/o enviar datos que requieren en la realización de sus operaciones optimizando de esta manera el funcionamiento de sus procesos.

El tema de nuestro trabajo de investigación es el desarrollo de un marco de trabajo para el intercambio de información entre una entidad educativa y financiera utilizando la metodologías SOA y KANBAN, los servicios desarrollados en el marco de trabajo utilizaron el formato JSON para modelar las solicitudes y respuestas que utilizan los servicios implementados en la entidad educativa; a partir del análisis de proyectos que han desarrollado servicios web en diferentes organizaciones, de igual forma se analizó la utilización de la arquitectura SOA en la prestación de servicios dentro de algunas instituciones. Los resultados obtenidos nos han permitido elaborar un marco para el intercambio de información entre dos entidades (una educativa y la otra financiera) para una mejor prestación de servicios. Nuestro trabajo de investigación teniendo en cuenta lo delicado en la temática ha sido sometida a constante análisis y evaluación cuyos resultados se pueden ver en los diagramas UML tanto en los servicios identificados como de las pruebas de verificación de rendimiento desde el momento de su aplicación interinstitucional, sin embargo, como todo trabajo científico en la práctica, debido al cambio constante de la actividad desarrollada por las instituciones es susceptible a la presencia de errores y variaciones en tiempos de ejecución, siendo lo primordial la prestación de un servicio eficiente para los usuarios y beneficioso para las entidades.

Palabras clave: JSON, intercambio de datos, servicios.

ABSTRACT

At present, inter-institutional data exchange is advancing more and more due to the appearance of new techniques and formats, which has given rise to almost the majority of institutions having one or more connections with different entities, allowing them to access their databases. data and obtain and/or send data that they require in carrying out their operations, thus optimizing the operation of their processes.

The subject of our research work is the development of a framework for the exchange of information between an educational and financial entity using the SOA and KANBAN methodologies, the services developed in the framework used the JSON format to model the requests and answers that use the services implemented in the educational entity; Based on the analysis of projects that have developed web services in different organizations, the use of the SOA methodology in the provision of services within some institutions was also analyzed. The results obtained have allowed us to develop a framework for the exchange of information between two entities (one educational and the other financial) for a better provision of services. Our research work, taking into account the sensitivity of the subject, has been subjected to constant analysis and evaluation, the results of which can be seen in the UML diagrams both in the identified services and in the performance verification tests from the moment of their inter-institutional application. However, like all scientific work in practice, due to the constant change of the activity carried out by the institutions, it is susceptible to the presence of errors and variations in execution times, with the provision of an efficient and beneficial service for users being paramount. for entities.

Keywords: JSON, data exchange, services

INTRODUCCIÓN

Las instituciones bancarias hoy en día no se ocupan únicamente, como antes, de transacciones comerciales como giros, letras de cambio, pagares, etc., como tampoco las instituciones educativas son únicamente de carácter estatal, la búsqueda de una mejor calidad educativa por parte de los padres de familia ha dado lugar a la aparición y proliferación de instituciones educativas de carácter particular y con ello un mayor movimiento económico debido al cobro por prestación de servicios por parte de estas últimas instituciones, con las consecuentes molestias y riesgos tanto en el cobro y pago de prestaciones de servicios como con la custodia de dichos fondos.

El sistema obsoleto utilizado todavía por algunas instituciones educativas demanda un cambio radical tanto en la actividad bancaria como en el sistema de cobranza de los centros educativos. Teniendo en cuenta que uno de los fines de nuestra carrera profesional es la de prestar alternativas de solución a los problemas de nuestra sociedad es que revisamos bibliografía especializada a fin de informarnos sobre trabajos realizados a nivel nacional como internacional.

La consulta bibliográfica y los resultados obtenidos de la nueva interrelación (intercambio de datos) entre instituciones bancarias y educativas nos ha llevado a elegir como tema de trabajo “Desarrollo de un Marco de Trabajo para el intercambio de Información entre una Entidad Educativa y Financiera Utilizando SOA y KANBAN” donde a través de la utilización de servicios API-REST facilita el intercambio y transferencia de datos utilizando el formato JSON. Teniendo en cuenta que, la incorporación permitirá la comunicación e intercambio de datos, presentando una oportunidad para abrir nuevos flujos de ingreso, maximizando los valores del cliente, gracias a la reutilización que se puede utilizar para nuevas interrelaciones con otras instituciones.

El estudio de Atencio Flores & Mamani Machaca (2018), en una municipalidad se logró optimizar la comunicación mediante el desarrollo de una interfaz que utiliza un servicio API-REST, mostrándonos como se modelo a través de los diagramas de Modelamiento Unificado (UML) y la metodología Scrum para la documentación de sus procesos, una vez finalizada la implementación de los servicios se obtuvo un prototipo de servicio (API-REST) que ayudo en la mejora de las aplicaciones de comunicación de la municipalidad optimizando sus tiempos en sus procesos. Esta interesante experiencia nos demuestra que las relaciones interinstitucionales no solo pueden darse entre entidades financieras y educativas.

Valencia Altamirano (2018), realizo una implementación de aplicaciones web que necesitaban tener un mejor rendimiento y funcionalidad motivo por el cual se vio la necesidad de implementar nuevas técnicas para su optimización (búsqueda de datos asíncrona, actualización de contenido, etc.) para lo cual el acoplamiento de servicios API-REST que son desarrolladas en backend debe ser fácil de acoplar al frontend, el cual a su vez ha permitido medir a través de las métricas

Response Time y Throughput el rendimiento de las aplicaciones y el comportamiento de los servicios API-REST que son desarrolladas con SPA (Single Page Application).

Uno de los objetivos que se tiene al incorporar servicios dentro de una organización, es ver que tan viable es “apificar” los procesos dentro de la arquitectura de TI actual, con tal fin se realizó un estudio acerca de los beneficios que conlleva adoptar una administración de API dentro de las empresas, mediante la incorporación de tres servicios backend del BBVA y el desarrollo de una aplicación de Android que la consuma. (*Liaño Díaz, 2017*)

El enorme incremento de clientes en las instituciones bancarias especialmente en los últimos años ha dado lugar a la demanda y uso de una mayor cantidad de aplicativos por estas instituciones y sus empleados, tal es el caso de la empresa de fondo Colectivo Maquisitema, la cual necesitaba la integración de un Sistema de Gestión de Colas al CRM (Sistema de Gestión de Relaciones) por la cantidad de aplicativos que sus empleados manejaban, para lo cual se gestionó una página web que generaba los tickets (a través de un servicio API-REST), con la finalidad de que se pueda actualizar el estado de los tickets atendidos desde el CRM, así mismo se desarrolló un servicio API-REST para realizar una encuesta de atención a la cual se podía acceder por una Tablet que estaba conectada por un puerto USB en la empresa. (*Meneses Rivera, 2021*)

Teniendo en cuenta el tipo de servicio prestado por las instituciones bancarias y los requeridos por los usuarios, los servicios y aplicativos deben buscar la eficiencia y la optimización de los procesos, pero, fundamentalmente lograr la evitar los datos sean accesibles para terceras personas adaptándose a las normativas de Servicios de Pago (PSD2), de tal manera que vayan de la mano eficiencia y seguridad. Un trabajo importante es el que nos presenta Romano Rodríguez quien a partir de la simulación de un entorno empresarial implementado de un microservicio, nos da a conocer su implementación y los resultados obtenidos. (*Romano Rodríguez, 2021*)

Lo expuesto hasta ahora nos permite ver que los servicios y aplicativos son cada vez más requeridos, siendo las entidades financieras las más beneficiadas con la implementación de servicios API-REST, y las instituciones públicas o privadas que se integran a este sistema enviando y recepcionando los datos procesados en tiempo real como lo detalla la investigación que se realizó en la Universidad de Zaragoza, en la cual se implementó una Plataforma Integrada de Pagos para la Universidad compuesta por 5 módulos, los cuales facilitan y optimizarían los pagos antes realizados por medio electrónico. (Gascón Gascón, 2018)

La experiencia obtenida en el desarrollo de la utilización de servicios y aplicativos en diferentes instituciones extranjeras y algunas nacionales nos ha llevado a establecer que el principal problema esta en la cantidad de cuentas por pagar y cobrar las cuales son trasladadas al sistema financiero de allí la necesidad de estructurar un marco de trabajo que permita interconectar las entidades educativas con las entidades financieras con lo cual se agilizarían los cobros y pagos por la prestación de servicios y se evitaría pagos y cobros indebidos ya que todas estas transacciones serian procesadas y consultadas de manera inmediata.

Nuestro trabajo de investigación tiene por objetivo principal desarrollar un marco de trabajo que permita la implementación de un servicio API-REST en entidades educativas que permita el intercambio de datos con una entidad financiera para optimizar los procesos de pago por la prestación de servicios de dichas entidades educativas. Y como objetivos específicos se tiene los siguientes:

- Analizar los requerimientos de los procesos que van a ser implementados dentro la interconexión con la entidad financiera.
- Modelar los datos que los procesos utilizaran durante la interconexión.
- Detallar la semántica de los procesos implementados mediante la elaboración de los diagramas UML (Lenguaje Unificado de Modelado).
- Establecer pruebas que permitan evaluar la velocidad y funcionamiento de los procesos implementados.

Nuestro trabajo de investigación, presentado en la modalidad de tesis, está compuesta por 4 capítulos y un apéndice cuyo contenido está organizado de la siguiente manera:

- Capítulo 1: Descripción del proyecto, incluye toda la información base del proyecto desarrollado y el marco teórico.
- Capítulo 2: Metodología, que incluye la explicación de las metodologías utilizadas y como se implementaron, así como el desarrollo del marco de trabajo.
- Capítulo 3: Caso de estudio, implementación del marco de trabajo en una entidad educativa y financiera.
- Capítulo 4: Resultados y discusión, en este capítulo presentamos los diagramas de Modelamiento Unificado UML, los resultados de las pruebas obtenidas de los servicios expuestos y la discusión de los resultados.

Aparte va detallada en primer lugar las referencias consultadas, en especial referencias en el cuerpo de nuestra tesis y en el acápite de anexos se incluyen documentos y formatos que sirven de base de sustento complementario a nuestro trabajo de investigación.

Índice General

DICTAMEN APROBATORIO

AGRADECIMIENTOS

RESUMEN

ABSTRACT

CAPÍTULO I: DESCRIPCIÓN DEL PROYECTO 1

1. Descripción y Objetivos del Proyecto 1

1.1. Descripción del Problema 1

1.2. Objetivos del Proyecto 1

1.2.1. Objetivo General 1

1.2.2. Objetivos Específicos..... 1

1.3. Alcances y Limitaciones..... 1

1.3.1. Alcances..... 1

1.3.2. Limitaciones..... 2

1.4. Fundamentos Teóricos..... 2

1.4.1. Webservices..... 2

1.4.2. Estándares en el desarrollo de WebServices..... 2

1.4.3. Localizador Uniforme de Recursos de Internet (URL)..... 3

1.4.4. JavaScript Object Notation (JSON) 3

1.4.6. Métodos HTTP 4

1.4.7. API-REST 4

1.4.8. Entorno de Desarrollo Integrado (IDE) 4

1.5. Técnicas y Herramientas 5

1.5.1. Técnicas..... 5

1.5.2. Herramientas 5

CAPÍTULO II: METODOLOGIA..... 7

2. Justificación, aporte y metodología del proyecto 7

2.1. Justificación y Aporte del proyecto 7

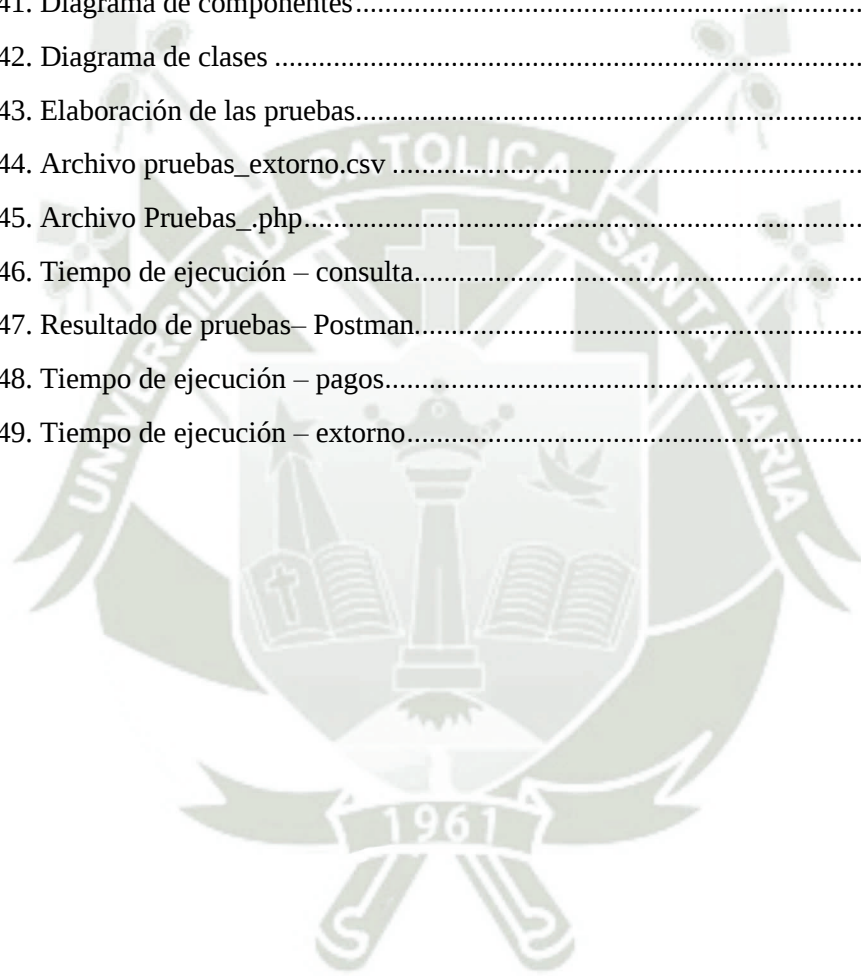
2.1.1. Justificación Teórica	7
2.1.2. Justificación Técnica	7
2.1.3. Justificación Operativa	8
2.1.4. Importancia e Impacto.....	8
2.2 Comparativa entre formatos de Datos	9
2.3 Comparativa entre metodologías de desarrollo ágil	10
2.4 Objetivos	11
2.5 Diseño del Proyecto (Marco de Trabajo)	12
2.5.1. Componentes del Marco de trabajo.....	12
2.5.2. Metodología Kanban.....	13
2.5.3. Arquitectura Orientada a los Servicios (SOA)	14
2.6 Implementación.....	18
2.6.1. Características del Servidor	18
2.6.2. Instalación de las herramientas.....	18
2.7 Plan de Pruebas	18
2.7.1. Propósito	18
2.7.2. Alcance	19
2.7.3. Estrategia de Pruebas	19
CAPÍTULO III: CASO DE ESTUDIO	20
3. Desarrollo del caso de estudio	20
3.1 Metodología Kanban.....	20
3.2 Arquitectura	26
3.2.1 Identificación de los servicios	26
3.2.2 Relación entre servicios y entidades	30
3.2.3 Requerimientos de los servicios identificados.....	33
3.2.4 Priorización de los servicios.....	35
3.2.5 Descripción de los servicios	38
3.4 Códigos de Respuesta.....	49

3.5 Funciones PostgreSQL.....	50
3.6 Implementación y pruebas	53
3.6.1 Características del servidor y programas	53
3.6.2 Instalación de los servicios en el servidor.....	54
3.6.3 Estrategia de Pruebas	55
Capítulo IV: Resultados.....	57
4. Diagramas y resultados de las pruebas	57
4.1 Diagramas UML.....	57
4.1.1 Diagrama de casos de uso	57
4.1.2 Diagrama de secuencia.....	69
4.1.3 Diagrama de componentes.....	72
4.1.4 Diagrama de clases	73
4.2 Resultado de las pruebas	75
4.2.1 Pruebas de Funcionalidad	75
4.2.2 Pruebas de Carga	82
4.3 Comparativa con trabajos relacionados.....	99
Conclusiones	101
Recomendaciones y trabajos futuros.....	102
Referencias.....	103

Índice de Figuras

Figura 1. Modelo de Comunicación Soap	2
Figura 2. Ejemplo de la Estructura de URL	3
Figura 3. Componentes del marco de trabajo	12
Figura 4. Tarjeta Kanban (Plantilla)	13
Figura 5. Tablero Kanban (Plantilla)	13
Figura 6. Ejemplo de BPMN	14
Figura 7. Ejemplo de un diagrama Entidad-Relación	15
Figura 8. Tablero Kanban 1ª Etapa	20
Figura 9. Tarjeta de Identificación de servicios	20
Figura 10. Tarjetas en estado “Por hacer”	21
Figura 11. Tablero Kanban Actualización 1	22
Figura 12. Tablero Kanban Actualización 2	22
Figura 13. Tablero Kanban Actualización 3	22
Figura 14. Actividades: análisis y elaboración de tabla	23
Figura 15. Tablero Kanban actualización 4	24
Figura 16. Tablero Kanban actualización 5	24
Figura 17. Tarjetas – Identificación y descripción de requerimientos	25
Figura 18. Tablero Kanban actualización 6	25
Figura 19. Mapa de Procesos – Entidad Educativa Superior	26
Figura 20. Proceso de solicitud de estado de cuenta	27
Figura 21. Proceso de Pago – 1	28
Figura 22. Proceso de Pago – 2	28
Figura 23. Proceso de reembolso de una deuda	29
Figura 24. Diagrama entidad relación – Solicitud de estado de cuenta	30
Figura 25. Diagrama entidad relación – Proceso de pago de deudas	31
Figura 26. Diagrama entidad relación – Proceso de reembolso de una deuda	32
Figura 27 Fragmento de función consulta	51
Figura 28. Tipo t_consulta	51
Figura 29. Fragmento de función pagos	52
Figura 30. Icono de Filezilla	54
Figura 31. Configuración usuario y contraseña	54
Figura 32. Directorio de proyectos	55
Figura 33. Subida de archivos al servidor	55

Figura 34. Diagrama de Casos de uso - General	57
Figura 35. Consultar Deudas.....	60
Figura 36. Pago de una deuda	63
Figura 37. Reembolso de una deuda (extornar)	66
Figura 38. Diagrama de secuencia – Consultar deuda	69
Figura 39. Diagrama de secuencia – Pagar deuda.....	70
Figura 40. Diagrama de secuencia – Reembolso de una deuda (extorno).....	71
Figura 41. Diagrama de componentes.....	72
Figura 42. Diagrama de clases	73
Figura 43. Elaboración de las pruebas.....	82
Figura 44. Archivo pruebas_extorno.csv	83
Figura 45. Archivo Pruebas_.php.....	83
Figura 46. Tiempo de ejecución – consulta.....	85
Figura 47. Resultado de pruebas– Postman.....	85
Figura 48. Tiempo de ejecución – pagos.....	94
Figura 49. Tiempo de ejecución – extorno.....	99



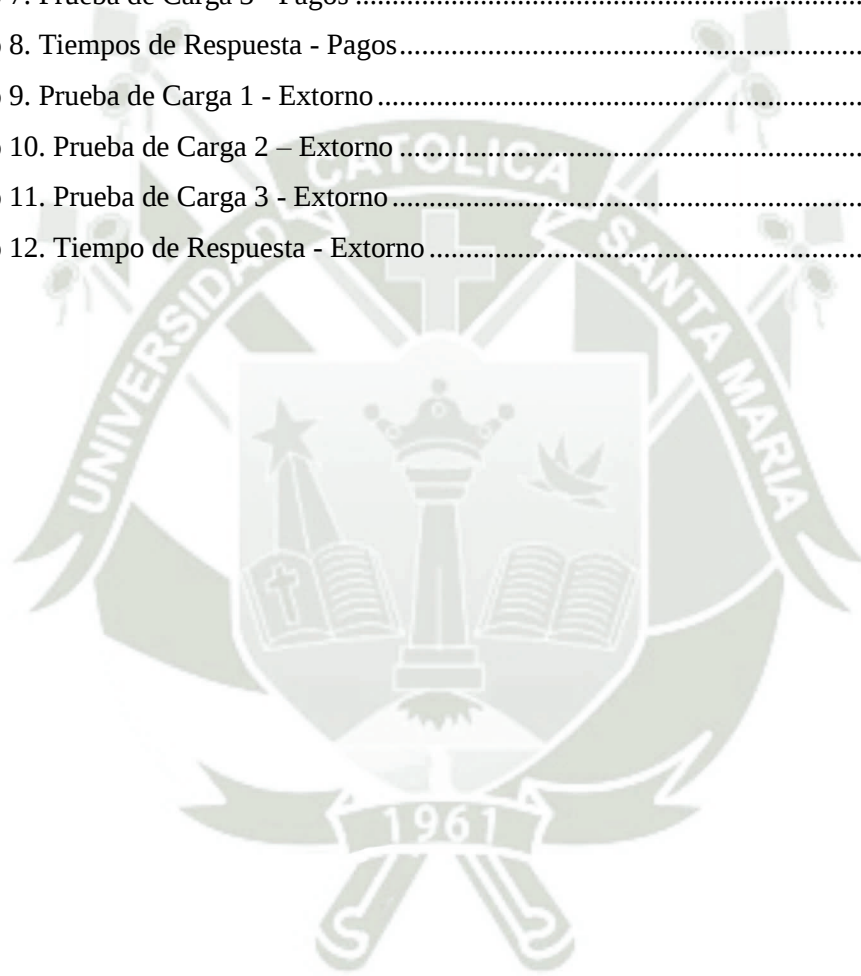
Índice de Tablas

Tabla 1. Cuadro comparativo de formatos de datos	9
Tabla 2. Cuadro comparativo Kanban vs Scrum.....	11
Tabla 3. Validación de Objetivos	11
Tabla 4. Plantilla de cuadro de procesos – servicios - Ejemplo	15
Tabla 5 Plantilla del cuadro de procesos – entidad - Ejemplo	16
Tabla 6. Plantilla del Cuadro de procesos – servicios - Ejemplo	16
Tabla 7. Plantilla del cuadro de objetivos - Ejemplo	16
Tabla 8. Plantilla del Cuadro de objetivos - Ejemplo.....	17
Tabla 9. Plantilla de la Estructura de una solicitud - Ejemplo	18
Tabla 10. Cuadro Procesos – servicios Proceso de solicitud de estado de cuenta	27
Tabla 11. Cuadro Procesos – servicios Proceso de Pago	29
Tabla 12. Cuadro Procesos – servicios Proceso de Reembolso de una deuda	30
Tabla 13. Cuadro Procesos – entidad Solicitud de una cuenta II	31
Tabla 14. Cuadro Procesos – entidad Pago de una deuda II	32
Tabla 15. Cuadro Procesos – entidad Reembolso de una deuda II	33
Tabla 16. Clasificación de Requerimientos.....	33
Tabla 17. Cuadro Servicio – entidad (Requerimientos).....	33
Tabla 18. Cuadro de Objetivos.....	36
Tabla 19. Puntaje de Objetivos	36
Tabla 20. Priorización de Servicios.....	37
Tabla 21. Lista de Servicios Priorizados	38
Tabla 22. Estructura Solicitud – Consultar deudas	38
Tabla 23. Estructura Respuesta – Consultar deudas.....	39
Tabla 24. Estructura Respuesta Error – Consultar deudas	41
Tabla 25. Estructura Solicitud – Pago de deudas	42
Tabla 26. Estructura Respuesta – Pago de deudas	43
Tabla 27. Estructura Respuesta Error – Pago de deudas	45
Tabla 28. Estructura Solicitud – Reembolso de una deuda	46
Tabla 29. Estructura Respuesta – Reembolso de una deuda	47
Tabla 30. Estructura Respuesta Error – Reembolso de una deuda.....	48
Tabla 31. Códigos de errores.....	50
Tabla 32. Parámetros función de Consulta.....	50
Tabla 33. Parámetros función de Pagos	52
Tabla 34. Parámetros función de Extorno	52
Tabla 35. Ficha técnica del servidor.....	53

Tabla 36. Pruebas Funcionales.....	56
Tabla 37. Pruebas de Carga.....	56
Tabla 38. Caso de uso N°1.....	57
Tabla 39. Caso de uso N°2.....	58
Tabla 40. Caso de uso N°3.....	59
Tabla 41. Caso de uso N°4.....	60
Tabla 42. Caso de uso N°5.....	61
Tabla 43. Caso de uso N°6.....	61
Tabla 44. Caso de uso N°7.....	62
Tabla 45. Caso de uso N°8.....	63
Tabla 46. Caso de uso N°9.....	64
Tabla 47. Caso de uso N°10.....	64
Tabla 48. Caso de uso N°11.....	65
Tabla 49. Caso de uso N°12.....	66
Tabla 50. Caso de uso N°13.....	67
Tabla 51. Caso de uso N°14.....	67
Tabla 52. Caso de uso N°15.....	68
Tabla 53. Estructura de la Tabla EstudianteDer.....	73
Tabla 54. Estructura de la Tabla EstudiantePrin.....	74
Tabla 55. Estructura de la Tabla Deuda.....	74
Tabla 56. Estructura de la Tabla Extorno_deuda.....	75
Tabla 57. Prueba Funcional N°1 Detalle.....	75
Tabla 58. Prueba Funcional N°1 Pruebas.....	76
Tabla 59. Prueba Funcional N°2 Detalle.....	78
Tabla 60. Prueba Funcional N°2 Pruebas.....	78
Tabla 61. Prueba Funcional N°3 Detalle.....	80
Tabla 62. Prueba Funcional N°3 Pruebas.....	80
Tabla 63. Tiempos de ejecución consulta.....	83
Tabla 64. Resumen de los resultados - Consultas.....	87
Tabla 65. Resultados de las pruebas consulta.....	88
Tabla 66. Tiempo Promedio por solicitud - Consulta.....	89
Tabla 67. Tiempos de ejecución pagos.....	91
Tabla 68. Resultados de las pruebas pagos.....	92
Tabla 69. Tiempo Promedio por solicitud - Pagos.....	93
Tabla 70. Tiempos de ejecución extorno.....	96
Tabla 71. Resultados de las pruebas extorno.....	97
Tabla 72. Tiempo Promedio por solicitud - Extorno.....	98

Índice de Gráficos

Gráfico 1. Prueba de Carga 1 - Consultas	86
Gráfico 2. Prueba de Carga 2 - Consultas	86
Gráfico 3. Prueba de Carga 3 - Consultas	87
Gráfico 4. Tiempo de respuestas - Consulta	88
Gráfico 5. Prueba de Carga 1 - Pagos	89
Gráfico 6. Prueba de Carga 2 - Pagos	90
Gráfico 7. Prueba de Carga 3 - Pagos	90
Gráfico 8. Tiempos de Respuesta - Pagos	93
Gráfico 9. Prueba de Carga 1 - Extorno	94
Gráfico 10. Prueba de Carga 2 – Extorno	95
Gráfico 11. Prueba de Carga 3 - Extorno	95
Gráfico 12. Tiempo de Respuesta - Extorno	98



CAPÍTULO I: DESCRIPCIÓN DEL PROYECTO

1. Descripción y Objetivos del Proyecto

1.1. Descripción del Problema

Uno de los principales problemas que se presentan dentro de las organizaciones, es que tienen muchas cuentas por pagar y por cobrar que son trasladadas al sistema financiero. Por la ausencia de un marco de trabajo que permita interconectar estas organizaciones con las entidades financieras hace que el proceso de los pagos y cobros de las cuentas sean más tediosos y demoren más tiempo para poder procesarse.

1.2. Objetivos del Proyecto

1.2.1. Objetivo General

Desarrollar un marco de trabajo que permita la implementación de un servicio API-REST en entidades educativas para el intercambio de datos con una entidad financiera para optimizar los procesos de pago dentro de las entidades educativas.

1.2.2. Objetivos Específicos

- Analizar los requerimientos de los procesos que van a ser implementados dentro de la interconexión con la entidad financiera.
- Modelar los datos que los procesos utilizarán durante la interconexión.
- Detallar la semántica de los procesos implementados mediante la elaboración de los diagramas UML (Lenguaje Unificado de Modelado).
- Establecer pruebas que permitan evaluar la velocidad y funcionamiento de los procesos implementados.

1.3. Alcances y Limitaciones

1.3.1. Alcances

El alcance del proyecto está enfocado en el desarrollo de un servicio API-REST utilizando PHP, que permita el intercambio de datos entre 2 entidades (institucional y financiera), permitiendo hacer el procesamiento de sus datos de manera eficaz, sin necesidad de recurrir a un elevado costo para la implementación de dicho o contar con una base de datos especializada para el almacenamiento y procesamiento de los datos, teniendo en cuenta que en el presente año (2022) se implementó un decreto (Decreto Legislativo N°1529) en el cual todas las personas naturales o jurídicas que prestan servicios deben de bancarizar sus operaciones a partir de S/2 000 o \$500, de manera que el marco propuesto ayudara a tener un control de las operaciones bancarizadas además contar con mayor seguridad al tener los fondos de las operaciones resguardados en las entidades financieras.

1.3.2. Limitaciones

- El presente proyecto, desarrolla la interconexión en una entidad educativa universitaria con una entidad financiera (1 a 1).
- El desarrollo de los servicios API-REST a implementar se hacen en backend, sin incluir el desarrollo de una plataforma web.
- El desarrollo del servicio API-REST está pensado para funcionar con JSON.
- El proyecto se desarrollará en una institución educativa superior en el departamento de Arequipa en el año 2022.

1.4. Fundamentos Teóricos

1.4.1. Webservices

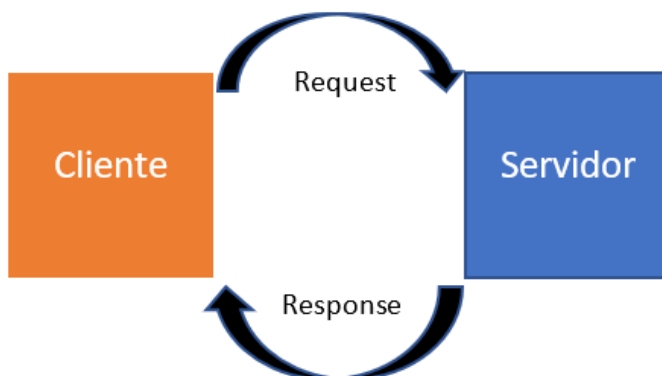
Es una aplicación que puede ser publicada e invocada a través de una red, normalmente Internet. A diferencia de tecnologías actuales, no se pueden acceder por protocolos específicos como el modelo de objetos, sino que se accede a ellos por medio de protocolos web como HTTP o XML. Además de que su interfaz está constituida por mensajes envío y respuesta, de tal manera que pueden ser consumidos e implementados en cualquier plataforma y lenguaje de programación, solo bastaría con la creación y consumo de los mensajes que se definan en su interfaz. (Lázaro, 2018)

1.4.2. Estándares en el desarrollo de WebServices

a) SOAP (Simple Object Access Protocol)

Protocolo utilizado para el intercambio de información en ambientes descentralizados y distribuidos, es el más utilizado al momento de realizar intercambio de información en los web services. Además, el protocolo está basado en XML, pudiendo combinarse con otros protocolos de comunicación (HTTP, es con el que se suele utilizar más), utilizando HTTP para el transporte de la información y XML para la presentación de esta. (Lázaro, 2018)

Figura 1. Modelo de Comunicación Soap



En la Figura 1, se observa el modelo de comunicación de SOAP, siendo parecido al de HTTP. El protocolo consiste en que un cliente hace una petición (request), el servidor escucha y atiende la petición y responde (response) enviando la información solicitada o el mensaje de error dependiendo el caso.

b) REST (Representational State Transfer)

Protocolo definido por Roy Fielding en el año 2002 en su tesis doctoral, el cual está basado en las URLs, en si el protocolo consiste en URLs a las que se pueden acceder por el protocolo HTTP para la obtención de información o realizar operaciones. Sus principales propiedades incluyen: capacidad de direccionamiento, interfaz limitada y uniforme, orientado a la representación y comunicación sin estado. (García Núñez, 2017)

1.4.3. Localizador Uniforme de Recursos de Internet (URL)

Según Corrales Rubiano y López Herrera (s.f), el URL es una secuencia de caracteres en un formato estándar utilizado para la localización de recursos en Internet, crea la conexión entre los recursos y el servidor, brindando información de del recurso y como acceder a él.

Figura 2. Ejemplo de la Estructura de URL



Fuente: Virginia Sánchez (2018), estructura de una URL.

1.4.4. JavaScript Object Notation (JSON)

Es un formato de datos basado en la sintaxis de los objetos en JavaScript, utilizado para el intercambio de información entre redes; describe los datos con una sintaxis que es independiente de cualquier plataforma, permitiendo que estos puedan utilizarse para intercambiar información entre diferentes aplicaciones, sin importar el lenguaje de programación utilizado en su desarrollo, gracias a que cada lenguaje tiene su propia librería para codificar y de-codificar en este estándar, permitiendo incluir diferentes tipos de datos en su estructura (números, cadenas, booleanos, arreglos, objetos y valores nulos). (Bengtsson, 2020)

1.4.5. Protocolo de Transferencia de Hipertexto (HTTP)

Es un protocolo de Internet usado para la transferencia y visualización de documento XML y de hipertexto (hipervínculos) en la web.

1.4.6. Métodos HTTP

García Nuñez (2017) describe que el protocolo HTTP utiliza generalmente 4 métodos para el intercambio de información:

- GET: Se usa para obtener un recurso del servidor, esta acción no cambia el estado de este. La petición GET devuelve un recurso en el campo entity-body.
- POST: Encargado de la creación de un nuevo recurso o modificarlo, se debe de evitar su uso con REST (en el caso de uso de formularios es inevitable su uso).
- DELETE: Permite la eliminación de un recurso, el servidor devuelve un mensaje de estado o puede no devolver nada.
- PUT: Utilizado para la creación o modificación de un recurso al igual que POST, la principal diferencia es que este método al ejecutarse varias veces, con los mismos valores genera el mismo resultado (por eso, es mejor para la actualización de datos).

1.4.7. API-REST

Una API de estado representacional (REST) o API de RESTFUL, es una interfaz de que se ajusta a la arquitectura REST y permite acceder a aplicaciones back-end de otra entidad.

1.4.8. Entorno de Desarrollo Integrado (IDE)

Un entorno de desarrollo integrado es un tipo de software compuesto de gran variedad de herramientas que faciliten el diseño y creación de nuevos softwares.

Según Dimitri Boboia (2018), un IDE está compuesto por: editor de código, compilador, interprete, depurador e interfaz gráfica.

Otro de sus aspectos más importantes es la de dar soporte a la productividad, permitiendo optimizarla reduciendo la configuración que se necesita, brindando distintas herramientas que permitan agilizar el ciclo de desarrollo de software.

1.5. Técnicas y Herramientas

1.5.1. Técnicas

a) Revisión Documental

Se utilizó esta técnica, para la recolección de información sobre el desarrollo de servicios API-REST y su implementación, asimismo se buscó información de servicios desarrollados para la consulta o pago de deudas en diferentes entidades.

1.5.2. Herramientas

a) JavaScript Object Notation

Estándar basado en texto plano utilizado para el intercambio de información, se utilizará para la recepción y envío de la información a través de los servicios API-REST.

b) Sublime Text

Según Florencio (2020), Sublime Text es un editor de texto que soporta gran variedad de lenguajes de programación como ASP, C, C++, PHP, Pascal, entre otros. Además, de que se encuentra disponible para diferentes plataformas como Linux, Windows y Mac.

c) Arquitectura Basada en Servicios (SOA)

Para Pascual (2020), la arquitectura basada en servicios permite la utilización de servicios (programas que realizan funciones específicas) dentro de los requisitos de la empresa. Además de basarse en principios que sirven de guía para la definición de las reglas de diseño: encapsulación, contrato, abstracción, reutilización, composición y descubrimiento.

d) Metodología Kanban

Según kanbanize (s.f), la metodología Kanban se enfoca en el desarrollo de las tareas pendientes y esta dividida en 4 principios con 6 prácticas.

- **Principio 1: Empezar con lo que hace ahora**

Kanban no requiere de una configuración y se puede aplicar sobre flujos de trabajo reales o procesos activos para la identificación de problemas, permitiendo implementarlo fácilmente en cualquier empresa ya que no se necesitan de cambios drásticos.

- **Principio 2: Comprometerse a buscar e implementar cambios incrementales y evolutivos**

Kanban se diseñó para implementarse con poca o mínima resistencia en las empresas, por lo que trata de pequeños y continuos cambios incrementales en el proceso actual, evitando los cambios radicales.

- **Principio 3: Respetar los procesos, las responsabilidades y los cargos actuales**

Kanban reconoce los procesos en curso (roles, responsabilidades, cargos, etc.) proveen a la empresa valor y vale la pena conservarlos; este método no restringe el cambio ni lo ordena, ya que puede generar miedo como para detener el progreso.

- **Principio 4: Animar el liderazgo en todos los niveles**

Principio más novedoso, donde la persona que está al frente de los equipos fomenten la mentalidad de mejora continua para alcanzar un rendimiento optimo a nivel de equipo o empresa.

e) PgAdmin4

Herramienta de código abierto que permite la administración de bases de datos en PostgreSQL, facilitando su administración. Entre algunas de sus funciones se encuentran:

- Herramienta de creación de consultas SQL.
- Editor de código (procedimientos almacenados).
- Terminal/batch/shell.

f) Postman

Plataforma para el desarrollo de Interfaz de programación de aplicaciones API-REST, originalmente desarrollado como una extensión para Chrome, actualmente se encuentra disponible para diferentes Sistemas Operativos (MAC, Windows), permite la ejecución de tareas como: creación de peticiones a APIs, elaboración de pruebas de verificación y creación de entornos de trabajo con definición de variables. (Redondo, 2021)

CAPÍTULO II: METODOLOGIA

En el presente capítulo mostramos la justificación, límites y alcances del proyecto, así mismo detallamos las diferencias entre los formatos de datos y metodologías de desarrollo ágil mediante cuadros comparativos, estas comparativas se hacen para mostrar porque se utilizó JSON como formato de datos y la metodología Kanban, la cual se eligió por ser más flexible que otras al no necesitar roles y reuniones, permitiendo el desarrollo en base a las tarjetas y el tablero Kanban donde se definen las actividades que se van a realizar durante nuestro proyecto.

Para finalizar el capítulo se detalla la estructura del marco de trabajo, donde explicamos la metodología Kanban, definimos las plantillas que usamos durante el Capítulo 3 y el plan de pruebas donde explicamos que tipos de pruebas se utilizaron para medir el rendimiento y funcionamiento de los servicios API-REST.

2. Justificación, aporte y metodología del proyecto

2.1. Justificación y Aporte del proyecto

2.1.1. Justificación Teórica

El desarrollo de un marco de trabajo para el intercambio de información entre una entidad educativa y financiera utilizando SOA y KANBAN favorecerá tanto a las entidades educativas, financieras, estudiantes y egresados en el optimizando los tiempos de pago al tener una o más interconexiones con diferentes entidades financieras, permitiendo que los alumnos puedan realizar los pagos de diferentes conceptos (matrícula, certificados, etc.) de manera más cómoda, además de poder controlar por parte de la entidad educativa cuando se realizaron los pagos para llevar un control más estable dentro de su institución. El servicio API-REST permitirá que se procesen los pagos de manera más segura con una conexión directa con la entidad financiera, evitando que pasen por otras verificaciones que atrasen el proceso.

2.1.2. Justificación Técnica

El proyecto empleará la metodología KANBAN y para la implementación de la interconexión se utilizarán las siguientes herramientas tecnológicas:

- Para el desarrollo de la capa lógica o acceso de datos:
 - PHP 7: Lenguaje de programación usado para la creación de páginas webs dinámicas, principalmente utilizado para la lectura de datos en formularios y convertirlos en variables de PHP.
 - JSON: Estándar de intercambio de datos, basado en el lenguaje de programación JavaScript.

- POSTGRESQL: Es un sistema gestor de base de datos relacional de código abierto que permite a todas las personas colaborar en el desarrollo de un proyecto o configurarlo según sus necesidades.
- PROCEDIMIENTOS ALMACENADOS: También llamadas funciones almacenadas, son un conjunto de sentencias o estructuras que se almacenan en los servidores de las bases de datos a fin de evitar la repetición de código, mejorando su eficiencia (Calvo, 2017).
- Para el desarrollo de las pruebas (velocidad y conexión):
 - cURL: Es una librería de PHP que permite realizar las peticiones HTTP mediante la transferencia de información con un URL (The PHP Group, s. f.).
 - Postman: Es una herramienta que permite realizar pruebas API, mediante “HTTP requests” con una interfaz gráfica de usuario, con la cual se obtendrá diferentes tipos de respuestas (Romero, 2021).

2.1.3. Justificación Operativa

- El servicio API-REST agilizará los pagos dentro de las instituciones educativas al permitir que los estudiantes puedan realizar pagos con agentes afiliados a la entidad financiera con la que tengan interconexión.
- El servicio API-REST posibilitará el pago mediante el aplicativo de los agentes y aplicativos móviles que la entidad financiera tenga disponible para realizar pagos.
- El servicio API-REST facilitará el pago de las deudas a los alumnos al brindar mayores opciones de entidades financieras con las cuales

2.1.4. Importancia e Impacto

La importancia e impacto que tiene el proyecto a desarrollar es que traerá beneficios para los alumnos al permitir realizar los pagos de manera más rápida a las entidades financieras con las que la entidad educativa tenga interconexión.

Para ello, se plantea el desarrollo de un marco de trabajo para la interconexión entre una entidad financiera y educativa, utilizando la metodología Kanban propuesta a inicios del siglo XXI, la cual buscaba mejorar la forma en la que se producían y entregaban los productos y servicios en la industria del software con la finalidad de optimizar los procesos de pago en las entidades educativas; y, en base al estudio de Joaquín Liaño Díaz (2017), revisar que tan viable es la “apifización” de los procesos de pago a través de la interconexión, los resultados del estudio van a permitir verificar que tan útil es su implementación dentro de las entidades educativas. Mientras que el estudio de Atencio Flores y Mamani Machaca (2017), al desarrollar una interfaz

de comunicación entre aplicaciones y un web service basado en API-REST, haciendo uso del lenguaje UML y la metodología SCRUM, validado a través del ISO 9126, nos permitirán ver como los procesos mejoran su calidad, hasta que el producto llegue a los estándares buscados por las entidades, de allí que planteamos desarrollar un marco de trabajo que permita llegar a ofrecer recomendaciones para el desarrollo de un servicio API-REST que utilicen JSON para la transferencia de datos, así mismo veremos la implementación y seguridad al servicio API-REST mediante un filtrado de IPs, así mismo como modelos guía para el desarrollo de los diagramas UML basados en servicios API-REST, donde se observará el flujo que se usará para los procesos de cada operación donde los principales beneficiados serán los estudiantes de las entidades educativas al poder realizar los pagos de manera más rápida además tendrán la opción de poder añadir más interconexiones pero evaluando la capacidad y rendimiento que necesitarían los servidores en base a las pruebas que se van a realizar.

2.2 Comparativa entre formatos de Datos

El marco de trabajo desarrollado utilizo el formato JSON para el envío y recepción de los datos a través de los servicios API-REST, a continuación, se muestra un cuadro comparativo donde se muestra las principales diferencias con otros formatos de datos (XML y YAML).

Tabla 1. Cuadro comparativo de formatos de datos

XML (eXtensible Markup Language)	JSON (JavaScript Object Notation)	YAML
Lenguaje basado en etiquetas (es una evolución de HTML) lo que dificulta su lectura.	Representa los datos en pares (key, value), donde la key, es el nombre de la variable y el value su valor.	No se enfoca en etiquetas, es un data encoding 'simple y claro'
Utilizado en servicios que manejen WSDL (Web Services Description Language), el cual es un estándar que se usa para describir los servicios desarrollados en XML.	Utilizado en servicios desarrollados que utilicen el concepto de listas y arreglos asociativos.	Utilizado para la gestión de redes, no es usado para desarrollo de servicios.
Difícil de comprender debido a la estructura que utiliza y las etiquetas que usa para dar el formato a los datos.	Mas simple de comprender gracias a su estructura y mapeado sobre estructuras de datos en lenguajes de programación más modernos.	Es fácil de comprender gracias a que incluye caracteres UNICODE, UTF-8 o UTF-16.

Pesa más a comparación de JSON en el envío de datos, debido a los caracteres que usa este formato.	Eficaz en la codificación, siendo más pequeño al enviar datos a comparación con otros formatos.	El peso de los archivos en este formato es muy liviano.
Los datos son representados con etiquetas que determinan la jerarquía.	La jerarquía de los datos se representa con doble espacio.	Los objetos se representan mediante corchetes. y los arreglos con llaves
Lleva tiempo en procesarse.	Se procesa con una velocidad alta.	-
Se pueden definir estructuras difíciles y reutilizarse.	Se pueden reutilizar sus estructuras.	Se pueden reutilizar sus estructuras.
Se adapta a los navegadores fácil debido a que está vinculado a HTML.	Se adapta fácilmente a navegadores debido a la relación que tiene con JavaScript.	No se suele utilizar en navegadore.
Desarrollado para el intercambio de datos gracias a la relacion que tiene con HTML y los archivos WSDL que manejan algunos lenguajes de programación.	Esta desarrollado para el intercambio de datos ya que permite la representación de varios elementos (objetos, matrices, cadenas, etc.)	No esta pensado para el intercambio de datos, ya que se dificulta al momento de parsear (formateo de dato).
Permite el uso de comentarios dentro de su estructura.	No acepta comentarios en su estructuración.	Permite el uso de comentarios dentro de su estructura.

2.3 Comparativa entre metodologías de desarrollo ágil

El marco de trabajo desarrollado utilizo la metodología Kanban para la implementación de los servicios, a continuación, se mostrarán las diferencias entre Kanban y Scrum, mostrando los motivos que llevaron a escoger Kanban como la metodología para nuestro proyecto.

Tabla 2. Cuadro comparativo Kanban vs Scrum

Kanban	Scrum
No hay iteraciones, es un trabajo continuo.	Trabaja con iteraciones de tiempo fijo (sprints)
El número de tareas está limitado por el flujo de trabajo.	Limita el número de tareas por las iteraciones
No existen roles.	Existen roles (scrum master, producto owner, etc.)
Permite que los equipos sean formados por especialistas.	Exige que los equipos sean multidisciplinarios
Permite la modificación de las tareas hasta que entran en el flujo.	No permite hacer cambio en las tareas de los sprints (se permite modificar la fecha de entrega, pero no la tarea en sí).
No mide ni tareas ni la velocidad.	Mide cuanto a llevar a realizar cumplir con cada historia de usuario, esfuerzo y tiempo en realizar las tareas.
No se considera la realización de reuniones ni de los gráficos.	Necesita realizar reuniones y gráficos.

2.4 Objetivos

Tabla 3. Validación de Objetivos

Objetivo General	
Objetivo	Validación
Desarrollar un marco de trabajo que permita la implementación de un servicio API-REST en entidades educativas para el intercambio de datos con una entidad financiera para optimizar los procesos de pago dentro de las entidades educativas.	El objetivo se validará en el Capítulo 2: Metodología, donde se presentará los pasos de la metodología SOA, para la identificación de los servicios para su implementación.
Objetivos Específicos	
Objetivo	Validación
Analizar los requerimientos de los procesos que van a ser implementados dentro la interconexión con la entidad financiera.	El objetivo se valida a través de la tabla que se presenta en el capítulo 3, que muestra los

	requerimientos que se necesitan para la implementación en el caso de estudio.
Modelar los datos que los procesos utilizarán durante la interconexión.	Este objetivo se valida en el capítulo 3, donde se detalla la estructura de la petición (request) y respuesta (response) de los servicios que se desarrollaron, así mismo se detalla el diccionario de datos para la implementación de cada uno de los servicios.
Detallar la semántica de los procesos implementados mediante la elaboración de los diagramas UML (Lenguaje Unificado de Modelado).	Este objetivo se valida en el capítulo 4, en el apartado de los diagramas UML, mostrando los componentes, entidades y la semántica que van a tener los servicios implementados.
Establecer pruebas que permitan evaluar la velocidad y funcionamiento de los procesos implementados.	El objetivo se valida en el capítulo 4, donde se muestran los resultados de las pruebas que se realizaron (pruebas de velocidad y funcionamiento de los servicios).

2.5 Diseño del Proyecto (Marco de Trabajo)

2.5.1. Componentes del Marco de trabajo

El marco de trabajo desarrollado está conformado por los siguientes componentes: la metodología Kanban y la arquitectura orientada a servicios (SOA).

Figura 3. Componentes del marco de trabajo



La figura 3, nos muestra los componentes utilizados en el marco de trabajo así como las actividades que se utilizan para la incorporación de la arquitectura SOA y la metodología KANBAN (descripción de los elementos utilizados).

2.5.2. Metodología Kanban

La metodología Kanban permite la gestión de proyectos por tableros y tarjetas Kanban, donde se definen las tareas que se van a desarrollar, están en desarrollo o se finalizaron, el proyecto hace uso de esta metodología al poner las actividades en las tarjetas, donde se define el nombre de la actividad, duración (en días) y la prioridad que va a tener en el proyecto.

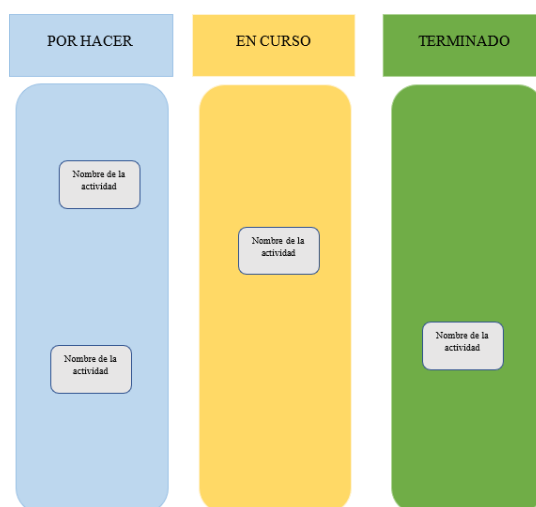
La siguiente imagen, muestra la plantilla de la tarjeta Kanban que se uso en el desarrollo del proyecto.

Figura 4. Tarjeta Kanban (Plantilla)

Nombre de la actividad
Prioridad:
Duración:
Descripción de la actividad que se va a desarrollar.

Acto seguido, se comenzaba a llenar el tablero Kanban con las tareas que se identificaron, como se muestra en la Figura 5.:

Figura 5. Tablero Kanban (Plantilla)



Una vez que una actividad se complete, pasa a la fila “Terminado” y al finalizar con una actividad se actualizarán las demás filas para posicionarse en donde le correspondería dentro de las actividades del proyecto.

2.5.3. Arquitectura Orientada a los Servicios (SOA)

El SOA es utilizado en la implementación de servicios en una organización, en el desarrollo de nuestro proyecto en la estructuración de SOA, se tomó como referencia las tablas presentadas en el trabajo de “Diseño de una arquitectura orientada a servicios para un establecimiento de salud de nivel de complejidad I-3”, *Aguilar, 2013*:

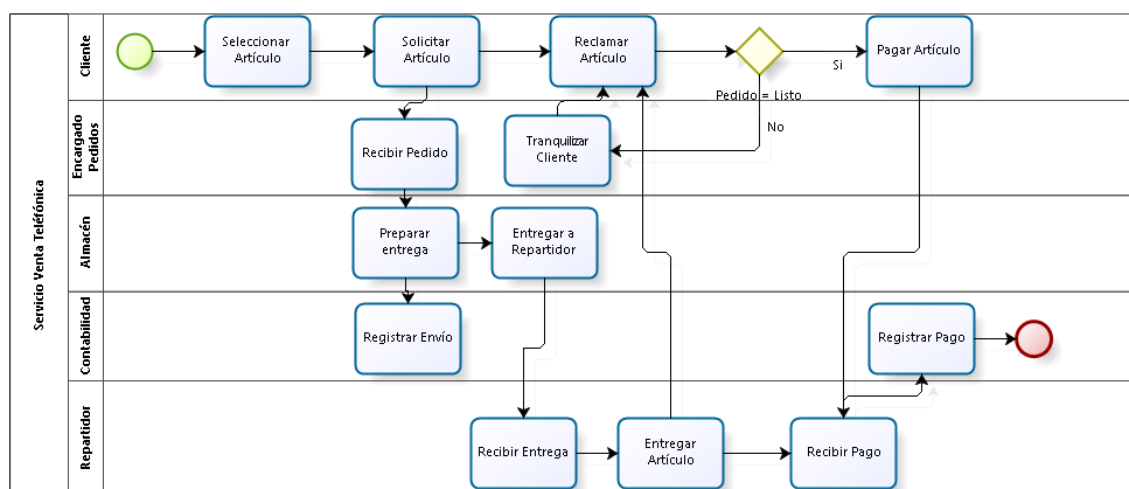
a) Identificación de los Servicios

Primera etapa de la implementación de la arquitectura SOA, donde se presenta el mapa de procesos de la organización, eligiéndose los procesos de acuerdo con el mapa de procesos de la institución sobre los cuales se va a trabajar para la identificar los procesos candidatos para el desarrollo de la interconexión.

Una vez se tienen los procesos, se procede a desarrollar el BPMN (Business Process Model and Notation), para la identificación de actividades que pueden pasar a convertirse a servicios.

Para nuestros fines y objetivos presentamos como modelo de BPMN de una compañía telefónica, el mismo que se puede observar en la Figura 5. Ejemplo de BPMN, donde se explica el funcionamiento del servicio de venta telefónica, así como los participantes que intervienen en dicho proceso.

Figura 6. Ejemplo de BPMN



Fuente: Deberrión (2018), *Análisis de Sistemas*.

Luego se procede a llenar la Tabla 4. Plantilla de cuadro de procesos – servicios, donde se muestran los servicios identificados para el proceso antes detallado.

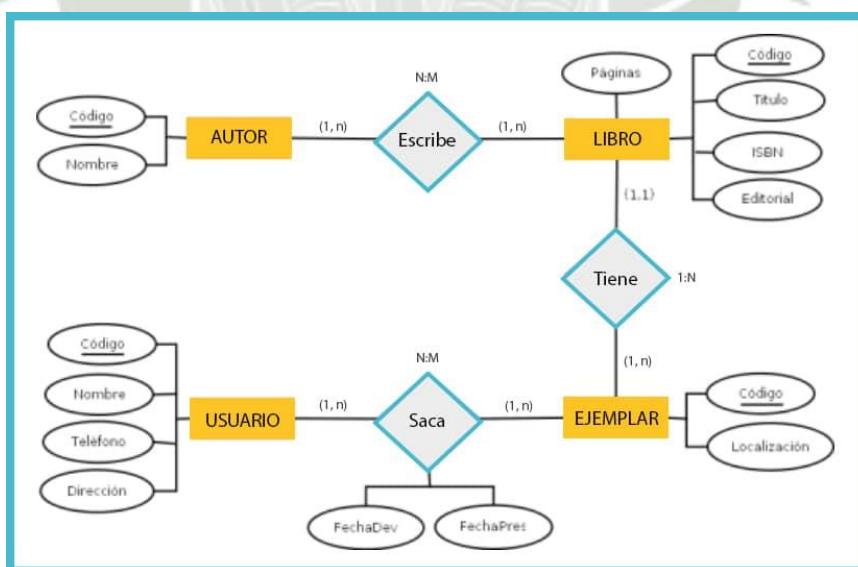
Tabla 4. Plantilla de cuadro de procesos – servicios - Ejemplo

Cuadro de procesos – servicios		
Proceso del negocio	Servicio identificado	
	Código	Nombre
P001 – Proceso 01	S01	wsS01
	S02	wsS02
	S03	wsS03

b) Relación entre servicios y entidades

En esta fase se desarrolló el diagrama entidad-relación de cada proceso para la identificación de los participantes, permitiendo conocer a mayor rasgo cómo funciona y quienes participan durante la ejecución de este, con la finalidad de observar quienes tienen y comparten entidades en su ejecución.

Figura 7. Ejemplo de un diagrama Entidad-Relación



Fuente: ILERNA ONLINE (2019), EJEMPLO DE LA REPRESENTACIÓN DE UN MODELO ENTIDAD-RELACIÓN.

Luego se procede a llenar la Tabla 5. Cuadro de procesos – entidad, donde se muestran las entidades y a que procesos pertenecen.

Tabla 5 Plantilla del cuadro de procesos – entidad - Ejemplo

Cuadro Proceso – Entidad			
Proceso del Negocio	Servicio		Entidad
	Código	Nombre	Nombre
P001 – Proceso 1	S01	wsS01	Entidad 1 Entidad 2
	S02	wsS02	Entidad 3 Entidad 2
	S03	wsS03	Entidad 4

c) Requerimientos de los servicios identificados

Fase donde lista los requerimientos que se necesitarán para la creación de los servicios identificados, para lo cual se necesita clasificarlos en funcionales y no funcionales dependiendo del tipo de requerimiento solicitado.

Tabla 6. Plantilla del Cuadro de procesos – servicios - Ejemplo

Servicio – Entidad				
Código	Nombre del servicio	Requerimientos		
		Código	Tipo	Descripción
S01	wsS01	RS01	F	
		RS02	F	
		RS03	F	
		RS04	NF	

d) Priorización de los Servicios

Fase donde inicialmente se valora los objetivos que van ligados a los procesos para poder identificar a cuáles procesos pertenecen.

Tabla 7. Plantilla del cuadro de objetivos - Ejemplo

Objetivo				
	Objetivo 1	Objetivo 2	Objetivo 3	Objetivo 4
wsS01	X	X	X	
wsS02	X	X		X
wsS03		X		
Total	2	3	1	1

Con los resultados de la tabla anterior, se procede a valorizar los objetivos (alto=3, medio=2 y bajo=1) como se muestra en la siguiente tabla, para poder obtener la lista de los servicios priorizados y ver qué servicios se van a desarrollar de acuerdo a lo que la empresa haya solicitado.

Tabla 8. Plantilla del Cuadro de objetivos - Ejemplo

Servicio	Objetivos				Total
	Permitir la consulta de datos con entidades externas	Agilizar procedimientos internos y/o externos en la organización	Optimizar los procesos de consultas y pagos en entidades financieras	Actualizar registros en las BD	
wsS01	0	3	2	0	5
wsS02	1	2	0	1	4
wsS03	0	2	0	0	2

e) Descripción de los servicios

Apartado donde se describe la función que tiene el servicio que se está implementando, de igual forma se detalla la estructura de la petición y respuesta que los servicios deben devolver.

Tabla 9. Plantilla de la Estructura de una solicitud - Ejemplo

#	Path	Obligatorio	Longitud	Tipo	Descripción
1	Param1	Si	2	Caracter	Parámetro numero 1
2	Param2	No		Numerico	Parámetro numero 1

2.6 Implementación

En esta sección, se definirá como se realizó la implementación de los servicios en el servidor principal de la entidad educativa.

2.6.1. Características del Servidor

Se muestran la ficha técnica, donde se puede observar la capacidad de procesamiento, memoria RAM, capacidad del disco duro, entre otros.

2.6.2. Instalación de las herramientas

Comienza con la verificación de los softwares, caso de contar con el software (editor de código, administrador de bases de datos, etc), se procede a actualizar los que se vean necesarios para la implementación, verificando que las actualizaciones de no pongan en riesgo otros servicios que se ejecuten dentro del servidor.

2.7 Plan de Pruebas

Las pruebas son un proceso fundamental en el desarrollo de software, permitiendo detectar las fallas, errores, que se presentan en los sistemas. Ejecutando las pruebas en los servicios desarrollados, se probará la eficiencia y la velocidad de respuesta durante que tienen cuando son invocados por otros sistemas.

2.7.1. Propósito

El plan de pruebas tiene como objetivo la creación de pruebas en los servicios (API-REST) necesarios para la interconexión con una entidad financiera, identificando los posibles errores que se puedan presentar, así como los tiempos de respuesta que se manejan.

2.7.2. Alcance

El presente plan de pruebas se llevará a cabo con todos los servicios desarrollados para la interconexión con la entidad financiera permitiendo la identificación de errores que afecten la operatividad para su corrección, así como la optimización del código en caso la ejecución de los servicios sea mayor de lo que se requiere para la interconexión.

2.7.3. Estrategia de Pruebas

En este momento, se define como realizar las pruebas y los tipos de pruebas, como se aplicarán, en qué momento serán aplicadas y como se realizará la evaluación.

- Pruebas Funcionales: Comprobarán que los servicios se ejecuten de manera correcta.
- Pruebas de Carga: Comprobarán que los servicios funcionen correctamente enviándole una gran cantidad de solicitudes, lo que permitirá verificar el tiempo de respuesta del servidor a cada petición realizada.

Cabe destacar que se pueden implementar las pruebas de estrés y seguridad; siendo la primera la que nos ayudaría a verificar cuantas solicitudes puede aguantar cada servicio, mientras que la de seguridad nos proporcionaría información de que tan seguro es cada servicio implementado en el servidor.

CAPÍTULO III: CASO DE ESTUDIO

El desarrollo de todo proyecto debe seguir una ruta pragmática, en donde se detalle las tareas y actividades (que se deben de realizar en la ejecución) de dicho proyecto. En esta etapa se detalla a fondo las necesidades (requerimientos), el uso que se dieron a las herramientas, la metodología, arquitectura, los componentes y las diversas actualizaciones.

3. Desarrollo del caso de estudio

3.1 Metodología Kanban

El proyecto se desarrolló utilizando la metodología Kanban, para la cual se planificaron actividades, las cuales se representaron gráficamente con tarjetas ordenadas en 3 columnas que representan los estados en los que se encuentran las actividades.

Cada tarjeta es agregada según las necesidades que surjan durante el desarrollo del proyecto, a continuación, se muestran las tarjetas, así como los estados en que se encontraban las demás actividades mientras se iban añadiendo más actividades durante la ejecución del proyecto.

Para una primera etapa se crearon 5, tarjetas siendo la de “Identificación de los servicios”, la única que se encontraba en curso durante la primera etapa.

Figura 8. Tablero Kanban 1ª Etapa

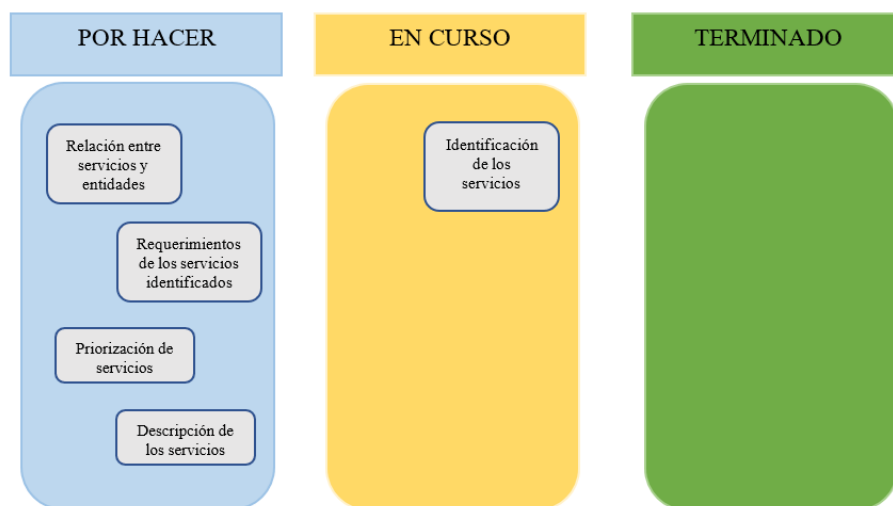


Figura 9. Tarjeta de Identificación de servicios

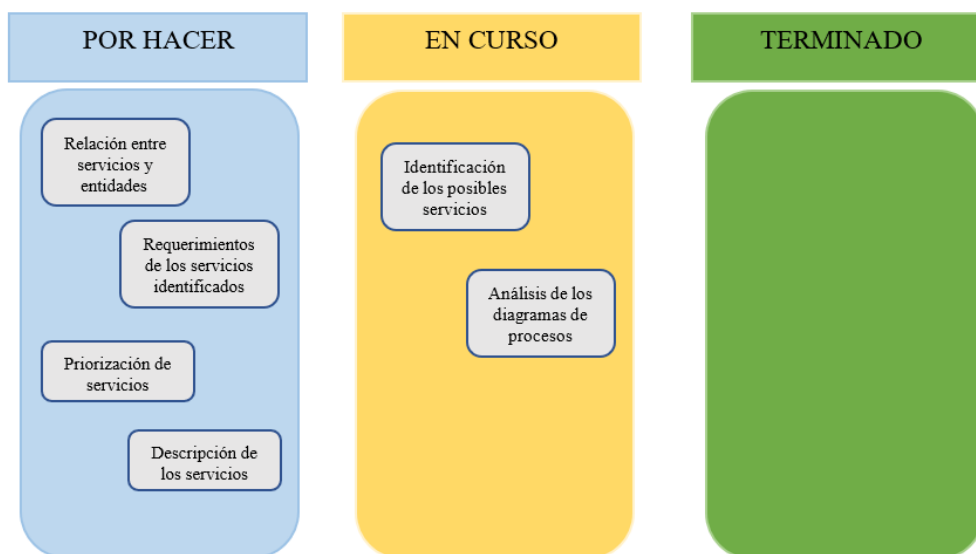
Identificación de los servicios
Prioridad: Alta Duración: 10 días
Proceso de partida para la implementación de la arquitectura SOA, donde se evalúan los procesos en el área donde se requieran servicios.

Así mismo durante la identificación de servicios, se procedió a dividir la actividad, la primera consistía en identificar los posibles servicios en base al mapa de procesos de la organización y la segunda, analizar los diagramas de procesos donde se haya detectado un posible servicio.

Figura 10. Tarjetas en estado “Por hacer”

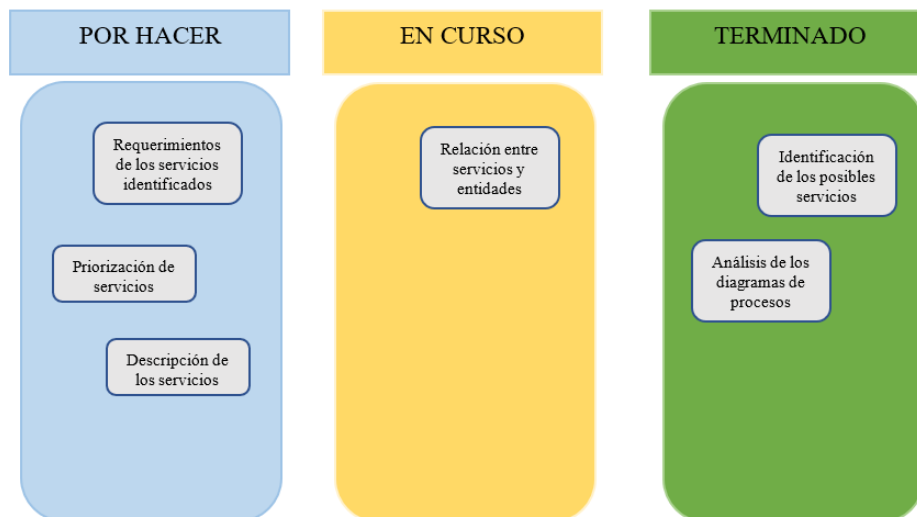
Identificación de posibles servicios	Análisis de los diagramas de procesos
Prioridad: Alta Duración: 1 semana	Prioridad: Alta - media Duración: 2 semanas
Con ayuda del mapa de procesos de la entidad educativa se comienza a identificar los posibles servicios en base a sus procesos.	A través de la documentación de la entidad, se comienza a analizar los procesos para la identificación de servicios candidatos a través de las actividades que se realizan.

Figura 11. Tablero Kanban Actualización 1



Una vez terminada ambas actividades, se procedió a realizar la tabla de servicios candidatos en base a las actividades de cada proceso identificado, dando por finalizado la primera etapa de la implementación de SOA.

Figura 12. Tablero Kanban Actualización 2



Una vez se tiene los servicios candidatos, se procede a identificar la relación que se tienen con las entidades que participan en los procesos, para lo cual se realizan 2 actividades, analizar los diagramas de entidad-relación y elaborar la tabla de relaciones de los servicios con las entidades.

Figura 13. Tablero Kanban Actualización 3

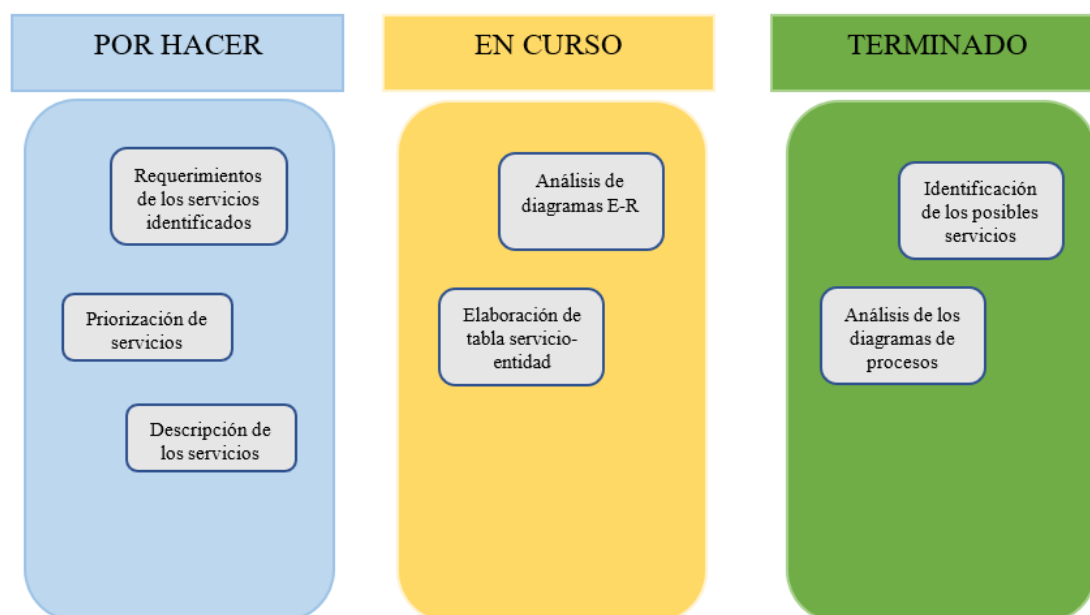
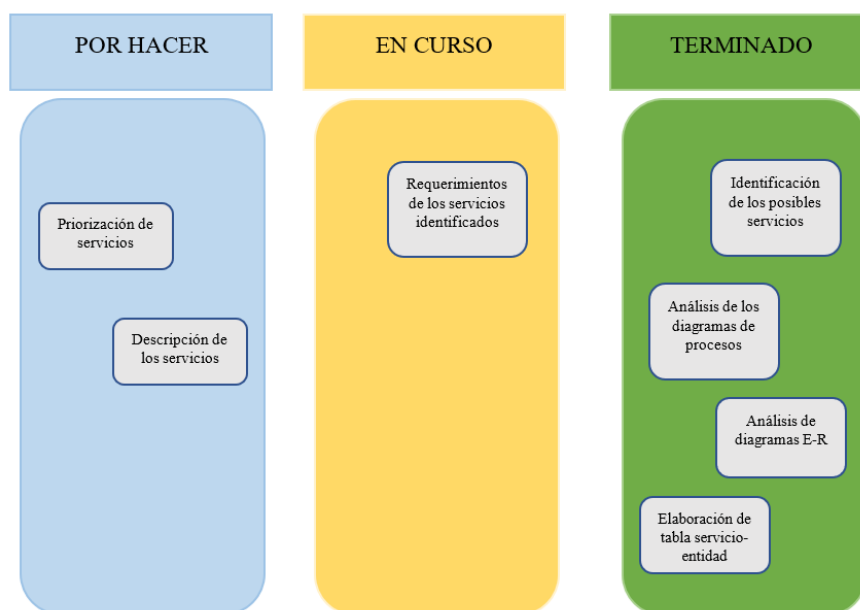


Figura 14. Actividades: análisis y elaboración de tabla

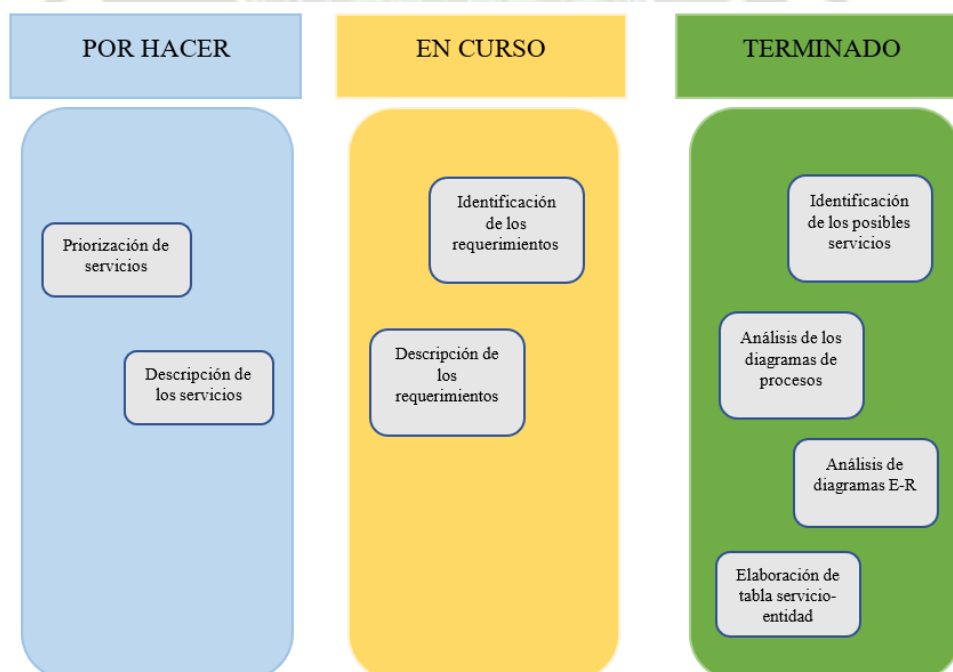
Análisis de diagramas E-R	Elaboración de tabla servicio-entidad
Prioridad: Media	Prioridad: Alta
Duración: 6 días	Duración: 1 semana
A través de los diagramas E-R obtenidos de la documentación, se verifican las entidades que participan en los procesos de los servicios candidatos.	Creación de una tabla donde se verifica las relaciones de las entidades y cuales servicios comparten entidades.

Figura 15. Tablero Kanban actualización 4



Luego de haber elaborado las tablas y tener identificado las entidades con sus respectivos servicios candidatos se procede a identificar y describir los requerimientos para los servicios identificados.

Figura 16. Tablero Kanban actualización 5



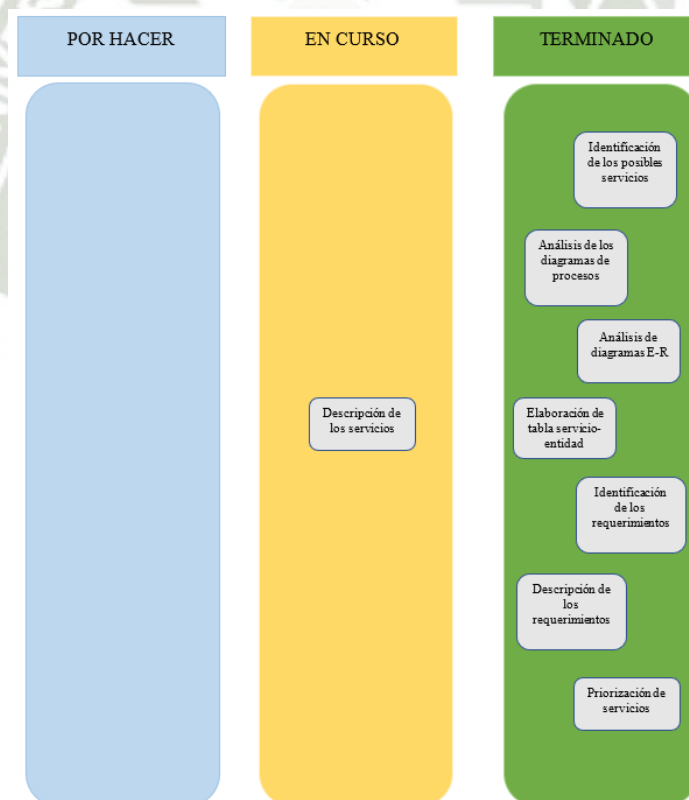
De igual forma se crean las funciones que van a ser invocadas dentro de los servicios API-REST, como se muestra a continuación.

Figura 17. Tarjetas – Identificación y descripción de requerimientos

Identificación de los requerimientos	Descripción de los requerimientos
Prioridad: Alta Duración: 3 días	Prioridad: Alta Duración: 2 días
Elaboración de la lista de los requerimientos en base a las necesidades de la entidad.	Describir cada requerimiento listado además de clasificarlo en funcional o no funcional.

Después de haber descrito y clasificado los requerimientos, se procede a priorizar los servicios identificados para elaborar los que la entidad educativa vea necesario, para este caso, se solicitó que fueran los 3 primeros.

Figura 18. Tablero Kanban actualización 6



Para finalizar con la implementación de SOA, se procede a describir cada servicio que vaya a ser elaborado para su implementación, para luego dar paso a la fase de elaboración de los servicios (API-REST).

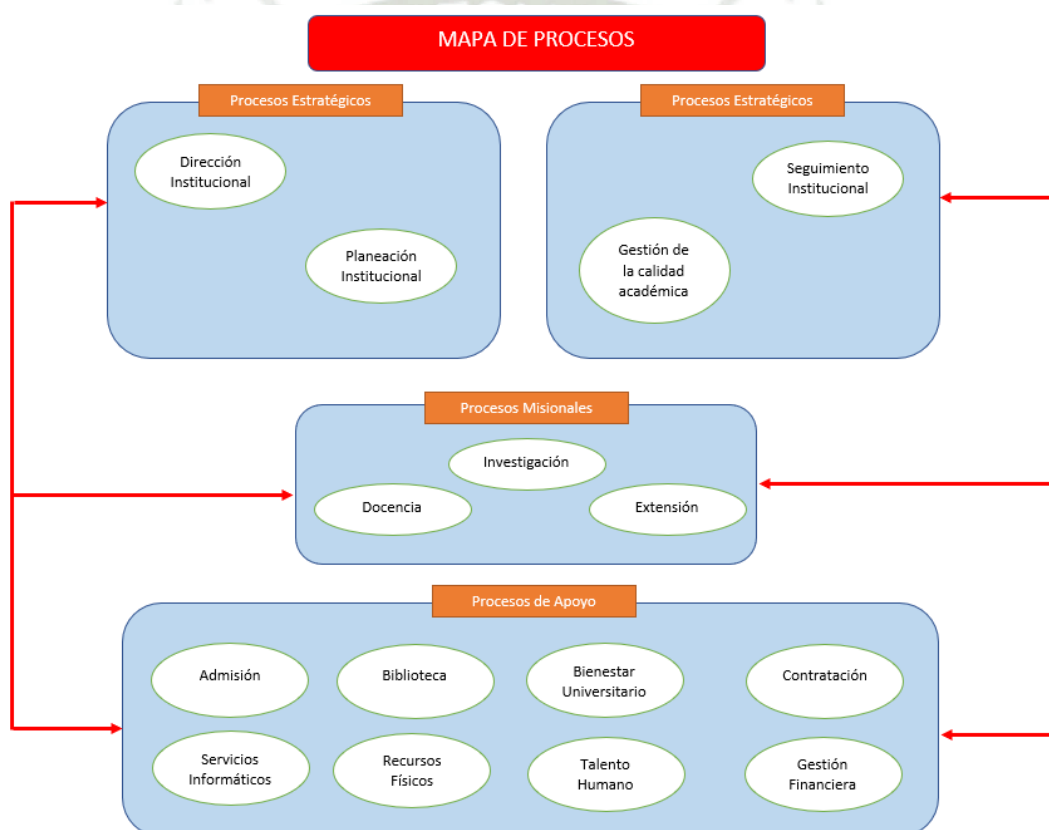
3.2 Arquitectura

La arquitectura utilizada en el desarrollo del Proyecto es la Orientada a Servicios (SOA), para lo cual se siguió los siguientes pasos para su implementación dentro de la entidad educativa:

3.2.1 Identificación de los servicios

Punto de partida para la implementación de una arquitectura Orientada a Servicios, donde se comienza con la identificación de los servicios a través de los procesos; en las entidades educativas se pudo identificar la presencia de 3 tipos de procesos: Procesos estratégicos, misionales y de apoyo.

Figura 19. Mapa de Procesos – Entidad Educativa Superior



Los 3 procesos a partir de los cuales se identificaron los servicios son:

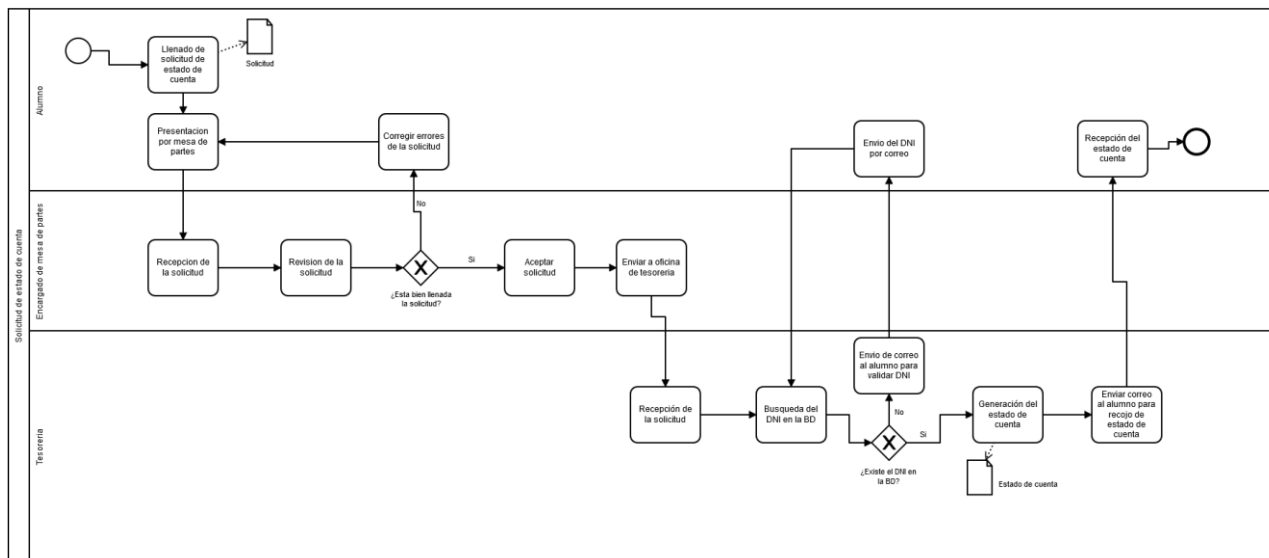
- Solicitud de estado de cuenta
- Pago de deudas
- Reembolso de gastos

A continuación, se listaron los procesos, detallando cada uno de los servicios candidatos identificados en cada uno de los procesos.

a) Solicitud de estado de cuenta

En este proceso, el alumno presenta por mesa de partes una solicitud para la verificación de las deudas que se tiene con la entidad educativa, mostrando a detalle que deudas tiene pendiente por pagar según el número de DNI que haya presentado en la solicitud.

Figura 20. Proceso de solicitud de estado de cuenta



En la Tabla 10. Cuadro Procesos – servicios Proceso de solicitud de estado cuenta, se presentan, los servicios identificados a partir del proceso presentado en la Figura 19.

Tabla 10. Cuadro Procesos – servicios Proceso de solicitud de estado de cuenta

Procesos – servicios		
Proceso del negocio	Servicio identificado	
	Código	Nombre
P001 – Solicitud de estado de cuenta	S01	wsConsultaDeudas
	S02	wsConsultaAlumnos
	S03	wsGenerarEstadoCuenta

b) Pago de deudas

Proceso por el cual los alumnos realizan el pago de las deudas que tienen pendientes con la entidad educativa, el proceso abarca desde la admisión a la ventanilla de tesorería hasta la recepción del voucher de pago.

Figura 21. Proceso de Pago – 1

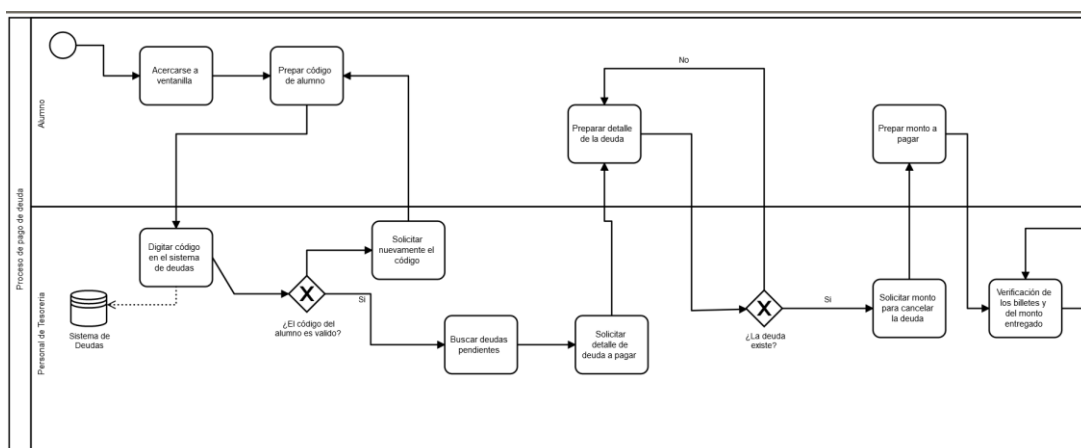
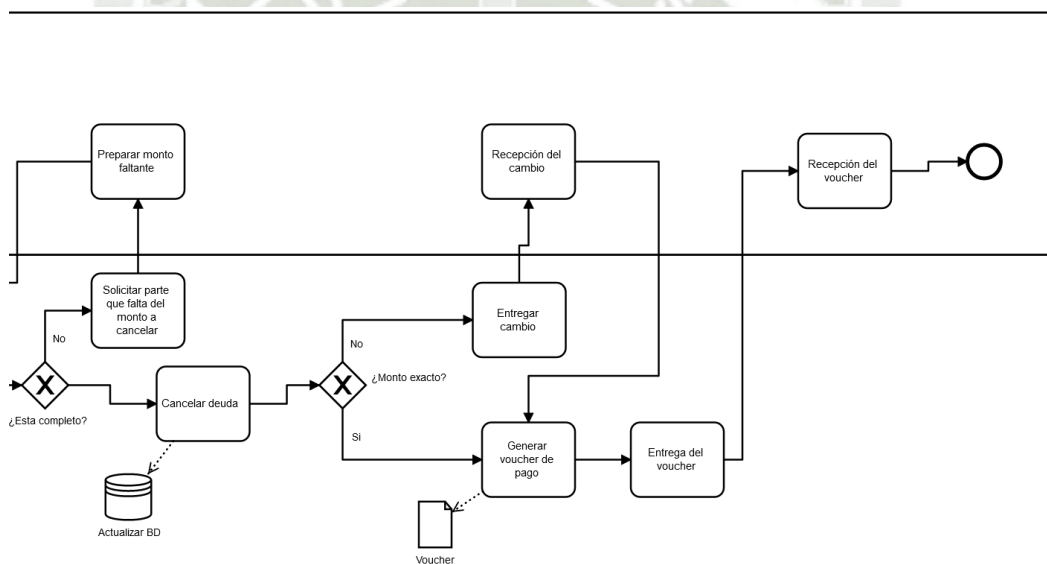


Figura 22. Proceso de Pago – 2



En la Tabla 11. Cuadro Procesos – servicios Proceso de Pago se presentan, los servicios identificados a partir del proceso presentado en las Figura 21 y 22.

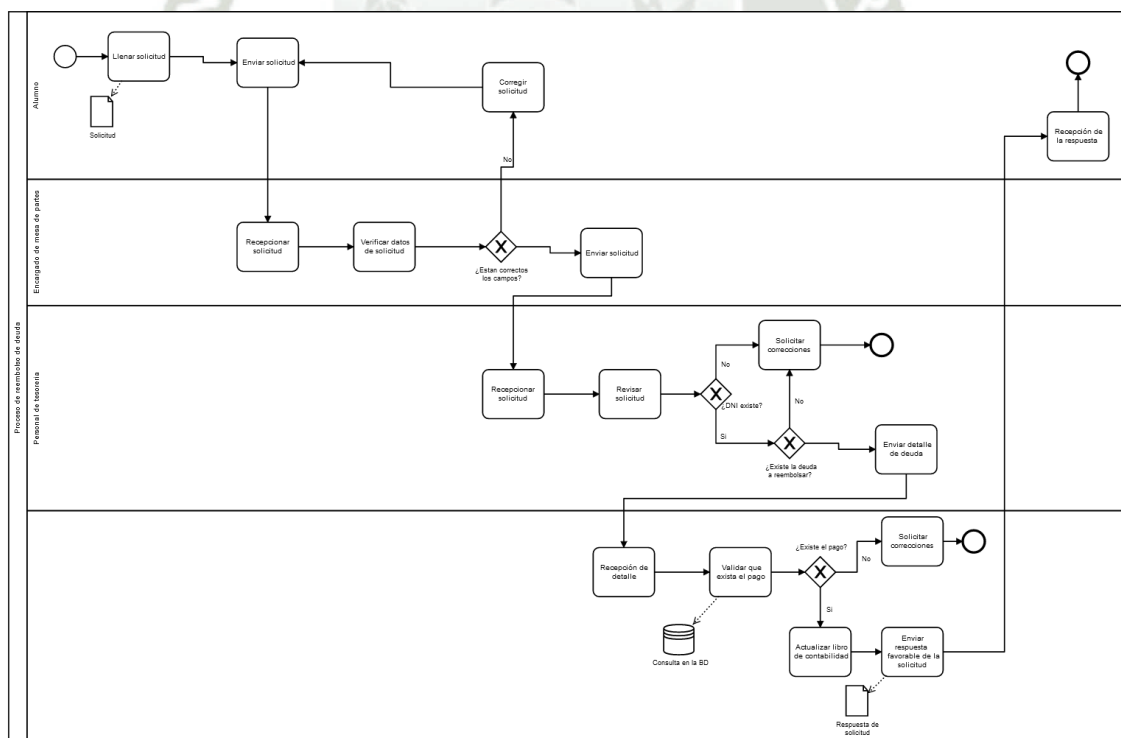
Tabla 11. Cuadro Procesos – servicios Proceso de Pago

Procesos – servicios		
Proceso del negocio	Servicio identificado	
	Código	Nombre
P002 – Pago de una deuda	S04	wsPagosDeuda
	S01	wsConsultaDeudas
	S05	wsGenerarVoucherPago

c) Reembolso de una deuda

Proceso por el cual el alumno solicita el reembolso de una deuda, este proceso abarca desde la presentación de la solicitud hasta la aprobación del reembolso de la deuda.

Figura 23. Proceso de reembolso de una deuda



En la Tabla 12. Cuadro de Procesos – servicios Proceso de reembolso de una deuda, se presentan, los servicios identificados a partir del proceso presentado en la Figura 23.

Tabla 12. Cuadro Procesos – servicios Proceso de Reembolso de una deuda

Procesos – servicios		
Proceso del negocio	Servicio identificado	
	Código	Nombre
P003 – Reembolso de una deuda	S06	wsExtornoPago
	S07	wsConsultarPagos
	S08	wsActualizarLibroContabilidad

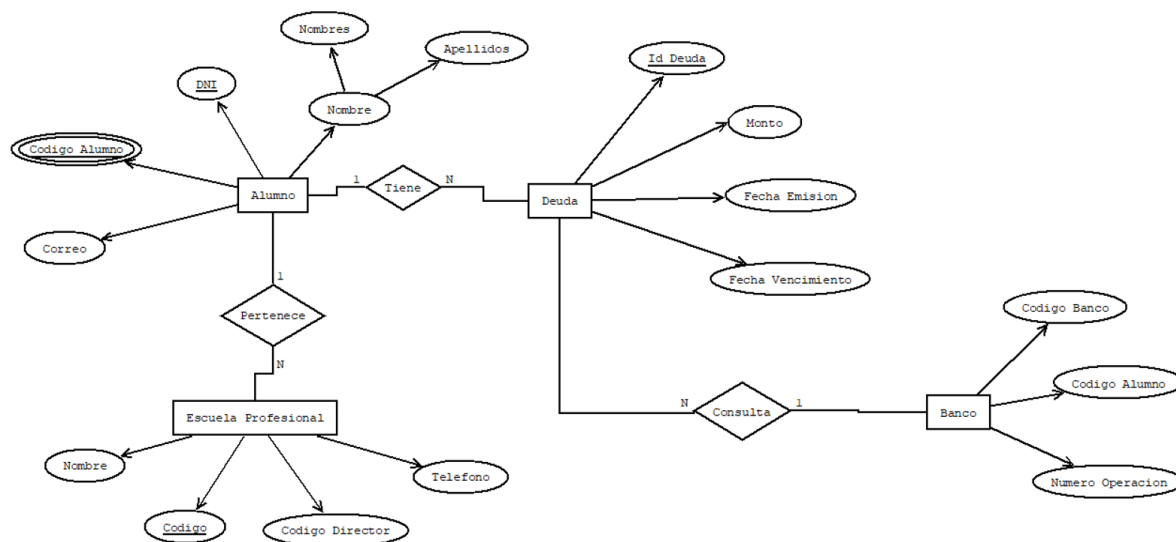
3.2.2 Relación entre servicios y entidades

Para cada servicio candidato que se encontró en los procesos seleccionados, es importante ver como se relacionan con las entidades que participan en el servicio.

a) Solicitud de estado de cuenta

En la Figura 24, se muestra el modelo de entidad-relación que corresponde al proceso de solicitud de estado de cuenta.

Figura 24. Diagrama entidad relación – Solicitud de estado de cuenta



A partir del diagrama entidad-relación, se pudo identificar a las entidades que participaron en los servicios identificados en el proceso de “Solicitud de estado de cuenta”

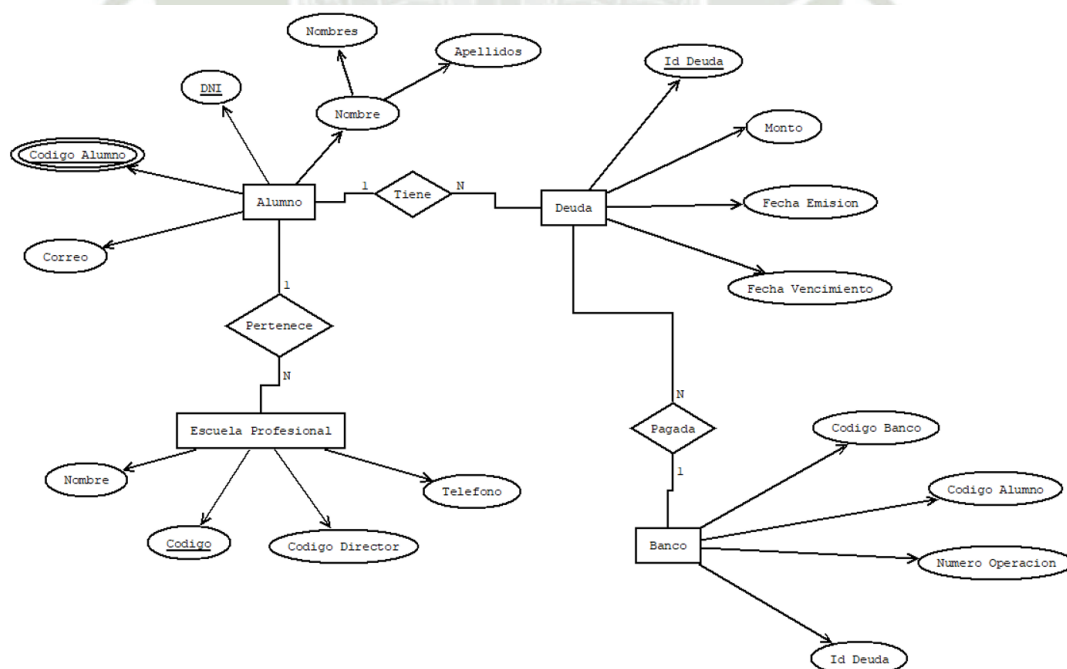
Tabla 13. Cuadro Procesos – entidad Solicitud de una cuenta II

Proceso – Entidad			
Proceso del Negocio	Servicio		Entidad
	Código	Nombre	Nombre
P001 – Solicitud de estado de cuenta	S01	wsConsultaDeudas	Alumno Deuda Escuela Profesional Banco
	S02	wsConsultaAlumnos	Alumno Escuela Profesional
	S03	wsGenerarEstadoCuenta	Alumno Deuda

b) Pago de deudas

En la Figura 25, se muestra el modelo de entidad-relación que corresponde al proceso de pago de deudas

Figura 25. Diagrama entidad relación – Proceso de pago de deudas



A partir del diagrama entidad-relación, se pudo identificar a las entidades que participaron en los servicios identificados en el proceso de “Proceso de pago de deudas”

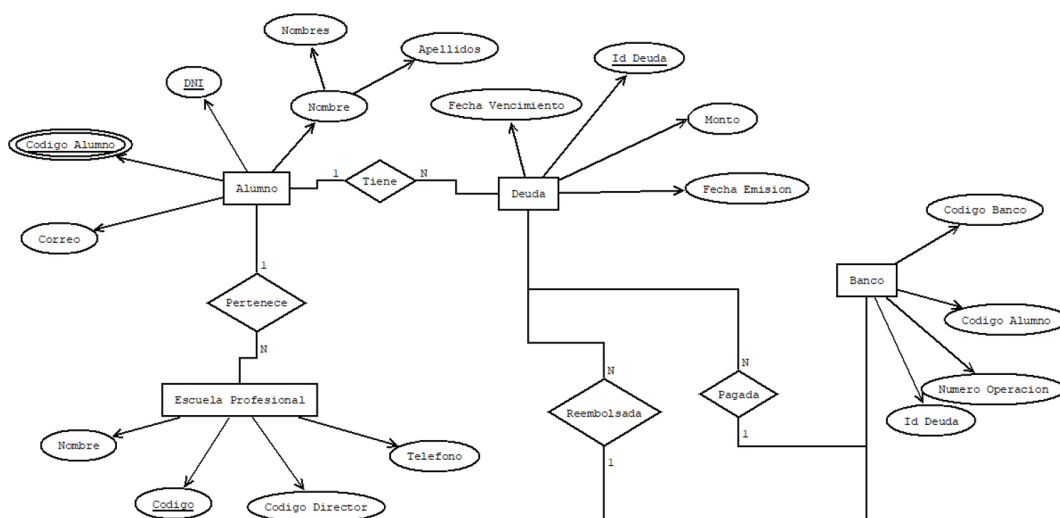
Tabla 14. Cuadro Procesos – entidad Pago de una deuda II

Cuadro Proceso – Entidad			
Proceso del Negocio	Servicio		Entidad
	Código	Nombre	Nombre
P002 – Pago de deudas	S04	wsPagosDeuda	Alumno Deuda Escuela Profesional Banco
	S01	wsConsultaDeudas	Alumno Escuela Profesional Deuda Banco
	S05	wsGenerarVoucherPago	Banco Deuda

c) Reembolso de una deuda

En la Figura 26, se muestra el modelo de entidad-relación que corresponde al proceso de reembolso de una deuda.

Figura 26. Diagrama entidad relación – Proceso de reembolso de una deuda



A partir del diagrama entidad-relación, se pudo identificar a las entidades que participaron en los servicios identificados en el proceso de “Reembolso de una deuda”

Tabla 15. Cuadro Procesos – entidad Reembolso de una deuda II

Cuadro Proceso – Entidad			
Proceso del Negocio	Servicio		Entidad
	Código	Nombre	Nombre
P003 – Reembolso de una deuda	S06	wsExtornoPago	Alumno Deuda Escuela Profesional
	S07	wsConsultarPagos	Alumno Escuela Profesional

3.2.3 Requerimientos de los servicios identificados

La clasificación de los requerimientos se usó la siguiente tabla:

Tabla 16. Clasificación de Requerimientos

Requerimiento funcional	F
Requerimiento no funcional	NF

Una vez que se tiene la clasificación de los requerimientos, se comienza a la determinación de los requerimientos por cada servicio, como se puede observar en la siguiente tabla:

Tabla 17. Cuadro Servicio – entidad (Requerimientos)

Servicio – Entidad				
Código	Nombre del servicio	Requerimientos		
		Código	Tipo	Descripción
S01	wsConsultaDeudas	RS01	F	Validar longitud de los parámetros enviados (DNI, fecha, etc.)
		RS02	F	Validar que el DNI enviado exista en la BD.
		RS03	F	Retornar la lista de deudas con su respectivo detalle e identificador (ID).

		RS04	NF	Retornar las ultimas 10 deudas que el alumno tiene pendiente.
S02	wsConsultaAlumnos	RS05	F	Verificar la longitud del DNI o código de alumno.
		RS06	F	Retornar el respectivo error en caso no se encuentre DNI o código en la BD.
S03	wsGenerarEstadoCuenta	RS07	F	Verificar la consulta correcta de todas las deudas pendientes.
		RS08	NF	Grabar la fecha actual en la que se va a generar el estado de cuenta
		RS09	F	Añadir la opción para filtrar las deudas en un determinado periodo de tiempo.
S04	wsPagosDeuda	RS01	F	Validar longitud de los parámetros enviados (DNI, fecha, etc.)
		RS02	F	Validar que el DNI enviado exista en la BD.
		RS10	F	Validar que el estudiante tenga deudas pendientes, antes de realizar la transacción.
		RS11	F	Generación del número de operación (Id de la deuda).
		RS12	F	Retorno de error y descripción, si no se pudo realizar el pago.
		RS13	F	Retorno de respuesta correcta al realizar el pago.

S05	wsGenerarVoucherPago	RS14	F	Validar el número de la operación exista.
		RS15	F	Validar que el pago se realizó correctamente.
		RS16	F	Obtención de la fecha en la que se realizó el pago.
S06	wsExtornoPago	RS01	F	Validar longitud de los parámetros enviados (DNI, fecha, etc.)
		RS02	F	Validar que el DNI enviado exista en la BD.
		RS17	F	Verificar que el número de la operación exista.
		RS18	F	Generación y envío de errores por numero de pago no encontrado o no pagado.
		RS19	F	Validar que el pago se haya realizado el mismo día que se quiere realizar el extorno.
S07	wsConsultarPagos	RS20	F	Verificación de la fecha en que se va a extornar.
		RS17	F	Validar que el número de la operación exista.
		RS21	NF	Verificar que se devuelvan los 10 últimos pagos realizados por el alumno.
S08	wsActualizarLibroContabilidad	RS22	NF	Realizar una opción que permita el filtrado de la consulta de pagos por año.
		RS17	NF	Verificar que el número de la operación exista.

3.2.4 Priorización de los servicios

Fase donde se van a listar todos, los servicios identificados, clasificándolos por prioridad de desarrollo, en esta lista se van a decidir qué servicios se van a desarrollar en base a las necesidades de la empresa y/o el cliente necesite.

Tabla 18. Cuadro de Objetivos

Objetivo					
	Permitir el intercambio de datos con entidades externas	Agilizar procedimientos internos y/o externos en la organización	Optimizar los procesos de consultas y pagos en entidades financieras	Actualizar registros en las BD	Reutilización en otros sistemas
wsConsultaDeudas	X	X	X		X
wsConsultaAlumnos	X	X			X
wsGenerarEstadoCuenta		X			
wsPagosDeuda	X	X	X	X	X
wsGenerarVoucherPago		X			
wsExtornoPago	X	X		X	X
wsConsultarPagos	X				X
wsActualizarLibroContabilidad		X			
Total	5	7	2	2	5

De tal manera que los objetivos quedarían priorizados de la siguiente forma:

Tabla 19. Puntaje de Objetivos

Objetivo	Puntaje
Permitir la consulta de datos con entidades externas	5
Agilizar procedimientos internos y/o externos en la organización	7
Optimizar los procesos de consultas y pagos en entidades financieras	2
Actualizar registros en las BD	2
Reutilización en otros sistemas	2

Una vez que se tienen los objetivos priorizados, se procede a la identificación de los servicios que participan en cada objetivo, para lo cual se establecen niveles: alto (3), medio (2) y bajo (1). Teniendo esto en cuenta se elabora el cuadro donde se establece la relación de los servicios con los objetivos, valorizándolos para su priorización final y ver cuales servicios se expondrán.

Tabla 20. Priorización de Servicios

Servicio	Objetivos					Total
	Permitir la consulta de datos con entidades externas	Agilizar procedimientos internos y/o externos en la organización	Optimizar los procesos de consultas y pagos en entidades financieras	Actualizar registros en las BD	Reutilización en otros sistemas	
wsConsultaDeudas	2	3	2	0	2	9
wsConsultaAlumnos	1	2	0	0	2	5
wsGenerarEstadoCuenta	0	2	0	0	0	2
wsPagosDeuda	3	3	3	3	2	14
wsGenerarVoucherPago	0	2	0	0	0	2
wsExtornoPago	2	3	0	3	2	10
wsConsultarPagos	1	0	0	0	2	3
wsActualizarLibroContabilidad	0	2	0	0	2	2

Con el resultado de la Tabla 21, se obtiene la lista de servicios priorizados como se muestra a continuación:

Tabla 21. Lista de Servicios Priorizados

Código	Servicio	Puntaje
S04	wsPagosDeuda	14
S06	wsExtornoPago	10
S01	wsConsultaDeudas	9
S02	wsConsultaAlumnos	5
S07	wsConsultarPagos	3
S03	wsGenerarEstadoCuenta	2
S08	wsActualizarLibroContabilidad	2
S05	wsGenerarVoucherPago	2

3.2.5 Descripción de los servicios

En esta etapa se describen los 3 primeros servicios de la tabla de priorización, a continuación, se detalla la descripción de los servicios desarrollados, las peticiones, las respuestas y los diferentes errores que devuelven dependiendo de las peticiones enviadas.

a) Consulta de deudas

Servicio que consulta la deuda asociada a un estudiante de la entidad educativa permitiendo ver las deudas activas (un máximo de 10 deudas por fecha en que se generaron), a continuación, se detalla la estructura de la petición que se le envía al servicio (API-REST):

Tabla 22. Estructura Solicitud – Consultar deudas

#	Path	Obligatorio	Longitud	Tipo	Descripción
1	ConsultaDeuda				
2	Operación				
3	codOperacion	Si	3	Carácter	Código de la operación a realizar
4	canOperacion	Si	2	Carácter	Canal por donde se

					realiza la operación
5	numOperacion	Si	9	Numérico	Numero único de la operación. (Banco)
6	fecOperacion	Si	8	Numérico	Fecha de la operación (yyyyMMdd)
7	horOperacion	Si	6	Numérico	Hora de la operación (HHmmss)
8	nroReferencia	Si	10	Numérico	Numero de referencia a consultar (DNI, código de alumno)

A continuación, se muestra la respuesta a la solicitud del servicio de consultas, en caso la solicitud no haya presentado errores:

Tabla 23. Estructura Respuesta – Consultar deudas

#	Path	Obligatorio	Longitud	Tipo	Descripción
1	ConsultaDeudaRsp				
2	Operación				
3	codOperacion	Si	3	Carácter	Mismo valor que el enviado por la petición.
4	numOperacion	Si	9	Numérico	Mismo valor que el enviado por la petición.
5	canOperacion	Si	2	Numérico	Mismo valor que el

					enviado por la petición.
6	fecOperacion	Si	8	Numérico	Mismo valor que el enviado por la petición.
7	horOperacion	Si	6	Numérico	Mismo valor que el enviado por la petición.
8	Respuesta				
9	Código	Si	3	Numérico	Código de respuesta
10	Descripción	Si	30	Alfanumerico	Descripción de la respuesta
11	transaccionDetalle				
12	nroReferencia	Si	10	Numérico	Mismo valor que el enviado por la petición.
13	nombreCli	Si	45	Numérico	Nombre del cliente
14	nroOperEmp	Si	9	Numérico	Numero de la operación por parte de la entidad educativa.
15	nroDocs	Si	1	Numérico	Cantidad de documentos devueltos por la consulta
16	listaDocs				
17	Docs				
18	numDoc	Si	15	Numérico	Número del documento

19	Descripción	Si	30	Alfanumerico	Descripción del documento
20	Monto	Si	10	Decimal	Monto de la deuda
21	fechaEmi	Si	8	Numérico	Fecha de emisión del documento (yyyyMMdd)
22	fechaVen	Si	8	Numerico	Fecha de vencimiento del documento (yyyyMMdd)

Para el caso de errores se presenta la siguiente estructura para la respuesta de las solicitudes:

Tabla 24. Estructura Respuesta Error – Consultar deudas

#	Path	Obligatorio	Longitud	Tipo	Descripción
1	ConsultaDeudaRsp				
2	Operación				
3	codOperacion	Si	3	Carácter	Mismo valor que el enviado por la petición.
4	numOperacion	Si	9	Numérico	Mismo valor que el enviado por la petición.
5	Canaloperacion	Si	2	Numérico	Mismo valor que el enviado por la petición.
6	fecOperacion	Si	8	Numérico	Mismo valor que el

					enviado por la petición.
7	horOperacion	Si	6	Numérico	Mismo valor que el enviado por la petición.
8	Respuesta				
9	Código	Si	3	Numérico	Código de respuesta
10	Descripción	Si	30	Alfanumerico	Descripción de la respuesta

b) Pago de Deudas

Servicio que realiza los pagos de las deudas de los alumnos de la entidad educativa por medio de la entidad financiera, a continuación, se detalla la estructura de la petición que se envía al servicio (API-REST).

Tabla 25. Estructura Solicitud – Pago de deudas

#	Path	Obligatorio	Longitud	Tipo	Descripción
1	NotiPago				
2	Operación				
3	codOperacion	Si	3	Carácter	Código de la operación a realizar
4	canOperacion	Si	2	Carácter	Canal por donde se realiza la operación
5	numOperacion	Si	9	Numérico	Numero único de la operación. (Banco)

6	fecOperacion	Si	8	Numérico	Fecha de la operación (yyyyMMdd)
7	horOperacion	Si	6	Numérico	Hora de la operación (HHmmss)
8	detalle				
9	nroReferencia	Si	10	Numérico	Numero de referencia a consultar (DNI, código de alumno)
10	numDoc	Si	15	Numérico	Número del documento
11	monto	Si	10	Decimal	Monto de la deuda
12	metPago	Si	2	Caracter	Método para realizar el pago
13	nroRecaudo	Si	9	Numérico	Número que identifica a la operación.
14	tipMoneda	Si	2	Caracter	Tipo de moneda: 01: Soles 02: dólares

A continuación, se muestra la respuesta a la solicitud del servicio de pago de deudas, en caso la petición no haya presentado errores en su envío:

Tabla 26. Estructura Respuesta – Pago de deudas

#	Path	Obligatorio	Longitud	Tipo	Descripción
1	NotiPagoRsp				
2	operacion				
3	codOperacion	Si	3	Carácter	Mismo valor que el

					enviado por la petición.
4	numOperacion	Si	9	Numérico	Mismo valor que el enviado por la petición.
5	canOperacion	Si	2	Numérico	Mismo valor que el enviado por la petición.
6	fecOperacion	Si	8	Numérico	Mismo valor que el enviado por la petición.
7	horOperacion	Si	6	Numérico	Mismo valor que el enviado por la petición.
8	respuesta				
9	codigo	Si	3	Numérico	Código de respuesta
10	descripción	Si	30	Alfanumerico	Descripción de la respuesta
11	transaccionDetalle				
12	nroReferencia	Si	10	Numérico	Mismo valor que el enviado por la petición.
14	nroOperEmp	No	9	Numérico	Numero de la operación por parte de la entidad educativa.

Para el caso de errores se presenta la siguiente estructura para la respuesta de las solicitudes:

Tabla 27. Estructura Respuesta Error – Pago de deudas

#	Path	Obligatorio	Longitud	Tipo	Descripción
1	NotiPagoRsp				
2	operacion				
3	codOperacion	Si	3	Carácter	Mismo valor que el enviado por la petición.
4	numOperacion	Si	9	Numérico	Mismo valor que el enviado por la petición.
5	canaloperacion	Si	2	Numérico	Mismo valor que el enviado por la petición.
6	fecOperacion	Si	8	Numérico	Mismo valor que el enviado por la petición.
7	horOperacion	Si	6	Numérico	Mismo valor que el enviado por la petición.
8	respuesta				
9	codigo	Si	3	Numérico	Código de respuesta
10	descripción	Si	30	Alfanumerico	Descripción de la respuesta

c) Reembolso de una deuda (Extorno)

Servicio que anula los pagos que se realizaron por medio de la entidad financiera, cabe destacar que para el reembolso (extorno) de una deuda este se debe de realizar el mismo día que se pagó o el extorno no procederá.

Tabla 28. Estructura Solicitud – Reembolso de una deuda

#	Path	Obligatorio	Longitud	Tipo	Descripción
1	ExtornoPago				
2	operacion				
3	codOperacion	Si	3	Carácter	Código de la operación a realizar
4	canOperacion	Si	2	Carácter	Canal por donde se realiza la operación
5	numOperacion	Si	9	Numérico	Numero único de la operación. (Banco)
6	fecOperacion	Si	8	Numérico	Fecha de la operación (yyyyMMdd)
7	horOperacion	Si	6	Numérico	Hora de la operación (HHmmss)
8	detalle				
9	nroReferencia	Si	10	Numérico	Numero de referencia (DNI, código de alumno)
10	monto	Si	10	Decimal	Monto de la deuda pagada
11	nroOperacion	Si	9	Numérico	Numero único de la operación

12	nroOperOri	Si	9	Numérico	Numero de la operación obtenido del pago.
13	fecOperOri	Si	8	Numérico	Fecha original de la operación de pago.
14	metPago	Si	2	Caracter	Método para realizar el pago
15	tipMoneda	Si	2	Caracter	Tipo de moneda: 01: Soles 02: dólares

A continuación, se muestra la respuesta a la solicitud del servicio de pago de deudas, en caso la petición no haya presentado errores en su envío:

Tabla 29. Estructura Respuesta – Reembolso de una deuda

#	Path	Obligatorio	Longitud	Tipo	Descripción
1	ExtornoPagoRsp				
2	operacion				
3	codOperacion	Si	3	Carácter	Mismo valor que el enviado por la petición.
4	numOperacion	Si	9	Numérico	Mismo valor que el enviado por la petición.
5	canOperacion	Si	2	Numérico	Mismo valor que el enviado por la petición.

6	fecOperacion	Si	8	Numérico	Mismo valor que el enviado por la petición.
7	horOperacion	Si	6	Numérico	Mismo valor que el enviado por la petición.
8	respuesta				
9	codigo	Si	3	Numérico	Código de respuesta
10	descripción	Si	30	Alfanumerico	Descripción de la respuesta
11	transaccionDetalle				
12	nroReferencia	Si	10	Numérico	Mismo valor que el enviado por la petición.
14	nroOperEmp	No	9	Numérico	Numero de la operación por parte de la entidad educativa.

Para el caso de errores se presenta la siguiente estructura para la respuesta de las solicitudes:

Tabla 30. Estructura Respuesta Error – Reembolso de una deuda

#	Path	Obligatorio	Longitud	Tipo	Descripción
1	ExtornarPagoRsp				
2	operacion				
3	codOperacion	Si	3	Carácter	Mismo valor que el enviado por la petición.

4	numOperacion	Si	9	Numérico	Mismo valor que el enviado por la petición.
5	canaloperacion	Si	2	Numérico	Mismo valor que el enviado por la petición.
6	fecOperacion	Si	8	Numérico	Mismo valor que el enviado por la petición.
7	horOperacion	Si	6	Numérico	Mismo valor que el enviado por la petición.
8	respuesta				
9	codigo	Si	3	Numérico	Código de respuesta
10	descripción	Si	30	Alfanumerico	Descripción de la respuesta

En caso de enviar un campo que no cumpla con las restricciones mencionadas, el path de operaciones mandará nulo todos sus campos, retornando el error “FORMATO NO VALIDO” (la misma estructura en caso de ese error es el mismo para los 3 servicios desarrollados).

3.4 Códigos de Respuesta

La Tabla 31. Códigos de errores se muestra los diferentes errores que se presentan en los servicios desarrollados:

Tabla 31. Códigos de errores

Código	Descripción
00001	Operación realizada con éxito
ERR02	No se pudo realizar la operación
ERR03	No se pudo realizar el extorno
ERR04	No se tiene deudas pendientes
ERR05	Deuda con estado pagado
ERR06	Se debe pagar la deuda más antigua
ERR07	No se pudo procesar extorno, no existe registro del pago
ERR08	Número de referencia no existe
ERR00	Formato de trama no es valido

3.5 Funciones PostgreSQL

Se elaboraron 3 funciones dentro de la base de datos, para la implementación de los servicios, cada una ligada a una operación (pago, consulta o extorno):

a) Función para la consulta

Función que se encarga de la validación de los parámetros enviados, así como del retorno correcto de los datos para la respuesta del servicio, en caso de un error, se devuelve el código del error y la descripción de este.

Los parámetros que recibe para su ejecución se muestran a continuación:

Tabla 32. Parámetros función de Consulta

Parámetro	Descripción
cod_alu	Código del estudiante a realizar la consulta de las deudas.
num_caj	Numero de cajero en el cual se realiza la operación.
cod_banco	código que identifica al banco dentro del sistema
canal	Canal por el cual se realiza la operación.

Una vez que la función se ejecute (sin errores), devolverá las deudas que el alumno tenga pendiente (máximo 10 deudas), en caso de devolver un error PHP, verificara el código devuelto por la función para su respectivo envío (código y descripción) a la respuesta del servicio.

Figura 27 Fragmento de función consulta

```
CREATE OR REPLACE FUNCTION public.Consulta(
    cod_alu character,
    num_caj character,
    cod_banco character ,
    canal character)
RETURNS SETOF t_consulta
LANGUAGE 'plpgsql'
COST 100
VOLATILE
ROWS 1000
```

Figura 28. Tipo t_consulta

```
CREATE TYPE public.t_consulta AS
(
    id_deuda integer,
    cod_alu character(10),
    nombre character varying(100),
    periodo character(8),
    descripcion character varying(100),
    moneda character(1),
    monto numeric(12,2),
    fecha_emi date,
    fecha_ven date
);
```

b) Función para el pago

Función que se encarga de la validación de los parámetros enviados, así como del retorno correcto de los datos para la respuesta del servicio, en caso de un error, se devuelve el código del error y la descripción de este.

Los parámetros que recibe para su ejecución se muestran a continuación:

Tabla 33. Parámetros función de Pagos

Parámetro	Descripción
cod_alu	Código del estudiante a realizar la consulta de las deudas.
id_deuda	Identificador de la deuda.
monto	Monto para cancelar la deuda.
moneda	Tipo de moneda usada en la operación.

Una vez que la función se ejecute (sin errores), devolverá 'Ok', para su respectiva validación por PHP, en caso de devolver un error, PHP identificara el error para devolver en la respuesta el código y la descripción adecuada al caso.

Figura 29. Fragmento de función pagos

```
CREATE OR REPLACE FUNCTION public.Pagos(
    cod_alu character,
    id_deuda character,
    monto numeric,
    moneda character)
RETURNS TABLE(cresult character, cdescri character varying)
LANGUAGE 'plpgsql'
COST 100
VOLATILE
ROWS 1000
AS $BODY$
```

c) Función para el extorno

Función que se encarga de la validación de los parámetros enviados, así como del retorno correcto de los datos para la respuesta del servicio, en caso de un error, se devuelve el código del error y la descripción de este.

Los parámetros que recibe para su ejecución se muestran a continuación:

Tabla 34. Parámetros función de Extorno

Parámetro	Descripción
cod_alu	Código del estudiante a realizar la consulta de las deudas.
id_deuda	Identificador de la deuda.

Id_extorno	Identificador del extorno.
monto	Monto para cancelar la deuda.

Una vez que la función se ejecute (sin errores), devolverá 'Ok', para su respectiva validación por PHP, en caso de un error se devolverá el código y descripción para su envío por PHP.

3.6 Implementación y pruebas

3.6.1 Características del servidor y programas

Para la implementación de los servicios primero, se tuvo que revisar las especificaciones técnicas donde se iba a realizar la instalación para acto seguido verificar las versiones de los programas que se iban a utilizar y que necesiten actualizarse.

Las especificaciones técnicas del servidor donde se realizó la instalación de los servicios es la que se muestra en la siguiente tabla:

Tabla 35. Ficha técnica del servidor

Procesador	3.3 GHz Xeon
RAM	8 Gb
Disco Duro	1 TB
Marca Chipset	Intel
Modelo	Dell PowerEdge
Sistema Operativo	Debian 11
Numero de Procesadores	4
Entradas USB	6
Fuente de Energía	Corriente Alterna (AC)

A continuación, se muestran los programas que estaban instalados en el servidor antes de la implementación de los servicios (API-REST):

- PgAdim4: Interfaz gráfica que permite la administración de las bases de datos que se tiene en PostgreSQL.
- PHP 7.4: Lenguaje de programación de backend usado para el desarrollo de los servicios.
- PostgreSQL 9: Sistema de gestión de la base de datos.
- Apache2: Servidor web HTTP.

Como se pudo observar el servidor contaba con los programas para la implementación de los servicios web, solo se necesitaba una maquina dentro de la red para ejecutar las pruebas instalando Postman.

3.6.2 Instalación de los servicios en el servidor

Realizada la verificación de la existencia de todos los programas necesarios para la implementación de los servicios se procede a pasar los programas al servidor, para lo cual utilizamos el cliente de Filezilla (conexión por FTP o SFTP), para la transferencia de archivos.

En el proceso de transferencia lo primero que hacemos es abrir Filezilla con la máquina que tiene los archivo

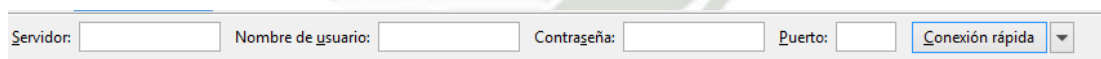
Figura 30. Icono de Filezilla



Fuente: bokehlicia (2016), icono de filezilla.

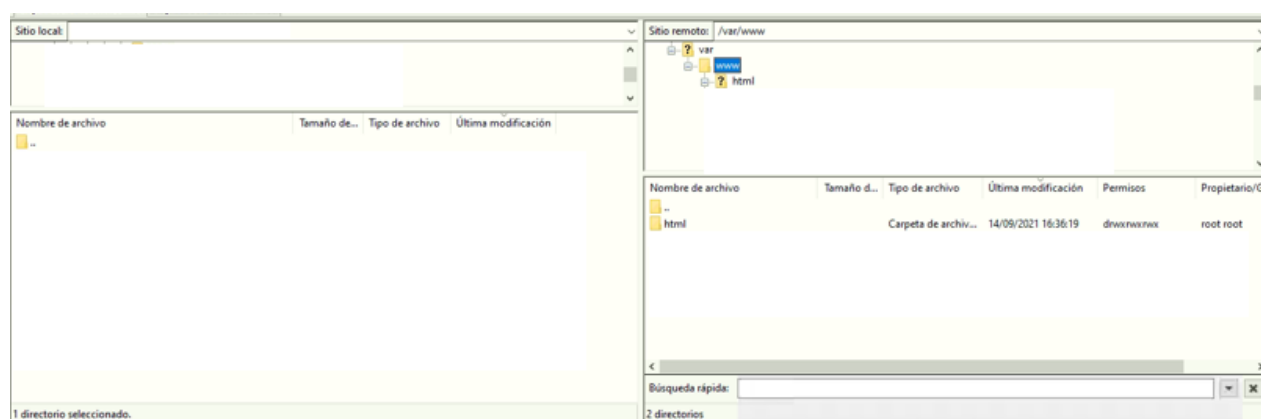
Una vez abierto Filezilla procedemos a conectarnos utilizando el usuario y contraseña brindada por la institución educativa, así como la IP para poder acceder al servidor.

Figura 31. Configuración usuario y contraseña



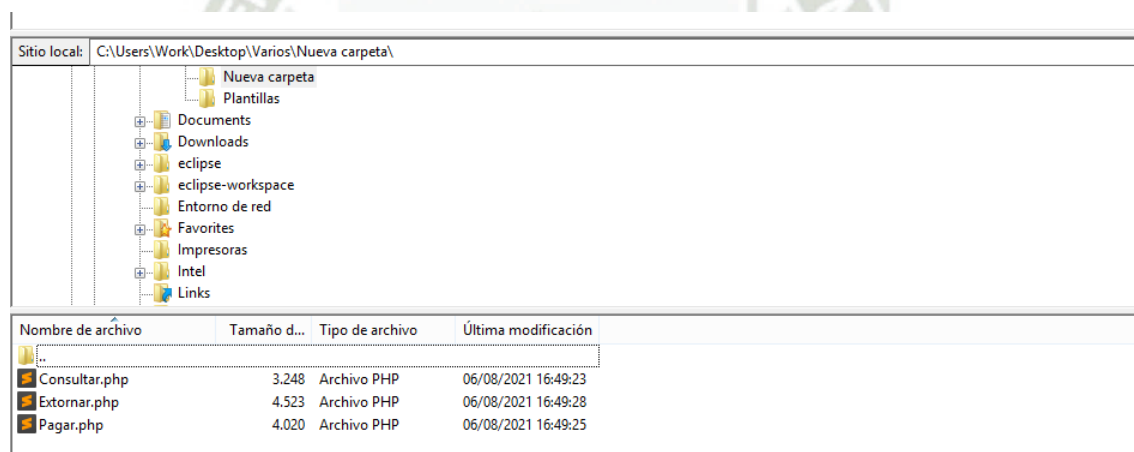
Luego de haber presionado en el botón de conexión rápida, esperamos a conectarnos con el servidor para luego dirigirnos a la ubicación donde se guardan proyectos en apache (/var/www/html/...) (ruta determinada en servidores con sistema operativo Linux/Debian, en caso de estar trabajando en un servidor con Windows, se tiene que guardar dentro de la carpeta raíz de los proyectos en apache).

Figura 32. Directorio de proyectos



Por último, procedemos a ubicar en el lado derecho, donde se encuentra el directorio de nuestra maquina local para ubicarnos en la carpeta que contiene los archivos necesarios para el funcionamiento de los servicios y hacemos clic derecho en cada uno para que se suban al servidor.

Figura 33. Subida de archivos al servidor



3.6.3 Estrategia de Pruebas

Las pruebas realizadas en los servicios implementados están divididas en 2, primero las pruebas funcionales, las cuales permiten ver que los servicios retornen las respuestas correctas a las solicitudes enviadas y las de carga que permiten verificar el tiempo de respuesta que demora el procesar la petición y generar la respuesta.

a) Pruebas Funcionales

Tabla 36. Pruebas Funcionales

Objetivo	Asegurar que los servicios (API-REST) devuelvan la respuesta correcta al enviar la petición (request) que cumpla con la estructura antes vista en el Capítulo 2.
Técnicas	• Verificación del LogEventos.txt una vez terminada la prueba • Revisar la base de datos una vez se la ejecución del proceso de extorno o pago,
Criterio de Aceptación	El servicio (API-REST) debe de retornar la respuesta, según el caso a evaluar y se debe de actualizar el LogEventos.txt
Consideraciones	Se utilizarán los requisitos funcionales detallados en el capítulo 2.

b) Pruebas de Carga

Tabla 37. Pruebas de Carga

Objetivo	Asegurar que los servicios se ejecuten de manera correcta, ejecutándose y devolviendo la respuesta (response) en menos de 1 segundo.
Técnicas	• Enviar peticiones con un archivo .csv que permitirá ver cuanto demora en ejecutar cada una. • Ejecutar las peticiones con el cliente cUrl de php, mostrando la ejecución por cada petición.
Criterio de Aceptación	Todas las peticiones demoren menos de 1 segundo en procesarse (ya sea por una notificación o ejecución correcta)
Consideraciones	Se utilizarán los requisitos funcionales detallados en la sección 3.1.1

Capítulo IV: Resultados

En este capítulo presentamos los resultados de las pruebas ejecutadas a los servicios implementados a la entidad financiera, los cuales nos han permitido verificar los tiempos de ejecución de cada servicio instalado, asimismo se podrá identificar los factores que afectan en el intercambio de datos para tenerlos en cuenta en futuras implementaciones y poder mejorar su estructura, de igual forma se elaboraron los diagramas UML (Lenguaje Unificado de Modelado) en base a los servicios permitiendo ver sus componentes y los casos que se presentan.

4. Diagramas y resultados de las pruebas

4.1 Diagramas UML

4.1.1 Diagrama de casos de uso

Primero se comenzó con el desarrollo de un diagrama de casos de uso general donde se engloban los servicios implementados.

Figura 34. Diagrama de Casos de uso - General

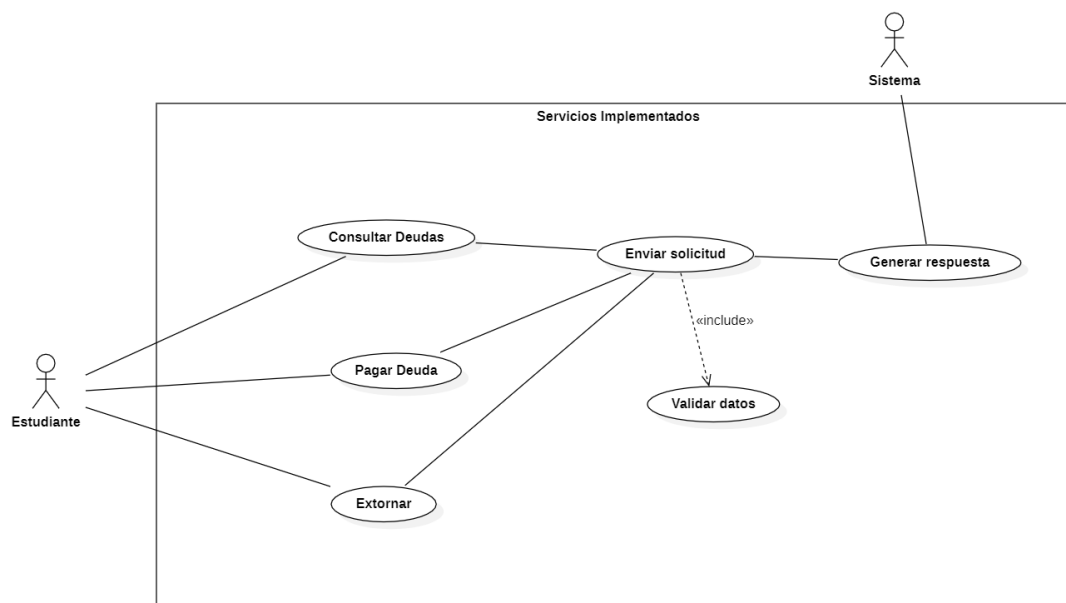


Tabla 38. Caso de uso N°1

CU001 – CONSULTAR DEUDAS, PAGO O EXTORNO

Descripción

El estudiante ingresa el número de referencia con para hacer la consulta (DNI, código de alumno)

Actores

- Estudiante

Precondiciones
Validación del parámetro enviado (número de referencia) y monto pagado en el caso de pago y extorno.
Post condiciones
Ninguna
Excepciones
No haya conexión con el sistema de la entidad bancaria
Flujo
- Recepcionar el parámetro enviado al servidor. - Generar la solicitud. - Enviar la solicitud al servidor de la entidad educativa.
Salida:
Generación de la solicitud para el servicio de consultas, pagos o extornos.

Tabla 39. Caso de uso N°2

CU002 – ENVIAR SOLICITUD

Descripción

El servidor de la entidad financiera recibe el parámetro enviado por el estudiante a través de la interfaz (cajero)

Actores

- Servidor

Precondiciones

- Validación del parámetro enviado (número de referencia)

Post condiciones

- Ninguna

Excepciones

- No haya conexión con el sistema de la entidad bancaria

Flujo

- Recepción del parámetro.
- generación de la solicitud.
- Enviar al servidor de la entidad bancaria.

Salida:

- Envío de la solicitud al servidor de la entidad educativa.

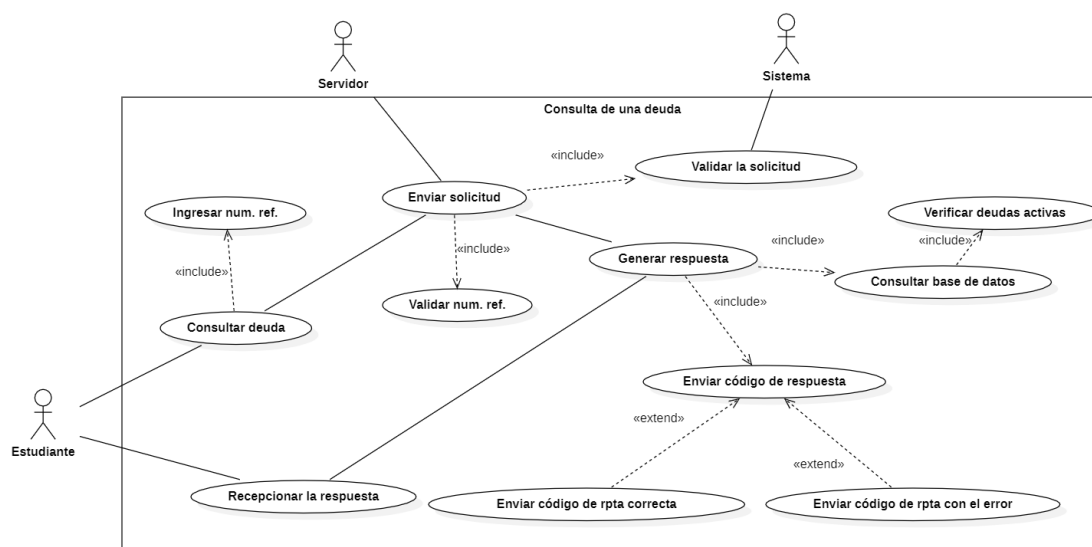
Tabla 40. Caso de uso N°3

CU003 – GENERAR RESPUESTA	
Descripción	El servidor de la entidad educativa recibe la solicitud generada a través del servicio API-REST.
Actores	• Sistema (servicio API-REST)
Precondiciones	• Ninguna
Post condiciones	• Ninguna
Excepciones	• No haya conexión con el servicio API-REST
Flujo	• Recepción de la solicitud. • Validación de la solicitud • Consultar en la base de datos, deuda del estudiante. • Generación de la respuesta. • Enviar la respuesta.
Salida:	• Respuesta de la solicitud enviada.

Una vez visto los casos de uso de manera general, a continuación, se muestran los diagramas de casos de uso elaborados para cada uno de los servicios implementados para la interconexión con la entidad financiera, donde se detalla los casos de uso para cada servicio.

a) Consulta de deudas

Figura 35. Consultar Deudas



La Figura 35. nos muestra las entidades que participan en el proceso de consulta de una deuda, en la siguiente tabla se detallan los casos de uso que se presentan:

Tabla 41. Caso de uso N°4

CU004 – CONSULTAR DEUDA

Descripción

El estudiante ingresa el número de referencia con para hacer la consulta (DNI, código de alumno)

Actores

- Estudiante

Precondiciones

- Ninguna

Post condiciones

- Ninguna

Excepciones

- No haya conexión con el sistema de la entidad bancaria

Flujo

- Ingresar número de referencia
- Validación del número de referencia
- Enviar número de referencia al servidor de la entidad bancaria

Salida:

- Envío del número de referencia al servidor de la entidad financiera

Tabla 42. Caso de uso N°5

CU005 – ENVIAR SOLICITUD
Descripción La entidad bancaria recibe el número de referencia enviado a través de la interfaz (agente) con el cual se genera la solicitud (request) para el servicio de consulta de deudas.
Actores • Sistema (servicio API-REST de consulta)
Precondiciones • Recepción del número de referencia.
Post condiciones • Ninguna.
Excepciones • No se encuentre disponible el servidor de la entidad financiera • Número de referencia no valido (longitud errónea).
Flujo • Recepcionar número de referencia • Validar número de referencia. • Generar solicitud. • Enviar solicitud al servidor de la entidad educativa.
Salida: • Envío de la solicitud.

Tabla 43. Caso de uso N°6

CU006 – GENERAR RESPUESTA
Descripción El servidor de la entidad educativa recibe la solicitud generada a través del servicio API-REST para consulta de deudas.
Actores • Sistema (servicio API-REST)
Precondiciones • Ninguna
Post condiciones

Ninguna
Excepciones No haya conexión con el servicio API-REST de consultas.
Flujo - Recepción de la solicitud. - Validación de la solicitud para consultas. - Consultar en la base de datos, deudas activas del estudiante. - Generación de la respuesta (correcta o con notificación de un error). - Enviar la respuesta.
Salida: Respuesta de la solicitud enviada.

Tabla 44. Caso de uso N°7

CU007 – RECEPCIONAR LA RESPUESTA
Descripción El servidor de la entidad financiera recibe la respuesta del servicio de consultas, el cual es mostrado por la interfaz (agente).
Actores Estudiante
Precondiciones Recepción de la respuesta al servicio de consultas.
Post condiciones Ninguna
Excepciones No haya conexión con el servicio API-REST de consultas.
Flujo - Recepción de la respuesta. - Validación de la respuesta para consultas. - Enviar la respuesta a la interfaz.
Salida: Interfaz (agente) muestra las deudas activas del estudiante.

b) Pago de una deuda

Figura 36. Pago de una deuda

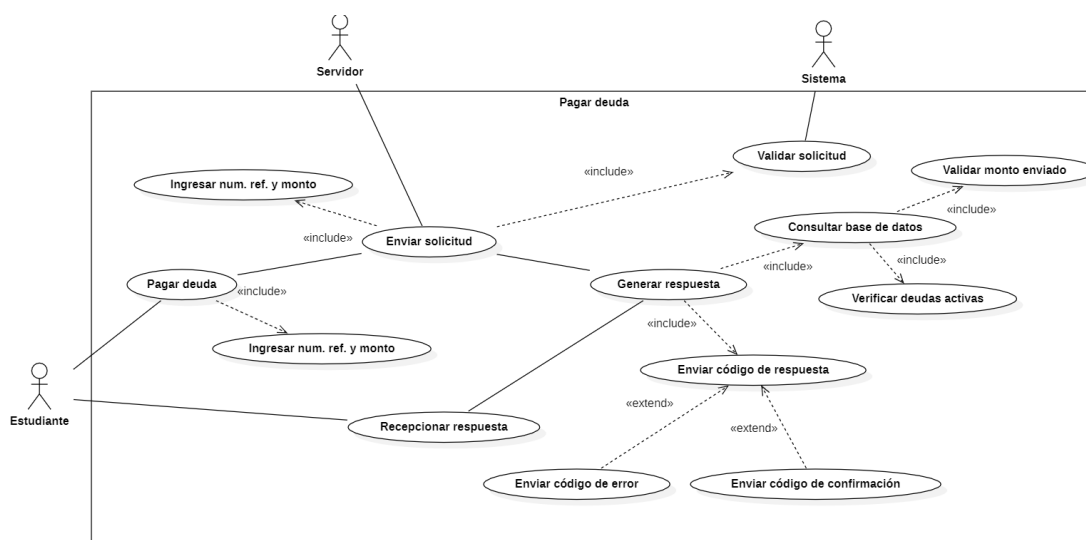


Tabla 45. Caso de uso N°8

CU008 –PAGAR DEUDA	
Descripción	El estudiante ingresa el número de referencia con para hacer el pago de la deuda más antigua que se tenga.
Actores	Estudiante
Precondiciones	Ninguna
Post condiciones	Ninguna
Excepciones	No haya conexión con el sistema de la entidad bancaria
Flujo	- Ingresar número de referencia y monto de la deuda. - Enviar al servidor de la entidad bancaria
Salida:	Envío del número de referencia y monto de la deuda al servidor de la entidad financiera.

Tabla 46. Caso de uso N°9

CU009 – ENVIAR SOLICITUD	
Descripción	La entidad financiera recibe los parámetros (número de referencia y monto de la deuda) para el servicio de pago de una deuda, acto seguido genera el request (solicitud) para el servicio mencionado anteriormente.
Actores	• Sistema (servicio API-REST de pago de una deuda)
Precondiciones	• Recepción de los parámetros (número de referencia y monto) para el servicio de pago.
Post condiciones	• Ninguna
Excepciones	• No haya conexión con el sistema de la entidad bancaria • Numero de referencia erróneo (longitud excedida)
Flujo	• Recepción del número de referencia y monto de la deuda • Generación de la solicitud • Enviar la solicitud al servidor de la entidad educativa
Salida:	• Envío de la solicitud al servidor de la entidad educativa.

Tabla 47. Caso de uso N°10

CU010 – GENERAR RESPUESTA	
Descripción	El servidor de la entidad educativa recibe la solicitud generada a través del servicio API-REST para pago de una deuda.
Actores	• Sistema (servicio API-REST)
Precondiciones	• Recepción de la solicitud.
Post condiciones	• Ninguna
Excepciones	• No haya conexión con el servicio API-REST de pagos.

No exista la deuda a pagar.
Flujo - Recepción de la solicitud. - Validación de la solicitud para pago de una deuda. - Consultar en la base de datos, deudas activas del estudiante. - Validación del monto enviado coincida con la deuda mas antigua. - Generación de la respuesta (correcta o con notificación de un error). - Enviar la respuesta.
Salida: Respuesta de la solicitud enviada.

Tabla 48. Caso de uso N°11

CU011 – RECEPCIONAR LA RESPUESTA
Descripción El servidor de la entidad financiera recibe la respuesta del servicio de pagos, el cual es mostrado por la interfaz (agente).
Actores Estudiante
Precondiciones Recepción de la respuesta al servicio de pagos.
Post condiciones Ninguna
Excepciones No haya conexión con el servidor de la entidad financiera.
Flujo - Recepción de la respuesta. - Validación de la respuesta para pagos. - Enviar la respuesta a la interfaz.
Salida: Interfaz (agente) muestra si se pudo realizar o no el pago (en caso de no poder muestra un mensaje con el motivo).

c) Reembolso de una deuda (extornar)

Figura 37. Reembolso de una deuda (extornar)

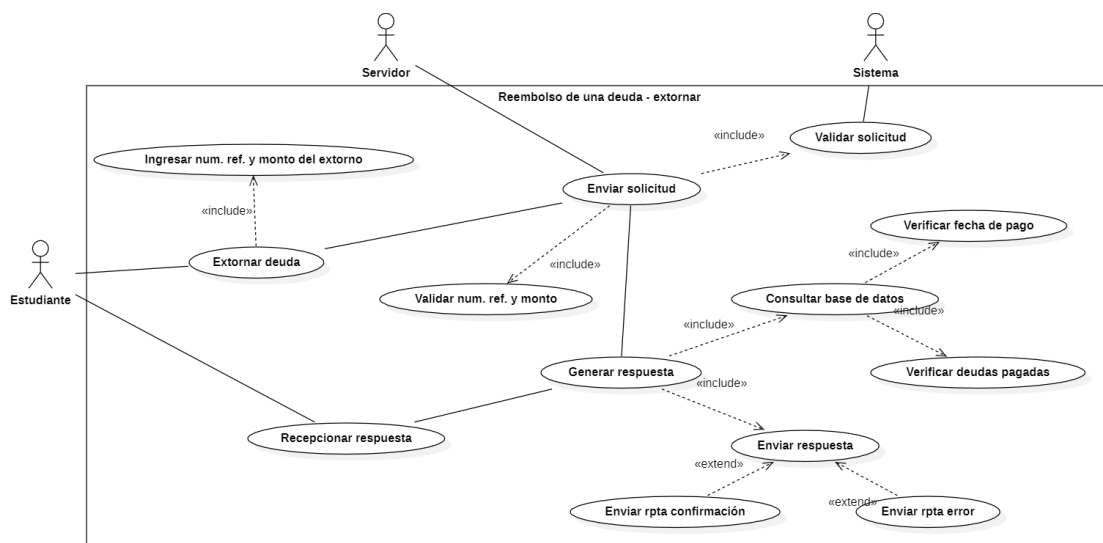


Tabla 49. Caso de uso N°12

CU012 –EXTORNAR DEUDA

Descripción

El estudiante ingresa el número de referencia con para hacer el extorno de una deuda pagada

Actores

- Estudiante

Precondiciones

- Ninguna.

Post condiciones

- Ninguna.

Excepciones

- No haya conexión con el sistema de la entidad bancaria

Flujo

- Ingresar número de referencia y monto de la deuda pagada.
- Envió al servidor de la entidad financiera.

Salida:

- Envió del número de referencia al servidor de la entidad financiera

Tabla 50. Caso de uso N°13

CU013 – SELECCIONAR DEUDA A EXTORNAR	
Descripción	La entidad financiera recepciona los parámetros enviados para el extorno, acto seguido generar la solicitud (request) para el servicio API-REST de extorno.
Actores	• Sistema (servicio API-REST de extorno)
Precondiciones	• Validación de los parámetros (número de referencia y monto de la deuda pagada) enviados para el extorno.
Post condiciones	• Ninguna
Excepciones	• No haya conexión con el sistema de la entidad bancaria. • Numero de referencia erróneo (longitud excedida).
Flujo	• Recepción del número de referencia y monto de la deuda pagada. • Generación de la solicitud para el extorno • Enviar la solicitud al servidor de la entidad educativa
Salida:	• Envío de la solicitud para el servicio de extorno.

Tabla 51. Caso de uso N°14

CU014 – GENERAR RESPUESTA	
Descripción	El servidor de la entidad educativa recibe la solicitud generada a través del servicio API-REST para extorno de una deuda.
Actores	• Sistema (servicio API-REST)
Precondiciones	• Recepción de la solicitud.
Post condiciones	• Ninguna
Excepciones	• No haya conexión con el servicio API-REST de extornos.

• No exista la deuda pagada. • Se haya superado el tiempo límite para hacer el extorno.
Flujo • Recepción de la solicitud. • Validación de la solicitud para extorno de una deuda. • Consultar en la base de datos, deudas pagadas del estudiante. • Verificación de la fecha en la que se realizó el pago. • Generación de la respuesta (correcta o con notificación de un error). • Enviar la respuesta.
Salida: • Respuesta de la solicitud enviada.

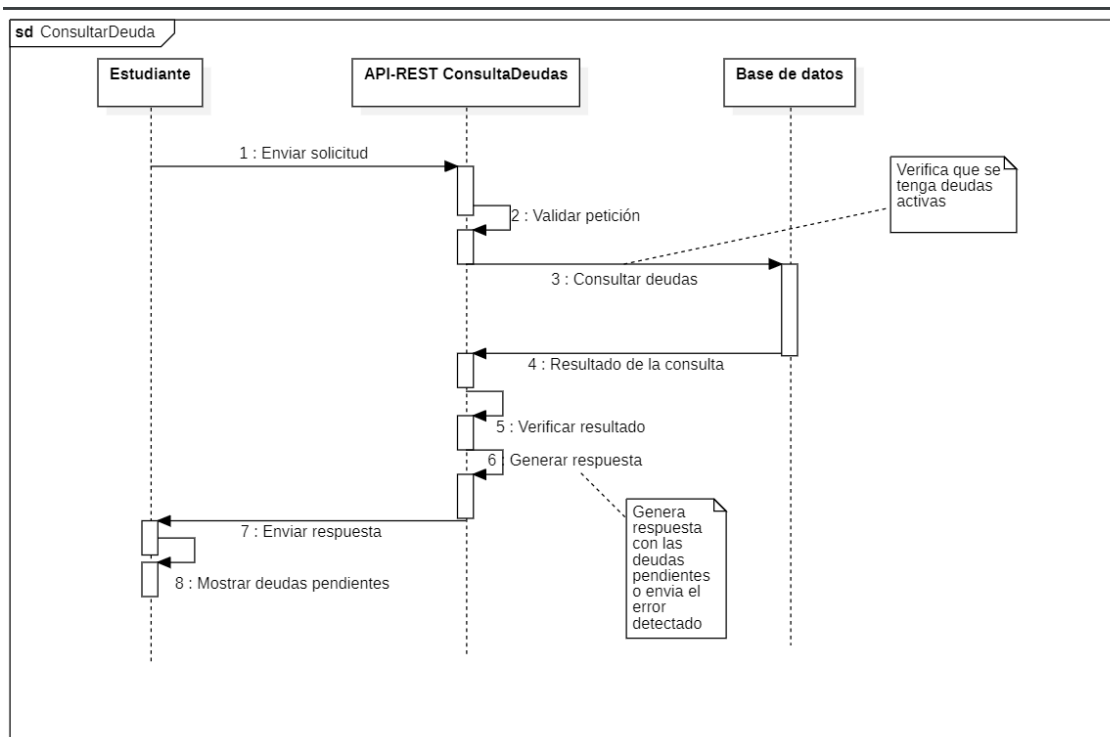
Tabla 52. Caso de uso N°15

CU015 – RECEPCIONAR LA RESPUESTA
Descripción El servidor de la entidad financiera recibe la respuesta del servicio de extornos, el cual es mostrado por la interfaz (agente).
Actores • Estudiante
Precondiciones • Recepción de la respuesta al servicio de extornos.
Post condiciones • Ninguna
Excepciones • No haya conexión con el servidor de la entidad financiera.
Flujo • Recepción de la respuesta. • Validación de la respuesta para pagos. • Enviar la respuesta a la interfaz.
Salida: • Interfaz (agente) muestra si se pudo realizar o no el extorno de la deuda (en caso de no poder muestra un mensaje con el motivo).

4.1.2 Diagrama de secuencia

a) Consulta de deudas

Figura 38. Diagrama de secuencia – Consultar deuda

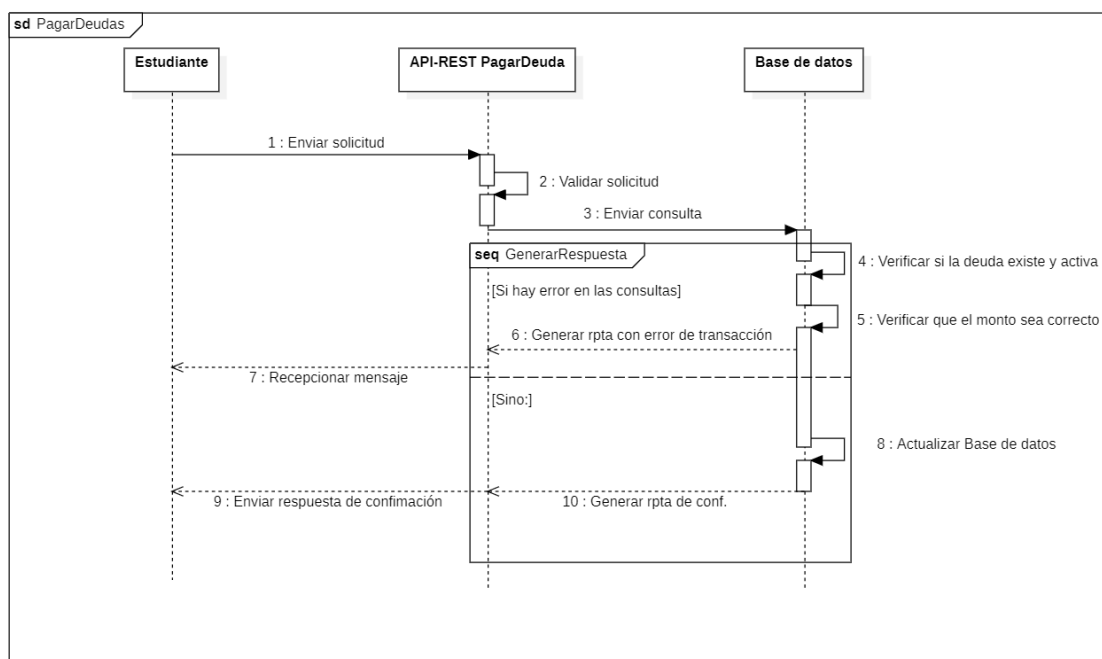


La Figura 38. nos muestra el funcionamiento del servicio Consultar deuda, donde se identificaron 3 (estudiante, el servicio API-REST y la base de datos) objetos que participan durante su funcionamiento.

El servicio de consultar deuda se inicia cuando el estudiante envía la solicitud (la cual es generada por la entidad bancaria, a través de un cajero), luego llega al servicio API-REST que está en el servidor (entidad educativa) para su respectiva validación, una vez que se verifico que la solicitud es correcta, se envía las consultas a la base de datos donde se verifica si el estudiante tiene deudas pendientes, una vez hecha la consulta se procede a generar la respuesta con la cantidad de deudas pendientes.

b) Pago de una deuda

Figura 39. Diagrama de secuencia – Pagar deuda

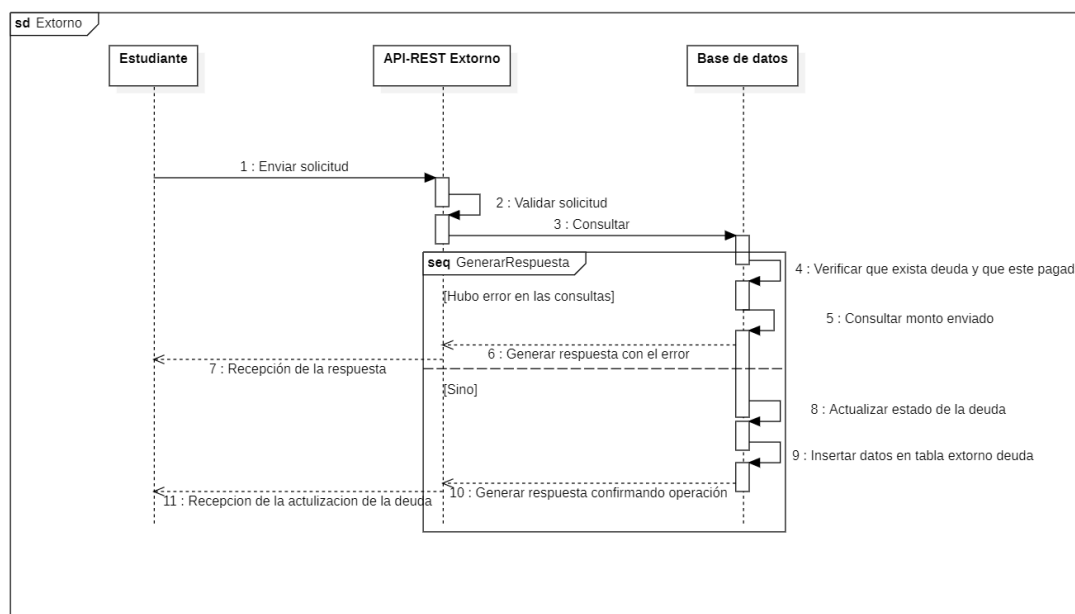


La Figura 39. nos muestra el funcionamiento del servicio Pagar deuda, donde se identificaron 3 objetos (estudiante, el servicio API-REST y la base de datos) que participan durante su funcionamiento, además de observar la presencia de un fragmento combinado, donde según los datos enviados en la solicitud, el servicio genere y envíe la respuesta al estudiante.

El servicio de pagar deuda se inicia cuando el estudiante envía la solicitud (la cual es generada por la entidad bancaria), luego llega al servicio API-REST que está en el servidor (entidad educativa) para su respectiva validación, una vez que se verificó que la solicitud es correcta, se envían las consultas a la base de datos donde se verifica si la deuda a pagar coincide con el monto, si es así se actualiza el estado de la deuda y se crea la respuesta para que enviárselo al estudiante, en caso el monto no coincida se genera la respuesta con el mensaje de que no existe la deuda a pagar.

c) Reembolso de una deuda (extornar)

Figura 40. Diagrama de secuencia – Reembolso de una deuda (extorno)

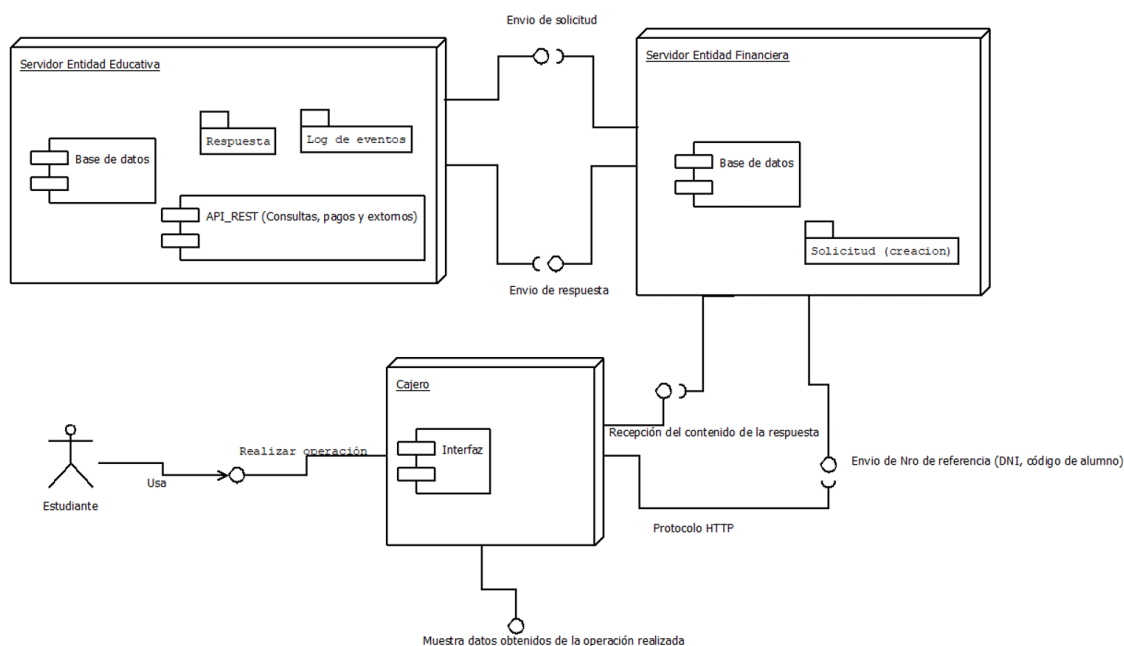


La Figura 40. nos muestra el funcionamiento del servicio Externo de una deuda, donde se identificaron 3 objetos (estudiante, el servicio API-REST y la base de datos) que participan durante su funcionamiento, además de observar la presencia de un fragmento combinado, generando la respuesta según los datos enviados en la solicitud.

El servicio de extorno se inicia cuando el estudiante envía la solicitud (la cual es generada por la entidad bancaria), luego llega al servicio API-REST que está en el servidor (entidad educativa) para su respectiva validación, una vez que se verificó que la solicitud es correcta, se envía las consultas a la base de datos donde se verifica si la deuda a extornar existe en el registro de deudas pagadas y está dentro de la fecha límite para extornar, si es así se actualiza el estado de la deuda, se genera un id para identificar el extorno y ambos ids se registran en la tabla de extorno deuda y se crea la respuesta para que enviárselo al estudiante; en caso no se pueda proceder con el extorno (porque la fecha para extornar se venció) se genera la respuesta indicando el motivo por el cual no se pudo realizar el extorno de la deuda.

4.1.3 Diagrama de componentes

Figura 41. Diagrama de componentes

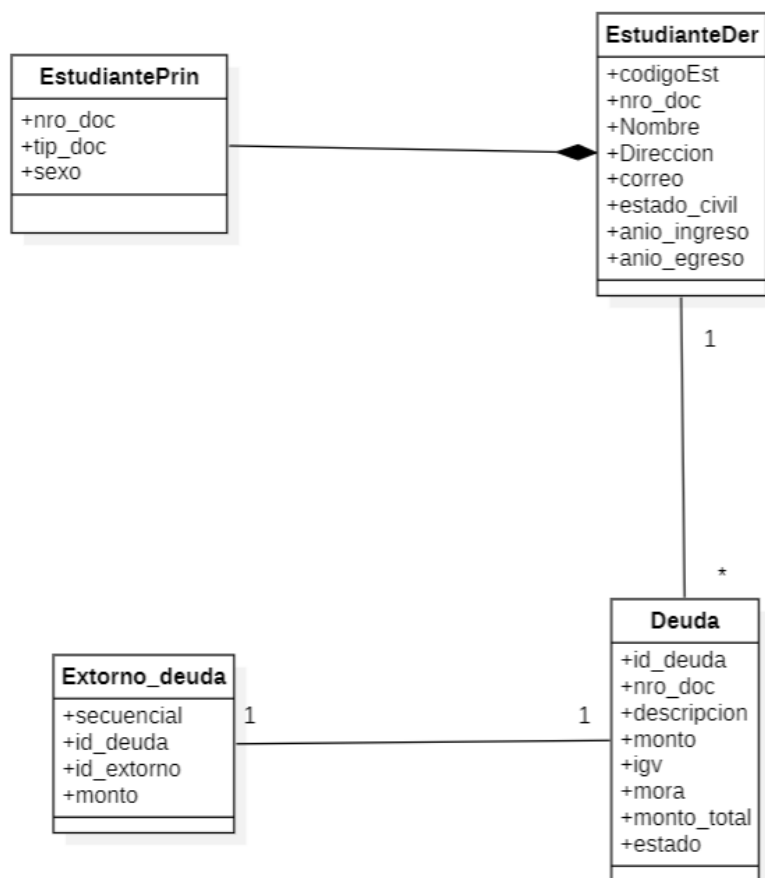


La Figura 41. nos muestra los componentes que utilizan los servicios implementados para la interconexión institucional educativa y financiera (consultas, pagos y extornos), a continuación, se detallan cada uno de ellos:

- **Cajero:** Encargado de recibir los datos por su interfaz para enviarlos al servidor de la entidad financiera.
- **Servidor entidad financiera:** Encargada de recibir los datos (dni, código de estudiante) para la generación de la solicitud (request) que se va a enviar al servidor de la entidad educativa. Dentro suyo encontramos a la solicitud y la base de datos de la entidad financiera.
- **Servidor de la entidad educativa:** Encargada de validar la petición y hacer las respectivas consultas en su base de datos, para la generación de la respuesta (response). Dentro encontramos los servicios API-REST, la base de datos y el log de eventos (registro de las consultas que se han hecho).

4.1.4 Diagrama de clases

Figura 42. Diagrama de clases



La Figura 42. nos muestra las clases que participan en los servicios de interconexión con el banco, a continuación, se describen cada una de las clases con sus principales atributos:

Tabla 53. Estructura de la Tabla EstudianteDer

Tabla EstudianteDer (Derivado de estudiantes)			
nro_doc	Varchar (10)	Número de Documento	PK
tip_doc	Character (3)	Tipo de Documento	
sexo	Character (1)	Género	FK

Tabla 54. Estructura de la Tabla EstudiantePrin

Tabla EstudiantePrin (Maestro de Estudiantes)			
codigoEst	Character (10)	Código de estudiante	PK
nro_doc	Varchar (10)	Numero de Documento	FK
Nombre	Varchar (60)	Apellidos y nombres del estudiante	NOT NULL
Direccion	Varchar (30)	Direccion del estudiante	DEFAULT ''
correo	Varchar (30)	Correo electrónico personal	DEFAULT ''
estado_civil	Character (1)	Estado civil	NOT NULL
anio_ingreso	DATE	Año de ingreso	NOT NULL
anio_egreso	DATE	Año de egreso	DEFAULT ''

Tabla 55. Estructura de la Tabla Deuda

Tabla Deuda			
id_deuda	Character (10)	Identificador de la deuda	UNIQUE
nro_doc	Varchar (10)	Numero de Documento	FK
descripcion	Varchar (20)	Detalle de la deuda	NOT NULL
monto	Number (10,2)	Monto de la deuda	DEFAULT ''
igv	Number (10,2)	Igv de la deuda	DEFAULT ''
mora	Number (10,2)	Mora por retraso en el pago	NOT NULL, DEFAULT 0
monto_total	DATE	Año de ingreso	NOT NULL
estado	DATE	Año de egreso	DEFAULT ''

Tabla 56. Estructura de la Tabla Extorno_deuda

Tabla Extorno_deuda			
secuencial	SERIAL	Identificador de la deuda	UNIQUE
id_deuda	Character (10)	Identificador de la deuda	NOT NULL
id_extorno	Character (10)	Identificador del extorno	NOT NULL
monto	Number (10,2)	Monto del extorno	NOT NULL

4.2 Resultado de las pruebas

En esta sección, se muestran los resultados obtenidos de las pruebas realizadas en los servicios implementados para la interconexión con la entidad financiera.

4.2.1 Pruebas de Funcionalidad

Este tipo de pruebas tiene como objetivo ver que los servicios implementados, devuelvan las respuestas correctas según la solicitud enviada.

Tabla 57. Prueba Funcional N°1 Detalle

Nombre del Sistema: Interconexión Banco	
Caso de Prueba	
ID de Caso de Prueba:	PF-001
Prioridad de prueba: Alta	Alta
Nombre del servicio:	Consultar deuda
Título de la Prueba:	Verificación de las respuestas generadas
Descripción de la Prueba:	Se verifica la respuesta del servidor sea la adecuada, dependiendo de la solicitud enviada.
Condiciones Previas:	Ninguna

La prueba PF-001, muestra la funcionalidad del servicio API-REST de consultas, al cual se le envió diferentes tipos de solicitudes, en el primer caso se envió una solicitud con los datos correctos, para lo cual el servicio devolvió el listado de deudas que tenía el estudiante, para el caso 2, se envió una solicitud con la fecha con un formato diferente al aceptado, para lo cual el

servicio devolvió la respuesta con la debida notificación “FORMATO DE TRAMA NO ES VALIDO”.

Tabla 58. Prueba Funcional N°1 Pruebas

N°	Paso	Datos	Resultado	Estado (pasa / falla)
1	Envío de la solicitud por postman.	{ "ConsultaDeuda": { "operacion": { "codOperacion":"CON", "canOperacion":"T", "numOperacion":454224128, "fecOperacion":"2022-04-15", "horOperacion":"160312", "nroReferencia": ***** } } }	-	Pasa
1.1	Espera de respuesta (response) por parte del servidor.		{ "ConsultaDeudaRsp": { "codOperacion": "", "canOperacion": "", "numOperacion": "", "fecOperacion": "", "horOperacion": "", "respuesta": { "codigo": 001, "descripcion": "FORMATO DE TRAMA NO ES VALIDO" } } }	Pasa
2	Envío de la solicitud por postman.	{ "ConsultaDeuda": { "operacion": { "codOperacion":"CON", "canOperacion":"TF", "numOperacion":454224128, "fecOperacion":"2022-04-15", "horOperacion":"160312", "nroReferencia": ***** } } }	-	Pasar

		<pre> } } } </pre>		
2.1	Espera de respuesta (response) por parte del servidor.	Fecha 2020-04-18	<pre> { "ConsultaDeudaRsp": { "operacion": { "codOperacion": "CON", "canOperacion": "TF", "numOperacion": 454224128, "fecOperacion": "2022-04-15", "horOperacion": "160312", "respuesta": { "codigo": "00001", "descripcion": "Operación realizada con éxito" }, "transacciondetalle": { "nroReferencia": "*****", *, "nombreCli": "*****", "nroOperEmp": 903212531, "nroDocs": , "listaDocs": [{ "numDoc": 124, "descripcion": "Pago, "monto": 233.00, "fechaEmi": "2022-03-12", "fechaVen": "2022-05-12" }] } } } } </pre>	Pasa

Tabla 59. Prueba Funcional N°2 Detalle

Nombre del Sistema: Interconexión Banco	
Caso de Prueba	
ID de Caso de Prueba:	PF-002
Prioridad de prueba: Alta	Alta
Nombre del servicio:	Pagar deuda
Título de la Prueba:	Verificación de las respuestas generadas
Descripción de la Prueba:	Se verifica la respuesta del servidor sea la adecuada, dependiendo de la solicitud enviada.
Condiciones Previas:	Ninguna

La prueba PF-002, muestra la funcionalidad del servicio API-REST de pagos a diferencia de la primera prueba, en esta se envió un numero de referencia que no existía en la base de datos, a lo cual el servicio debía mandar el error ERR08 “Número de referencia no existe”, pero nos retorna el error ERR04 “No se tiene deudas pendientes”.

Tabla 60. Prueba Funcional N°2 Pruebas

N°	Paso	Datos	Resultado	Estado (pasa / falla)
1	Envío de la solicitud por postman.	<pre>{ "NotiPago": { "operacion": { "codOperacion":"PAG", "canOperacion":"TF", "numOperacion":325476231, "fecOperacion":"2022-04-16", "horOperacion":"162302", "detalle": { "nroReferencia":"*****", "numDoc":1242663, "monto": 123.00, "metPago":"EF", "nroRecaudo":342356345, "tipMoneda": "01" } } } }</pre>	-	Pasa

1.1	Espera de respuesta (response) por parte del servidor.		<pre>{ "NotiPagoRsp": { "codOperacion":"PAG", "canOperacion":"TF", "numOperacion":325476231, "fecOperacion":"2022-04-16", "horOperacion":"162302", "respuesta": { "codigo":"ERR04", "descripcion": " No se tiene deudas pendientes " } } }</pre>	Falla
2	Envío de la solicitud por postman.	<pre>{ "NotiPago": { "operacion": { "codOperacion":"PAG", "canOperacion":"TF", "numOperacion":325476231, "fecOperacion":"2022-04-16", "horOperacion":"162302", "detalle": { "nroReferencia": *****, "numDoc":1242663, "monto": 123.00, "metPago":"EF", "nroRecaudo":342356345, "tipMoneda": "01" } } } }</pre>	-	Pasar
2.1	Espera de respuesta (response) por parte del servidor.	-	<pre>"NotiPagoRsp": { "operacion": { "codOperacion":"PAG", "canOperacion":"TF", "numOperacion":325476231, "fecOperacion":"2022-04-16", "horOperacion":"162302", "respuesta": { "codigo":"00001", "descripcion":"Operacion realizada con exito" }, "transaccionDetalle":</pre>	Pasa

			<pre> { "nroReferencia": "*****", "nroOperEmp": 342356345 } } } } </pre>	
--	--	--	--	--

Tabla 61. Prueba Funcional N°3 Detalle

Nombre del Sistema: Interconexión Banco	
Caso de Prueba	
ID de Caso de Prueba:	PF-003
Prioridad de prueba: Alta	Alta
Nombre del servicio:	Extornar deuda
Título de la Prueba:	Verificación de las respuestas generadas
Descripción de la Prueba:	Se verifica que la respuesta del servidor sea la adecuada, dependiendo de la solicitud enviada.
Condiciones Previas:	Realizar el pago de una deuda

La prueba PF-003, muestra la funcionalidad del servicio API-REST de extornos donde se envían las solicitudes a base de pagos que se realizaron anteriormente, como se observa en ambos casos las respuestas devueltas corresponden a los casos que envía la solicitud, en el primero, se envía un extorno de un pago que no existe, devolviendo el error ERR07 “No se pudo procesar extorno, no existe registro de pago”.

Tabla 62. Prueba Funcional N°3 Pruebas

N°	Paso	Datos	Resultado	Estado (pasa / falla)
1	Envío de la solicitud por postman.	<pre> { "ExtornoPago": { "codOperacion": "EXT", "canOperacion": "TF", "numOperacion": 453264, "fecOperacion": "2022-04-17", "horOperacion": "130115", "detalle": { "nroReferencia": "*****", "monto": 820.00, </pre>	-	Pasa

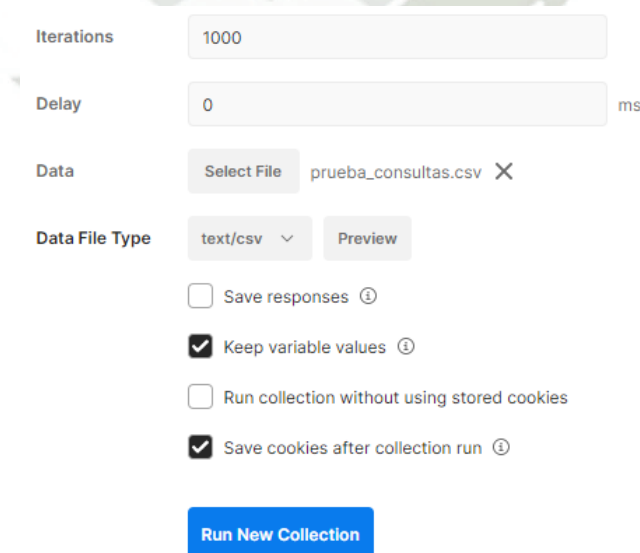
		<pre> "nroOperacion": 876543758, "nroOperOri":876543757, "fecOperOri":"2022-04-17", "metPago":"EF", "tipMoneda":"01" } } } </pre>		
1.1	Espera de respuesta (response) por parte del servidor.		<pre> { "ExtornoPagoRsp ": { "codOperacion":"EXT", "numOperacion":453264, "canOperacion":"TF", "fecOperacion":"2022-04-17", "horOperacion":"130200", "respuesta": { "codigo":"ERR07", "descripcion": "No se pudo procesar extorno, no existe registro del pago" } } } </pre>	Pasa
2	Envío de la solicitud por postman.	<pre> { "ExtornoPago": { "codOperacion":"EXT", "canOperacion":"TF", "numOperacion":453264, "fecOperacion":"2022-04-17", "horOperacion":"130115", "detalle": { "nroReferencia":*****, "monto":820.00, "nroOperacion": 876543758, "nroOperOri":876543753, "fecOperOri":"2022-04-17", "metPago":"EF", "tipMoneda":"01" } } } </pre>	-	Pasa
2.1	Espera de respuesta (response) por parte del servidor.	-	<pre> { "ExtornoPagoRsp": { "operacion": { "codOperacion":"EXT", "numOperacion":453264, "canOperacion":"TF", "fecOperacion":"2022-04-17", "horOperacion":"130200", </pre>	Pasa

			<pre> "respuesta": { "codigo":"00001", "descripcion":"Operación realizada con éxito" }, "transaccionDetalle": { "nroReferencia":*****, "nroOperEmp":876543758 } } } </pre>	
--	--	--	--	--

4.2.2 Pruebas de Carga

Este tipo de prueba es aplicada a cada uno de los servicios implementados anteriormente, para la determinación del tiempo de ejecución, esto es, desde que el servidor de la entidad educativa recepciona la solicitud hasta el instante que el usuario recibe la respuesta por parte del servidor de la entidad bancaria. Para lograr este objetivo, se diseñó diferentes pruebas de carga con la herramienta de Postman, el cual permite mandar a través de un archivo .csv (Comma Separated Values), los datos de la solicitud para enviarlos y verificar el estado y la respuesta que nos devuelve cada servicio, de igual forma se desarrolló un programa en PHP con ayuda del cliente de cUrl, para la determinación de los tiempos de ejecución y verificarlos con los de Postman. (Según se puede observar en las Figuras. 43,44 y 45)

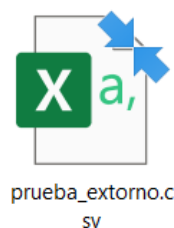
Figura 43. Elaboración de las pruebas



The image shows the Postman interface for running a collection. The configuration is as follows:

- Iterations:** 1000
- Delay:** 0 ms
- Data:** Select File prueba_consultas.csv
- Data File Type:** text/csv
- Options:**
 - ☐ Save responses
 - ☒ Keep variable values
 - ☐ Run collection without using stored cookies
 - ☒ Save cookies after collection run
- Run New Collection** (button)

Figura 44. Archivo pruebas_extorno.csv



El archivo .csv que se muestra en la Figura 42. contiene las variables que van a utilizarse para la ejecución de las diferentes pruebas que se realizan con Postman.

Figura 45. Archivo Pruebas_.php



El archivo Pruebas.php que se muestra en la Figura 45, utiliza el cliente cUrl donde se envía el URL del servicio a conectarse y la solicitud.

a) Consulta de deudas

Las pruebas de carga se realizaron en 3 grupos (10,100 y 1000 solicitudes enviadas al servidor), a continuación, se muestran los resultados obtenidos del segundo grupo (30 primeros registros).

Tabla 63. Tiempos de ejecución consulta

Número de Prueba	Tiempo de ejecución (seg)	Consumo de memoria (kb/seg)
1	0.088295	1047
2	0.099619	1050
3	0.091794	1063
4	0.099449	1045
5	0.088535	1076
6	0.062293	1018
7	0.079513	1030
8	0.099676	1020

9	0.101239	1056
10	0.093186	1031
11	0.077579	1008
12	0.065420	1019
13	0.081553	1053
14	0.067712	1019
15	0.087833	1035
16	0.067144	1006
17	0.055081	1026
18	0.060262	1038
19	0.065903	1069
20	0.055071	1015
21	0.060870	1029
22	0.061671	1045
23	0.059955	1063
24	0.061826	1055
25	0.055312	1019
26	0.057278	1065
27	0.056200	1030
28	0.061283	1045
29	0.059588	1047
30	0.088295	1050

La Fig. 46 nos muestra una parte del código utilizado para la obtención de los tiempos de ejecución con el cliente cUrl en PHP para las consultas, se puede observar que el programa va a enviar 1000 solicitudes al servicio de consulta de deudas.

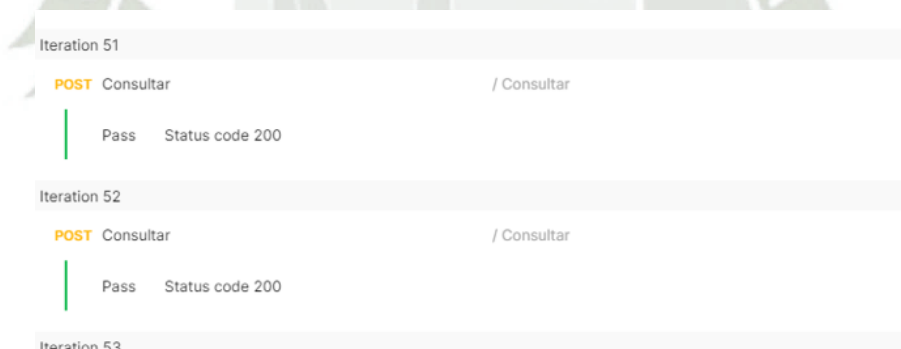
Figura 46. Tiempo de ejecución – consulta

```
$ch = curl_init($url);
$i=0;
while ($i<1000){
    $time_init = microtime_service();
    curl_setopt($ch, CURLOPT_POSTFIELDS,$jsonData);
    curl_setopt($ch, CURLOPT_HTTPHEADER,array('Content-Type:application/json'));
    curl_setopt($ch, CURLOPT_RETURNTRANSFER,true);
    $result = curl_exec($ch);
    $time_fn = microtime_service();
    $time = $time_fn - $time_init;
    echo $i.". "."Tiempo de ejecucion:".$time.PHP_EOL."<br>";
    $i++;
}
curl_close($ch)
```

Log de Eventos: Prueba de Carga – Consultas

Para el control del funcionamiento del sistema, se ha validado las respuestas del servidor de 2 formas, la primera con ayuda de Postman donde para cada una de las solicitudes enviadas nos devuelve el estado del servidor que devuelve (Figura 47) y la segunda mediante el log de eventos donde se guardan las solicitudes y respuestas a cada solicitud.

Figura 47. Resultado de pruebas– Postman



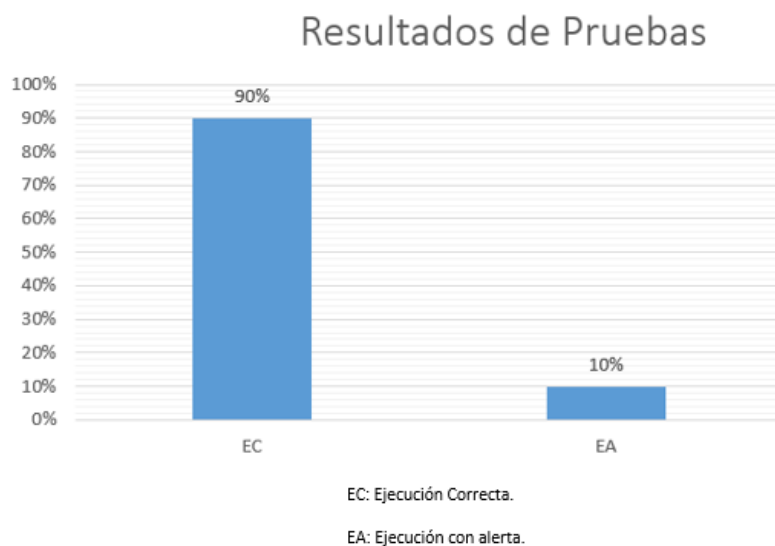
A continuación, se muestran los resultados obtenidos de las 3 pruebas antes mencionadas (10,100 y 1000 solicitudes) para el servicio de consulta.

Prueba N°1 – Consultas:

➤ Prueba de Carga 1 – Consultas: 10 solicitudes

De las 10 solicitudes enviadas al servidor, 9 se ejecutaron correctamente (EC), devolviendo la lista de las deudas activas del estudiante con el estado del servidor (código 200), mientras que una no se ejecutó correctamente (EA) y devolvió la notificación del error encontrado junto al estado del servidor (400).

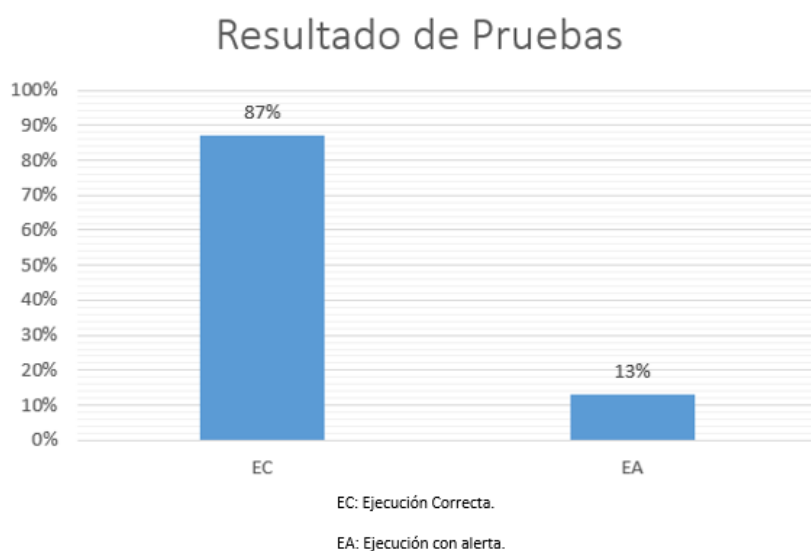
Gráfico 1. Prueba de Carga 1 - Consultas



➤ Prueba de Carga 2 – Consultas: 100 solicitudes

Siguiendo el mismo procedimiento sobre un total de 100 solicitudes procesadas, se obtuvieron 87 ejecuciones correctas (EC) mientras que 13 fueron notificadas como no correctas (EA). (Ver el siguiente gráfico)

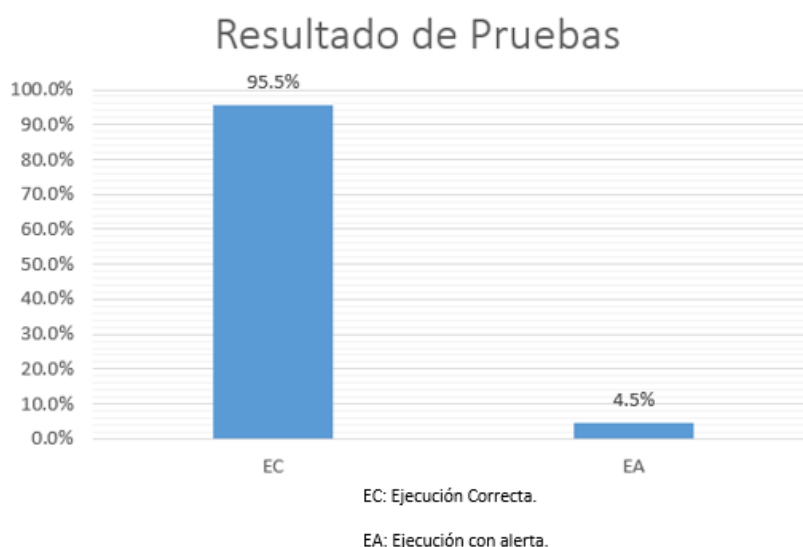
Gráfico 2. Prueba de Carga 2 - Consultas



➤ Prueba de Carga 3 – Consultas: 1000 solicitudes

En el procesamiento de las 1000 solicitudes recibidas se obtuvo como resultado 955 solicitudes ejecutadas correctamente (EC) y 45 ejecutadas con notificación (EA) o incorrecta. (Ver el siguiente gráfico)

Gráfico 3. Prueba de Carga 3 - Consultas



Sintetizando los resultados obtenidos en las tres pruebas de carga tenemos:

Tabla 64. Resumen de los resultados - Consultas

CONSULTAS					
CARGA 10 solicitudes		CARGA 100 solicitudes		CARGA 1000 solicitudes	
Correctas (EC)	Incorrectas (EA)	Correctas (EC)	Incorrectas (EA)	Correctas (EC)	Incorrectas (EA)
9	1	87	13	955	45

El análisis de los resultados nos permite establecer que el número de solicitudes correctas totalmente procesadas (EC) en relación a las incorrectas o con notificación (EA) es relativamente proporcional entre la carga de 10 y la de 100 solicitudes mientras que conforme aumenta el número de solicitudes hasta llegar a la carga de 1000 solicitudes el número de solicitudes correctas (EC) aumenta casi un 9% debido a que los usuarios van aprendiendo sobre el uso del servicio (el envío de los datos que permite generar la solicitud), la aclaración sobre qué datos deben de enviarse, entre otros. La información incorrecta, falta de datos en la solicitud, fallo en la conexión con el servidor o bases de datos darán lugar a que la solicitud sea clasificada como incorrecta (EA).

Respecto a los resultados de las pruebas de consulta relacionados con el tiempo de ejecución, consumo de recursos y tiempo de respuesta los consignamos en la siguiente tabla:

Tabla 65. Resultados de las pruebas consulta

Numero de Prueba	Cantidad de solicitudes	Tiempo de ejecución por cada solicitud	Consumo de recursos (kb/seg)	Tiempo de respuesta	Solicitudes recibidas por segundo
1	10	0.068717	1071	0.416 seg	21.3
2	100	0.069424	1295	4.642 seg	21.2
3	500	0.071295	1374	23.023 seg	20.9
4	1000	0.074091	1484	46.345 seg	20.4

Los datos obtenidos en la prueba de consulta (Tabla 65) nos permiten establecer que el envío de 1000 solicitudes al servidor, para su procesamiento afecta de manera mínima al rendimiento del servidor, ya que el tiempo de procesamiento de las solicitudes es estable existiendo una milésima de segundo entre solicitud y solicitud, tiempo mínimo requerido para su procesamiento, salvo en el caso de las 500 ultimas donde se observa que el tiempo incrementa en dos milésimas de segundo lo cual no repercute de gran forma en la eficiencia del servicio.

Al graficar la información sobre el tiempo de respuesta (Tabla 65), podemos ver que conforme aumenta el ingreso de solicitudes al servidor (representado por la línea azul continua), mayor es el tiempo que demanda su procesamiento, y por lo tanto mayor es la demora en dar a conocer la respuesta lo cual queda representado por la línea punteada de color azul.

Gráfico 4. Tiempo de respuestas - Consulta



En el grafico se puede observar la línea de tendencia que va en aumento entre más solicitudes se envíen al servidor, haciendo que aumente el tiempo que va a demorar en procesar cada una de las solicitudes.

Tabla 66. Tiempo Promedio por solicitud - Consulta

N° Solicitudes	Tiempo Promedio
10 solicitudes	0.0416
100 solicitudes	0.04612
1000 solicitudes	0.046345

La Tabla 66. Tiempo Promedio por solicitud – Consulta, nos muestra el tiempo promedio de envío que se hizo en las 3 pruebas, de igual manera que se observó en el Gráfico Tiempo de Respuesta – Extorno, los tiempos fueron aumentando, en el caso de 10 a 100 solicitudes, aumento en un 10.87%, mientras que de 10 a 1000 aumento en un 11.41%, viéndose que los tiempos no son muy afectados debido a que solo se realiza una consulta a las deudas activas del estudiante.

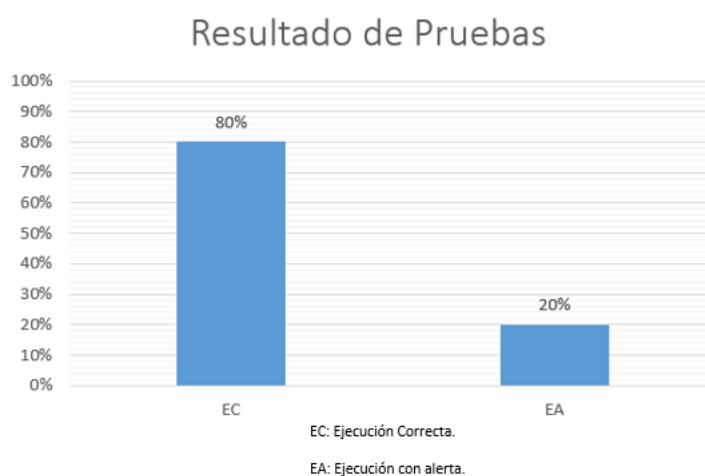
b) Pago de una deuda

Para la prueba del servicio de pagos hemos utilizado las mismas cargas utilizadas en el servicio de consulta esto es en grupos de 10, 100 y 1000 solicitudes enviadas al servidor, los resultados fueron los siguientes:

➤ Prueba de Carga 1 – Pagos: 10 solicitudes

Procesadas las 10 solicitudes se obtuvieron los siguientes resultados: 8 solicitudes fueron procesadas en su totalidad por encontrarse dentro de la calificación correctamente (EC), devolviendo al usuario “OPERACIÓN REALIZADA CON EXITO” con el estado 200 en el servidor, mientras que 2 solicitudes recibieron la anotación incorrecta (EA) con el estado 400 en el servidor (Ver el siguiente gráfico).

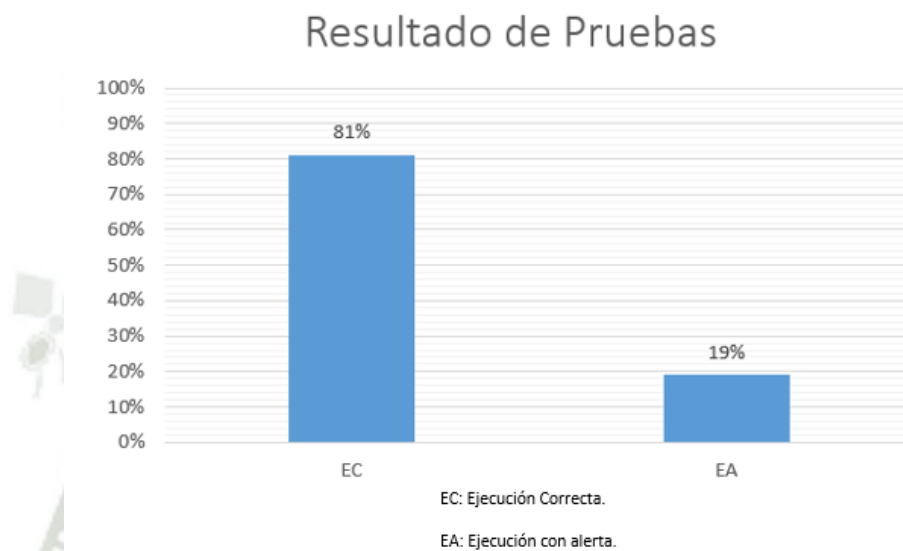
Gráfico 5. Prueba de Carga 1 - Pagos



➤ Prueba de Carga 2 – Pagos: 100 solicitudes

De igual manera en la prueba realizada sobre 100 solicitudes del servicio de Pago los resultados obtenidos fueron 81 ejecuciones correctas (EC) mientras que 19 ejecuciones no culminaron al recibir un error o notificación en el proceso (EA). (Ver el siguiente gráfico).

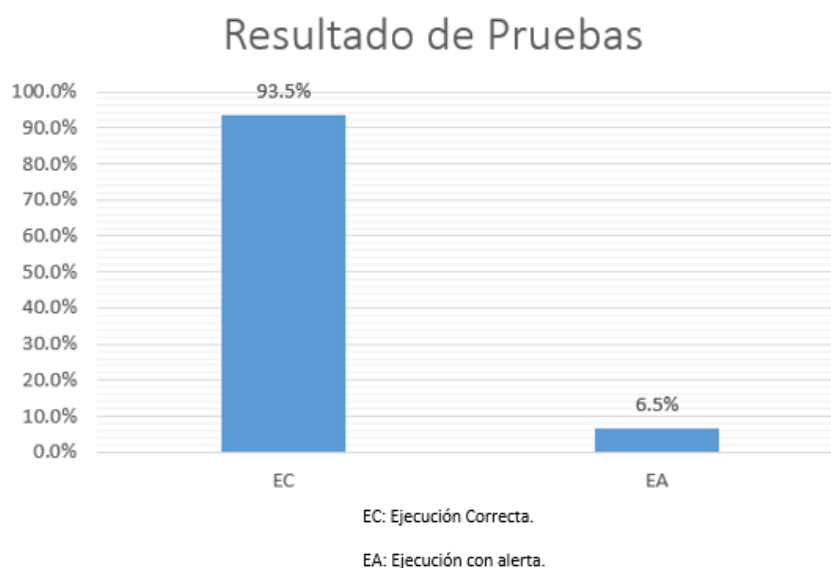
Gráfico 6. Prueba de Carga 2 - Pagos



➤ Prueba de Carga 3 – Pagos: 1000 solicitudes

En la prueba de solicitudes de Pago remitidas al servidor sobre 1000 solicitudes, en el grafico podemos ver que 935 consultas se ejecutaron correctamente (EC) y 65 solicitudes se recibieron con notificaciones o errores (EA). (Ver el siguiente gráfico)

Gráfico 7. Prueba de Carga 3 - Pagos



Al realizarse la Prueba de Carga 1 – Pagos, en el procesamiento de las solicitudes se tomó nota del Tiempo de ejecución y consumo de Memoria requerida por el servidor en el procesamiento de cada solicitud. El registro de dicho proceso lo presentamos a continuación:

Tabla 67. Tiempos de ejecución pagos

Número de Prueba	Tiempo de ejecución (seg)	Consumo de memoria (kb/seg)
1	0.183222	1361
2	0.196051	1304
3	0.184820	1372
4	0.193412	1392
5	0.218654	1374
6	0.180527	1318
7	0.183222	1353
8	0.195168	1363
9	0.227043	1309
10	0.189197	1365
11	0.229934	1400
12	0.191481	1389
13	0.191796	1317
14	0.183789	1338
15	0.199981	1365
16	0.208703	1382
17	0.185920	1321
18	0.198925	1340
19	0.193219	1329
20	0.189892	1355
21	0.202966	1371
22	0.190292	1390
23	0.185388	1354
24	0.194853	1350
25	0.183417	1331
26	0.211063	1362
27	0.183222	1329
28	0.196051	1326
29	0.184820	1361

30

0.316230

1304

Los datos consignados en la Tabla 55 nos dan a conocer que el tiempo de ejecución o procesamiento de las solicitudes oscila 18 y 19 décimas de segundo existiendo tiempos pico de 22 y 31 décimas de segundo, siendo casos especiales donde se tiene que hacer la verificación de deudas antiguas, la más reciente y la que cubre exactamente el monto ingresado a pagar.

El rubro Consumo de Memoria lo consideramos estándar ya está entre 1304 y 1400 Kb/seg. lo que pone en manifiesto que el servicio es eficiente.

A los datos tomados y consignados en la Tabla 68 referente al tiempo de ejecución o procesamiento de cada una de las solicitudes se agregaron la cantidad de solicitudes, tiempo de respuesta y solicitudes recibidas por segundo dentro de una carga de 10, 100, 500 y solicitudes a fin de observar el rendimiento del servicio, datos que dan forma a la siguiente Tabla:

Tabla 68. Resultados de las pruebas pagos

Numero de Prueba	Cantidad de solicitudes	Tiempo de ejecución por cada solicitud	Consumo de recursos (kb/seg)	Tiempo de respuesta	Solicitudes recibidas por segundo
1	10	0.192196	1351	1.982 seg	22,3
2	100	0.200221	1435	20.04 seg	21.6
3	500	0.243700	1553	121.08 seg	20,3
4	1000	0.354452	1647	353.98 seg	19,4

Como se puede observar los datos consignados en la Tabla 56, obtenidos de la prueba de pagos, el rendimiento del servicio es óptimo y eficiente ya que los tiempos de ejecución de las solicitudes y el consumo de recursos está dentro de los estándares establecidos por la cantidad de solicitudes, fecha de creación de la solicitud, la actualización de los datos (en el caso de pago y extorno), revisión de deudas activas, entre otros, todo lo cual como es lógico demanda mayor consumo de recursos y mayor tiempo de respuesta.

El tiempo de respuesta de acuerdo a los datos obtenidos los podemos graficar de la siguiente manera:

Gráfico 8. Tiempos de Respuesta - Pagos



El grafico y las tablas nos da a conocer que el servicio de Pagos requiere de mayor tiempo para el procesamiento de las solicitudes, así como mayor consumo de memoria debido a que este servicio a parte de consultar la base de datos del servidor de la entidad educativa y financiera realiza actualizaciones de acuerdo a la solicitud enviada al servidor.

Tabla 69. Tiempo Promedio por solicitud - Pagos

N° Solicitudes	Tiempo Promedio
10 solicitudes	0.1982
100 solicitudes	0.2004
1000 solicitudes	0.3540

La tabla anterior, nos muestra los tiempos promedio que demora entre solicitud, el servicio de pagos, gracias a eso se pudo obtener que los tiempos fueron aumentando, en el caso de 10 a 1000 solicitudes aumento en un 78.60%, mientras que en el caso de 10 a 100 aumento en 1.11%, lo que nos da a entender que el servicio entre más solicitudes este procesando va a demorar mas en procesar otra, esto debido a la búsqueda de la deuda que cubra totalmente el monto que se está pagando.

De la misma manera que en el servicio de consulta, en el servicio de pagos se ha utilizado el cliente cUrl de PHP, para la obtención de los tiempos de ejecución del servicio. Una pequeña parte del código utilizado se muestra en la Figura 46 donde se puede observar que el programa va enviar los datos de la consulta con 100 solicitudes al servicio de Pagos.

Figura 48. Tiempo de ejecución – pagos

```
$ch = curl_init($url);
$i=0;
while ($i<100){
    $time_init = microtime_service();
    curl_setopt($ch, CURLOPT_POSTFIELDS,$lcJson);
    curl_setopt($ch, CURLOPT_HTTPHEADER,array('Content-Type:application/json'));
    curl_setopt($ch, CURLOPT_RETURNTRANSFER,true);
    $result = curl_exec($ch);
    $time_fn = microtime_service();
    $time = $time_fn - $time_init;
    echo $i."."."Tiempo de ejecución:". $time.PHP_EOL.'<br>';
    $i++;
}
```

c) Reembolso de una deuda (extornar)

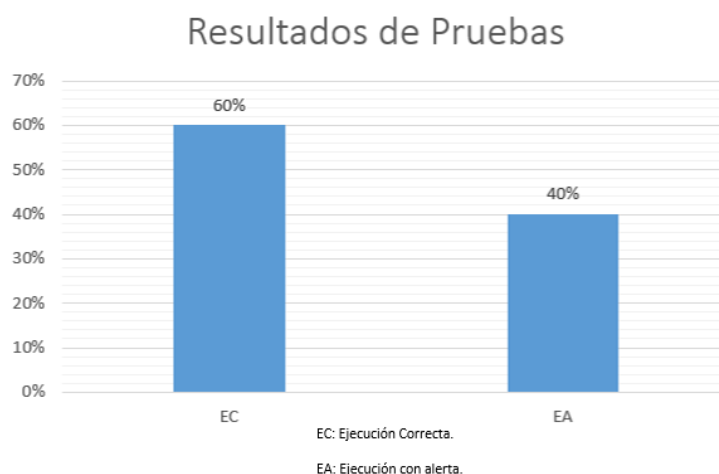
El extorno o reembolso de una deuda es el servicio que se presta al usuario a partir del procesamiento de una solicitud enviada al servidor para hacer el reembolso a una deuda anteriormente pagada. El proceso inicia con el envío de la solicitud al servidor y culmina con la respuesta que le es enviada al usuario.

En la prueba del servicio del extorno hemos tomado los mismos tipos de carga utilizadas en el servicio de consulta y en el servicio de pagos, esto es, grupos de 10,100 y 1000 solicitudes enviadas al servidor, los resultados obtenidos fueron los siguientes:

➤ Prueba de Carga 1 – Extorno de una deuda

Procesadas las 10 solicitudes remitidas al servidor se obtuvieron los siguientes resultados: 6 solicitudes fueron calificadas como correctas (EC) por lo tanto fueron procesadas en su totalidad mostrando el mensaje de “OPERACIÓN REALIZADA CON EXITO” al usuario con el estado 200 en el servidor, mientras que 4 solicitudes recibieron la anotación incorrecta (EA) con estado 400 en el servidor. (Ver el siguiente gráfico)

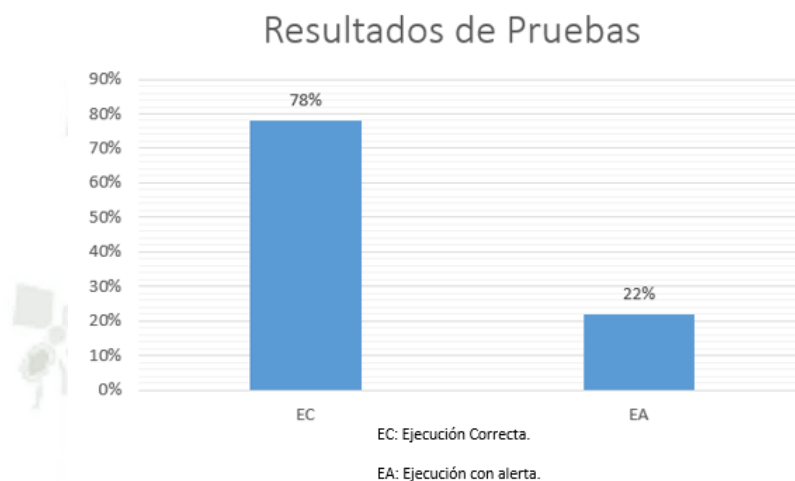
Gráfico 9. Prueba de Carga 1 - Extorno



➤ Prueba de Carga 2 – Extorno: 100 solicitudes

El resultado obtenido en esta prueba fue de 78 solicitudes correctas (EC) mientras que 22 fueron calificadas como incorrectas (EA) o no fueron procesadas devolviendo en el mensaje de respuesta “DEUDA CON ESTADO PAGADO” con estado 400 en el servidor.

Gráfico 10. Prueba de Carga 2 – Extorno

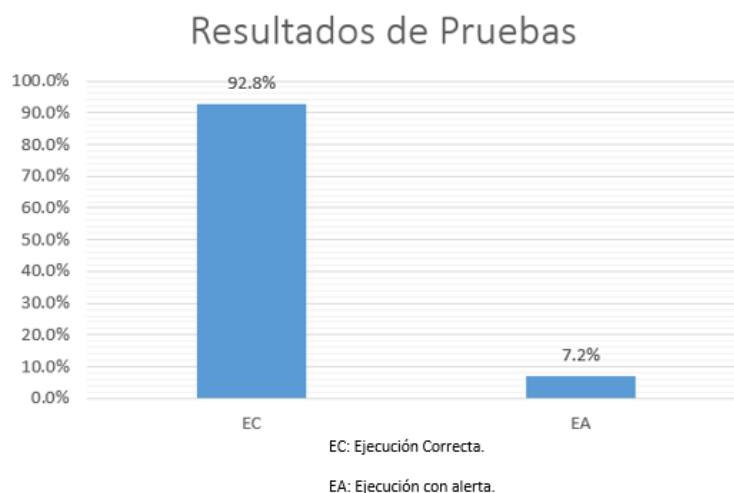


El resultado obtenido en esta prueba fue de 78 solicitudes correctas (EC) mientras que 22 fueron calificadas como incorrectas (EA) o no fueron procesadas devolviendo en el mensaje de respuesta “DEUDA CON ESTADO PAGADO” con estado 400 en el servidor.

➤ Prueba de Carga 3 – Extorno: 1000 solicitudes

En esta oportunidad el resultado fue de 928 solicitudes procesadas correctamente y 72 no procesadas por ser consideradas incorrectas (EA).

Gráfico 11. Prueba de Carga 3 - Extorno



Para finalizar, se observa el rendimiento que tiene el servicio para lo cual se elaboró una tabla donde se pusieron los promedios de los tiempos de ejecución, el consumo de recursos al ejecutar los servicios en las diferentes pruebas y el tiempo de respuesta a las solicitudes, observando cómo se incrementaba la cantidad de recursos utilizados entre más peticiones se hacían.

Lo delicado del servicio de extorno, mediante el cual se puede proceder a la devolución de dinero, nos llevó a tener que reunir información que permita determinar el rendimiento del servicio de extorno en especial el tiempo de ejecución de cada una de las solicitudes y el consumo de memoria que demandaba dicho proceso. Esta labor dio como resultado la elaboración de la siguiente Tabla:

Tabla 70. Tiempos de ejecución extorno

Número de Prueba	Tiempo de ejecución (seg)	Consumo de memoria (kb/seg)
1	0.015996	1015
2	0.012401	951
3	0.017430	1038
4	0.017300	983
5	0.015391	1012
6	0.014975	984
7	0.020773	1056
8	0.011077	1081
9	0.012582	1098
10	0.016694	1029
11	0.014327	968
12	0.012177	990
13	0.013571	1005
14	0.014022	957
15	0.013781	1040
16	0.011105	956
17	0.015028	1010
18	0.015551	1023
19	0.016827	1087
20	0.012206	980
21	0.011520	997

22	0.013050	1019
23	0.011965	984
24	0.012399	1034
25	0.015543	1050
26	0.018257	1002
27	0.010510	989
28	0.011547	1073
29	0.017345	1015
30	0.012945	951

Los datos consignados en la Tabla 70 nos llevan a concluir que el tiempo de ejecución y consumo de memoria están en estrecha relación, así mismo nos permite ver la diferencia de tiempo en el procesamiento de las solicitudes que oscila entre 0.015391 y 0.017345 segundos con un consumo de memoria entre 1012 y 1015 (kb/seg) respectivamente, sin embargo, esto no quiere decir que no se presenten casos que demanden menor y menor tiempo o consumo de memoria dependiendo de las validaciones a las que se vaya a ingresar dependiendo de la solicitud ingresada.

Para una mayor seguridad y seguir con nuestra metodología de validación del servicio en esta etapa y en procesamiento del servicio de Externo se tomaron datos relacionados con la cantidad de solicitudes, tiempo de ejecución de cada solicitud, consumo de recursos, tiempo de respuesta y solicitudes recibidas por segundo, información que se ha reunido sistemáticamente en la siguiente Tabla.

Tabla 71. Resultados de las pruebas externo

Numero de Prueba	Cantidad de solicitudes	Tiempo de ejecución por cada solicitud	Consumo de recursos (kb/seg)	Tiempo de respuesta	Solicitudes recibidas por segundo
1	10	0.015462	1012	0.15 seg	22.3
2	100	0.01654	1058	1.66 seg	22.1
3	500	0.01961	1096	9.81 seg	21.9
4	1000	0.02443	1135	20.54 seg	21.6

Los datos de la Tabla 70 nos permite ver que existe un aumento proporcional en el tiempo de ejecución de recursos y tiempo de respuesta determinado por la cantidad de solicitudes recibidas por el servidor.

Gráfico 12. Tiempo de Respuesta - Extorno



La Tabla 72. Tiempo Promedio por solicitud – Extorno, nos muestra el tiempo promedio de envío que se hizo en las 3 pruebas, de igual manera que se observó en el Gráfico Tiempo de Respuesta – Extorno, los tiempos fueron aumentando, en el caso de 10 a 100 solicitudes, aumento en un 10.67%, mientras que de 10 a 1000 aumento en un 36.93%, viéndose afectado de manera considerable, debido a que en los casos de extorno se necesita validar la fecha del pago para verificar si es posible realizar el extorno, a aparte registrar en el historial de extornos.

Tabla 72. Tiempo Promedio por solicitud - Extorno

N° Solicitudes	Tiempo Promedio
10 solicitudes	0.0150
100 solicitudes	0.0166
1000 solicitudes	0.0201

Al igual que en los estudios anteriores de los otros servicios analizados presentamos una pequeña muestra del código utilizado para la obtención de los tiempos de ejecución con el cliente cUrl en PHP para el servicio de Extorno.

Figura 49. Tiempo de ejecución – extorno

```
$ch1 = curl_init($urlP);
$ch2 = curl_init($urlE);
$i=0;
while ($i<1000){
    curl_setopt($ch1, CURLOPT_POSTFIELDS,$lcJsonP);
    curl_setopt($ch1, CURLOPT_HTTPHEADER,array('Content-Type:application/json'));
    curl_setopt($ch1, CURLOPT_RETURNTRANSFER,true);
    $resultP = curl_exec($ch1);
    $time_init = microtime_service();
    curl_setopt($ch2, CURLOPT_POSTFIELDS,$lcJsonE);
    curl_setopt($ch2, CURLOPT_HTTPHEADER,array('Content-Type:application/json'));
    curl_setopt($ch2, CURLOPT_RETURNTRANSFER,true);
    $resultE = curl_exec($ch2);
    $time_fn = microtime_service();
    $time = $time_fn - $time_init;
    echo $i."."."Tiempo de ejecucion:".$time.PHP_EOL."<br>";
    $i++;
}
curl_close($ch1);
curl_close($ch2);
```

4.3 Comparativa con trabajos relacionados

La investigación desarrollada tiene por objetivo el desarrollo de un marco de trabajo que permita la implementación de servicios entre 2 entidades (educativa y financiera), teniendo en cuenta que los principales procesos puedan convertirse en servicios API-REST seguros y óptimos para lo cual se identificaron 8 servicios candidatos en base a procesos de la institución, pero posteriormente se determinó que para la implementación del marco de trabajo propuesto solo se necesitarían los 3 primeros según la tabla de priorización que se vio en el capítulo 3 de nuestro trabajo.

En la investigación “Análisis de frameworks de desarrollo de API-REST y su impacto en el rendimiento de aplicaciones web con arquitectura Spa”, Valencia Altamirano, 2018, se realiza la comparativa de los tiempos de ejecución de los servicios en diferentes frameworks utilizando JMeter para la obtención de las métricas, obteniendo resultados bajos en consumo de memoria así como tiempos de ejecución de 2 segundos para las pruebas de 50 usuarios, mientras que el servicio desarrollado para las entidades bancarias por la gran cantidad de datos que manejan utilizan una mayor cantidad de recursos bordeando los 1306 kb por segundo (promedio) en cada operación realizada, en un tiempo promedio de 0,07 segundos en cada operación (se usó como referencia el servicio de consulta).

En la investigación, “Un framework como solución para mejorar la gestión de servicios rest requeridos en los pases a producción de una entidad bancaria”, Carolay Piedad, 2021, se detalla cómo mejorar el servicio de pases de una entidad bancaria, y propone la incorporación de un sistema de tickets, mostrando los resultados, la cantidad de tickets procesados en un periodo

de 3 meses, clasificando en el servicio la cantidad de tickets que no pudieron generarse debido a un error en el servicio, identificando los principales motivos por los cuales no se pudieron generar dichos tickets, en la propuesta de nuestro trabajo la investigación, se detalla en el capítulo 3, los diferentes códigos que los servicios API-REST retornan al detectar un error, durante la prueba de carga, esta evaluación se realiza al verificar solicitudes con datos erróneos donde el servicio devolverá las respuestas con sus respectivos errores.



Conclusiones

- Se desarrolló un marco de trabajo capaz de identificar los servicios a implementarse en una entidad educativa sin tener que asignar un servidor dedicado que procese las transacciones con la entidad financiera a interconectarse. Además de permitir la bancarización de las operaciones (Decreto Legislativo 1529) que pasen de S/.2 000 y la optimización de los procesos de consultas, pagos y extorno permitiendo realizarse con un tiempo promedio de 0.0621, 0.1945 y 0.0144 segundos respectivamente.
- Se hizo la identificación de los requerimientos, donde se pudo identificar los requerimientos funcionales y no funcionales para cada uno de los servicios candidatos, lo cual nos ayudó a establecer las principales funciones de los servicios a implementarse, así como el desarrollo de las funciones en PostgreSQL.
- Se hizo el modelamiento de los datos, inicialmente el modelado se hizo con un diagrama E-R, donde se identificaron 4 entidades (alumno o estudiante, escuela profesional, la deuda y el banco) para luego ser pasado a un diagrama de clases donde cada una de las entidades paso a convertirse en una clase. Además del modelado de datos, se tuvo que modelar la estructura de las solicitudes para cada servicio (consulta, pago y extorno) que se implementó; todo esto va a permitir tener documentado la estructura (de los datos, solicitud y respuesta) en cada uno de los servicios implementados en caso se necesite reestructurarlo o modificarlo para la agregar una capa extra de seguridad.
- El uso de los diagramas UML, nos permitió detallar la semántica de los servicios implementados, al poder identificar las entidades y componentes que participan en cada servicio; en el caso del diagrama de componentes, mostramos los componentes (cajero, servidor de la entidad financiera y educativa) y la manera que se comunican para llevar a cabo las actividades del servicio API-REST al que se esté invocando para consumir.
- Se realizaron 2 tipos de pruebas, funcionales y de carga; en el caso de las pruebas de carga mostramos que tanto incremento hubo al realizar las 3 cargas a los servicios implementados, obteniendo de resultado en un incremento de 10.86% al incrementar de 10 a 100 solicitudes, mientras que hubo un incremento de 11.41 al aumentar a 1000 solicitudes para el servicio de consultas; en el caso de pagos, el incremento fue de 1.11% y 78.60%, respectivamente siendo el servicio más afectado; en el caso de extornos la variación no fue mucha incrementando de 10.67% y 36.94% respectivamente.

Recomendaciones y trabajos futuros

- Implementar un token digital que permitan mejorar la seguridad de los servicios API-REST vistos.
- Desarrollar una plataforma web que permita la utilización de los servicios API-REST realizados durante la interconexión con el objetivo de realizar las operaciones vistas en nuestro proyecto.
- Desarrollar los servicios API-REST en lenguajes de programación que estén dirigidos al desarrollo de webservices (API-REST, SOAP).
- Reducir la cantidad de datos que son enviados en la solicitud, lo que agilizaría las respuestas en el servidor.



Referencias

- Aguilar, M. N. E. (2012). Diseño de una arquitectura orientada a servicios para un establecimiento de salud de nivel de complejidad I-3 [Tesis de Grado, Universidad Peruana de Ciencias Aplicadas]. Repositorio Académico UPC.
- Atencio Flores, A. F. D., & Mamani Machaca, M. M. D. (2018). Interconectividad basado en API REST en aplicaciones de la Municipalidad Provincial de Lampa. [Tesis de Grado, Universidad Nacional del Altiplano]. Repositorio Institucional UNAP
- Bengtsson, P. (2020, 8 diciembre). *Trabajando con JSON - Aprende sobre desarrollo web* | MDN. MDN Plus.
<https://developer.mozilla.org/es/docs/Learn/JavaScript/Objects/JSON>
- Calvo, D. (2017, 22 noviembre). Procedimientos y funciones en PostgreSQL. Diego Calvo. <https://www.diegocalvo.es/procedimientos-y-funciones-en-postgresql/>
- Carolay Piedad, D. G. (2021, 9 septiembre). Un framework como solución para mejorar la gestión de servicios rest requeridos en los pases a producción de una entidad bancaria [Trabajo de Suficiencia Profesional, Universidad Nacional Mayor de San Marcos]. Cybertesis Repositorio de Tesis UNMSM.
- Corrales Rubiano, A. & López Herrera, C. Y. (s. f.). Identificadores digitales: Una herramienta que apoya la recuperación de información [Artículo]. E-prints in Library & Information Science. <http://eprints.rclis.org/10599/>
- Decreto Legislativo N°1529. (3 de marzo del 2022) que modifica la ley para la lucha contra la evasión y para la formalización de la economía. Diario Oficial El Peruano, 3 de marzo del 2022
<https://busquedas.elperuano.pe/normaslegales/decreto-legislativo-que-modifica-la-ley-para-la-lucha-contradecreto-legislativo-n-1529-2044433-2/>
- Dimitru Boboia, R. (2018). Herramientas y buenas prácticas para el desarrollo, mantenimiento y evolución de Software en Java. [Proyecto Fin de Grado, Universidad Politécnica de Madrid]. Archivo Digital UPM
- Florencio, A. (2020, 27 octubre). Instalación de Sublime Text 3. Paquetes para HTML, CSS y JavaScript. Cursos GIS | TYC GIS Formación.

<https://www.cursosgis.com/instalacion-de-sublime-text-3-paquetes-para-html-css-y-javascript/>

García Núñez, A. R. (2017, enero). Restful: un caso de uso de gestión de bibliotecas [Trabajo de Fin de Grado, Universidad Politécnica de Madrid]. Archivo Digital UPM.

Gascón Gascón, S. G. G. (2018). Plataforma integrada de pagos para la Universidad de Zaragoza - Repositorio Institucional de Documentos [Trabajo de Fin de Grado, Universidad de Zaragoza]. ZAGUAN Repositorio Institucional de Documentos.

kanbanize. (s. f.). *¿Qué es Kanban? Principales características y funciones*. Kanban Software for Agile Project Management. <https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-kanban>

Lázaro, D. (2018). Introducción a los Web Services. Introducción a los Web Services. <https://diego.com.es/introduccion-a-los-web-services>

Liaño Díaz, Joaquín (2017, julio). Modelado de comunicaciones entre servicios internos y aplicaciones entre servicios internos y aplicaciones externas mediante APIS en el entorno bancario [Trabajo de Fin de Grado, Universidad Pontificia Comillas]. Repositorio Comillas.

Meneses Rivera, K. A. (2021, 23 abril). Desarrollo de una API REST para la integración de CRM con el sistema de gestión de colas en empresa de fondos colectivos [Trabajo de Suficiencia Profesional, Universidad Nacional Mayor de San Marcos]. Cybertesis Repositorio de Tesis UNMSM.

Pascual, J. R. (2020, enero 4). Arquitectura SOA y Composición de Servicios - Disrupción Tecnológica. Disrupción Tecnológica; <https://www.disrupciontecnologica.com/arquitectura-soa-y-composicion-de-servicios/>

Redondo, F. (2021, 5 julio). Postman: gestiona y construye tus APIs rápidamente. Paradigma. <https://www.paradigmadigital.com/dev/postman-gestiona-construye-tus-apis-rapidamente/>

Romano Rodríguez, S. (2021, junio). Desarrollo de una API de agregación de cuentas bancarias según normativa PSD2 [Proyecto de Fin de Grado, Universidad Politécnica de Madrid]. Archivo Digital UPM.

Romero, G. (2021, 29 junio). Cómo realizar pruebas automatizadas con Postman. Encora.
<https://www.encora.com/es/blog/como-realizar-pruebas-automatizadas-con-postman>

The PHP Group. (s. f.). *PHP: Introducción - Manual*.
<https://www.php.net/manual/es/intro.curl.php>

Valencia Altamirano, V. A. D. G. (2018, 8 junio). Análisis de frameworks de desarrollo de api rest y su impacto en el rendimiento de aplicaciones web con arquitectura Spa [Tesis de Postgrado, Universidad Técnica del Norte]. Repositorio Digital Universidad Técnica del Norte.

