



UNIVERSIDAD CATOLICA SANTA MARIA
FACULTAD DE CIENCIAS E INGENIERIAS FÍSICAS Y FORMALES
ESCUELA PROFESIONAL DE INGENIERIA DE SISTEMAS

PROPUESTA DE AMPLIACIÓN DE UML CON ELEMENTOS PARA LA REPRESENTACIÓN DE MÉTRICAS Y CONTROL EN LA FASE DE DISEÑO DE SOFTWARE

**Tesis presentada para optar el:
Título Profesional de INGENIERO DE SISTEMAS**

RENE ALONSO NIETO VALENCIA

Asesor: ANGEL MONTESINOS MURILLO
Arequipa, 2016

PRESENTACIÓN

Sra. Directora del Programa Profesional de Ingeniería de Sistemas.

Srs. Miembros del Jurado.

De conformidad con las disposiciones del Reglamento de Grados y Títulos del Programa Profesional de ingeniería de Sistemas, ponemos a vuestra consideración el presente trabajo de investigación titulado: “PROPUESTA DE AMPLIACIÓN DE UML CON ELEMENTOS PARA LA REPRESENTACIÓN DE MÉTRICAS Y CONTROL EN LA FASE DE DISEÑO DE SOFTWARE”, el mismo que de ser aprobado me permitirá optar el Título Profesional de Ingeniería de Sistemas.

Rene Alonso Nieto Valencia

RESUMEN

El uso y la construcción de productos de software, permite que los usuarios finales sean los referentes potenciales para la evaluación de calidad de los mismos. La industria respondió a la demanda de un mercado global en crecimiento, pero esa respuesta no siempre tuvo en cuenta las exigencias de calidad y calidad en uso de usuarios cada vez más numerosos y mejor formados.

En esta tesis se propone una ampliación del enfoque de UML para obtener medidas y control en la etapa de diseño de software. El marco ofrece una ampliación que permite determinar las características internas del producto a construir sobre detalles de mediciones y control. Para llevar adelante el caso de estudio se empleó una metodología en particular, diseñada para evaluación de la ampliación de UML. Un reto importante fue adaptar la simbología a los artefactos diseñados por UML.

El caso de estudio se realizó con la participación de usuarios reales de una aplicación para el estudio jurídico del Dr. David Dong Ortega. El diseño de las métricas e indicadores empleados en el caso de estudio constituye uno de los aportes significativos de esta tesis ya que, si bien los resultados de este tipo de evaluaciones no son generalizables, se generaron metadatos provenientes de definiciones exhaustivas que pueden constituir líneas base para proyectos similares.

ABSTRACT

The use and construction of software products, allows end users are those concerning potential for quality assessment thereof. The industry responded to the demand of a growing global market, but the answer is not always taken into account the requirements of quality and quality in use ever more numerous and better trained users.

In this thesis an extension of UML approach is proposed for measurement and control in the software design stage. The framework provides an extension that allows to determine the internal characteristics of the product to build on details of measurements and control. To carry out the case study a particular methodology designed to evaluate the extension of UML was used. A major challenge was to adapt the symbology designed by UML artifacts.

The case study was conducted with the participation of actual users of an application for legal study by Dr. David Dong Ortega. The design metrics and indicators used in the case study is one of the significant contributions of this thesis since, although the results of these assessments are not generalizable, metadata from exhaustive definitions that may constitute baselines were generated for similar projects.

INTRODUCCION

Tras varios meses de trabajo dentro del metamodelo de UML y después de conversar con personas experimentadas en el desarrollo de productos de software, se determinó cuáles eran las necesidades de la investigación:

- La información de los productos a construir no se encontraba controlada ni protegida sino distribuida por las diferentes portátiles, de los que son responsables los propios usuarios internos. Esto supone un importante riesgo de pérdida del conocimiento sobre los clientes en el momento que un encargado abandone la organización.
- Las Gerencias no tienen la información actualizada con la frecuencia que sería deseable.
- No se dispone de una herramienta informática que permita una colaboración eficaz de todos los integrantes del proyecto en desarrollo.

Para ello se propuso la ampliación de UML para permitir llevar a cabo un control de la situación en la etapa de diseño del producto computacional. El capítulo 1 trata sobre las generalidades del trabajo de investigación mientras que el capítulo 2 enmarca el aspecto conceptual de la tesis. En el capítulo 3 se define la ampliación de UML mientras que en el capítulo 4 se narra la aplicación de la ampliación al caso de estudio. Finalmente se arriba a las conclusiones y recomendaciones del caso.

INDICE

Contenido	Pag.
Presentación	ii
Resumen	iii
Abstract	iv
Introducción	v
Índice	vi
CAPITULO 1: Planteamiento teórico	1
1.1 Título del proyecto	1
1.2 Descripción del problema	1
1.3 Delimitaciones y definición del problema	1
1.3.1 Delimitaciones	1
1.3.2 Definición del problema	2
1.4 Formulación del problema	4
1.4.1 Problema principal	4
1.5 Objetivos de la investigación	4
1.5.1 objetivo general	4
1.5.2 Objetivos específicos	4
1.6 Viabilidad de la investigación	5
1.6.1 Económica	5
1.6.2 Técnica	5
1.6.3 Operativa	6
1.7 Justificación e importancia de la investigación	6
1.7.1 Justificación	6
1.7.2 Importancia	7
1.8 Limitaciones de la investigación	7
1.9 Hipótesis de la investigación	7
1.9.1 Hipótesis general	7

1.10 Variables e indicadores	7
1.10.1 Variable independiente	7
1.10.2 variable dependiente	7
1.10.3 Indicadores	8
1.11 Área, línea, tipo y nivel de la investigación	8
1.11.1 Área de investigación	8
1.11.2 Línea de investigación	8
1.11.3 Tipo de investigación	8
1.11.4 Nivel de investigación	8
1.12 Método y diseño de la investigación	8
1.12.1 Método de la investigación	8
1.12.2 Diseño de la investigación	8
1.13 Cobertura del estudio	10
1.13.1 Universo	10
1.13.2 Muestra	10
1.14 Técnicas e instrumentos de recolección de información	10
1.14.1 Técnicas	10
1.14.2 Instrumentos	11
CAPITULO 2: Marco teórico	12
2.1 Antecedentes investigativos	12
2.2 Marco conceptual	15
2.2.1 UML	15
2.2.2 Desarrollo de software	34
CAPITULO 3: Propuesta de ampliación de UML	40
3.1 Introducción	40
3.2 Descripción de la propuesta	43
3.2.1 Etapa de educación de requisitos	43
3.2.2 Etapa de elicitación de requisitos	45

3.2.3 Etapa de especificación de requisitos de software	48
3.2.4 Etapa de diseño del diagrama de casos de uso	50
3.2.5 Etapa de diseño del modelo conceptual	52
3.2.6 Etapa de diseño de los diagramas de secuencias	54
3.2.7 Etapa de diseño de interfaces gráficas de usuario	55
3.2.8 Etapa de diseño de la arquitectura de software	59
3.2.9 Etapa de diseño de los diagramas de colaboración	62
3.2.10 Etapa de diseño del diagrama de clases	64
3.2.11 Medidas e indicadores	68
CAPITULO 4: Caso de estudio	71
4.1 Introducción	71
4.2 Desarrollo del producto	72
Conclusiones	143
Recomendaciones	144
Bibliografía	145
Anexo N° 1	147

INDICE DE FIGURAS

Contenido	Pag.
Figura N° 1: Diagrama de clase	21
Figura N° 2: Relación de dependencia en diagramas de clase	23
Figura N° 3: Auto-agregación	23
Figura N° 4: Diagrama de componentes	24
Figura N° 5: Diagrama de despliegue	25
Figura N° 6: Diagrama de casos de uso nivel 1	25
Figura N° 7: Diagrama de casos de uso nivel 2 A	26
Figura N° 8: Diagrama de casos de uso nivel 2B	26
Figura N° 9: Diagrama de casos de uso nivel 1 detallado	28
Figura N° 10: Diagrama de secuencia	29
Figura N° 11: Diagrama de colaboración	31
Figura N° 12: Máquinas de estados, estados simples	32
Figura N° 13: Máquinas de estados, estados compuestos	33
Figura N° 14: Diagrama de actividades	34
Figura N° 15: Proceso de desarrollo de software	35
Figura N° 16: Elementos del proceso del software	37
Figura N° 17: Relación entre elementos del proceso del software	38
Figura N° 18: Ciclo de vida de la ampliación UML propuesta	42
Figura N° 19: Ampliación UML para la educación de requisitos	44
Figura N° 20: Ampliación UML para la elicitación de requisitos	47
Figura N° 21: Ampliación UML para la especificación de requisitos de software	49
Figura N° 22: Ampliación UML para el diagrama de casos de uso	51
Figura N° 23: Ampliación UML para el modelo conceptual	53
Figura N° 24: Ampliación UML para el diagrama de secuencias	54
Figura N° 25: Ampliación UML para los interfaces gráficos de usuario	57
Figura N° 26: Ampliación UML para los interfaces gráficos de usuario por componentes	58
Figura N° 27: Ampliación UML para la arquitectura de software	61

Figura N° 28: Ampliación UML para la arquitectura de software	63
Figura N° 29: Ampliación UML para el diagrama de clases	65
Figura N° 30: Ampliación UML para componentes de un diagrama de clases	66
Figura N°31: Conectores que funcionan como elementos de estereotipo	67
Figura N° 32: Clases que redireccionan a metaclases basadas en estereotipo	67
Figura N° 33: Clases que redireccionan servicios de una clase o interfaz	67



INDICE DE TABLAS

Contenido	Pag.
Tabla N° 0: Costos de la investigación	5
Tabla N° 1: Variables y técnicas para la investigación	9
Tabla N° 2: Elementos de construcción en UML	16
Tabla N° 3: Elementos de relación en UML	17
Tabla N° 4: Diagramas de UML	18
Tabla N° 5: Multiplicidad en diagramas de clases	21
Tabla N° 6: variables y técnicas para la investigación	30
Tabla N° 7: Especificación para la educación de requisitos	45
Tabla N° 8: Especificación para la elicitación de requisitos	48
Tabla N° 9: Especificación para la especificación de software	50
Tabla N° 10: especificación para los casos de uso	51
Tabla N° 11: especificación del modelo conceptual	53
Tabla N° 12: especificación para el diagrama de secuencias	55
Tabla N° 13: especificación para los interfaces gráficos de usuario	58
Tabla N° 14: Especificación para la arquitectura de software	61
Tabla N° 15: Especificación para el diagrama de colaboración	63
Tabla N° 16: Especificación para el diagrama de clases	68
Tabla N° 17: Medidas e indicadores	69

Capítulo 1: Planteamiento de la investigación

1.1 Título del proyecto

Propuesta de ampliación de UML con elementos para representación de métricas y control en la fase de diseño de software.

1.2 Descripción del problema

La Ingeniería de Software está reconocida por la aplicación de métricas durante el ciclo de vida de desarrollo de los productos de software; lo que permite predecir atributos externos de la calidad como de la mantenibilidad. Generalmente se espera cumplir todo el ciclo de vida de desarrollo para cumplir con la obtención de estas métricas; pero se pueden aproximar estas medidas a partir de algunos artefactos concebidos durante la construcción del producto de software.

Las frecuentes pérdidas que se registran en los proyectos de desarrollo de software reclaman una constante mejora en el control de los proyectos de software, de los procesos de desarrollo y de los métodos de estimación. Los errores cometidos en las estimaciones suelen estar estrechamente relacionados con las pérdidas que ocasiona el proyecto. Por lo tanto disminuir el error, como lo son errores de análisis, diseño y codificación, implica evitar grandes pérdidas económicas.

Los resultados de las estimaciones están lejos de satisfacer las necesidades de la industria del software, puestos que los mismos suelen presentar un valor que no es aceptable para los inversores, esto es, una magnitud del error relativo mayor a lo predicho. La importancia de las métricas ha sido reconocida por la definición de los estándares de métricas de tamaño funcional (Somerville, 2005).

1.3 Delimitaciones y definición del problema

1.3.1 Delimitaciones

a. Delimitación espacial

El presente trabajo de investigación se lleva a cabo en la ciudad de Arequipa.

b. Delimitación temporal

El trabajo se inicia en octubre del 2013 y culmina en enero del 2013.

c. Delimitación social

Está orientado a resolver problemas de construcción de software, lo cual implica a Ingenieros de Sistemas, Ingenieros Informáticos, Ingenieros en Ciencias de la Computación, Ingenieros de Software, Ingenieros de Requerimientos y Arquitectos de Software así como Ingenieros con especialidad en Tecnologías de Información y Comunicaciones.

d. Delimitación conceptual

Métricas, activos de procesos, Ingeniería de software, diagrama de clases, casos de uso.

1.3.2 Definición del problema

Normalmente empleamos UML para el modelamiento del producto de software a construir logrando insertar el concepto de calidad. Un factor crítico en el éxito de los proyectos es que los desarrolladores no solamente entiendan qué tienen que desarrollar, sino que lo que están desarrollando contiene elementos de calidad que les permite entender que los procesos están bien realizados. El enfoque de métricas se concentra en por qué el sistema es construido, proveyendo motivación y fundamentos para justificar la construcción de productos software (Pressman, 2002)(Somerville, 2005).

Los sistemas de métricas de software tradicionales se han centrado fundamentalmente en las métricas de procesos, de productos y de recursos.

Los principales objetivos de todos los sistemas de métricas desarrollados hasta el momento son el control y la estimación de los proyectos, la calidad del desarrollo o los costes y esfuerzos estimados para la fase de mantenimiento de la aplicación. En algunas ocasiones se han utilizado las métricas en tiempo de ejecución, normalmente como estimadores de velocidad, testing de ejecución, profiling o para estimación del coste de servicios, pero en ningún caso con intención de control en el proceso de desarrollo (Pressman, 2002)(Somerville, 2005).

Por otro lado, se debe recordar que las ventas totales estimadas para el año 2003 fueron de 69 millones de dólares con una tasa de crecimiento cercana al 30%, mientras que la exportación fue de 7.3 millones de dólares con una tasa de crecimiento del 44%. Las empresas de software destinan el 13% de sus costos totales a la Investigación y Desarrollo y el 52% de las empresas encuestadas considera que las certificaciones de calidad son muy caras; sin embargo, el 87% estaría dispuesto a invertir en la implementación de programas de mejoramiento de procesos de software (Pressman, 2002).

Estos datos nos demuestran que son pocas las empresas que se apoyan en estimaciones del esfuerzo para obtener calidad en sus productos, por lo que el presente trabajo de investigación cubre un aspecto importante en la calidad del producto de software. Asimismo, define un marco de trabajo en el cual integra métricas de complejidad en ámbitos diferentes a la complejidad algorítmica basados en el contenido y los principios. También define un conjunto de métricas para medir la complejidad estructural de los diagramas de clase UML debido al uso de relaciones de asociación, generalización, agregación y dependencias; estudio que lo llevan a cabo bajo el análisis de Componentes Principales (Moreno, 2000).

Las estadísticas demuestran que el modelamiento de los sistemas comerciales falla debido a que no se lleva un control del esfuerzo, ello implica

que las metodologías propuestas no contienen la suficiente semántica para representar los elementos del modelamiento (Somerville, 2005). UML fue diseñado para resolver el modelamiento de un subconjunto de sistemas de información, dejando de lado los sistemas métricos y de control (Pressman, 2002), mientras que las metodologías alternativas intentan suplir el modelamiento de un subconjunto mucho más reducido de sistemas de información. A partir de ello, incrementar elementos auxiliares a UML permitirá, que aprovechando los elementos existentes se pueda diseñar sistemas métricos y de control (Pressman, 2002).

1.4 Formulación del problema

1.4.1 Problema principal

Proponer una ampliación del Lenguaje de Modelado Unificado que utilizando algunos elementos auxiliares conjuntamente con los propuestos por UML permita una representación de métricas y control en la fase del diseño de software.

1.5 Objetivos de la investigación

1.5.1 Objetivo general

Elaborar una propuesta de ampliación del Lenguaje de Modelado Unificado que permita la representación de métricas y control en la fase del diseño de software.

1.5.2 Objetivos específicos

- Proponer un marco de trabajo básico que permita insertar un conjunto de elementos en el Lenguaje de Modelado Unificado.
- Diseñar un conjunto de símbolos que permita una ampliación del Lenguaje de Modelado Unificado (UML) para soportar la representación de métricas y control en la fase de diseño de software.
- Validar la propuesta de ampliación de UML, mediante un caso de estudio, que nos permita obtener métricas para el análisis respectivo.

1.6 Viabilidad de la investigación

1.6.1 Económica

Se cuenta con los recursos económicos necesarios para solventar el presente trabajo de investigación. Los mismos que serán afrontados, en su integridad, por el investigador. Los detalles se muestran en la tabla siguiente:

Tabla 0

Costos de la investigación

RECURSOS	COSTO UNITARIO	CANTIDAD	COSTO TOTAL
HUMANOS:			
• Consultor 1	S/. 32.00	25 Horas	S/. 800.00
• Consultor 2	S/. 32.00	25 Horas	S/. 800.00
• Especialistas	S/. 100.00	2 Horas	S/. 200.00
ECONÓMICOS:			
• Laptop core i5	S/. 2180.00	1 unid.	S/.2180.00
• Horas de Internet	S/. 105.00	5 meses	S/. 525.00
• Scanner	S/. 1.50	50 hojas	S/. 75.00
• Libros	S/. 80.00	6 libros	S/. 480.00
• Discos Compactos	S/. 20.00	1 cajas	S/. 20.00
• Discos Flexibles	S/. 10.00	1 cajas	S/. 10.00
• Papel	S/. 17.00	4 paquetes	S/. 68.00
• Tinta	S/. 42.00	4 cartuchos	S/. 168.00
• Fotocopias	S/. 0.10	200 hojas	S/. 20.00
TOTAL			S/. 5346.00

Fuente: Elaboración propia

Los beneficios se encuentran enmarcados dentro del aspecto personal, ya que la propuesta permitirá que los profesionales que construyan software tengan herramientas que les permita llevar un mejor control del mismo mediante la confección de un conjunto de medidas que aseguren parte de la calidad del software.

1.6.2 Técnica

Se cuenta con la capacidad académica para resolver el problema.

1.6.3 Operativa

Medios bibliográficos como libros, artículos, tesis de pregrado, tesis de postgrado, Internet, bibliotecas y otras universidades con infraestructura, laboratorios, entre otros.

1.7 Justificación e importancia de la investigación

1.7.1 Justificación

La Ingeniería de Software, busca producir una descripción detallada de un problema, con el fin de construir un Sistema de Software, que satisfaga las “necesidades y objetivos” de la organización donde funcionará dicho sistema. En la comunidad de Ingeniería de Software, estos objetivos constituyen el fundamento del sistema, y son usualmente definidos como las metas a ser cumplidas por el sistema y su entorno, aunque algunos autores distinguen los objetivos del sistema de los objetivos de la organización.

Debido a que los documentos de construcción cumplen también una función contractual entre analistas y stakeholders, es importante para los stakeholders que el contenido de las métricas esté en un lenguaje entendible con el cual ellos puedan interactuar activamente.

Los stakeholders son aquellas personas afectadas de un modo u otro por el sistema a construir, es decir, tienen alguna clase de interés en el proyecto. Mediante la construcción de diagramas con un nuevo enfoque los desarrolladores se comunican con los stakeholders usando un lenguaje basado en conceptos con los que se sienten familiarizados y confortables.

A los stakeholders puede resultarle más fácil comprender los objetivos a cumplir que la funcionalidad que se exhibiría en el sistema deseado. Las

métricas a menudo son difíciles de entender, pero ellos pueden ser justificados y explicados a través de la discusión de calidad. Debe notarse que la construcción de otros productos de software, a partir del análisis de las métricas de complejidad estructural, es más estable pudiendo obtener costos más reales.

1.7.2 Importancia

Permitir a los diseñadores de proyectos de software obtener una ampliación del Lenguaje de Modelado Unificado permitiéndole diseñar sistemas métricos y de control proporcionándole una visión sustancial del proyecto de desarrollo a partir de las métricas de complejidad estructural.

1.8 Limitaciones de la investigación

El trabajo de investigación no contempla a aquellas metodologías o lenguajes de modelado diferentes a UML. Asimismo, no se contempla emplear otro tipo de artefacto usado en la concepción de medidas de software en su fase de diseño.

1.9 Hipótesis de la investigación

1.9.1 Hipótesis general

Dada la importancia que representa el diseño del software para los sistemas de información, es probable que a partir de la ampliación de los elementos del Lenguaje de Modelado Unificado permita representar las métricas y control como elementos suficientes para su diseño.

1.10 Variables e indicadores

1.10.1 Variable independiente

- Lenguaje de modelado unificado.

1.10.2 Variable dependiente

- Técnica propuesta.

1.10.3 Indicadores

- Cantidad de artefactos que presentan dificultades de construcción en el momento del diseño del sistema.
- Cantidad de procesos en los sistemas métricos y control que no pueden ser representados mediante artefactos, propuestos por la técnica, con el Lenguaje de Modelado Unificado.

1.11 Área, línea, tipo y nivel de la investigación

1.11.1 Área de investigación

El área de investigación es la de Ingeniería de Software.

1.11.2 Línea de investigación

La línea de investigación es la de métricas y calidad de Software.

1.11.3 Tipo de investigación

Aplicada.

1.11.4 Nivel de investigación

Experimental porque se ensaya una nueva solución con el fin de resolver, modificar o mejorar alguna situación concreta en el modelamiento del software. Este tipo de investigación tiene la función de evaluar o auditar si alguna acción, programa o política está teniendo los resultados esperados.

1.12 Método y diseño de la investigación

1.12.1 Método de la investigación

Investigación aplicada, empleando el método lógico deductivo.

1.12.2 Diseño de la investigación

Para la investigación se utilizará las siguientes técnicas, instrumentos y materiales de verificación, como se señala en la siguiente tabla:

Tabla 1

Variables y técnicas para la investigación

VARIABLES	TECNICAS	INTRUMENTOS DOCUMENTALES
UML	Observación	Ficha de observación documental
Técnica propuesta.	Observación	Ficha de observación documental

Fuente: Elaboración propia

1.12.2.1 Forma de Tratamiento de los Datos

Para el proceso de los datos provenientes de la aplicación de los instrumentos, se utilizará:

a) Matriz de Sistematización de Datos

Para consolidar los datos de la aplicación de la técnica, estos se sistematizarán de acuerdo a los rangos conceptuales previstos en la definición.

b) Matriz de Tabulación

Con el fin de contabilizar las respuestas a las observaciones hechas al caso de estudio que se aplicará.

c) Cuadros Estadísticos

Elaboración de cuadros estadísticos descriptivos que permitan visualizar las respuestas correspondientes en términos de indicadores.

d) Análisis

Se hará un análisis estadístico descriptivo aplicado a los resultados obtenidos.

1.13 Cobertura del estudio

1.13.1 Universo

Uno de los mayores problemas que presenta nuestra ciudad es la confección de estadísticas sobre el área de tecnología, a pesar de los esfuerzos que viene realizando el gobierno peruano por medio del Instituto de Estadística e Informática, no se ha podido consolidar esta información por lo que nuestro universo se encuentra conformado por personas dedicadas al desarrollo de software que se encuentren trabajando actualmente y que empleen las técnicas de desarrollo en su totalidad; asimismo se tomará en cuenta las estadísticas proporcionadas por la Cámara de Comercio e Industria para detectar a aquellas empresas dedicadas a esta labor.

1.13.2 Muestra

Al no existir estadísticas o métricas sobre información referente a la construcción de software y a todos sus activos; se realiza un muestreo por cuotas. Se visita a las empresas dedicadas a ésta área y se considera a las personas orientadas a estos menesteres, que hayan trabajado y que trabajen con la documentación correcta en la construcción de software. Las cuotas que ingresan a la muestra están referidas a la empresas definidas en (Ruelas, 2013), donde se presentan estadísticas actualizadas de las empresas desarrolladoras de software en la ciudad de Arequipa.

1.14 Técnicas e instrumentos de recolección de información

1.14.1 Técnicas

- Entrevista
- Revisión bibliográfica
- Revisión documental

1.14.2 Instrumentos

- Entrevista personal
- Textos
- Documentos propios de la organización



Capítulo 2: Fundamentos teóricos

2.1 Antecedentes investigativos

Pérez Lavalle Sonia & Orejas Valdés Fernando (Pérez, 2006) mencionan que el lenguaje de modelado UML (Unified Modelling Language) debe ser extendido para lograr representar dominios específicos, por ejemplo Inteligencia Artificial, Sistemas Expertos, Sistemas Distribuidos, entre otros. Esto conlleva aplicar mecanismos permitidos para este fin. En su trabajo de investigación presentan una extensión al metamodelo de UML, por medio de la creación de metaclases y metaasociaciones con la finalidad de lograr representar la arquitectura de componentes genéricos.

Arias Barragán Luís & Fernando García Diego (Arias, 2013) propone, en su artículo, un modelo de gestión del proceso de despacho en sistemas alimentados por generación eléctrica convencional y distribuida. Para ello implementa un sistema automatizado que permite, de manera dinámica, adaptarse a escenarios de cargas variables, aleatoriedad de los parámetros de eficiencia y calidad de energía del sistema, restricciones técnicas, económicas y regulatorias, y la inclusión activa de agentes generadores distribuidos.

Tabares Martha y Pineda Juan (Tabares, 2008) menciona que actualmente en los ambientes de desarrollo de software existe un gran interés en buscar y desarrollar técnicas que puedan integrar los sistemas transaccionales con los flujos de trabajo que soportan los procesos del negocio de las organizaciones. Sin embargo, en la industria del software no es común encontrar técnicas o prácticas que faciliten el desarrollo de los modelos del sistema en función de los procesos del negocio. En este artículo definen un patrón de desarrollo que estandariza la interacción entre diagramas de actividades de UML 2.0, que representan las operaciones de un sistema, y procesos del negocio automatizados bajo tecnologías workflow. La trazabilidad de dicha interacción se mantiene por medio de modelos de trazabilidad que controlan la evolución de las operaciones del negocio y del sistema. Para mostrar la aplicación del

patrón se desarrolla un caso de estudio.

Gacitúa Bustos Ricardo (Gacitúa, 2003) describe la evolución de los métodos de desarrollo de software. El foco está centrado en el desacuerdo en cómo debe crearse el software. El tema es como se considera el desarrollo de software: como un proceso de ingeniería o un proceso centrado en las personas. Se presenta el lenguaje de modelamiento unificado (UML) como una notación estándar del desarrollo de software. Actualmente es considerado como base para una metodología monumental (que incluye muchas reglas y prácticas) – RUP. Se menciona la reacción a las metodologías monumentales: los métodos ágiles. La cual es característica de un estado inmaduro del desarrollo de software como una disciplina. No solo hay desacuerdo en terminologías, enfoques y detalles de diferentes métodos, sino incluso en un esquema de clasificación común. La siguiente estructura está basada en la evolución de los principales conceptos y las distinciones claves que reflejan los cambios de paradigmas en la filosofía de métodos.

Zapata Carlos & Gonzáles Guillermo (Zapata, 2008) indican que en el ciclo de vida del software, durante las fases de definición y análisis, se realiza una especificación de los requisitos. Para ello, es necesario llevar a cabo un proceso de captura de las necesidades y expectativas de los interesados, que se traduce posteriormente en un conjunto de modelos que representan tanto el problema como su solución. Por lo general, la mayoría de los modelos se expresan en el lenguaje de modelado unificado -UML-, que define un conjunto de artefactos que permiten especificar los requisitos del software, los cuales deberían guardar consistencia, cuando se trate del mismo modelo. Sin embargo, la consistencia entre diferentes artefactos no se encuentra definida en la especificación de UML. En el artículo se propone un método para verificar la consistencia entre el diagrama de clases y el diagrama de casos de uso de UML de una manera formal. Dicho proceso se lleva a cabo evaluando una serie de reglas definidas en el lenguaje de restricciones de objetos -OCL- que se deben cumplir para garantizar que la información brindada por dichos modelos sea consistente.

Moreno Espino Maily & Rosete Suárez Alejandro (Moreno, 2006) hace hincapié que los estándares de ingeniería de software, RUP como proceso de desarrollo y UML como lenguaje de modelado, enfrentan retos debido a los paradigmas distribuidos: Servicios Web y agentes. Los agentes, paradigma prometedor para la ingeniería de software, tienen un gran reto, la falta de estándares metodológicos para su desarrollo. En este trabajo se realiza un estudio comparativo de nueve metodologías de agentes, se analizan roles de personas en la ingeniería de software orientada a agentes y se proponen algunos roles no presentes en RUP y recomendaciones para otros sí presentes que parecen necesarios con este nuevo paradigma.

Zapata Carlos & Giraldo Gloria (Zapata, 2009) mencionan que UML, SysML y WebML son lenguajes de modelado gráfico (LMG) similares que no se pueden interpretar conjuntamente, pues tienen diferencias en tipos de modelos y diagramas. En esta investigación se encuentran técnicas que estudian las características de algunos LMG, pero se aplican sobre lenguajes particulares, sin considerar sus características comunes. En el presente trabajo de investigación se propone el diseño e implementación de una ontología que resuma los principales conceptos y relaciones de los LMG.

Rodríguez Rosseline y Goncalves Marlene (Rodríguez, 2009) mencionan que modelar dominios de aplicación que involucren requisitos con términos vagos puede ser un proceso complejo, debido a que las herramientas de modelado disponibles en la actualidad no están concebidas para ello. Sin embargo, muchos requisitos de información en sistemas y aplicaciones actuales podrían involucrar términos vagos del lenguaje natural que expresan las preferencias del usuario. Algunos esfuerzos previos se han hecho en el área de consultas a bases de datos incorporando la lógica difusa para la expresión de tales preferencias. En este trabajo, se define un perfil bajo UML que permite el modelado visual de requisitos de información basados en términos lingüísticos, ejemplo de ellos son los llamados requerimientos fuzzy o

difusos. El perfil propuesto se basa en estereotipos y lógica difusa extendida extendido OCL (Object Constraint Language). Dicho perfil posee una semántica formal que permite eliminar ambigüedades, que puede ser representado gráficamente. Se ejemplifica la aplicación del perfil a través de un caso de estudio de la vida real que muestra la sencillez y potencialidad de su uso.

2.2 Marco conceptual

2.2.1 UML

El UML es un lenguaje de modelado cuyo vocabulario y sintaxis están ideados para la representación conceptual y física de un sistema. Sus modelos son precisos, no ambiguos, completos y pueden ser trasladados directamente a una gran variedad de lenguajes de programación, pero también a tablas de bases de datos relacionales y orientados a objetos. Es posible generar código a partir de un modelo UML y también construirse un modelo a partir de la implementación.

2.2.1.1 Bloques básicos de construcción de UML

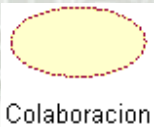
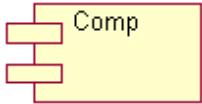
Los bloques básicos de construcción de UML son tres:

Los *elementos* son abstracciones que actúan como unidades básicas de construcción. Hay cuatro tipos, los *estructurales*, los de *comportamiento*, los de *agrupación* y los de *notación*. En cuanto a los elementos estructurales son las partes estáticas de los modelos y representan aspectos conceptuales o materiales. Los elementos de comportamiento son las partes dinámicas de los modelos y representan comportamientos en el tiempo y en el espacio. Los elementos de agrupación son las partes organizativas de UML, establecen las divisiones en que se puede fraccionar un modelo. Sólo hay un elemento de agrupación, el paquete, que se emplea para organizar otros elementos en grupos. Los elementos de notación son las partes explicativas de UML, comentarios que pueden describir textualmente

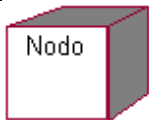
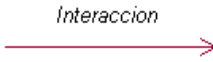
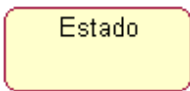
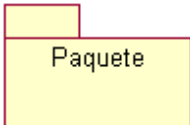
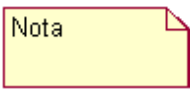
cualquier aspecto de un modelo. Sólo hay un elemento de notación principal, la nota. La figura N° xx muestra los elementos de UML.

Tabla 2

Elementos de construcción en UML

E L E M E N T O S E S T R U C T U R A L E S	Clase		Describe un conjunto de objetos que comparten los mismos atributos, métodos, relaciones y semántica. Las clases implementan una o más interfaces.
	Clase activa		Se trata de una clase, en la que existen procesos o hilos de ejecución concurrentes con otros elementos. Las líneas del contorno son más gruesas que en la clase “normal”
	Interfaz		Agrupación de métodos u operaciones que especifican un servicio de una clase o componente, describiendo su comportamiento, completo o parcial, externamente visible. UML permite emplear un círculo para representar las interfaces, aunque lo más normal es emplear la clase con el nombre en cursiva.
	Colaboración		Define una interacción entre elementos que cooperan para proporcionar un comportamiento mayor que la suma de los comportamientos de sus elementos.
	Caso de uso		Describe un conjunto de secuencias de acciones que un sistema ejecuta, para producir un resultado observable de interés. Se emplea para estructurar los aspectos de comportamiento de un modelo.
	Componente		Parte física y por tanto reemplazable de un modelo, que agrupa un conjunto de interfaces, archivos de código fuente, clases, colaboraciones y proporciona la implementación de dichos elementos.

Fuente: [BOOCH, 2005]

	Nodo		Elemento físico que existe en tiempo de ejecución y representa un recurso computacional con capacidad de procesar.
Elementos de comportamiento	Interacción		Comprende un conjunto de mensajes que se intercambian entre un conjunto de objetos, para cumplir un objetivo específico.
	Máquinas de estados		Especifica la secuencia de estados por los que pasa un objeto o una interacción, en respuesta a eventos.
Elementos de agrupación	Paquete		Se emplea para organizar otros elementos en grupos.
Elementos de notación	Nota		Partes explicativa de UML, que puede describir textualmente cualquier aspecto del modelo

Las relaciones son abstracciones que actúan como unión entre los distintos *elementos*. Hay cuatro tipos, la *dependencia*, la *asociación*, la *generalización* y la *realización*.

Tabla 3

Elementos de relación en UML

Dependencia		Es una relación entre dos elementos, tal que un cambio en uno puede afectar al otro.
Asociación		Es una relación estructural que resume un conjunto de enlaces que son conexiones entre objetos.
Generalización		Es una relación en la que el elemento generalizado puede ser substituido por cualquiera de los elementos hijos, ya que comparten su estructura y comportamiento.
Realización		Es una relación que implica que la parte realizante cumple con una serie de especificaciones propuestas por la clase realizada (interfaces).

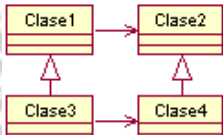
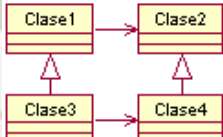
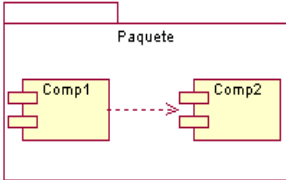
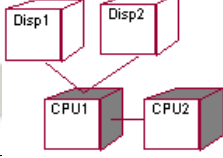
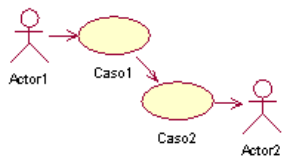
Fuente: [BOOCH, 2005]

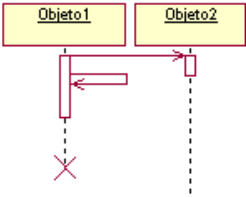
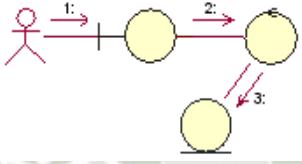
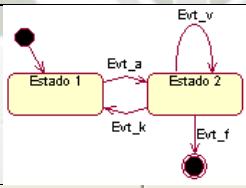
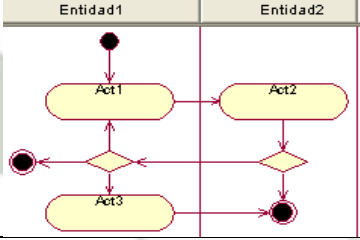
Los diagramas son la disposición de un conjunto de elementos,

que representan el sistema modelado desde diferentes perspectivas. UML tiene nueve diagramas fundamentales, agrupados en dos grandes grupos, uno para modelar la estructura estática del sistema y otro para modelar el comportamiento dinámico. Los **diagramas estáticos son:** el de *clases*, de *objetos*, de *componentes* y de *despliegue*. Los **diagramas de comportamiento son:** el de *Casos de Uso*, de *secuencia*, de *colaboración*, de *estados* y de *actividades*.

Tabla 4

Diagramas de UML

M O D E L O E S T R U C T U R A	Clases		Muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones, cubriendo la vista de diseño estática del sistema.
	Objetos		Análogo al diagrama de clases, muestra un conjunto de objetos y sus relaciones, pero a modo de vista instantánea de instancias de una clase en el tiempo.
	Componentes		Muestra la organización y dependencias de un conjunto de componentes. Cubren la vista de implementación estática de un sistema. Un componente es un módulo de código, de modo que los diagramas de componentes son los análogos físicos a los diagramas de clases.
	Despliegue		Muestra la configuración del hardware del sistema, los nodos de proceso y los componentes empleados por éstos. Cubren la vista de despliegue estática de una arquitectura.
M O D E L	Casos de Uso		Muestra un conjunto de casos de uso, los actores implicados y sus relaciones. Son diagramas fundamentales en el modelado y organización del sistema.

A N C O M P O R T A M I E N T O	Secuencia		Son diagramas de interacción, muestran un conjunto de objetos y sus relaciones, así como los mensajes que se intercambian entre ellos. Cubren la vista dinámica del sistema. El diagrama de secuencia resalta la ordenación temporal de los mensajes, mientras que el de colaboración resalta la organización estructural de los objetos, ambos siendo equivalentes o isomorfos. En el diagrama de colaboración de la figura de la izquierda, se puede ver que los elementos gráficos no son cajas rectangulares, como cabría esperar, y en su lugar encontramos sus versiones adornadas. Estas versiones tienen como finalidad evidenciar un rol específico del objeto siendo modelado. En la figura encontramos de izquierda a derecha y de arriba abajo un Actor, una Interfaz, un Control (modela un comportamiento) y una Instancia (modela un objeto de dato).
	Colaboración		
	Estados		Muestra una máquina de estados, con sus estados, transiciones, eventos y actividades. Cubren la vista dinámica de un sistema. Modelan comportamientos reactivos en base a eventos.
	Actividades		Tipo especial de diagrama de estados que muestra el flujo de actividades dentro de un sistema.

Fuente: [BOOCH, 2005]

2.2.1.2 Diagrama de clases y diagrama de objetos

Los diagramas de clases muestran un resumen del sistema en términos de sus clases y las relaciones entre ellas. Son diagramas estáticos que muestran **qué** es lo que interactúa, pero no cómo interactúa o qué pasa cuando ocurre la interacción. Por ejemplo, el siguiente diagrama modela los pedidos de un cliente a una tienda de venta por catálogo. La clase principal es “Pedido”, asociada a un cliente, una forma de pago y un conjunto de artículos.

La clase “Pago” es abstracta, en UML los nombres de clases abstractas se representan en *Itálica*. Las clases abstractas actúan a modo de **interfaz**, proporcionando únicamente un listado de métodos a ser “realizados” por las clases que las implementan o realizan. “Pago” es una superclase especializada, y a la vez realizada, por sus formas más comunes “Credito” y “Efectivo”. Un “Pedido” tiene una única forma de pago, expresada por su multiplicidad, 1, mientras que una forma de pago puede estar presente en uno o más pedidos, como sugiere su multiplicidad, 1..*.

En cuanto a las asociaciones, observamos que algunas vienen representadas como una *flecha navegable*, cuya orientación expresa el sentido en que se consultan los datos. Las asociaciones sin flecha son bi-direccionales. Las agregaciones expresan “conjunto de”; la relación entre “Pedido” y “Articulo” es de conjunto. Un pedido es una agregación de una o más líneas de pedido, donde cada una hace alusión a un artículo concreto, así mismo una línea de pedido puede estar presente en varios pedidos y un artículo puede no haber sido solicitado nunca.

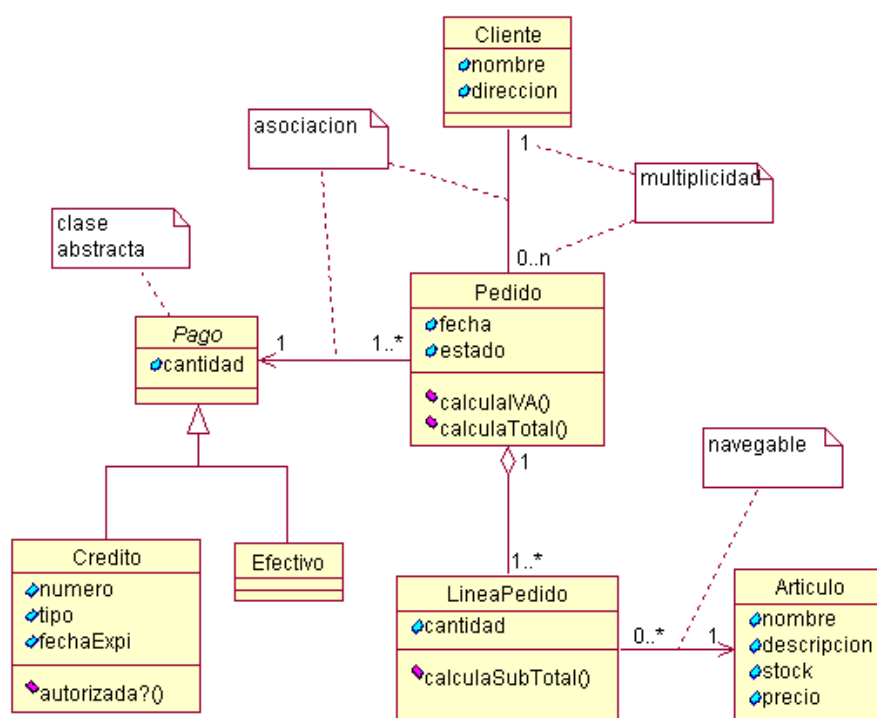


Figura 1. Diagrama de clase

Fuente: [BOOCH, 2005]

En cuanto a la multiplicidad, la siguiente tabla resume las más comunes. Hay que tener en cuenta que la multiplicidad se expresa “en el lado opuesto” de la relación y es el número de posibles instancias de una clase asociadas con una única instancia de la clase en el otro extremo.

Tabla 5

Multiplicidad en diagramas de clases

Multiplicidad	Significado
1	Una única instancia
N / *	N instancias
0..N / 0..*	Entre ninguna y N instancias
1..N / 1..*	Entre una y N instancias
0..1	Ninguna o una instancia
N..M	Entre N y M instancias

Fuente: [BOOCH, 2005]

El siguiente diagrama muestra una dependencia existente entre las clases “Pedido” y “Fecha”. Cualquier cambio en la clase dependida, “Fecha”, afectará la clase dependiente, “Pedida”.

Así mismo se puede observar que las clases vienen representadas por cajas en las que hay tres separaciones, o compartimentos. El **primero** se emplea siempre para indicar el *nombre* de la clase, el **segundo** para mostrar los *atributos* y el **tercero** para los *métodos*. Tanto los atributos como los métodos vienen precedidos por un símbolo de acceso, que normalmente suele ser un “+” para el acceso *público*, un “-” para el acceso *privado*, (sólo por otros métodos de la clase) y un “#” para el acceso *protegido* (sólo por clases hija), aunque la herramienta empleada en la elaboración del tutorial traduce estos elementos en iconos.

Los atributos tienen un tipo que puede mostrarse a continuación de su nombre separado por “:”. De igual manera, los métodos pueden devolver un elemento de un tipo determinado y recibir parámetros, expresados entre paréntesis mediante el nombre del parámetro y el tipo, separados por “:”. Para el caso de múltiples parámetros, se separan por comas ($p1:t1, p2:t2 \dots pn:tn$). Los parámetros que tienen un valor por defecto se expresan mediante un “=” y el valor, a continuación del tipo ($p1:t1=v1$) y si un parámetro en la posición “i” de la lista de parámetros tiene valor por defecto, todos los parámetros que le sigan, es decir que ocupen posiciones sucesivas a “i” en la lista, deberán tener también un valor por defecto.

Los atributos y métodos estáticos (de clase) se representan mediante un subrayado (en el caso de los métodos se puede emplear el estereotipo <<static>>, los estereotipos se ven más adelante).

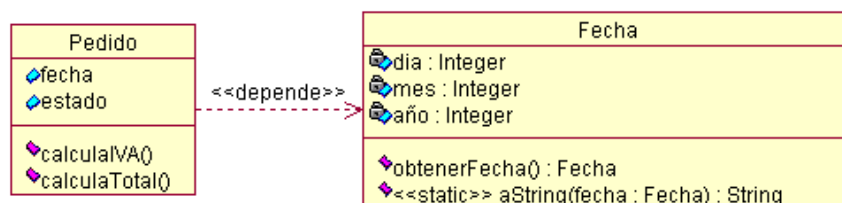


Figura 2. Relación de dependencia en diagramas de clase

Fuente: [BOOCH, 2005]

El siguiente diagrama muestra una auto-relación de agregación. Un “Departamento” puede estar compuesto a su vez por más sub-departamentos, o ninguno, con la restricción de que el mínimo número de personas en los sub-departamentos debe ser dos. Las *restricciones* son condiciones que deben ser cumplidas siempre, se expresan *entre llaves* “{condición}”.

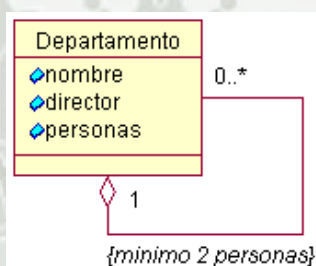


Figura 3. Auto-agregación

Fuente: [BOOCH, 2005]

Los diagramas de objetos son análogos a los de clases, con la particularidad de que en lugar de encontrar clases, encontramos instancias de éstas. Son útiles para explicar partes pequeñas del modelo en las que hay relaciones complejas.

2.2.1.3 Diagrama de componentes y diagrama de despliegue

Los componentes son módulos de código, así que los

diagramas de componentes vienen a ser los análogos físicos a los diagramas de clases. Muestran como está organizado un conjunto de componentes y las dependencias que existen entre ellos.

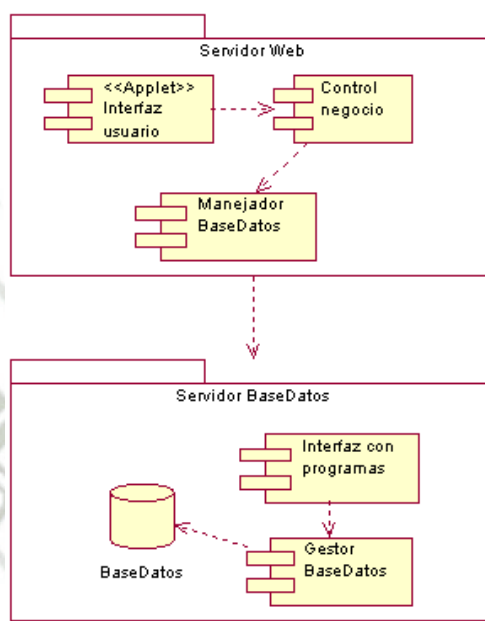


Figura 4. Diagrama de componentes

Fuente: [BOOCH, 2005]

Los diagramas de despliegue sirven para modelar la configuración hardware del sistema, mostrando qué nodos lo componen.

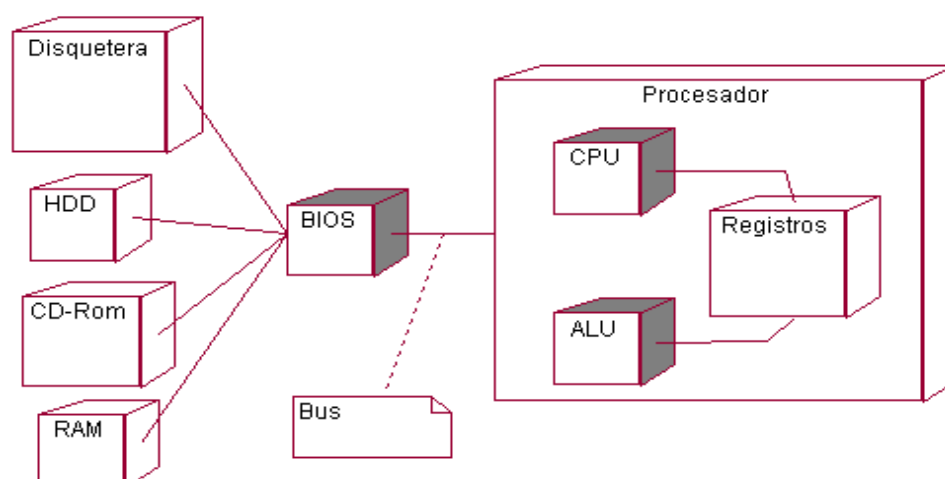


Figura 5. Diagrama de despliegue

Fuente: [BOOCH, 2005]

2.2.1.4 Diagrama de casos de uso

Los diagramas de Casos de Uso describen lo que hace un sistema desde el punto de vista de un observador externo, enfatizando el **qué** más que el cómo. Plantean escenarios, es decir, lo que pasa cuando alguien interactúa con el sistema, proporcionando un resumen para una tarea u objetivo. El siguiente Caso de Uso describe como Carlos va a desayunar (este es su objetivo), para lo que se plantea el escenario de preparar su café y el pan tostado

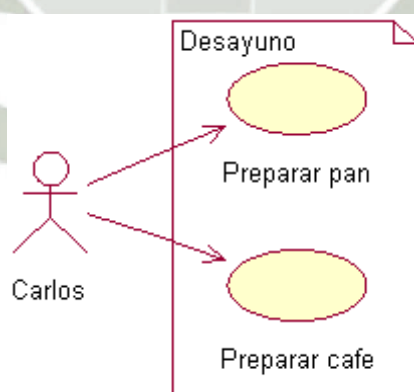


Figura 6. Diagrama de casos de uso nivel 1

Fuente: [BOOCH, 2005]

En los Casos de Uso, los *Actores* son papeles que determinadas personas u objetos desempeñan. Se representan mediante un “hombre de palitos”, de modo que en el ejemplo, Carlos es un Actor. Los Casos de Uso se representan por medio de *óvalos* y las líneas que unen Actores con Casos de Uso representan una asociación de comunicación.

Por su puesto, un Caso de Uso puede ser descrito en mayor profundidad. Por ejemplo si tomamos por separado “Preparar pan” y “Preparar café”, podemos bajar un nivel de descripción y llegar a los siguientes Casos de Uso.

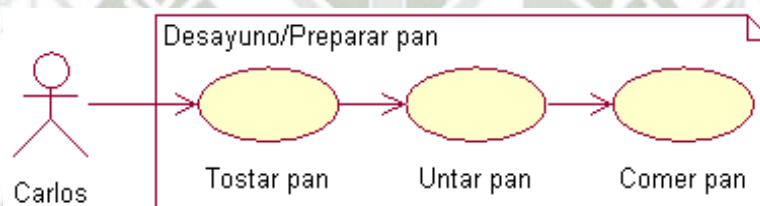


Figura 7. Diagrama de casos de uso nivel 2 A

Fuente: [BOOCH, 2005]

“Carlos tuesta el pan en la tostadora, después lo unta con mantequilla y mermelada de fresa y se lo come, posiblemente mojándolo en un café.”

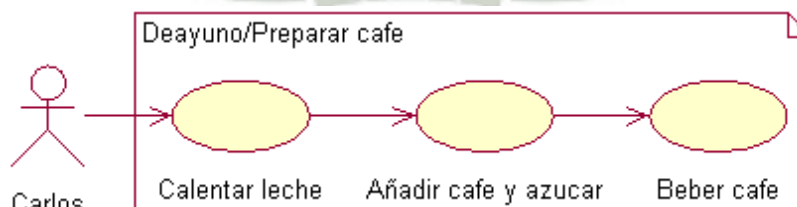


Figura 8. Diagrama de casos de uso nivel 2 B

Fuente: [BOOCH, 2005]

“Carlos calienta leche, añade café y azúcar al gusto y se lo bebe.”

Los Casos de Uso suelen venir delimitados por fronteras o límites, que definen una separación entre lo que es realmente la funcionalidad del sistema y los actores que la usan o colaboran en su desempeño. En las figuras, esta separación viene representada por medio de la caja que encapsula los óvalos.

Los Casos de Uso son acompañados por una explicación textual que clarifica las posibles cadencias del lenguaje meramente gráfico. De esta manera, combinando Casos de Uso y explicación textual, se puede obtener escenarios no ambiguos, que resultan ideales en la captura de requisitos de usuario, dada su sencillez de comprensión incluso por quien no está familiarizado con UML. Los Casos de Uso se emplean también en la preparación de escenarios de pruebas con que verificar el software una vez ha sido construido.

El siguiente Caso de Uso es equivalente al primero, “Desayuno”, sólo que en él se ha condensado la máxima cantidad posible de información. En él se muestra un nuevo elemento que hasta ahora no se había mostrado, el “**estereotipo**”, que viene entre sendos símbolos angulados “<<” y “>>” y concreta un paso más allá el tipo de relación existente entre dos Casos de Uso. Encontramos dos estereotipos <<include>> y <<extend>>. El primero indica que el Caso de Uso “Tostar pan” requiere de “Usar tostadora” para poder ser llevado a cabo. Esta es una forma muy adecuada de sacar factor común entre Casos de Uso, o incluso de fraccionar Casos de Uso muy grandes. El segundo indica que el Caso de Uso “Untar pan” es una variación de “Untar”. Observamos también que “Comer pan” y “Beber cafe” son una generalización de “Alimentarse”.

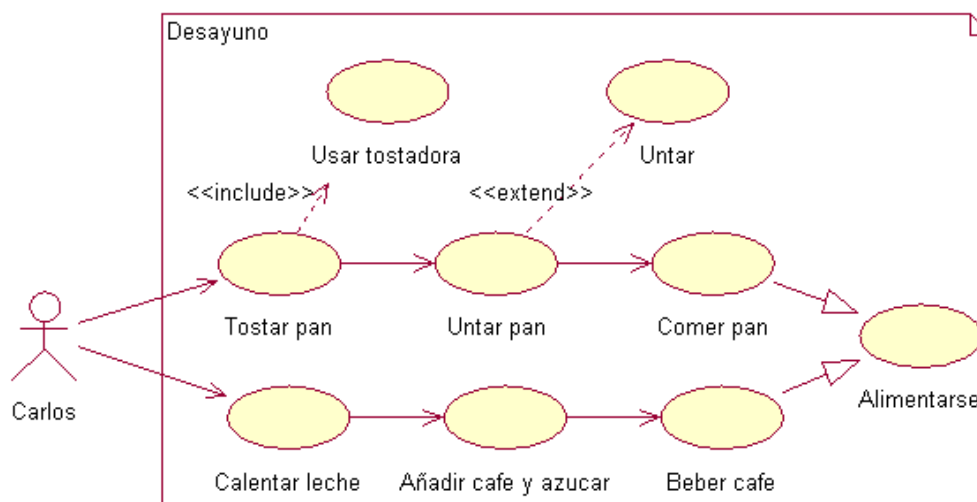


Figura 9. Diagrama de casos de uso nivel 1 detallado

Fuente: [BOOCH, 2005]

“Carlos va a desayunar. Para ello debe hacer dos actividades distintas, pero relacionadas. La primera consiste en tostar pan, para lo cual necesita emplear una tostadora. Una vez tostado el pan, lo unta de mantequilla y mermelada de fresa (untar pan no es muy distinto de untar otro tipo de alimentos). La segunda consiste en preparar el café, para lo cual necesita calentar leche y añadir café y azucar. Terminadas ambas actividades, Carlos puede proceder a alimentarse, comiendo el pan tostado y bebiendo el café. El orden en que realice las actividades da igual y también da igual si se realizan a la vez.”

2.2.1.5 Diagrama de secuencia y diagrama de colaboración

Los diagramas de secuencia describen como los objetos del sistema colaboran. Se trata de un diagrama de interacción que detalla como las operaciones se llevan a cabo, qué mensajes son enviados y cuando, organizado todo en torno al tiempo. El tiempo avanza “hacia abajo” en el diagrama. Los objetos involucrados en la operación se listan de izquierda a derecha de acuerdo a su orden de participación dentro de la secuencia de mensajes.

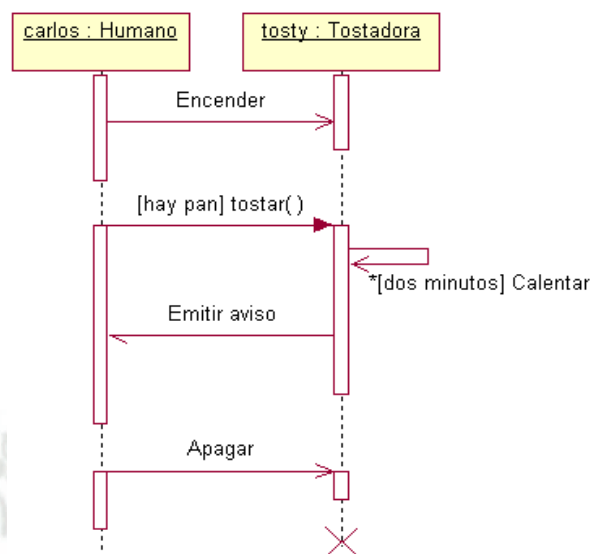


Figura 10. Diagrama de secuencia

Fuente: [BOOCH, 2005]

Las líneas verticales o “líneas de la vida” representan el tiempo de vida del objeto. La vida del objeto “carlos” no termina en este diagrama, sin embargo la del objeto “tosty” sí y esto viene representado mediante el aspa al final de su línea de la vida.

Los rectángulos verticales son barras de activación y representan la duración de la ejecución del mensaje. El mensaje “Encender”, posiblemente implementado mediante la introducción del enchufe en una toma de pared, tiene una duración escasa y similar a la de “Apagar”. No ocurre lo mismo con la llamada al método “tostar()”, que dura desde la pulsación del botón de tostar hasta que el pan es retirado de la bandeja y además interviene la emisión de un aviso cuando el pan está lo suficientemente caliente, a fin de evitar que se queme.

Como se puede observar, la acción tostar viene condicionada por la presencia de pan en la bandeja de la tostadora. En UML los corchetes “[]” expresan condición y si están precedidos de un asterisco

indican interacción mientras se cumpla la condición.

Los mensajes que son intercambiados entre los objetos de un diagrama de secuencia pueden ser *síncronos* o *asíncronos*. Los mensajes asíncronos son aquellos tal que el emisor puede enviar nuevos mensajes mientras el original está siendo procesado. El mensaje asíncrono ocurre en el tiempo de manera independiente a otros mensajes. Los mensajes síncronos son todo lo contrario, el emisor debe esperar a que termine el tiempo de proceso del mensaje antes de que pueda emitir nuevos mensajes. UML emplea los siguientes convenios para representar el tipo de mensaje.

Tabla 6
Variables y técnicas para la investigación

Símbolo	Significado
	Mensaje simple que puede ser síncrono o asíncrono.
	Mensaje simple de vuelta (opcional).
	Mensaje síncrono.
	Mensaje asíncrono.

Fuente: [BOOCH, 2005]

Los diagramas de colaboración son otro tipo de diagramas de interacción, que contiene la misma información que los de secuencia, sólo que se centran en las responsabilidades de cada objeto, en lugar de en el tiempo en que los mensajes son enviados. Cada mensaje de un diagrama de colaboración tiene un número de secuencia. El primer nivel de la secuencia es 1, y los mensajes que son enviados durante la misma llamada a un método se numeran 1.1, 1.2 y así sucesivamente para tantos niveles como sea necesario.



Figura 11. Diagrama de colaboración

Fuente: [BOOCH, 2005]

2.2.1.6 Diagrama de estados y diagrama de actividades

Los diagramas de estados muestran los posibles estados en que puede encontrarse un objeto y las transiciones que pueden causar un cambio de estado. El estado de un objeto depende de la actividad que esté llevando a cabo o de alguna condición.

Las transiciones son las líneas que unen los diferentes estados. En ellas se representa la condición que provoca el cambio, seguida de la acción oportuna separada por “/”. En un estado en que el objeto está pendiente de algún tipo de validación que dependa de un proceso en curso, no es necesario evento externo alguno para que se produzca la transición, ya que ésta ocurrirá cuando termine el proceso, en función

del resultado de éste. En estos casos es conveniente, por claridad, incluir la condición que de la que depende la transición (entre corchetes).

Los estados inicial, a partir del que se “entra” en la máquina de estados, y final, que indica que la máquina de estados termina, no tienen otro significado adicional, son elementos ornamentales y se representan mediante un círculo negro y un círculo negro resaltado respectivamente.

Los estados de un diagrama de estados pueden anidarse, de forma que los estados relacionados pueden ser agrupados en un estado compuesto. Esto puede ser necesario cuando una actividad involucra sub-actividades asíncronas o concurrentes.

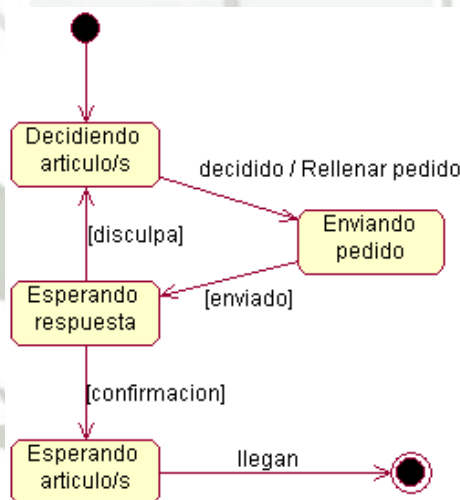


Figura 12. Máquina de estados, estados simples

Fuente: [BOOCH, 2005]

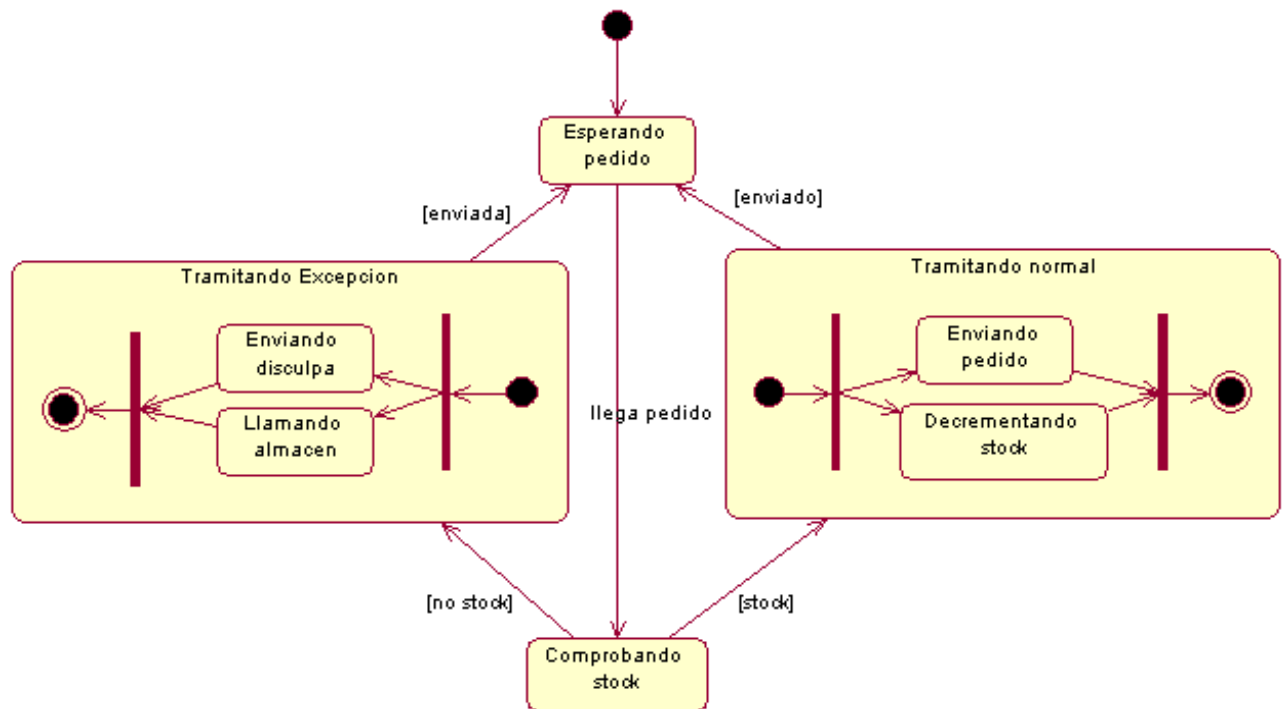


Figura 13. Máquina de estados, estados compuestos

Fuente: [BOOCH, 2005]

Los diagramas de actividades son básicamente diagramas de flujo adornados, que guardan mucha similitud con los diagramas de estados. Mientras que los diagramas de estados centran su atención en el proceso que está llevando a cabo un objeto, los diagramas de actividades muestran como las actividades fluyen y las dependencias entre ellas.

Los diagramas de actividades pueden dividirse en “calles” que determinan qué objeto es responsable de qué actividad. Las actividades vienen unidas por transiciones, que pueden separarse en ramas en función del resultado de una condición expresada entre corchetes. Cada rama muestra la condición que debe ser satisfecha para que el flujo opte por ese camino. Igualmente, las transiciones se pueden bifurcarse en dos o más actividades paralelas.

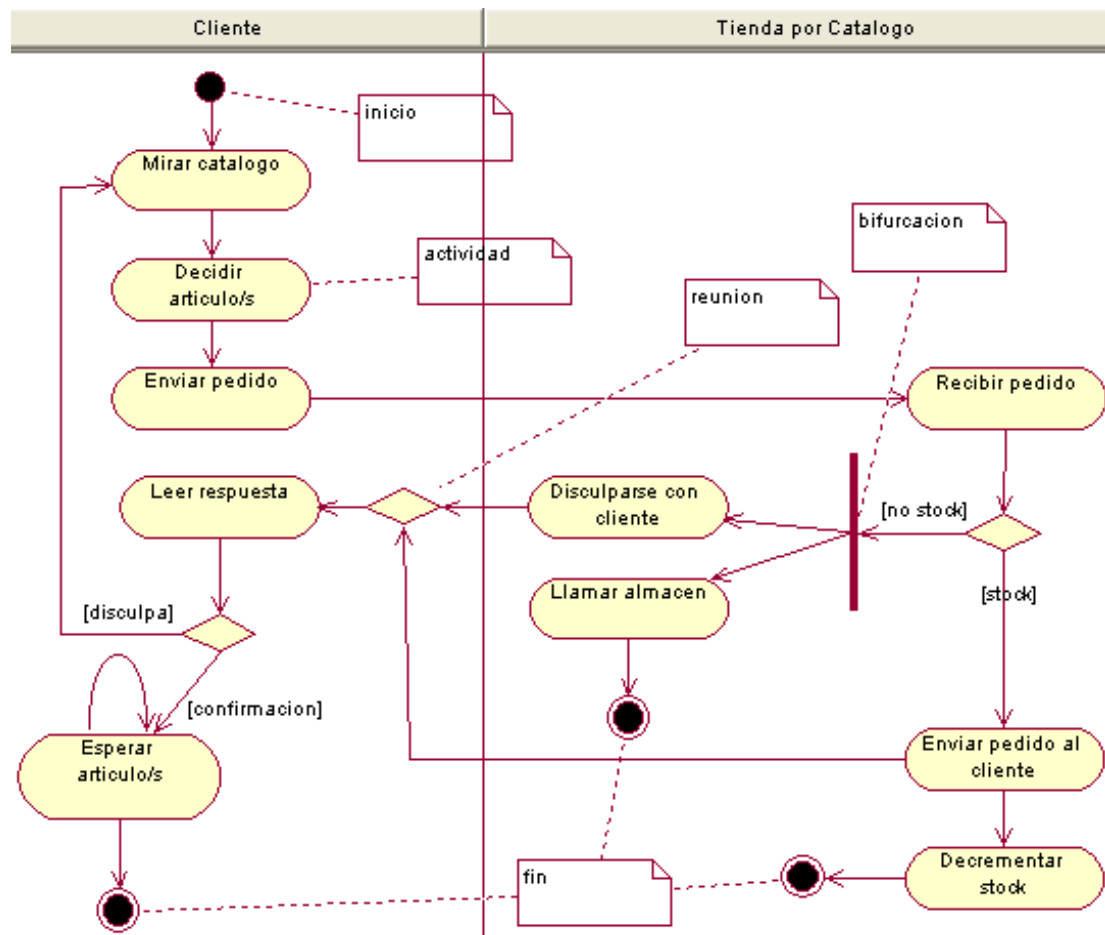


Figura 14. Diagrama de actividades

Fuente: [BOOCH, 2005]

2.2.2 Desarrollo de software

El proceso de desarrollo de software tiene como función principal la producción eficaz y eficiente de un producto software que respete los requerimientos del cliente. Este proceso es intelectual, afectado por la creatividad y juicio de las personas que analizan el modelo de negocio. Aunque este tipo de proyectos es idéntico en muchos aspectos a cualquier otro proyecto de ingeniería, en el desarrollo de software nos encontramos con varios desafíos, relacionados esencialmente a la naturaleza del producto. Algunas particularidades asociadas al desarrollo de software y que influyen en su proceso de construcción.

Un producto software en sí es complejo, es prácticamente inviable conseguir un 100% de confiabilidad de un programa por pequeño que sea. Existe una inmensa combinación de factores que impiden una verificación exhaustiva de las todas posibles situaciones de ejecución que se puedan presentar.

Un producto de software es intangible y por lo general abstracto, esto dificulta la definición del producto y de los requisitos, sobre todo cuando no se tiene precedentes en productos software similar. Esto hace que los requisitos sean difíciles de consolidar tempranamente. Así, los cambios en los requisitos son inevitables, no sólo después de entregado en producto sino también durante el proceso de desarrollo.

Además, siempre puede señalarse la inmadurez de la ingeniería del software como disciplina, justificada por su corta vida comparada con otras disciplinas de la ingeniería.

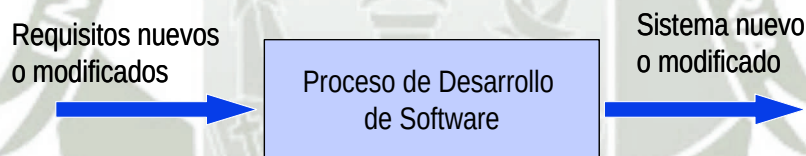


Figura 15. Proceso de desarrollo de software

Fuente: Elaboración propia

El proceso de desarrollo de software no es único. No existe un proceso de software universal que sea efectivo para todos los contextos de proyectos de desarrollo. Debido a esta diversidad, es difícil automatizar todo un proceso de desarrollo de software.

A pesar de la variedad de propuestas de proceso de software, existe un conjunto de actividades fundamentales que se encuentran presentes en todos ellos [SOMMERVILLE, 2005]:

Especificación de software: Se debe definir la funcionalidad y restricciones operacionales que debe cumplir el software.

Diseño e Implementación: Se diseña y construye el software de acuerdo a la especificación.

Validación: El software debe validarse, para asegurar que cumpla con lo que quiere el cliente.

Evolución: El software debe evolucionar, para adaptarse a las necesidades del cliente.

Además de estas actividades fundamentales, las actividades protectoras que se aplican a lo largo de todo el proceso del software son:

- Seguimiento y control de proyecto de software.
- Revisiones técnicas formales.
- Garantía de calidad del software.
- Gestión de configuración del software.
- Preparación y producción de documentos.
- Gestión de reutilización.
- Mediciones.

Gestión de riesgos.

En [PRESSMAN, 2014] caracteriza un proceso de desarrollo de software como se muestra en la Figura 2.16. Los elementos involucrados se describen a continuación:

- **Un marco común del proceso**, definiendo un pequeño número de actividades del marco de trabajo que son aplicables a todos los proyectos de software, con independencia del tamaño o complejidad.
- **Un conjunto de tareas**, cada uno es una colección de tareas de ingeniería del software, hitos de proyectos, entregas y productos de trabajo del software, y puntos de garantía de calidad, que permiten que las actividades del marco de trabajo se adapten a las características del proyecto de software y los requisitos del equipo del proyecto.
- **Las actividades de protección**, tales como garantía de calidad del software, gestión de configuración del software y medición, abarcan el modelo del proceso. Las actividades de protección son independientes de cualquier actividad del marco de trabajo y aparecen durante todo el proceso.

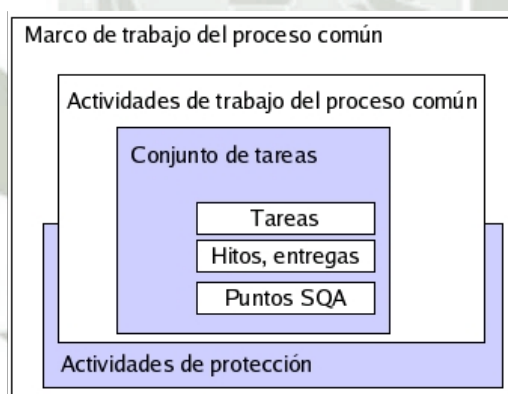


Figura 16. Elementos del proceso del software

Fuente: [PRESSMAN, 2014]

Otra perspectiva utilizada para determinar los elementos del proceso de desarrollo de software es establecer las relaciones entre elementos que permitan responder **Quién** debe hacer **Qué**, **Cuándo** y **Cómo** debe hacerlo.

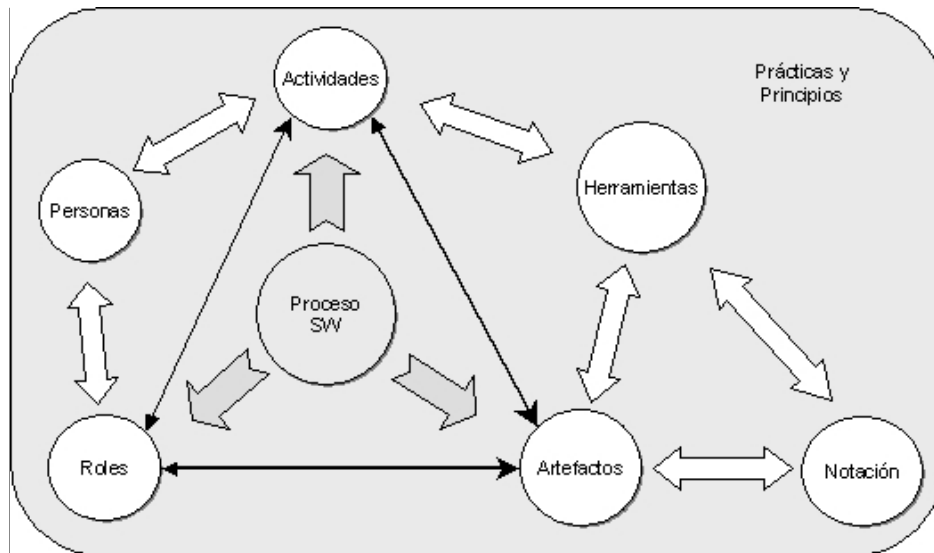


Figura 17. Relación entre elementos del proceso del software

Fuente: [PRESSMAN, 2014]

En la Figura 17 se muestran los elementos de un proceso de desarrollo de software y sus relaciones. Así las interrogantes se responden de la siguiente forma:

Quién: Las Personas participantes en el proyecto de desarrollo desempeñando uno o más Roles específicos.

Qué: Un Artefacto es producido por un Rol en una de sus Actividades. Los Artefactos se especifican utilizando Notaciones específicas. Las Herramientas apoyan la elaboración de Artefactos soportando ciertas Notaciones.

Cómo y Cuándo: Las Actividades son una serie de pasos que lleva a cabo un Rol durante el proceso de desarrollo. El avance del proyecto está controlado

mediante hitos que establecen un determinado estado de terminación de ciertos Artefactos.

La composición y sincronía de las actividades está basada en un conjunto de Principios y Prácticas. Las Prácticas y Principios enfatizan ciertas actividades y/o la forma como deben realizarse, por ejemplo: desarrollar iterativamente, gestionar requisitos, desarrollo basado en componentes, modelar visualmente, verificar continuamente la calidad, gestionar los cambios, entre otros.



Capítulo 3: Marco metodológico

3.1 Introducción

El lenguaje de modelado unificado (UML) es un lenguaje para especificar, construir, visualizar y documentar los artefactos de un sistema de software orientado a objetos. Un artefacto es una información que es utilizada o producida mediante un proceso de desarrollo de software. UML se quiere convertir en un lenguaje estándar con el que sea posible modelar todos los componentes del proceso de desarrollo de aplicaciones. Sin embargo, hay que tener en cuenta un aspecto importante del modelo: no pretende definir un modelo estándar de desarrollo, sino únicamente un lenguaje de modelado. Otros métodos de modelaje como OMT (Object Modeling Technique) o Booch sí definen procesos concretos (Booch, 2005).

En UML los procesos de desarrollo son diferentes según los distintos dominios de trabajo; no puede ser el mismo el proceso para crear una aplicación en tiempo real, que el proceso de desarrollo de una aplicación orientada a gestión. Las diferencias son muy marcadas y afectan a todas las fases del proceso. El método del UML recomienda utilizar los procesos que otras metodologías tienen definidos (Booch, 2005).

Si bien es cierto que su popularidad se ha incrementado grandemente, y que es muy utilizado para modelar estos sistemas de información, UML posee un conjunto de deficiencias o desventajas producto de que sus artefactos no logran modelar todos los temas de construcción de software. Por ejemplo, UML no presenta artefactos para modelar sistemas inteligentes, sistemas expertos o sistemas orientados a la inteligencia artificial; también no posee artefactos para modelar sistemas distribuidos o sistemas orientados a la web (Booch, 2005).

Por otro lado, los diagramas propuestos por UML son suficientes para modelar sistemas que sí lo requieran pero no adopta características de calidad. El uso de UML

ha sido aceptado mundialmente por lo que sus características de calidad han sido probadas largamente, Podemos aprovecharnos de esta nomenclatura para insertar en sus artefactos elementos que permitan una trazabilidad adecuadas y así reducir los errores cometidos en la construcción de productos software (Booch, 2005).

La ampliación propuesta presenta el siguiente ciclo de vida basada solo en la fase de diseño de productos de software (figura 18). En la misma se observa que la ingeniería de requisitos presenta un ciclo de vida personalizada ya que la educación, elicitación y especificación de requisitos forma un bloque que se retroalimenta por la forma como se van resolviendo las ambigüedades e inconsistencias de los requisitos (Booch, 2005).

Cuando los requisitos se encuentran bien consistenciados, se puede avanzar con la etapa de diseño en donde se generan una serie de artefactos producto de la buena especificación de los requisitos. Si existiera alguna inconsistencia en la etapa de diseño se debe de revisar la ingeniería de requisitos, y de ser posible corregir o incrementar los requisitos observados (Booch, 2005).

Dentro de la etapa de diseño también se puede notar una retroalimentación que permite corregir defectos de modelamiento siempre y cuando se vean sustentados por requisitos adecuadamente entendidos y consistenciados. Ambas etapas presentan una correlación producto de los posibles defectos que se pudieran encontrar en el modelamiento del sistema de información (Booch, 2005).

Mientras existan inconsistencias en la etapa de ingeniería de requisitos, no se puede avanzar en la etapa de diseño. Una vez completa esta etapa se continúa con la etapa de diseño comprando el modelamiento con los requisitos consistenciados (Booch, 2005).

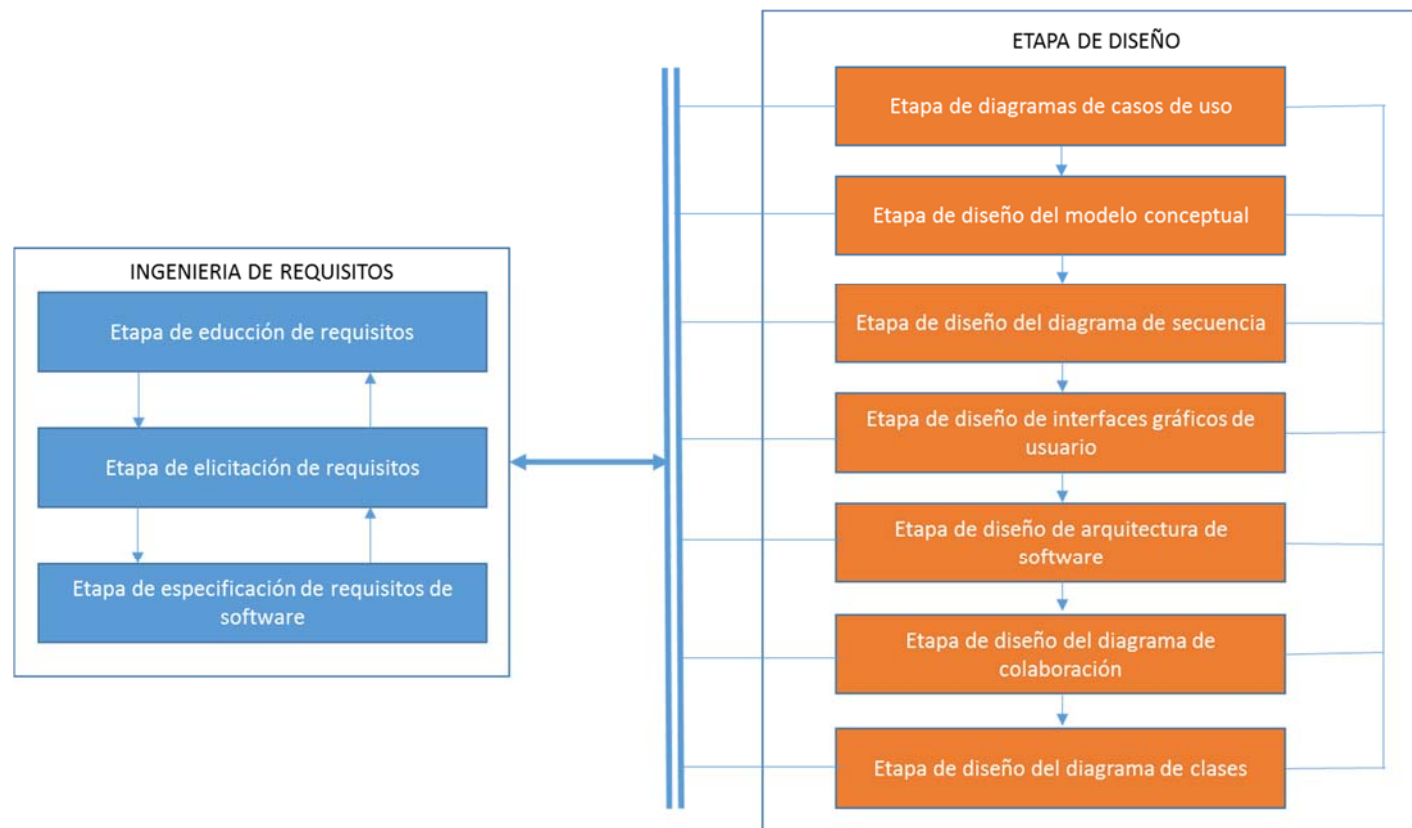


Figura 18. Ciclo de vida de la ampliación UML propuesta

Fuente: Elaboración propia

3.2 Descripción de la propuesta

3.2.1 Etapa de educación de requisitos

En la etapa de educación de requisitos se acopia, encausa y establece toda la información necesaria que tiene como objetivo formalizar las necesidades de información de los interesados. La educación es una técnica, que se obtiene empleando una serie de artefactos, destinada a entregar el insumo principal para el desarrollo de una aplicación de software (Pressman, 2014) (Somerville, 2005).

Este proceso puede ser complicado, especialmente si la organización, donde llevará a cabo esta tarea, es desconocida para el ingeniero de requisitos. Este problema ha permitido que desde hace años se lleven a cabo investigaciones que tienden a hacer este proceso más eficiente y preciso. En la práctica, cuando un ingeniero de requisitos se enfrenta a la tarea de determinar los requisitos de un sistema software, únicamente se emplean entrevistas para capturar esta información.

Este proceso de educación de requisitos, se fracciona en actividades de: búsqueda, descubrimiento de hechos, reuniones de trabajo e integración de la información. El procedimiento para obtener la información es el siguiente:

- Identificación de los orígenes de los requisitos, para lo cual se debe primeramente identificar a aquellas personas que son consideradas importantes en el manejo del mismo.
- Hacer interrogantes, que sean apropiadas, para lograr obtener y comprender necesidades reales de las organizaciones.
- Analizar toda la información reunida para encontrar ambigüedades e inconsistencias en la toma de información.
- Corroborar el entendimiento del dominio con los usuarios principales para corroborar los requisitos.
- Definir la educación de manera correcta.

El resultado de esta tarea es un conjunto de requisitos que deben ser utilizados por los analistas y diseñadores de software. UML no muestra un gráfico que represente esta etapa y menos aún algún indicador que mencione o identifique métrica alguna o medidas de control. Una ampliación UML para ello se muestra en la figura 19.

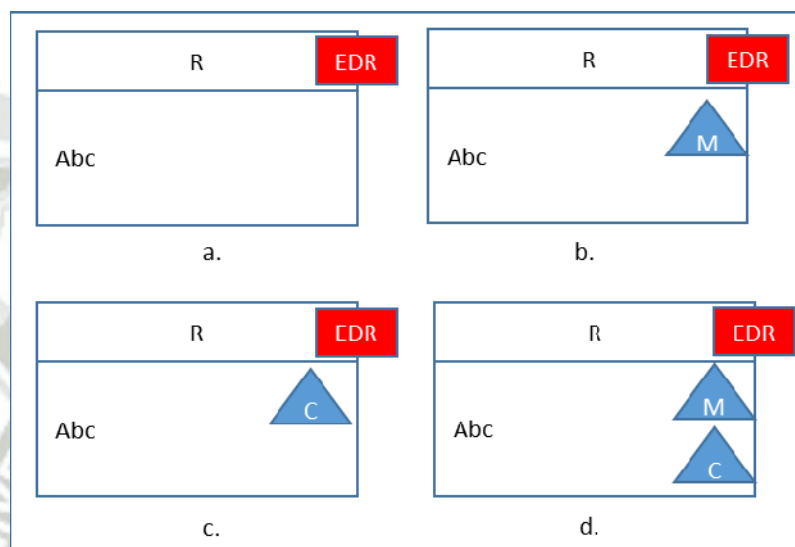


Figura 19. Ampliación UML para la educación de requisitos

Fuente: Elaboración propia

Cuando se trata de referenciar a la etapa de educación de requisitos se usa el ítem a.; el ítem b. y c. hacen referencia cuando en la educación de requisitos solo se emplea la medición y el control mientras que en el ítem d. en la educación se emplean ambos elementos. Se debe notar que se emplea la plantilla de casos de uso entonces se debe de agregar una “M” o “C” si es que el requisito necesita una medición o un control respectivamente.

Si se emplea una plantilla para obtener la educación de requisitos, esto es la IEEE-830 entonces se puede agregar una “M” o “C” a los requisitos que requieran una medida o un control. Esto va a permitir que los requisitos educacionados reflejen claramente las necesidades de medición o control asignados. La especificación para esta etapa se muestra en la tabla 7.

Tabla 7
Especificación para la educación de requisitos

EDU N°	Nombre del requisito a educionar	
M/C	Métrica, control o normal (A=Métrica, B=Control)	
Versión	Digito.DigitoDigito	
Autor	Apellidos y Nombres del Autor	
Cargo	Cargo que ocupa el autor	
Stakeholder	Apellidos y Nombres del Stakeholder	
Especialista	Nombres y Apellidos de la persona que llevará a cabo la educación.	
	Especialidad: Ing.	Tipo: Analista, etc
	Requerimientos	
	Experiencia Laboral (Alta, Media, Baja): El tiempo de experiencia le proporciona la cualidad	
Edución	Tipo: Primario, modificado	
	Objetivo: Objetivo del requerimiento educionado	
	Fecha: (dd/mm/aa), Fecha cuando se educciona el requerimiento	
Edución		
Descripción: Descripción del requerimiento a educionar.		
Observaciones: Observaciones del requerimiento educionado.		

Fuente: Elaboración propia

3.2.2 Etapa de elicitación de requisitos

En esta etapa, los ingenieros de requisitos logran abstraer la comprensión del problema para plasmarlos en documentos que permitan resolver el producto software a construir. Principalmente es una actividad donde quedan identificadas a las personas por parte del equipo de desarrollo del producto y personas de la organización estableciéndose un conjunto de relaciones personales (Pressman, 2014) (Somerville, 2005). También para esta etapa quedan identificados aquellos buenos requisitos y se dan respuesta a una serie

de inconsistencias obtenidas en la etapa previa. Los problemas a los que se enfrentan los ingenieros de requisitos son los siguientes:

- Problemas de alcance. Cuando los límites del problema se encuentran mal definidos producto de información innecesaria entregada por los clientes o por las personas relacionadas con información importante.
- Problemas de comprensión. Cuando los clientes no saben expresar sus necesidades provocando inconsistencias en el entendimiento de los límites del problema.
- Problemas de volatilidad. Cuando los requisitos cambian con el tiempo producto de una mala explicación del problema o producto del giro de la organización.

Para ayudar a superar estos problemas, se deben ejecutar de forma organizada la captura de requisitos además de solicitar la agregación de medidas que permitan una medición y control efectivo. Se debe recordar que esta etapa es un componente importante en la construcción de los productos de software porque presenta impactos importantes en las fases de construcción posteriores. Si se lleva a cabo de manera correcta, puede ayudar a reducir los cambios y las correcciones en los requisitos. UML tampoco presenta una expresión gráfica para esta etapa. En la figura 20 se presenta la propuesta.

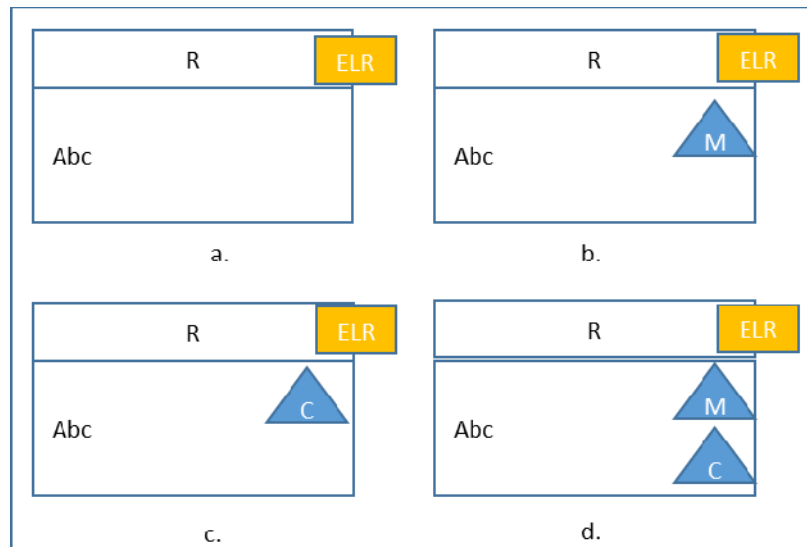


Figura 20. Ampliación UML para la elicitación de requisitos

Fuente: Elaboración propia

Cuando se trata de referenciar a la etapa de elicitación de requisitos se usa el ítem a.; el ítem b. y c. hacen referencia cuando en la elicitación de requisitos solo se emplea la medición y el control mientras que en el ítem d. en la elicitación se emplean ambos elementos. Se debe notar que se emplea la plantilla de casos de uso entonces se debe de agregar una “M” o “C” si es que el requisito necesita una medición o un control respectivamente.

Si se emplea una plantilla para obtener la elicitación de requisitos, esto es la IEEE-830 entonces se puede agregar una “M” o “C” a los requisitos que requieran una medida o un control. La tabla 8 muestra su correspondiente especificación.

Tabla 8
Especificación para la elicitación de requisitos

ELI N°	Nombre del requerimiento elicitado
M/C	Métrica, control o normal (A=Métrica, B=Control)
Educción N°	Código de los requisitos educionados
Versión	Versión del tema, Digito.DigitoDigito
Fecha	(dd/mm/aa) Fecha que se lleva a cabo la elicitación
Autor(es)	Apellidos y nombres del autor (es) de la elicitación
Fuentes	Apellidos y nombres de los actores que intervienen en la elicitación
Dependencias	Áreas con los que se relaciona
Descripción	Descripción de la elicitación
Precondición	Pre-condiciones para que funcione la elicitación
	1. Descripción de la secuencia de la escena
Secuencia normal	
Poscondición	Pos-condiciones de la elicitación
	1. Excepciones por cada elemento de la secuencia normal
Excepciones	
Comentarios	Comentarios con respecto al requerimiento elicitado

Fuente: Elaboración propia

3.2.3 Etapa de especificación de requisitos de software

La especificación de requisitos de software es la etapa culminante del proceso de obtención de información ya que todo lo entendido y consistente es entregado a los desarrolladores de código para que éstos lo plasmen en código de algún lenguaje de programación (Pressman, 2014) (Somerville, 2005). La traducción implica lograr entender las necesidades de recursos computacionales que son empleadas en la codificación del mismo.

Si los niveles de entendimiento son claros entonces los niveles de codificación no sufrirán consecuencias algunas, pero si la especificación es inconsistente entonces se logrará un producto mal codificado quedando el proyecto como un fracaso. La teoría no hace mención alguna sobre las

necesidades de vigilancia por medio de mediciones y controles; UML tampoco amplía, en sus esquemas, estas necesidades. La figura 21 muestra la ampliación en contexto de UML para lograr esta representación.

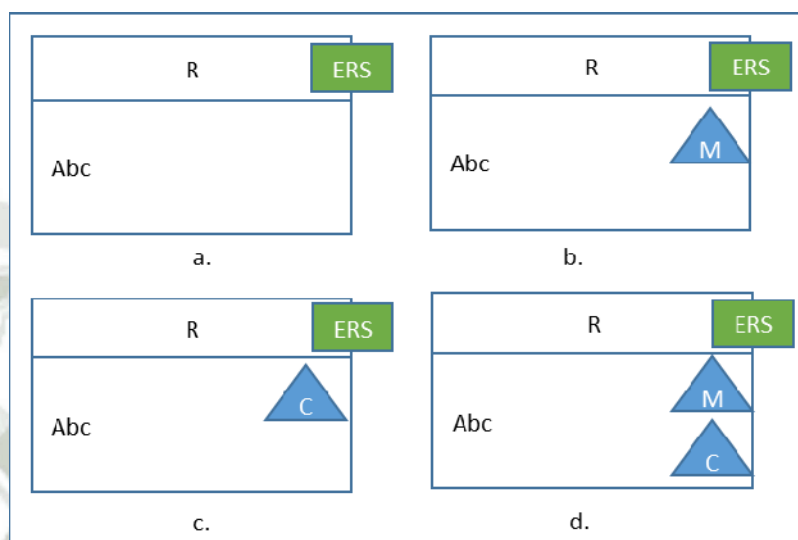


Figura 21. Ampliación UML para la especificación de requisitos de software

Fuente: Elaboración propia

Cuando se trata de referenciar a la etapa de especificación de requisitos de software se usa el ítem a.; el ítem b. y c. hacen referencia cuando en la especificación de requisitos de software solo se emplea la medición y el control mientras que en el ítem d. en la especificación se emplean ambos elementos. Se debe notar que se emplea la plantilla de casos de uso entonces se debe de agregar una “M” o “C” si es que el requisito necesita una medición o un control respectivamente.

Si se emplea una plantilla para obtener la especificación de requisitos de software, esto es la IEEE-830 entonces se puede agregar una “M” o “C” a los requisitos que requieran una medida o un control. La tabla 9 muestra la especificación para la especificación de requisitos de software.

Tabla 9

Especificación para la especificación de requisitos de software

ERS N°	Nombre de la especificación de requerimiento de software
M/C	Métrica, control o normal (A=Métrica, B=Control)
Elicitación N°	Código de las elicitaciones relacionadas
Versión	Versión de la ERS. (Digito.DigitoDigito)
Fecha	(dd/mm/aa) Fecha que se lleva a cabo la elicitación
Autor	Autores de la especificación
Fuente	Stakeholders de la especificación
Dependencias	Áreas o módulos con los que se relaciona
Descripción	Descripción de la especificación
Precondición	Pre-condiciones iniciales para que funcione el requerimiento
Poscondición	Pos-condiciones de salida del requerimiento
Excepciones	1. Excepciones que produce el requerimiento
Comentarios	Comentarios del requerimiento especificado

Fuente: Elaboración propia

3.2.4 Etapa de diseño del diagrama de casos de uso

A partir del nacimiento de UML, muchos investigadores consideran a los casos de uso como una buena manera de lograr una especificación del comportamiento de un sistema a automatizar. UML tiende a convertirse en un estándar para el modelado de sistemas de información. Es importante resaltar que los casos de uso permiten una representación clara de los requisitos ya sea de la fase de educación, elicitación o especificación de software. En ese sentido, los casos de uso nos hacen comprender un sistema a partir de los servicios o funciones que ofrece a su entorno; de esta manera permite definir funcionalidades adecuadas que se adhieren al concepto como tal (Booch, 2005).

Los casos de uso son independientes del método de diseño y del método de programación (Booch, 2005). UML expone una gráfica para este instrumento pero no evidencia las mediciones y controles que se pudieran hacer

para lograr confianza en el modelamiento de los sistemas. La figura 22 muestra la propuesta.

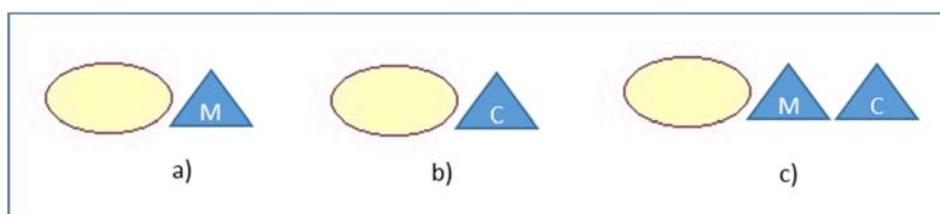


Figura 22. Ampliación UML para el diagrama de casos de uso

Fuente: Elaboración propia

Se respeta la notación original definida por UML sobre los casos de uso, pero se amplía la notación dependiendo de las necesidades del analista. Un caso de uso sin ningún componente extra es tratado de la misma forma como lo trata UML para el modelamiento de sistemas de información. Un caso de uso al cual se le adhiere una “M” significa que se debe de prestar atención al caso de uso ya que se deben definir medidas de soporte. Una “C” significa que este caso de uso necesita de un control; y si al mismo tiempo presenta una “M” y “C” significa que el caso de uso en estudio necesita ambos componentes: una medida y un control. La tabla 10 muestra la especificación para el caso de uso correspondiente.

Tabla 10
Especificación para los casos de uso

UC- N°	Código del caso de uso
M/C	Métrica, control o normal (A,B=Métrica, C=Control)
Versión	Versión del caso de uso (Dígito.DígitoDígito)
Fecha	Fecha de la creación del caso de uso
Autores	Grupo de trabajo
Fuentes	Stakeholders
Dependencias	Dependencias del caso de uso (Educación, elicitación y/o especificación de requisitos de software)
Descripción	Descripción del caso de uso
Precondición	Precondiciones de ejecución
Secuencia normal	Paso Acción - -
Postcondición	Postcondiciones de ejecución

Excepciones	Paso	Acción
	-	-
Importancia	Importancia del caso de uso	
Urgencia	Tipo de urgencia de construcción	
Estado	Estado en la construcción del producto	
Estabilidad	Estabilidad del caso de uso	
Comentarios	Comentarios sobre las acciones del caso de uso	

Fuente: Elaboración propia

3.2.5 Etapa de diseño del modelo conceptual

El modelo conceptual permite tener una visión clara del grado de correlación entre un conjunto de componentes, de primera vista, del sistema computacional a construir (Pressman, 2014) (Somerville, 2005). Esta se encuentra en función de la ingeniería de requisitos, para cuyo modelado se deben seguir los siguientes pasos:

- Hacer una lista de conceptos candidatos empleando una lista de categorías relacionados con los requisitos. La relación debe estar dada de tal manera que se puedan identificar las inconsistencias.
- Representarlos en un diagrama propuesto por UML
- Añadir asociaciones para preparar las relaciones entre conceptos que es necesario conocer y entender.
- Añadir atributos que abarquen toda la información que se necesite conocer de cada uno de los conceptos.

En este aspecto UML brinda una forma clara de diseñar un modelado conceptual pero no incluye mediciones ni controles para lograr trazabilidades exitosas o por lo menos que permitan reducir errores en la construcción del producto final. Este modelado permitirá que logremos un seguimiento adecuado de todos aquellos requisitos que requieren de una atención especial; es decir de aquellos que necesitamos controlar o necesitamos incrementar alguna medición. La figura 23 muestra la ampliación correspondiente.

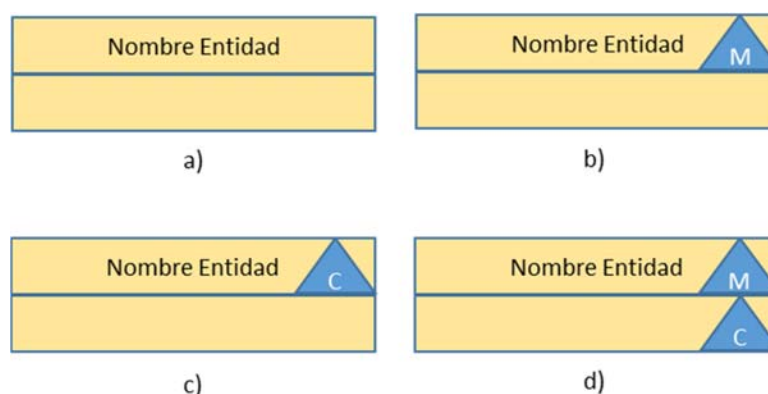


Figura 23. Ampliación UML para el modelo conceptual

Fuente: Elaboración propia

Se toma la base del artefacto original de UML para la representación del modelo conceptual (a)). Si una entidad necesita una medida o un control se emplea la notación b) o c). Si se necesita que en la entidad se contemple la medición y control los indicadores deben ser agregados como en d). Los elementos de Cardinalidad y asociación se respetan del UML original. La tabla 11 muestra la especificación para el modelo conceptual.

Tabla 11
Especificación del modelo conceptual

M/C	Métrica, control o normal (A=Métrica, B=Control)
Versión	Versión del modelo conceptual (Dígito.DígitoDígito)
Fecha	Fecha de la creación del modelo conceptual
Autores	Autor del modelo conceptual
Descripción	Descripción del modelo conceptual
Bloques de construcción	Elementos, relaciones y diagramas
Descripción del elemento	Nombre del bloque
Descripción de relaciones	Palabra que asocia un bloque de información con otro
Descripción del diagrama	Gráfico que conecta un conjunto de bloques con elementos y/o relaciones
Comentarios	Comentarios sobre el modelo conceptual

Fuente: Elaboración propia

3.2.6 Etapa de diseño de los diagramas de secuencias

Este diagrama, también llamado diagrama de interacción, muestra las interacciones entre un conjunto de objetos (clases, actores), ordenadas según el tiempo en que tienen lugar. Es decir, muestra el orden de las llamadas en el sistema. Se utiliza un diagrama para cada llamada a representar. Es imposible representar en un solo diagrama la secuencia de todas las llamadas posibles del sistema, es por ello que se escoge un punto de partida (Booch, 2005).

El objeto puede existir sólo durante la ejecución de la interacción, se puede crear o puede ser destruido durante la ejecución de la interacción. En este tipo de diagramas también intervienen los mensajes, que son la forma en que se comunican los objetos: el objeto origen solicita una operación del objeto destino. El diagrama de secuencia puede ser obtenido de dos partes, desde el diagrama estático de clases o desde el diagrama de casos de usos.

Si bien es cierto que UML tiene un diagrama para este artefacto, dentro del mismo no emplean elementos de medición y control, a pesar que este diagrama logra un control exhaustivo del paso de mensajes. La figura 24 muestra la ampliación para UML.

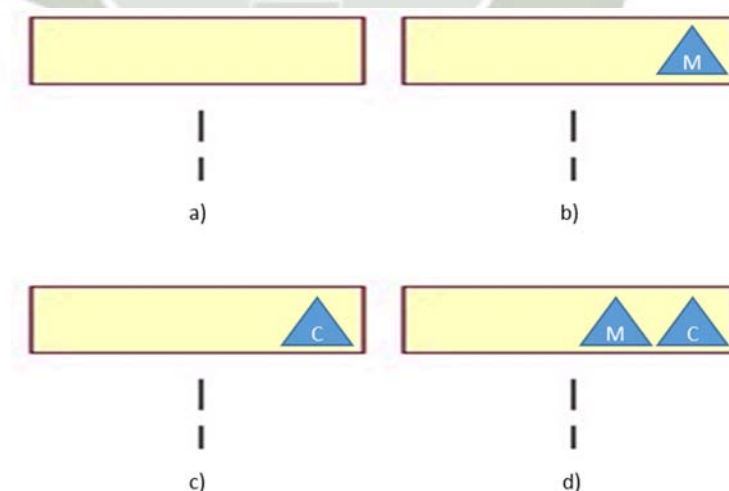


Figura 24. Ampliación UML para el diagrama de secuencias

Fuente: Elaboración propia

Se respeta el artefacto original de UML para el diagrama de secuencias tal como se muestra en a). Si un objeto que integra el diagrama de secuencias necesita de alguna medida entonces se agrega el indicador “M” tal como se ve b); ídem si es que se necesita integrar un control como en c). Si un objeto merece una atención especial ya que necesita ambos indicadores entonces el mismo queda expresado como en d).

Las líneas de vida y el paso de mensajes no sufren alteración alguna de la filosofía original de UML. La tabla 12 muestra la especificación para el diagrama de secuencias.

Tabla 12
Especificación para el diagrama de secuencias

M/C	Métrica, control o normal (A,B=Métrica, C=Control)
Versión	Versión del diagrama de secuencia (Dígito.DígitoDígito)
Fecha	Fecha de la creación del diagrama de secuencia
Autores	Autor del diagrama de secuencia
Descripción	Descripción del diagrama de secuencia bajo las dos siguientes características: 1. La línea de vida de un objeto es una línea discontinua vertical que representa la existencia de un objeto a lo largo de un período de tiempo. Los objetos se colocan en la parte superior del diagrama, con sus líneas de vida dibujadas de arriba hacia abajo. Las líneas de vida comienzan con la recepción del mensaje y los mismos pueden destruirse durante la interacción. 2. El foco de control es un rectángulo delgado y estrecho que representa el periodo de tiempo durante el cual un objeto ejecuta una acción, bien sea directamente o por medio de un procedimiento.
Descripción del diagrama	Gráfico que conecta un conjunto de objetos con sus líneas de vida por medio de mensajes
Comentarios	Comentarios sobre el diagrama de secuencias

Fuente: Elaboración propia

3.2.7 Etapa de diseño de interfaces gráficas de usuario

Durante la presente etapa, se deben de definir un conjunto de estructuras gráficas que permitan adecuar la interacción con el usuario final. Para este diseño se toma las iniciáticas del usuario final plasmadas en la etapa de

ingeniería de requisitos permitiendo lograr un modelo de navegabilidad de acuerdo con las necesidades de información del cliente (Pressman, 2014) (Somerville, 2005). Para lograr esto se debe de tomar en cuenta lo siguiente:

- El sistema debe presentar la información claramente al usuario, es decir mostrar un título adecuado por cada pantalla, minimizando el uso de abreviaturas y proporcionando una retroalimentación de usuario clara.
- El sistema debe ser consistente a lo largo de sus pantallas.
- El Sistema debe realizar un reconocimiento de la aceptación de entrada, es decir, el reconocimiento que la entrada está en la forma correcta. Antes del envío de datos a servidor, se debe validar en la medida de lo posible que lo que se desea ingresar sea correcto, o, por lo menos, sea un valor posible.
- El sistema debe informar de forma clara y explícita del problema con los datos ingresados de forma clara al usuario, en los casos que sea necesario.
- Los usuarios deben saber cuándo su petición está completa y si pueden efectuar nuevas solicitudes para ello el sistema desplegará un mensaje de retroalimentación.
- El sistema debe mostrar mensajes que ha de incluir regreso a la definición de la petición para que el usuario pueda revisar si los datos fueron ingresados correctamente, en lugar de dar comandos que no pueden ser ejecutados.
- El título de la página debe permitir al usuario identificar claramente en qué lugar del sistema se encuentra ubicado.
- En caso de interfaces secuenciales se debe permitir al usuario acceso a la siguiente o anterior interfaz de la guía consecutiva.
- Las páginas deben tener como tamaño máximo una pantalla proporcional a las necesidades del usuario final.
- Los campos de solo lectura aparecerán sombreados, mientras que los campos editables serán de fondo blanco. Otra característica de los campos no editables es que al estar deshabilitados no reciben el enfoque.

- El enfoque de los controles van de arriba hacia abajo y de izquierda a derecha, se tendrá en cuenta que los campos no editables no recibirán el enfoque.

La usabilidad depende de tres factores: 1) Usuario, quien interactuará con el artefacto en función de sus conocimientos y de la capacidad de uso del artefacto. 2) Artefacto: que será manejado por el usuario según el diseño de interfaz que el diseñador le proporcione. 3) Contexto: el cual influirá y, en algunos casos, determinará el uso del artefacto modificando, en determinadas oportunidades, su significado. UML no propone esquema alguno para la forma de diseñar los interfaces gráficos, solo muestra un artefacto que indica el lugar del interfaz gráfico; asimismo no muestra elementos de medición y control para la satisfacción del usuario final. La figura 25 muestra la propuesta para este tema.

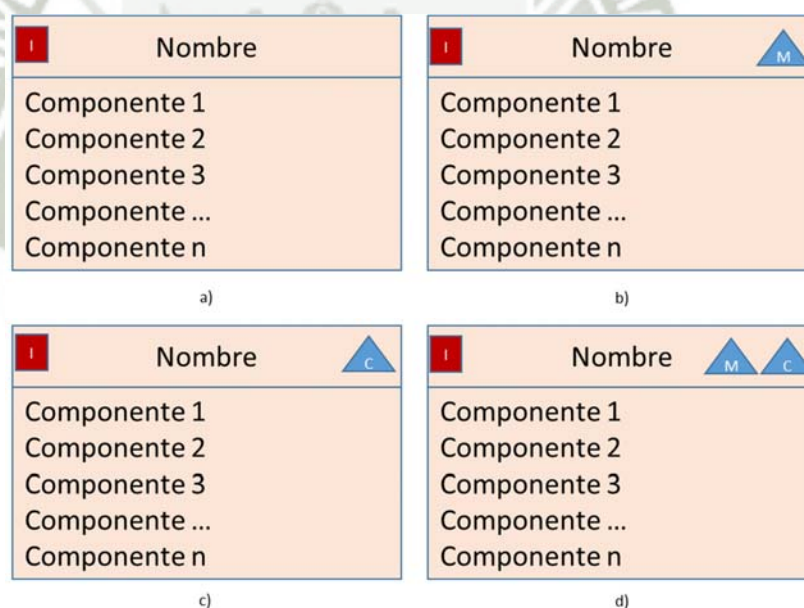


Figura 25. Ampliación UML para los interfaces gráficos de usuario

Fuente: Elaboración propia

El artefacto gráfico que representa al interface gráfico de usuario queda definido por la letra “I” que se encuentra en el extremo superior izquierdo del

gráfico. Si todo el interfaz requiere de una medida o control entonces se agrega la letra “M” o “C” en el extremo superior derecho tal como se muestra en b), c) y d).

En el caso de que solamente alguno de los componentes requieren de alguna medida o control, se adjunta el indicador “M” o “C” o “M””C” al componentes fuente tal como se muestra en la figura 26. En ella se muestra que el componente 1 necesita de control mientras que el componente n necesita de medidas. Asimismo, el componente 3 necesita de una medida y control. La tabla 13 muestra la especificación para los interfaces gráficos de usuario.

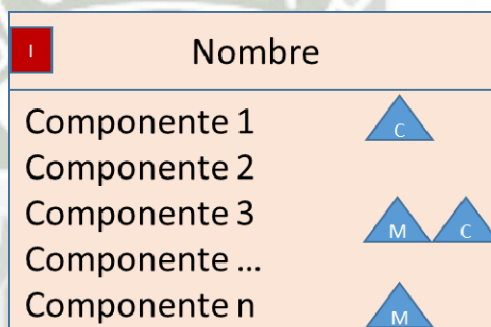


Figura 26. Ampliación UML para los interfaces gráficos de usuario por componentes

Fuente: Elaboración propia

Tabla 13
Especificación para los interfaces gráficos de usuario

M/C	Métrica, control o normal (A=Métrica, B=Control)
Versión	Versión del correspondiente interfaz gráfico de usuario (Dígito.DígitoDígito)
Fecha	Fecha de la creación del interfaz gráfico de usuario
Autores	Autor del diseño del interfaz gráfico de usuario
Descripción	Descripción del interfaz gráfico de usuario bajo las cinco siguientes características: 1. La interfaz gráfica de usuario está compuesto por un rectángulo sólido, con características flexibles en su tamaño. En la parte superior del rectángulo se insertan los íconos de minimización, eliminación y aparición del interfaz gráfico de usuario. 2. En el interior del rectángulo se agrupan íconos que corresponden a un conjunto de figuras que permitan insertar la información específica del caso como por ejemplo: combobox, listbox, textfields, botones, entre otros.

	3. Los íconos y el rectángulo deben de colorearse siguiendo la teoría del color.
	4. Los íconos deben contener formas siguiendo las características de la teoría de la forma.
	5. Al destruir un rectángulo, queda destruido absolutamente todo su contenido.
Descripción del diagrama	Gráfico que conecta un conjunto de objetos, íconos, colores y formas.
Comentarios	Comentarios sobre el diseño del interfaz gráfico de usuario

Fuente: Elaboración propia

3.2.8 Etapa de diseño de la arquitectura del software

En general, las definiciones entremezclan despreocupadamente (1) el trabajo dinámico de estipulación de la arquitectura dentro del proceso de ingeniería o el diseño, (2) la configuración o topología estática de sistemas de software contemplada desde un elevado nivel de abstracción y (3) la caracterización de la disciplina que se ocupa de uno de esos dos asuntos, o de ambos (Pressman, 2014) (Somerville, 2005).

La arquitectura de software es, a grandes rasgos, una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se la percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema (Pressman, 2014) (Somerville, 2005). La vista arquitectónica es una vista abstracta, aportando el más alto nivel de comprensión y la supresión o diferimiento del detalle inherente a la mayor parte de las abstracciones.

La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución. La aplicación de una estrategia sistemática, disciplinada y cuantificable al desarrollo, aplicación y mantenimiento del software. Los modelos a tomar en cuenta son los siguientes (Pressman, 2014) (Somerville, 2005):

- Modelos estructurales: Sostienen que está compuesta por componentes, conexiones entre ellos y (usualmente) otros aspectos tales como

configuración, estilo, restricciones, semántica, análisis, propiedades, racionalizaciones, requerimientos, necesidades de los participantes. El trabajo en esta área está caracterizada por el desarrollo de lenguajes de descripción arquitectónica.

- Modelos de framework: Similares a la vista estructural, pero su énfasis primario radica en la (usualmente una sola) estructura coherente del sistema completo, en vez de concentrarse en su composición. Los modelos de framework a menudo se refieren a dominios o clases de problemas específicos.
- Modelos dinámicos: Enfatizan la cualidad conductual de los sistemas. “Dinámico” puede referirse a los cambios en la configuración del sistema, o a la dinámica involucrada en el progreso de la computación, tales como valores cambiantes de datos.
- Modelos de proceso: Se concentran en la construcción de la arquitectura, y en los pasos o procesos involucrados en esa construcción. En esta perspectiva, la arquitectura es el resultado de seguir un argumento (script) de proceso. Esta vista se ejemplifica con el actual trabajo sobre programación de procesos para derivar arquitecturas.
- Modelos funcionales: Una minoría considera la arquitectura como un conjunto de componentes funcionales, organizados en capas que proporcionan servicios hacia arriba. Es tal vez útil pensar en esta visión como un framework particular.

Ninguno de estos modelos excluye a los otros, ni representa un conflicto fundamental sobre lo que es o debe ser. Por el contrario, representan un espectro sobre distintos énfasis que pueden aplicarse a la arquitectura: sobre sus partes constituyentes, su totalidad, la forma en que se comporta una vez construida, o el proceso de su construcción. Tomadas en su conjunto, destacan más bien un consenso.

Al respecto UML proporciona esquemas gráficos que permiten representar la arquitectura de software pero en ellos no se encuentran involucrados aspectos de medición y control (Booch, 2005). La figura 27 muestra la ampliación propuesta a UML.

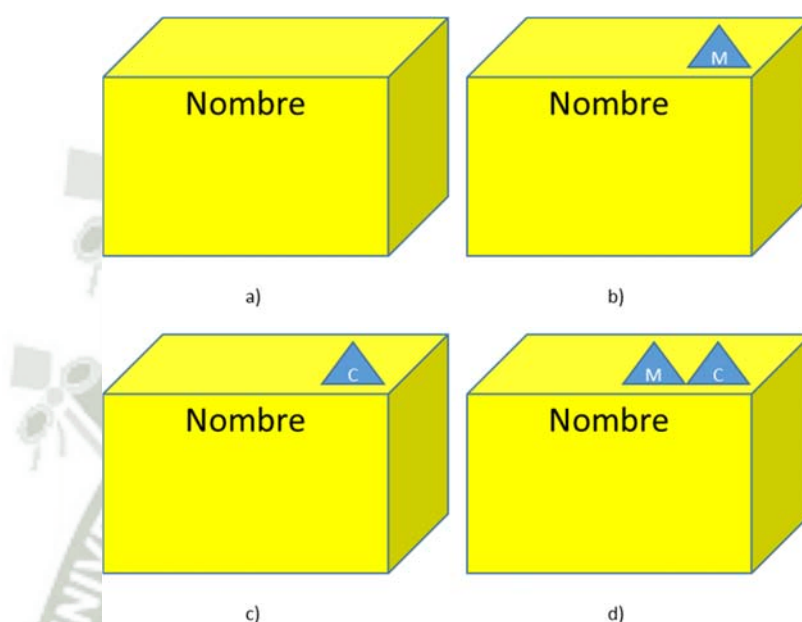


Figura 27. Ampliación UML para la arquitectura de software

Fuente: Elaboración propia

Se respeta el componente propuesto por UML. En este caso si el componente necesita de alguna medición entonces se agrega el indicador “M” como en el inciso b). Si necesita un control se agrega una “C” como en b) y si necesita ambos se agregan los indicadores “M””C” como en d). La tabla 14 muestra la especificación para la arquitectura.

Tabla 14

Especificación para la arquitectura de software

M/C	Métrica, control o normal (A=Métrica, B=Control)
Versión	Versión correspondiente a la arquitectura (Dígito.DígitoDígito)
Fecha	Fecha de la creación de la arquitectura
Autores	Autor del diseño de la arquitectura

Descripción	Descripción de la arquitectura bajo las siguientes características: 1. Es un tipo especial de diagrama que contiene un nombre y un contenido gráfico que representa la proyección del modelo. 2. El diagrama contiene: componentes, interfaces y relaciones de dependencias, relaciones de generalización, relaciones de asociación y relaciones de realización. 3. Estos diagramas pueden contener notas y restricciones. 4. Para modelar sistemas adaptables se debe considerar la distribución física de los componentes que pueden migrar de un nodo a otro. La localización de la instancia de un componente se puede especificar marcándola con un valor etiquetado de localización, que se puede representar en un diagrama. 5. Para modelar las acciones de migración de un componente, se debe crear el correspondiente diagrama que contenga instancias de componentes. Un cambio de localización se puede describir dibujando la misma instancia más de una vez, pero con diferentes valores para su valor etiquetado de localización.
Descripción del diagrama	Gráfico que conecta un conjunto de componentes.
Comentarios	Comentarios sobre la arquitectura.

Fuente: Elaboración propia

3.2.9 Etapa de diseño de los diagramas de colaboración

Este diagrama muestra la interacción entre varios objetos y los enlaces que existen entre ellos. Representa las interacciones entre objetos organizadas alrededor de los objetos y sus vinculaciones. A diferencia de un diagrama de secuencias, un diagrama de colaboraciones muestra las relaciones entre los objetos, no la secuencia en el tiempo en que se producen los mensajes (Booch, 2005). Los diagramas de secuencias y los diagramas de colaboraciones expresan información similar, pero en una forma diferente.

Para este tipo de diagramas, UML propone claramente esquemas gráficos para representar en forma clara el envío de un mensaje. Se representa mediante una flecha dirigida cercana a un enlace. Los mensajes que se envían entre objetos pueden ser de distintos tipos, también según como se producen en el tiempo; existen mensajes simples, sincrónicos, balking, timeout y asíncronos. Durante la ejecución de un diagrama de colaboración se crean y destruyen objetos y enlaces. No existe elementos de medición pero si existen los elementos de control. La figura 28 muestra la ampliación UML.

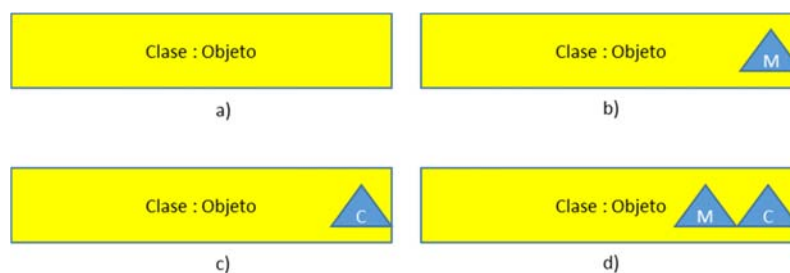


Figura 28. Ampliación UML para la arquitectura de software

Fuente: Elaboración propia

Los objetos derivadas de las clases que conforman el diagrama de colaboración presentan los indicadores “M” o “C” dependiendo de las necesidades que se tengan de los objetos tal como se muestran en los puntos b), c) y d). Debe existir una correspondencia biunívoca con las definiciones presentadas en los diagramas de clases. La tabla 15 muestra la especificación para este diagrama.

Tabla 15
Especificación para el diagrama de colaboración

M/C	Métrica, control o normal (A,B=Métrica, C=Control)
Versión	Versión correspondiente al diagrama de colaboración (Dígito.DígitoDígito)
Fecha	Fecha de creación del diagrama de colaboración
Autores	Autor del diagrama de colaboración
Descripción	Descripción del diagrama de colaboración: 1. Una colaboración es una sociedad de clase, interfaces y otros elementos que colaboran para proporcionar un comportamiento cooperativo mayor que la suma de los comportamientos de sus elementos. 2. Cada colaboración tiene un nombre que la distinga de las otras colaboraciones. Un nombre es una cadena de texto. Se dibuja mostrando su nombre. 3. Las colaboraciones tienen dos aspectos: Una estructural que especifica las clases, interfaces y otros elementos que colaboran para llevar a cabo la colaboración a la que da nombre, y una parte de comportamiento que especifica la dinámica de cómo interactúan esos elementos. 4. La parte estructural de una colaboración se representa mediante diagramas de clases, la parte de comportamiento se representa por un diagrama de interacción.
Descripción del diagrama	Gráfico que conecta un conjunto de componentes en el diagrama de colaboración.
Comentarios	Comentarios sobre el diagrama de colaboración.

Fuente: Elaboración propia

3.2.10 Etapa de diseño del diagrama de clases

En el diagrama de clases será donde definiremos las características de cada una de las clases y relaciones de dependencia y generalización. Es decir, es donde daremos rienda suelta a nuestros conocimientos sobre el contexto de la organización, definiendo las clases e implementando las ya típicas relaciones de herencia y agregación (Booch, 2005).

Este diagrama sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso y de contenido. Se utiliza cuando se necesita realizar un análisis de dominio: El analista se entrevista con el cliente con el objetivo de conocer las entidades principales en el dominio del cliente (Pressman, 2014) (Somerville, 2005). Durante la conversación entre el cliente y el analista se deben tomar apuntes. Desde estos apuntes, se buscarán las clases para los objetos del modelo buscando los sustantivos y convirtiéndolos en clases.

UML propone un amplio repertorio de conocimientos para este artefacto pero en ningún estado involucra los elementos de medición y control. El agregar estos indicadores permitirá saber cuáles son los atributos o métodos que merecen una especial atención y lograr una trazabilidad que compromete desde la ingeniería de requisitos en la etapa de análisis hasta la confección de diagramas acordes en la etapa de diseño.

Este artefacto respeta el artefacto original definida por UML en donde se indica que toda clase presenta un nombre, un conjunto de atributos y un conjunto de métodos. La figura 29 muestra la propuesta de modificación para este esquema gráfico.

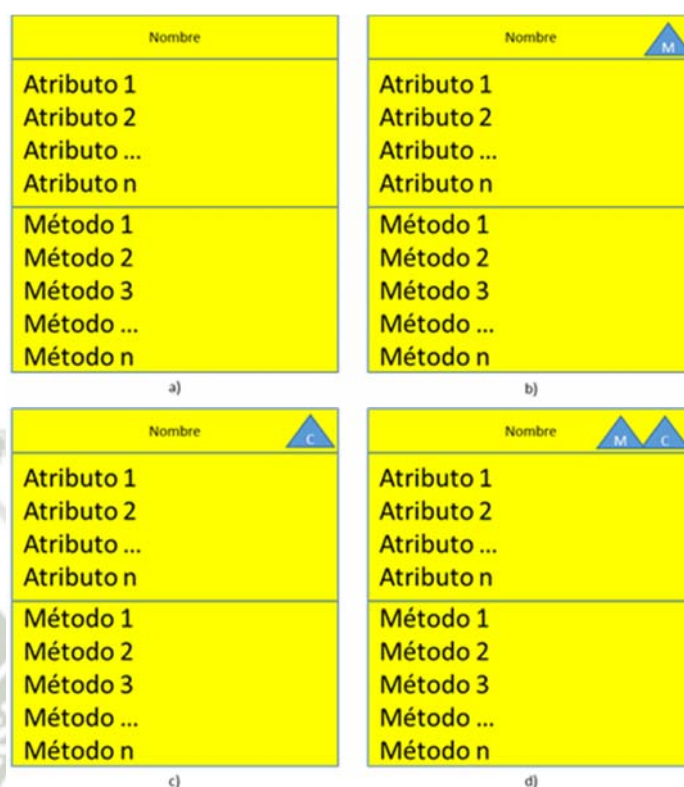


Figura 29. Ampliación UML para el diagrama de clases

Fuente: Elaboración propia

Este gráfico muestra el artefacto que define una clase según UML. Cuando se necesita que una clase tenga una medida se agrega el indicador “M” o “C” si es control tal como se muestra en b), c) y d).

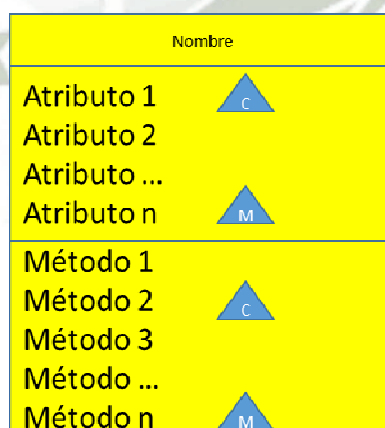


Figura 30. Ampliación UML para componentes de un diagrama de clases

Fuente: Elaboración propia

La figura 30 muestra que si solo se necesita obtener controles o medidas de algún atributo o método, el correspondiente indicador debe ir al lado derecho del atributo o indicador que así lo requiera.

En UML, se conoce como estereotipo a un elemento de texto que al ser aplicado a otro elemento define su categoría. Es uno de los mecanismos de extensión del lenguaje toda vez que permite cambiar o complementar la semántica de cualquier elemento (Booch, 2005).

La especificación UML estandariza un conjunto de estereotipos que se pueden aplicar a los elementos de un modelo. En general, cualquier diseñador puede crear conjuntos de estereotipos (Booch, 2005).

Para el caso de la propuesta cuando se tenga que relacionar clases propias de UML (definidas según su metamodelo) y las clases modificadas propuesta, estas deben conectarse por medio de estereotipos para lograr identificar las clases propias del sistema. Estas clases quedarán definidas porque incluyen el símbolo M o C dependiendo del contexto de trabajo en que se encuentre llevando a cabo el modelamiento.



Figura 31. Conectores que funcionan como elementos de estereotipo

Fuente: Elaboración propia

Así, la forma de llevar a cabo una ampliación de clase vía empleo de estereotipos se muestra en la figura 32. En ella también se indica que a las clases estereotipos convocados se les puede aplicar la métrica y/o control correspondiente.

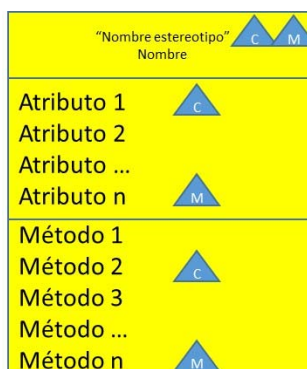


Figura 32. Clases que redireccionan a metaclasses basadas en estereotipos

Fuente: Elaboración propia

Asimismo, los interfaces son representados por la forma correcta de UML con la salvedad de que en su implementación puede estar orientado a encontrar un control o medida tal como se muestra en la figura 33.



Figura 33. Clases que redireccionan servicios de una clase o interfaz

Fuente: Elaboración propia

La tabla 16 muestra la especificación para el diagrama de clases.

Tabla 16

Especificación para el diagrama de clases

M/C	Métrica, control o normal (A=Métrica, B=Control)
Multiplicidad	M_i / C_i (A_i = Métrica, B_i = Control)
Versión	Versión correspondiente al diagrama de clases (Dígito.DígitoDígito)
Fecha	Fecha de creación del diagrama de clases
Autores	Autor del diagrama de clases

Descripción	Descripción del diagrama de colaboración:
	1. Un diagrama de clases muestra un conjunto de interfaces, colaboraciones y sus relaciones. Es colección de arcos y nodos. 2. El diagrama de clases contiene: Clases, interfaces, colaboraciones, relaciones de dependencia, relaciones de generalización y relaciones de asociación 3. Los diagramas de clases pueden contener paquetes o subsistemas, los cuales se emplean para agrupar los elementos de un modelo en partes más grandes. 4. El diagrama de clases modela el vocabulario del sistema, colaboraciones simples, esquema lógico de la base de datos. 5. Para modelar una colaboración se debe de hacer: ▪ Identificar mecanismos que se quieren modelar. ▪ Para cada mecanismo se debe de identificar las clases, interfaces y otras colaboraciones que participan en esta colaboración. ▪ Se usan los escenarios para recorrer la interacción entre estos elementos. ▪ Se debe de rellenar los elementos con su contenido. 6. Para las bases de datos se debe de hacer lo siguiente: ▪ Identificar aquellas clases del modelo cuyo estado debe trascender el tiempo de vida de las aplicaciones. ▪ Se deben de marcar las clases como persistentes. ▪ Se expanden los detalles estructurales de las clases. ▪ Se buscan patrones comunes que compliquen el diseño físico de bases de datos, tales como asociaciones cíclicas, asociaciones uno a uno y asociaciones múltiples.
Descripción del diagrama	Gráfico que conecta un conjunto de componentes en el diagrama de clases
Comentarios	Comentarios sobre el diagrama de clases

Fuente: Elaboración propia

3.2.11 Medidas e indicadores

Las medidas e indicadores centrados en la evaluación cuantitativa se basan generalmente en un conteo de los elementos de una lista de características (atributos o guías), a ser analizadas para un producto o productos competitivos. La lista puede contener características de distinto tipo, tales como cantidad de requisitos educionados, cantidad de diagramas de casos de uso empleados, entre otros.

Luego del proceso de evaluación, en la etapa de análisis, quienes toman las decisiones frecuentemente crean para, cada producto de software, una lista de con las cantidad de los atributos seleccionados. Finalmente, mediante un

mecanismo intuitivo, comparando las cantidades obtenidas para cada característica de cada producto de software, se arriba a una conclusión final. Este enfoque es obviamente conveniente y atractivo cuando el objeto de la evaluación y el proceso de decisión son lo suficientemente simple. Las medidas e indicadores a tomarse en cuenta, en el presente trabajo de investigación se muestran en la tabla 17.

Tabla 17

Medidas e indicadores

Conglomerado	Etapas	Medidas	Indicadores	Parámetro (correcto si)
Ingeniería de Requisitos	Educción de requisitos (métricas, control)	- A = Cantidad de requisitos educionados - B = Cantidad de requisitos que no han sido educionados	$IEd = (B / A) * 100$	< 10%
	Elicitación de requisitos (métricas, control)	- A = Cantidad de requisitos elicitados - B = Cantidad de requisitos que no han sido elicitados	$IEl = (B / A) * 100$	< 10%
	Especificación de requisitos de software (métricas, control)	- A = Cantidad de requisitos especificados - B = Cantidad de requisitos que no han sido especificados	$IEs = (B / A) * 100$	< 10%
Diseño	Diagrama de casos de uso (métricas, control)	- A = Número de diagramas de casos de uso empleados para cada ítem - B = Número promedio de casos de uso empleados en los diagramas de casos de uso de un ítem - C = Variación de los casos de uso con respecto a Icu	$Iin = (\sum A_i B_i) / (\sum A_i)$ $Icu = (C / Iin) * 100$	< 15%
	Modelo conceptual (métricas, control)	- A = Número de entidades del modelo conceptual - B = Número promedio de entidades modificadas	$Imc = (B / A) * 100$	< 10%
	Diagrama de secuencias (métricas, control)	- A = Número de diagramas de secuencias empleados por cada ítem - B = Número promedio de pasos de mensajes empleados en los diagramas de secuencias de un ítem - C = Variación de los diagramas de secuencias con respecto a Iin	$Iin = (\sum A_i B_i) / (\sum A_i)$ $Icu = (C / Iin) * 100$	< 15%
	Interfaces gráficas de usuario (métricas, control)	- A = Número total de interfaces gráficas diseñadas - B = Número total de interfaces gráficas modificadas	$Iig = (B / A) * 100$	< 10%
	Arquitectura de software (métricas, control)	- A = Número de componentes en el diagrama arquitectónico - B = Número promedio de componentes modificadas	$Iar = (B / A) * 100$	< 10%
	Diagrama de colaboración (métricas, control)	- A = Número de diagramas de colaboración empleados por cada ítem - B = Número promedio de pasos de mensajes empleados en los diagramas de colaboración de un ítem - C = Variación de los diagramas de colaboración con respecto a Iin	$Iin = (\sum A_i B_i) / (\sum A_i)$ $Icu = (C / Iin) * 100$	< 15%
	Diagrama de clases (métricas, control)	- A = Número de clases del diagramas de clases final - B = Número promedio de clases modificadas	$Icl = (B / A) * 100$	< 10%

Fuente: Elaboración propia

Capítulo 4: Caso de estudio

4.1 Introducción

En el presente caso de estudio se mostrará el desarrollo de una aplicación web denominada: “Estudio Jurídico Virtual”, cuyo cliente es el Dr. David F. Dongo Ortega, quien observó la necesidad de brindar sus servicios a la comunidad haciendo uso de la internet, y por medio de ella atender las consultas de la misma. Asimismo se muestra con detalle las diferentes actividades realizadas como la programación de dicha aplicación y las diferentes utilidades que la misma muestra; además se agrega todos los elementos correspondientes a la comprobación del objetivo del presente trabajo de investigación.

Los procesos de desarrollo de software que permiten elaborar el producto se encuentra especificado bajo RUP (Rational Unified Process) y el producto, en sí mismo, fue modelado bajo UML (Unified Modelling Language) respetando cada una de las etapas en su construcción.

Para llevar a cabo la comprobación del trabajo de investigación, y en vista de que no todas las casas desarrolladoras de software permitían acceder a sus productos para aplicar la propuesta de ampliación de UML con elementos para la representación de métricas y control en la fase de diseño de software, se tomó la decisión de acceder a una empresa que facilitara su catálogos de requisitos de un producto determinado. El Gerente General Roy Pérez Pinto, por medio de la empresa Smart Reasons S.R.L. tuvo la gentileza de conceder el permiso para trabajar sobre su producto “EJV – Estudio Jurídico Virtual”.

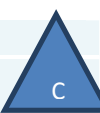
Asimismo, se tuvo que respetar las condiciones impuestas por la empresa Smart Reasons S.R.L. con respeto a la confidencialidad de la información.

4.2 Desarrollo del producto

4.2.1 Etapa de educación de requisitos

EDR

CLIENTE

ED – 001	Sección para gestionar la presentación inicial de la aplicación.			
Versión	1.0			
M/C	C: Medida B			
Autor	Grupo de Trabajo.			
Descripción	El Dr. Dongo solicita un espacio en el cual tanto él como los usuarios puedan ver sus datos profesionales entre otras secciones; de tal manera que él pueda modificar esta información cuando lo crea oportuno.			
Importancia	Vital.			
Estado	En construcción.			
ED – 002	Sección de consultas de usuarios.			
Versión	1.0			
M/C	C: Medida B			
Autor	Grupo de Trabajo.			
Descripción	Se necesita un espacio donde el Dr. David F. Dongo Ortega pueda revisar las consultas emitidas por los usuarios. Así como también, responder a dichas consultas. Las consultas deben estar clasificadas en consultas respondidas y consultas por responder. Adicionalmente se debe tener una clasificación de consultas rechazadas.			
Importancia	Vital.			
Estado	En construcción.			
ED – 003	Sección de reservaciones de citas de usuarios.			
Versión	1.0			
M/C	C: Medida B			
Autor	Grupo de Trabajo.			
Descripción	El Dr. Dongo solicita una sección donde pueda revisar y confirmar las reservaciones de citas emitidas por los usuarios.			
Importancia	Importante.			
Estado	En construcción.			
ED – 004	Sección para la gestión de usuarios.			
Versión	2.0			
M/C	C: Medida B			
Autor	Grupo de Trabajo.			
Descripción	El Dr. Dongo requiere un espacio donde pueda agregar un nuevo usuario; modificar información de usuarios existentes y eliminar un usuario. Asimismo, se puede activar y desactivar una cuenta de un usuario.			
Importancia	Vital.			
Estado	En construcción.			

ED – 005	Sección para obtener reportes.	
Versión	2.2	
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Descripción	Se necesita una sección en la cual se pueda obtener reportes de los ingresos netos, ingresos totales y los ingresos destinados al pago de la SUNAT.	
Importancia	Importante.	
Estado	En construcción.	

ED – 006	Sección de agenda personal.	
Versión	1.0	
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Descripción	Se requiere una sección en la cual el Dr. Dongo; pueda distribuir su tiempo de manera eficiente; de tal forma que pueda gestionar eventos con la ayuda de un calendario.	
Importancia	Importante.	
Estado	En construcción.	

ED – 007	Almacenamiento de información.	
Versión	1.2	 
M/C	M: Medida A, C: Medida B	
Autor	Grupo de Trabajo.	
Descripción	El Dr. Dongo desea que se almacene la información emitida por su persona; de tal forma que pueda tener acceso a la misma cuando él lo requiera.	
Importancia	Importante.	
Estado	En construcción.	

ED – 008	Aplicación amigable y fácil de usar.	
Versión	1.0	 
M/C	M: Medida A, C: Medida B	
Autor	Grupo de Trabajo.	
Descripción	El Dr. Dongo solicita una aplicación virtual que sea fácil de entender, para realizar una gestión adecuada de la aplicación.	
Importancia	Vital.	
Estado	En construcción.	

USUARIO

ED – 001	Sección para realizar consultas.	
Versión	1.2	
M/C	C: Medida B	
Autor	Grupo de Trabajo.	

Descripción	Se necesita un espacio donde los usuarios puedan realizar consultas jurídicas dirigidas al Dr. David F. Dongo Ortega con un pago previo únicamente en el caso de ser un usuario natural.	
Importancia	Vital.	
Estado	En construcción.	
ED – 002	Sección de presentación inicial.	
Versión	2.0	
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Descripción	El Dr. Dongo solicita un espacio en el cual los usuarios puedan ver sus datos profesionales y un calendario de referencia; así como también colocar información de preguntas que se le han emitido frecuentemente y noticias relacionadas al ámbito jurídico; considerando todos estos datos como sección inicial de la aplicación.	
Importancia	Vital.	
Estado	En construcción.	
ED – 003	Sección para reservar una cita.	
Versión	1.2	
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Descripción	El Dr. Dongo solicita una sección donde los usuarios puedan reservar una cita para ser atendidos personalmente.	
Importancia	Importante.	
Estado	En construcción.	
ED – 004	Almacenamiento de información.	
Versión	1.2	 
M/C	M: Medida A, C: Medida B	
Autor	Grupo de Trabajo.	
Descripción	El Dr. Dongo desea que se almacene la información emitida por los usuarios.	
Importancia	Importante.	
Estado	En construcción.	
ED – 005	Aplicación amigable y fácil de usar.	
Versión	1.2	 
M/C	M: Medida A, C: Medida B	
Autor	Grupo de Trabajo.	
Descripción	El Dr. Dongo solicita una aplicación virtual que sea fácil de usar por los usuarios; presentando colores y distribución de información adecuada.	
Importancia	Vital.	
Estado	En construcción.	

4.2.2 Etapa de elicitación de requisitos

ELR

CLIENTE

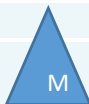
EL - 001	Editar información profesional y laboral.	
Versión	2.0	C
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	ED-001	
Descripción	El sistema deberá permitir al Dr. Dongo gestionar su información profesional y laboral en la página de presentación inicial; la cual cuenta con su fotografía y texto de sus datos informativos.	
Importancia	Vital.	
Estado	En construcción.	
EL - 002	Editar información de presentación o bienvenida de la aplicación.	
Versión	2.0	C
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	ED-001	
Descripción	El sistema deberá permitir que se pueda gestionar la presentación o mensaje de bienvenida en la página de inicio; de tal manera que dicha información se pueda mantener actualizada.	
Importancia	Vital.	
Estado	En construcción.	
EL - 003	Administrar información de preguntas frecuentes.	
Versión	2.0	C
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	ED-001	
Descripción	El sistema deberá permitir al Dr. Dongo gestionar el contenido de las preguntas frecuentes; de tal forma que pueda actualizar, agregar y eliminar.	
Importancia	Vital.	
Estado	En construcción.	
EL - 004	Administrar información de noticias de ámbito jurídico.	
Versión	2.0	C
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	ED-001	
Descripción	El sistema deberá permitir al Dr. Dongo gestionar el contenido de las noticias; de tal forma que pueda actualizar, agregar o eliminar. Asimismo, se debe tener en cuenta que para la gestión de estas noticias se deben considerar enlaces a fuentes como: "Diario el Peruano", "El Tribunal Constitucional".	
Importancia	Vital.	
Estado	En construcción.	

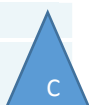
EL - 005	Administrar información del pie de página.	
Versión	2.0	
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	ED-001	
Descripción	El sistema deberá permitir al Dr. Dongo gestionar el contenido del pie de página de la aplicación; de tal forma que pueda actualizar dicha información cuando sea necesario.	
Importancia	Vital.	
Estado	En construcción.	

EL - 006	Revisar consultas.	
Versión	1.0	
M/C	M: Medida A, C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	ED-002	
Descripción	El sistema deberá clasificar las consultas de tres formas: • Consultas respondidas • Consultas por responder • Consultas rechazadas Considerando, que por defecto, al ingresar a las consultas se mostrará el listado de consultas por responder.	
Importancia	Vital.	
Estado	En construcción.	

EL - 007	Listar consultas.	
Versión	1.0	
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	ED-002	
Descripción	El sistema debe mostrar el listado de las consultas; donde, la información que se mostrará por cada una de ellas será la siguiente: • Número de operación • Fecha • Nombre de usuario • Asunto Para ver el contenido de dicha consulta se debe seleccionar en la consulta respectiva; de tal forma que se pueda ver la consulta completa considerando la información proporcionada por el usuario. Esta información debe estar completada automáticamente y en caso de ser un usuario bajo un contrato establecido debe omitirse <i>Número de operación</i>.	

Considerar un enlace al estado de cuenta del Dr. David Dongo; de tal forma que él pueda verificar el pago correspondiente de dicha consulta haciendo uso del Número de Operación.	
Importancia	Vital.
Estado	En construcción.

EL - 008	Responder una consulta.	 
Versión	1.0	
M/C	M: Medida A, C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	ED-002	
Descripción	El sistema debe permitir responder una determinada consulta, seleccionada previamente. Esta sección consta de dos partes: - Datos del usuario: ☐ Número de operación ☐ Nombre del usuario ☐ DNI del usuario ☐ Correo electrónico del Usuario ☐ Tipo de usuario - Datos de consulta: ☐ Asunto de la consulta ☐ Cuerpo de la consulta ☐ Archivos adjuntos (si es que lo hubiese) ☐ Respuestas emitidas (si es que lo hubiese) ☐ Área de texto para responder a la consulta. Se debe de tener en cuenta que en la respuesta se puede emitir archivos adjuntos, donde la respuesta es enviada a la dirección de correo electrónico del usuario. Una vez enviada dicha respuesta se debe confirmar el envío de la misma.	
	Importancia	Vital.
Estado	En construcción.	

EL - 009	Listar reservaciones de citas.	
Versión	1.0	
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	ED-003	
Descripción	El sistema debe mostrar el listado de las reservaciones de las citas; donde la información que se mostrará por cada una de ellas será la siguiente: - Fecha de emisión. - Nombre de usuario - Asunto de la reservación	

Para ver el contenido de dicha reservación se debe seleccionar la misma; de tal forma que se pueda ver los detalles completos, considerando la información proporcionada por el usuario. Esta información debe estar completada automáticamente.

Si la reservación es emitida por un usuario bajo un contrato establecido se debe agregar un cuarto atributo que indique dicha información.

Importancia Vital.
Estado En construcción.

EL – 010 Responder una reservación de una cita.

Versión 1.0

M/C C: Medida B

Autor Grupo de Trabajo.

Fuente ED-003

Descripción

El sistema debe permitir responder a una determinada reservación, seleccionada previamente.

Esta sección consta de dos partes:

- Datos del usuario:
 - Nombres completos.
 - DNI
 - Teléfono
 - Dirección
 - Correo electrónico.
 - Fecha y hora de la reservación (sugerida por el usuario).
- Datos de reservación:
 - Asunto de la reservación de la cita.
 - Área de texto en la que se responda o se confirme la reservación.

La respuesta de confirmación para la reservación es enviada a la dirección de correo electrónico del usuario.

Una vez enviada dicha respuesta se debe confirmar el envío de la misma.

Importancia Vital.
Estado En construcción.

EL – 0011 Considerar opciones para la administración de un usuario.

Versión 1.0

M/C M: Medida A, C: Medida B

Autor Grupo de Trabajo.

Fuente ED-004

Descripción

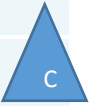
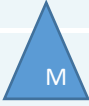
El sistema deberá tener opciones a las siguientes partes de la sección usuario:

- Agregar usuario.

	• Modificar datos de un usuario. • Eliminar un usuario. • Control de acceso de usuarios.
Importancia	Considerando por defecto la visualización de los usuarios.
Estado	Vital. En construcción.

EL - 012	Agregar un nuevo usuario.
Versión	1.0
M/C	C: Medida B
Autor	Grupo de Trabajo.
Fuente	ED-004
Descripción	Para agregar a un usuario, se debe considerar la siguiente información: • Nombres y Apellidos completos • Número de DNI • Número telefónico • Dirección • Correo electrónico • Id de usuario • Password Una vez ingresado estos datos se debe registrar al usuario y mostrar un mensaje de confirmación.
Importancia	Vital.
Estado	En construcción.

EL – 013	Modificar datos de un usuario.
Versión	1.0
M/C	C: Medida B
Autor	Grupo de Trabajo.
Fuente	ED-004
Descripción	El sistema debe permitir modificar los siguientes datos de un determinado usuario de acuerdo a la elección del cliente:
	• Nombres y Apellidos completos • Número telefónico • Dirección • Correo electrónico • Id de usuario • Password
	Una vez ingresado estos los nuevos datos se debe actualizar esta información y mostrar un mensaje de confirmación.
Importancia	Vital.

Estado	En construcción.	
EL - 014	Eliminar un usuario.	
Versión	10	
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	ED-004	
Descripción	El sistema debe permitir listar todos los usuarios, y realizar una búsqueda de los mismos; con la finalidad de eliminarlo.	
Importancia	Vital.	
Estado	En construcción.	
EL - 015	Administrar acceso de usuarios.	
Versión	10	
M/C	M: Medida A, C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	ED-004	
Descripción	El sistema debe permitir habilitar y deshabilitar el ingreso de un usuario a la aplicación como usuario registrado. Esta opción se llevará a cabo de acuerdo a reglas de negocio del Dr. Dongo.	
Importancia	Vital.	
Estado	En construcción.	
EL - 016	Considerar opciones de obtención de reportes.	
Versión	10	
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	ED-005	
Descripción	El sistema debe tener una sección de reportes; en la cual se pueda tener las siguientes opciones para la obtención de reportes: • Reportes de ingresos netos. • Reportes de ingresos totales. • Reportes de ingresos destinados al pago de la SUNAT.	
Importancia	Considerando por defecto la visualización de ingresos totales.	
Estado	Vital. En construcción.	
EL - 017	Obtener reportes de ingresos netos.	
Versión	10	
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	ED-005	

Descripción	El sistema debe mostrar los ingresos netos obtenidos de acuerdo a los siguientes criterios: • Mensuales. • De una fecha inicial hasta una fecha final.
Importancia	Vital.
Estado	En construcción.

EL - 018	Obtener reportes de ingresos totales.	
Versión	1.0	
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	ED-005	
Descripción	El sistema debe mostrar los ingresos totales obtenidos de acuerdo a los siguientes criterios: • Mensuales. • De una fecha inicial hasta una fecha final.	
Importancia	Vital.	
Estado	En construcción.	



EL - 019	Obtener reportes de ingresos destinados al pago de la SUNAT.	
Versión	1.0	
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	ED-005	
Descripción	El sistema debe mostrar los ingresos destinados al pago de la SUNAT de acuerdo a los siguientes criterios: • Mensuales. • De una fecha inicial hasta una fecha final.	
Importancia	Vital.	
Estado	En construcción.	



EL - 020	Administrar información de la agenda personal.	
Versión	1.0	
M/C	M: Medida A, C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	ED-006	
Descripción	El sistema debe tener una sección denominada agenda, en la cual se pueda acceder al listado de los eventos que se tienen. Asimismo se debe considerar las siguientes opciones a realizar: • Agregar un evento. • Modificar un evento.	



Importancia	Eliminar un evento.	
Estado	Vital. En construcción.	
EL - 021	Almacenar información de la aplicación.	
Versión	1.0	M
M/C	M: Medida A	
Autor	Grupo de Trabajo.	
Fuente	ED-007	
Descripción	El sistema deberá contar con una base de datos que almacene de manera estructurada toda la información relevante de la aplicación.	
Importancia	Vital.	
Estado	En construcción.	
EL - 022	Considerar una aplicación virtual con colores adecuados para el cliente.	
Versión	1.0	M
M/C	M: Medida A	
Autor	Grupo de Trabajo.	
Fuente	ED-008	
Descripción	El sistema deberá tener una aplicación virtual con colores adecuados para cada sección de la aplicación; de tal manera que el cliente se sienta cómodo.	
Importancia	Vital.	
Estado	En construcción.	
EL - 023	Considerar nombres apropiados y descriptivos para los menús y enunciados de la aplicación.	
Versión	1.0	C
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	ED-008	
Descripción	El sistema deberá tener una aplicación virtual fácil de utilizar; para ello se debe tener en cuenta el uso de nombres descriptivos y adecuados para cada sección; de tal manera que orienten al cliente.	
Importancia	Vital.	
Estado	En construcción.	
EL - 024	Tener secciones para la estructura de la aplicación.	
Versión	1.0	C
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	ED-008	

Descripción	La aplicación deberá tener diferentes secciones para tener una estructura definida de la aplicación; de tal manera que cada sección realice una determinada funcionalidad. Deberá consignar las siguientes secciones: • Cabecera • Menú • Contenido • Pie de página
Importancia	Vital.
Estado	En construcción.

EL - 025	Tener una sección de ayuda para el cliente.	C
Versión	1.0	
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	ED-008	
Descripción	La aplicación debe tener una sección denominada "Ayuda" en la cual el cliente pueda tener acceso a información que lo oriente en el uso de la aplicación.	
Importancia	Vital.	
Estado	En construcción.	

USUARIO

EL - 001	Enviar una consulta de un usuario.	C
Versión	1.0	
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	ED-001	
Descripción	El sistema deberá permitir que un usuario pueda realizar el envío de una consulta respecto a un determinado caso jurídico. Para ello se debe llenar un formulario con la siguiente información dependiendo sea el caso: Usuario natural: • Código o número del recibo de pago para realizar la consulta. (campo obligatorio a llenar) • Nombres y Apellidos completos • Número de DNI • Número telefónico • Dirección • Correo electrónico (campo obligatorio a llenar) Usuario bajo un contrato: • Iniciar Sesión previamente.	

Finalmente, para ambos casos se debe ingresar la información de la consulta; cuya estructura está definida mediante un asunto y cuerpo de consulta. Asimismo, se podrá adjuntar archivos si el usuario lo desea.

Importancia Vital.
Estado En construcción.

EL – 002	Confirmar el envío de la consulta por parte del usuario.	
Versión	2.0	C
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	ED-001	
Descripción	La aplicación deberá emitir un mensaje que confirme si el envío de la consulta fue satisfactoria o no satisfactoria.	
Importancia	Vital.	
Estado	En construcción.	

EL-003	Considerar información profesional por parte del Dr. Dongo en la sección de presentación inicial.		
Versión	2.0		
M/C	C: Medida B		
Autor	Grupo de Trabajo.		
Fuente	ED-002		
Descripción	El sistema deberá permitir a los usuarios ver la información profesional y laboral del Dr. David F. Dongo Ortega en la página de presentación inicial. La estructura de la información consta de la fotografía del Dr. Dongo y texto de sus datos.		
Importancia	Vital.		
Estado	En construcción.		

EL - 004	Colocar un calendario de referencia para los usuarios en la sección de presentación inicial.		C
Versión	2.0		
M/C	C: Medida B		
Autor	Grupo de Trabajo.		
Fuente	ED-002		
Descripción	El sistema deberá permitir a los usuarios tener acceso a un calendario que le indique tanto la fecha actual como fechas previas. El calendario deberá estar ubicado en la sección de inicio.		
Importancia	Vital.		
Estado	En construcción.		

EL - 005	Considerar información de preguntas frecuentes en la sección de presentación inicial.	C
Versión	2.0	
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	ED-002	
Descripción	El sistema deberá permitir a los usuarios ver las preguntas que se han realizado con mayor frecuencia al Dr. Dongo; de tal forma que aclare algunas dudas de los usuarios sin necesidad de realizar una consulta. Esta información deberá estar ubicada en la sección de inicio.	
Importancia	Vital.	
Estado	En construcción.	

EL - 006	Considerar noticias del ámbito jurídico en la sección de presentación inicial.	C
Versión	2.0	
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	ED-002	
Descripción	El sistema deberá permitir a los usuarios ver noticias relacionadas al ámbito jurídico que el Dr. Dongo considere. Las noticias deberán ser de fuentes como: Diario El Peruano, El Tribunal Constitucional; entre otros. Las noticias deberán estar ubicadas en la sección de inicio.	
Importancia	Vital.	
Estado	En construcción.	

EL - 007	Reservar citas.	C
Versión	2.0	
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	ED-003	
Descripción	El sistema deberá permitir a los usuarios acceder a una sección que les permita reservar una cita personal. Para reservar dicha cita se debe consignar la siguiente información: • Nombres completos. • Documento Nacional de Identidad • Teléfono • Dirección • Correo electrónico. • Fecha y hora sugeridas; de acuerdo al horario de atención establecido por parte del Dr. Dongo. • Asunto de la reservación de la cita.	
Importancia	Vital.	
Estado	En construcción.	

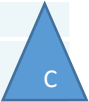
EL - 008	Almacenar información de la aplicación.		
Versión	1.0		
M/C	M: Medida A, C: Medida B	M	C
Autor	Grupo de Trabajo.		
Fuente	ED-004		
Descripción	El sistema deberá contar con una base de datos que almacene de manera estructurada toda la información relevante de la aplicación.		
Importancia	Vital.		
Estado	En construcción.		

EL - 009	Considerar una aplicación virtual con colores adecuados para los usuarios.		
Versión	1.0		
M/C	M: Medida A, C: Medida B	M	C
Autor	Grupo de Trabajo.		
Fuente	ED-005		
Descripción	El sistema deberá tener una aplicación virtual con colores adecuados para cada sección de la aplicación; de tal manera que los usuarios se sientan cómodos.		
Importancia	Vital.		
Estado	En construcción.		

EL - 010	Considerar nombres apropiados y descriptivos para los menús y enunciados de la aplicación.		
Versión	1.0		
M/C	C: Medida B		C
Autor	Grupo de Trabajo.		
Fuente	ED-005		
Descripción	El sistema deberá tener una aplicación virtual fácil de utilizar; para ello se debe tener en cuenta el uso de nombres descriptivos y adecuados para cada sección; de tal manera que orienten al usuario.		
Importancia	Vital.		
Estado	En construcción.		

EL - 011	Tener secciones para la estructura de la aplicación.		
Versión	1.0		
M/C	C: Medida B		C
Autor	Grupo de Trabajo.		
Fuente	ED-005		
Descripción	El sistema deberá tener diferentes secciones para tener una estructura definida de la aplicación; de tal manera que cada sección realice una determinada funcionalidad. Deberá consignar las siguientes secciones: • Cabecera • Menú • Contenido		

Importancia	• Pie de página
Estado	Vital. En construcción.

EL-012	Tener una sección de ayuda para el usuario	
Versión	1.0	
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	ED-005	
Descripción	La aplicación debe tener una sección denominada “Ayuda” en la cual el usuario pueda tener acceso a información que lo oriente en el uso de la aplicación.	
Importancia	Vital.	
Estado	En construcción.	

4.2.3 Etapa de especificación de requisitos de software

ERS

CLIENTE

ERS-001	Edición de información profesional y laboral.	
Versión	1.01	
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	EL-001	
Descripción	Para editar la información de esta sección, se debe cargar los datos que se encuentran en la base de datos; de tal forma que se pueda seleccionar un campo específico. Si el campo seleccionado es el texto y este se modifica se activa un evento onchange que llama a la función editarAtributo() que altera la tabla consultorio_datos con los datos modificados. Si el campo seleccionado es la fotografía, se debe abrir una ventana emergente para seleccionar la nueva imagen; una vez seleccionada, la ruta de la imagen se debe guardar en la base de datos.	
Importancia	Vital.	
Estado	En construcción.	

ERS-002	Edición de información de presentación o bienvenida de la aplicación.	
Versión	1.01	
M/C	C: Medida B	

Autor	Grupo de Trabajo.
Fuente	EL-002
Descripción	Para editar la información de esta sección, se debe cargar los datos que se encuentran en la base de datos; de tal forma que se pueda seleccionar un campo específico. - Si se modifica el título o el cuerpo de se activa un evento onchange que llama a la función editarBienvenida() que altera la tabla consultorio_bienvenida con los datos modificados. - Al seleccionar la imagen de bienvenida, se debe abrir una ventana emergente para cargar la nueva imagen; una vez seleccionada, la ruta de la imagen se debe guardar en la base de datos.
Importancia	Vital.
Estado	En construcción.

ERS-003	Edición de información de preguntas frecuentes.
Versión	1.01
M/C	C: Medida B
Autor	Grupo de Trabajo.
Fuente	EL-003
Descripción	Las preguntas frecuentes son mostradas con la etiqueta <input type = text> para poder escribir nuevamente sobre el campo. Si se hace una modificación en su contenido, se activa el evento onchange que llama a la función editarPregunta(valorAnterior, 87ndice); la cual tiene dos parámetros: - valorAnterior: tiene guardado el valor del campo antes de editarlo. - Índice: indica el número de la pregunta que se está editando. La función editarPregunta verificará que al editar el campo; este no se encuentre vacío. Si está vacío enviará un mensaje de error siguiente: "El campo de la pregunta no debe quedar vacío"; y luego vuelve a escribir en el campo el valor anterior. Si el campo no está vacío, se envía el valor del campo editado para actualizar la base de datos.
Importancia	Vital.
Estado	En construcción.

ERS-004	Editar pie de página
Versión	1.01
M/C	C: Medida B
Autor	Grupo de Trabajo.
Fuente	EL-005
Descripción	Para editar la información del pie de página se hace click de alguno de los campos (teléfono, dirección, email); si se edita se llama al evento onchange que llamará a la función editarPiePagina() que modificará la base de datos con el dato cambiado.

Importancia	Vital.
Estado	En construcción.

ERS-005	Estados de las consultas	
Versión	101	
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	EL-006	
Descripción	En la base de datos, las consultas deben guardarse con un estado representado por el atributo ConEstado. Cuando se envían las consultas se guardan con el valor 0, que indica "Consulta sin responder". Cuando se responde a una consulta ConEstado será igual a 1. Si se determina que la consulta no amerita respuesta por algún incumplimiento, el ConEstado será igual a 2.	
Importancia	Vital.	
Estado	En construcción.	

ERS-006	Menú de Consultas	
Versión	101	
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	EL-006	
Descripción	En la página admin/consulta/index.php habrá un menú con 3 botones: "No respondidas", "Respondidas", "Rechazadas". Los 3 poseen una referencia(<a>) a la página, mandando en el url el atributo ConEstado con los valores 0,1 y 2 respectivamente (por ejemplo: index.php?ConEstado=1)	
Importancia	Vital.	
Estado	En construcción.	

ERS-007	Menú de Consultas	
Versión	101	
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	EL-006	
Descripción	La página admin/consulta/index.php recogerá con el método recogerGet('ConEstado'), los valores pasados en el url al presionar algún botón del menú de consultas. Selecciona de la base de datos la tabla consultorio_consulta donde ConEstado sea igual al recogido del url. Mostrará las consultas no respondidas, respondidas y rechazadas si se le mandó los valores 0,1 y 2 respectivamente.	
Importancia	Vital.	
Estado	En construcción.	

ERS-008		Tabla de Consultas		
Versión	1.01			
M/C	C: Medida B			
Autor	Grupo de Trabajo.			
Fuente	EL-007			
Descripción	La página admin/consulta/index.php mostrará las consultas en una (<table></table>). Esta tabla tendrá las cabeceras N°, Número de Transacción, Usuario, Asunto, Fecha y Detalles. Los primeros 5 campos mostrará la información respectiva por cada consulta lo cual se hace usando: \$records = mysql_query("SELECT * FROM consultorio_consultas where ConEstado=\$conEstado"). En la columna de Detalles, se tiene en cada consulta una referencia a la página responderConsultas.php; se pasa como referencia en el url el valor del id de la consulta para saber cuál de todas hay que responder.			
Importancia	Vital.			
Estado	En construcción.			
ERS-009		Enlace a la página del banco. Interbank		
Versión	1.01		 	
M/C	M: Medida A, C: Medida B			
Autor	Grupo de Trabajo.			
Fuente	EL-007			
Descripción	En detalles de la consulta; hay una imagen que tiene un enlace hacia la página del banco Interbank: https://www.netinterbank.com.pe/web-interbank/interbanking/newWelcome.jsp Este enlace ayudará al cliente a la verificación del número de operación emitida por el usuario. A fin de validar la correctitud de la consulta emitida.			
Importancia	Vital.			
Estado	En construcción.			
ERS-010		Visualizar Datos completos de la Consulta		
Versión	1.01			
M/C	C: Medida B			
Autor	Grupo de Trabajo.			
Fuente	EL-008			
Descripción	Se visualiza los datos de la consulta (datos del usuario que la realizó y de la consulta misma). Se incluye el archivo plantillaUsuario.php y plantillaConsulta.php mediante: include("../plantillaConsulta.php"); include("../plantillaUsuario.php"); Se llama a los métodos imprimirUsuarioLleno(\$dniUsuario) e imprimirConsultaVacía(\$idConsulta) de estos archivos respectivamente; los cuales generará los datos de usuario y consultas al haberles pasado el dni de usuario y el id de consulta para generar la información a mostrar.			
Importancia	Vital.			

Estado	En construcción.
ERS-011	Responder Consulta
Versión	1.01
M/C	C: Medida B
Autor	Grupo de Trabajo.
Fuente	EL-008
Descripción	Al presionar sobre un botón “Responder consulta”, hará visible un cuadro de texto (<textarea></textarea>) donde se podrá responder a la consulta. Aparecerá una opción para adjuntar archivos (<input type=“file”>). Se enviará la respuesta al correo del usuario al presionar el botón “Enviar”. El correo se envía usando la clase phpMailer.php. Se guardará en la base de datos las respuestas de las consultas y los archivos adjuntos de éstas. Se actualiza el estado de la consulta a 1.
Importancia	Vital.
Estado	En construcción.
ERS-012	Visualizar respuestas de consultas
Versión	1.01
M/C	C: Medida B
Autor	Grupo de Trabajo.
Fuente	EL-008
Descripción	Si el estado de la consulta es 1, al ir a detalles de la consulta, se cargará la tabla “consultorio_respuestas” y posteriormente a la tabla “consultorio_adjuntosrespuestas” que tenga el id de las respuestas a las preguntas. <pre>\$sql1 = mysql_query("SELECT * FROM consultorio_adjuntosrespuestas WHERE IdRespuesta = '\$idRespuesta'");</pre> Después se visualizan los archivos enviados en la respuesta Si no hay respuestas a la consulta se imprimirá en pantalla “No se ha respondido a esta consulta”.
Importancia	Vital.
Estado	En construcción.
ERS-013	Tabla de Citas
Versión	1.01
M/C	C: Medida B
Autor	Grupo de Trabajo.
Fuente	EL-009
Descripción	Se tendrá una tabla (<table></table>) de 5 columnas: número, usuario, asunto, fecha de emisión, detalles.

	Los datos de la tabla se carga llamando a la base de datos con la siguiente consulta: \$records = mysql_query("SELECT * FROM consultorio_cita"). En la columna de detalles se tendrá enlaces a la página de confirmación de citas pasando en el url el valor del id de la cita: Detalles
Importancia	Vital.
Estado	En construcción.

ERS-014	Confirmar Cita	
Versión	1.01	
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	EL-010	
Descripción	Se visualizará la información de la cita llenada por el usuario, donde el campo de fecha de reservación como activado para modificar. Al hacer click llama al evento onclick y a la función javascript "mostrarCalendario()", luego se selecciona una fecha para la cita en caso de que no se esté de acuerdo con la establecida por el usuario. En otro campo se indicará la hora de la cita y la hora en que ésta se acaba. Al presionar el botón enviar actualiza la tabla "consultorio_citas" reemplazando los valores de CitFecha, CitInicio y CitFinal. Se cambia el estado de la cita a 1 para decir que se ha confirmado. Se envía un correo electrónico al usuario utilizando la clase php.mailer.	
Importancia	Vital.	
Estado	En construcción.	

ERS-015	Gestión de Usuarios	
Versión	1.01	
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	EL-011	
Descripción	Se gestiona al usuario de maneras diferentes en admin/usuarios/index.php; para acceder a estas opciones se realiza lo siguiente: Insertar usuario: Hacer click en el botón "Registrar Usuario", envía a la página registroUsuario.php. Modificar Usuario: Hacer click en "Detalles" en la fila de la tabla que contiene al usuario que hay que modificar. Envía a la página gestionarUsuario.php. Eliminar Usuario: En la página gestionarUsuario.php al hacer click sobre el botón "Eliminar Usuario".	
Importancia	Vital.	
Estado	En construcción.	

ERS-016	Formulario de Usuario	
----------------	------------------------------	--

Versión	101
M/C	C: Medida B
Autor	Grupo de Trabajo.
Fuente	EL-012
Descripción	En la página registroUsuario.php se encuentra el formulario “registroUsuario” con los campos D.N.I., Nombres, Apellidos, Teléfono, Email, Password; todos son <code><input /></code> con atributo <code>type=“text”</code> a excepción de Password que es <code>type=“password”</code> . Al final del formulario hay un botón “SUBMIT” que al ser presionado activa un evento onclick que llama a la función enviarFormulario().
Importancia	Vital.
Estado	En construcción.

ERS-017	Función enviarFormulario()	
Versión	101	
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	EL-012	
Descripción	La función enviarFormulario llamará a la función setErrorUsuario() para marcar todos los campos de formulario que presenten errores. Si no hay errores enviará un mensaje de confirmación. Si se acepta, se enviará el formulario y si no, se cancela la operación.	
Importancia	Vital.	
Estado	En construcción.	

ERS-018	Registrar nuevo usuario	
Versión	101	
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	EL-012	
Descripción	La función enviarFormulario llamará a la función setErrorUsuario() para marcar todos los campos de formulario que presenten errores. Si no hay errores enviará un mensaje de confirmación. Si se acepta, se enviará el formulario y si no, se cancela la operación.	
Importancia	Vital.	
Estado	En construcción.	

ERS-019	Recoger datos de un determinado formulario.	
Versión	102	
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	EL-021	
Descripción	Después de llenar un respectivo formulario, se presiona el botón correspondiente a dicho formulario, los datos del formulario se enviarán a otra página y se almacenarán utilizando variables como por ejemplo: <code>\$variable1 = \$_GET['nom_variable'];</code>	
Importancia	Vital.	

Estado	En construcción.
ERS-020	Editar usuario
Versión	1.01
M/C	C: Medida B
Autor	Grupo de Trabajo.
Fuente	EL-013
Descripción	Para editar un usuario en la página <code>gestionUsuario.php</code> , se mostrará el formulario de registro con los campos llenos con los valores actuales del usuario. Si se modifican se activa el evento <code>onchange</code> que llamara a la función <code>editarUsuario()</code> que enviará el formulario indicando una edición y se actualiza la tabla de usuarios.
Importancia	Vital.
Estado	En construcción.
ERS-021	Eliminar Usuario
Versión	1.01
M/C	C: Medida B
Autor	Grupo de Trabajo.
Fuente	EL-014
Descripción	En la página <code>gestionUsuario.php</code> al hacer click sobre el botón “Eliminar Usuario” se activará un evento <code>onclick</code> que llamara a la función <code>eliminarUsuario</code> que enviará el dni del usuario para que borre de la base de datos el usuario que presenta ese dni.
Importancia	Vital.
Estado	En construcción.
ERS-022	Estructura de la sección de reportes.
Versión	1.01
M/C	C: Medida B
Autor	Grupo de Trabajo.
Fuente	EL-016
Descripción	Para la estructura de la sección reportes se considera crear un menú con los siguientes botones: • Reporte de ingresos netos • Reporte de ingresos totales • Reporte de SUNAT Cada uno de los botones enlazan a su respectivo reporte mediante una referencia: <code>href="nombre-ubicación"</code> . La ubicación del menú se encontrará al lado derecho.
Importancia	Vital.
Estado	En construcción.
ERS-023	Reportes en formato PDF
Versión	1.02
M/C	C: Medida B
Autor	Grupo de Trabajo.

Fuente	EL-016
Descripción	Una vez que se obtuvo los datos deseados el cliente tiene la posibilidad de guardar los datos en formato pdf, esto se logra mediante la librería jsPDF.
Importancia	Vital.
Estado	En construcción.

ERS-024	Etiqueta div con ID para el reporte.	
Versión	1.02	
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	EL-016	
Descripción	La forma más fácil de saber cuál es la información que se guardara en el reporte PDF es ponerle un identificador a la etiqueta <div> que contendrá los datos que queremos visualizar en el reporte.	
Importancia	Vital.	
Estado	En construcción.	

ERS-025	Reporte de Ingresos Netos.	
Versión	1.02	
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	EL-017	
Descripción	Después de haber seleccionado el menú de reportes y haber seleccionado la opción de ingresos netos, se mostrará la página ingresos_netos.php.	
Importancia	Vital.	
Estado	En construcción.	

ERS-026	Obtención Ingresos Netos.	
Versión	1.02	
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	EL-017	
Descripción	El reporte de ingresos netos se podrá mostrar gracias al uso de consultas sobre la base de datos, el cliente especificara los ingresos que desea visualizar seleccionando el mes y el año que desea de una lista de Combobox.	
Importancia	Vital.	
Estado	En construcción.	

ERS-027	Uso de Variables POST en Ingresos Netos.	
Versión	1.02	
M/C	C: Medida B	
Autor	Grupo de Trabajo.	

Fuente	EL-017
Descripción	Para poder saber qué mes y que año el cliente selecciono se hará uso de variables POST que serán obtenidas una vez el cliente seleccione los datos y los envíe presionando el botón Ver Ingresos.
Importancia	Vital.
Estado	En construcción.
ERS-031	Reporte de Ingresos Totales.
Versión	1.02
Autor	Grupo de Trabajo.
Fuente	EL-018
Descripción	Después de haber seleccionado el menú de reportes y haber seleccionado la opción de ingresos totales, se mostrará la página ingresos_totales.php.
Importancia	Vital.
Estado	En construcción.
ERS-028	Obtención Ingresos Totales
Versión	1.02
M/C	C: Medida B
Autor	Grupo de Trabajo.
Fuente	EL-018
Descripción	El reporte de ingresos netos se podrá mostrar gracias al uso de consultas sobre la base de datos, el cliente especificara los ingresos que desea visualizar seleccionando el mes y el año que desea de un combobox.
Importancia	Vital.
Estado	En construcción.
ERS-029	Uso de Variables POST en Ingresos Totales.
Versión	1.02
M/C	C: Medida B
Autor	Grupo de Trabajo.
Fuente	EL-018
Descripción	Para poder saber qué mes y que año el cliente selecciono se hará uso de variables POST que serán obtenidas una vez el cliente seleccione los datos y los envíe presionando el botón Ver Ingresos.
Importancia	Vital.
Estado	En construcción.
ERS-030	Impuesto a la SUNAT
Versión	1.02
M/C	C: Medida B
Autor	Grupo de Trabajo.

Fuente	EL-019
	La página web será capaz de calcular el impuesto de acuerdo a la suma total de todos las consultas establecidas en el mismo mes.
	De acuerdo a la Resolución de Superintendencia de la SUNAT N° 373-2013. Los trabajadores independientes (Cuarta Categoría) que emiten recibos por honorarios y perciban ingresos mensuales que no superen los S/.2,771 no estarán afectos a la retención y/o pagos a cuenta del Impuesto a la Renta (IR).
	Caso contrario los contribuyentes deberán declarar y efectuar el pago a cuenta del Impuesto a la Renta por la totalidad de los ingresos de Cuarta Categoría que obtenga en el referido periodo.
	En este caso dicha persona efectuará la retención del 10% del total de los recibos emitidos tanto por superar los S/.2,771 como también por montos que excedan los S/.1500 nuevos soles.
Descripción	Por tanto el sistema constara de las siguientes sentencias: La función se encuentra en la carpeta admin/reportes/reportesSUNAT en la línea de código 161 <pre> if (\$sumatotal > 1500) { \$sunat = (\$sumatotal * 10) / 100; if (\$sumatotal > 2771) { echo ('<tr>'); echo " <td>Declaracion</td><td>a la</td><td>SUNAT </td><td>\$sunat</td>"; echo ('</tr>'); } else { echo ('<tr>'); echo " <td>---</td><td>---</td><td>--- </td><td>\$sunat</td>"; echo ('</tr>'); } } </pre>
Importancia	Vital.
Estado	En construcción.

ERS-031	Reportes de ingresos destinados a la SUNAT
Versión	1.01
M/C	C: Medida B
Autor	Grupo de Trabajo.
Fuente	EL-019
Descripción	Para obtener los ingresos destinados a la SUNAT; se debe seleccionar la fecha desde un ComboBox, posteriormente se debe hacer click en el botón aceptar; de tal manera que se activa un evento para que se muestre una tabla con la información respectiva.

	La tabla tiene los siguientes atributos: • Id de la consulta • DNI del usuario • Nombre de usuario • Apellidos del usuario • Monto de pago • Porcentaje destinado a la SUNAT. Una vez creada la tabla, se crea un botón llamado “Exportar tabla” el cual exportará un pdf; para ello se hace uso de la función tableExport() hecha en java script.
Importancia	Vital.
Estado	En construcción.
ERS-032	Guardar información.
Versión	102
M/C	C: Medida B
Autor	Grupo de Trabajo.
Fuente	EL-021
Descripción	Después de haber llenado un formulario, se toman los datos del formulario y se guardan en una determinada tabla que se encuentra dentro de la Base de Datos “Estudio_Juridico_Virtual”. Para ello se hará la consulta de inserción en la tabla, por ejemplo: <code>\$sql=mysql_query("INSERT INTO `Consultas` VALUES (\$nombres, '\$apellidos', '\$dni', '\$telefono', '\$direccion', '\$email', '\$asunto', '\$cuerpo');");</code>
Importancia	Vital.
Estado	En construcción.
ERS-033	Almacenar la información en una base de datos en MySQL
Versión	102
M/C	M: Medida A, C: Medida B
Autor	Grupo de Trabajo.
Fuente	EL-021
Descripción	Se creará una base de datos “Estudio_Juridico_Virtual” utilizando MySQL con las tablas correspondientes. La conexión con la base de datos se hará usando: <code>@mysql_connect("localhost","root","")</code> <code>@mysql_select_db("Estudio_Juridico_Virtual")</code>
Importancia	Vital.
Estado	En construcción.
ERS-034	Colores de página web.
Versión	102

M/C	C: Medida B
Autor	Grupo de Trabajo.
Fuente	EL-022
Descripción	La página web tendrá los siguientes colores de acuerdo a la estructura: • Cabecera: color celeste degradado. • Menú: color negro degradado. • Contenido: color blanco. Los colores se pueden modificar si el cliente así lo dispone.
Importancia	Vital.
Estado	En construcción.

ERS-035	Menú
Versión	102
M/C	C: Medida B
Autor	Grupo de Trabajo.
Fuente	EL-023
Descripción	El sistema contara con un menú con los siguientes nombres, los cuales intentan describir de la mejor manera su contenido, el menú será desarrollado con una lista, la estructura y nombres serán los siguientes: • <code>InicioRealizar ConsultaReservar CitaAyuda</code> Cada lista hará una referencia a una página correspondiente a cada sección.
Importancia	Vital.
Estado	En construcción.

ERS-036	Secciones para cada menú
Versión	102
M/C	C: Medida B
Autor	Grupo de Trabajo.
Fuente	EL-023
Descripción	El proyecto de la aplicación estará dividido por carpetas para cada menú, a excepción del inicio, el cual será nuestro index principal de la página. La estructura se muestra de la siguiente forma • Proyecto -Ayuda (Carpeta) -RealizarConsulta (Carpeta) -ReservarCita (Carpeta) -index.php (Archivo) Cada carpeta tendrá dentro un archivo index.php referente a la información de su carpeta correspondiente.
Importancia	Vital.

Estado	En construcción.
ERS-037	Estructura de la página
Versión	102
M/C	C: Medida B
Autor	Grupo de Trabajo.
Fuente	EL-024
Descripción	La aplicación estará formada por la siguiente estructura utilizando divisiones: • División que contiene a toda la aplicación. ○ División para la cabecera. ▪ División para el logo ▪ División para el título de la aplicación ○ División para el menú. ○ División para el contenido. ○ División para el pie de página Para las divisiones se utilizara la etiqueta <code><div></div></code>, como se ve lo siguiente: <pre><div id=página> <div id="divCabecera"> <div id="divLogo"></div> <div id="divTitulo"></div> </div> <div id="divMenu"></div> <div id="divContenido"></div> <div id="divPie"></div> </div></pre>
Importancia	Vital.
Estado	En construcción.
ERS-038	Sección ayuda
Versión	102
M/C	C: Medida B
Autor	Grupo de Trabajo.
Fuente	EL-025
Descripción	En esta sección se mostrará la ayuda en un pdf incrustado en la página; para ello se utiliza la siguiente estructura: <pre><object type="application/pdf" data="../documentos/MANUAL DE USUARIO.pdf" style="width: 100%; height: 350px;"> ERROR (no puede mostrarse el objeto) </object></pre>
Importancia	Vital.
Estado	En construcción.

USUARIO

ERS-001 Características del formulario de Consulta.		C
Versión	1.01	
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	EL-001	
Descripción	Para crear los formularios de consulta se utilizará código html; empleando las etiquetas <code><form></code> y <code><table></code> para el diseño. Todos los campos del formulario a excepción del cuerpo de la consulta se realizarán en cajas de texto de una línea, las cuales son creadas dentro de la tabla del formulario con la etiqueta <code><input type="text"></code>. El cuerpo de la consulta se hará usando una caja de texto multilinea mediante la etiqueta <code><textarea></textarea></code>. Se podrá adjuntar un documento del disco duro y enviarlo utilizando el control <code><input type="file" /></code> Se contará con un botón "Enviar Consulta" para enviar los datos del formulario, para ello se usa la etiqueta <code><input type="submit" value="Enviar"></code>. El formulario debe definirse de la siguiente forma: <code>form<method="get" action="consulta.php"></code>, para que los valores de los campos de formulario se vean por la barra de direcciones y sean enviados al archivo consulta.php.	
Importancia	Vital.	
Estado	En construcción.	
ERS-002 Formulario para una consulta de un usuario natural.		C
Versión	2.0	
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	EL-001	
Descripción	El formulario para que un usuario natural pueda realizar una consulta consta de la siguiente estructura: Debe estar contenido en la estructura: <code><form> </form></code> Etiquetas con los nombres siguientes: • Número de recibo de pago por concepto de la consulta a realizar. • Nombres • Apellidos • DNI • Correo electrónico • Teléfono • Dirección • Asunto de consulta	

- Cuerpo de consulta

Cada etiqueta deberá estar asociado a un elemento de entrada o caja de texto (definido mediante input), tal como sigue:

- Número de recibo: campo obligatorio a ser llenado. Asimismo, este campo se debe repetir una vez más para mayor seguridad.
- Nombres:
- Apellidos
- DNI
- Correo electrónico: campo obligatorio a ser llenado.
- Teléfono
- Dirección
- Asunto de consulta: campo obligatorio a ser llenado.
- Cuerpo de consulta: campo obligatorio a ser llenado.

Al final del formulario debe haber un botón con el nombre *“Enviar consulta”* que al ser seleccionado envíe la consulta solo si se ha llenado los campos requeridos obligatoriamente.

Importancia Vital.
Estado En construcción.

ERS-003 Formulario para inicio de sesión de un usuario bajo un contrato establecido.

Versión 2.0
M/C C: Medida B
Autor Grupo de Trabajo.
Fuente EL-001

Descripción

El formulario para que un usuario con un contrato definido pueda realizar una consulta primero debe de iniciar sesión. El formulario de inicio de sesión debe tener las siguientes características:

Estar contenido en la estructura: `<form> </form>`

Etiquetas con los nombres siguientes:

- Usuario
- Contraseña

Cada etiqueta deberá estar asociada a un elemento de entrada correspondiente; donde, cada elemento de entrada debe llenarse obligatoriamente y correctamente para el inicio de sesión.

Al final del formulario debe haber un botón con el nombre *“Iniciar Sesión”* que al ser seleccionado permita el inicio de sesión del usuario solo si se ha llenado los campos requeridos obligatoriamente.

Importancia Vital.
Estado En construcción.

ERS-004 Formulario para una consulta de un usuario bajo un contrato establecido.

C

Versión	2.0
M/C	C: Medida B
Autor	Grupo de Trabajo.
Fuente	EL-001
Descripción	El formulario para un usuario que tiene un contrato definido pueda realizar una consulta consta de la siguiente estructura: Debe estar contenido en la estructura: <code><form> </form></code> Etiquetas con los nombres siguientes: • Nombre • Correo electrónico Cada etiqueta deberá estar asociada a un elemento de entrada correspondiente; donde, cada elemento de entrada mencionado anteriormente debe llenarse automáticamente. Al final del formulario debe haber un botón con el nombre “<i>Enviar consulta</i>” que al ser seleccionado envíe la consulta solo si se ha llenado los campos requeridos obligatoriamente.
Importancia	Vital.
Estado	En construcción.

ERS-005	Enviar notificación al correo electrónico del Dr. Dongo.	
Versión	1.02	
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	EL-002	
Descripción	Después de guardar la consulta en la base de datos se enviará un correo electrónico al cliente mandándole la información del remitente, el asunto y cuerpo de la consulta y los documentos que se han adjuntado. Para ello se utilizará la función mail: • mail(\$para,\$asunto,\$mensaje) La variable <i>\$para</i> indica el correo del cliente; la variable <i>\$asunto</i> indicará: Nueva consulta del Estudio Jurídico Virtual; y el <i>\$mensaje</i> indicará la información del usuario y la consulta.	
Importancia	Vital.	
Estado	En construcción.	

ERS – 006	Mensaje de confirmación del envío de la consulta.	
Versión	2.0	

M/C	C: Medida B
Autor	Grupo de Trabajo.
Fuente	EL-002
Descripción	El sistema debe mostrar un mensaje que confirme el envío de una consulta dependiendo sea el caso; siguiendo la siguiente estructura: - Envío satisfactorio: <code><h3>“Su consulta fue enviada satisfactoriamente; recibirá respuesta su consulta tan pronto como sea posible mediante un mensaje a su correo electrónico”</h3></code> <code><h3>“Su consulta no fue enviada. Por favor revise sus datos e intente nuevamente. Disculpe los inconvenientes.”</h3></code>
Importancia	Vital.
Estado	En construcción.

ERS-007	División para la fotografía del Dr. Dongo.	
Versión	2.0	
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	EL-003	
Descripción	Se debe considerar una división que contenga una imagen; de tal forma que mediante esta imagen se haga una referencia a la fotografía del Dr. Dongo. Para ello se utiliza lo siguiente: <code><div></div></code>	
Importancia	Vital.	
Estado	En construcción.	

ERS-008	División para la información profesional y laboral del Dr. Dongo.	
Versión	2.0	
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	EL-003	
Descripción	Se debe considerar una división que contenga una lista de la información profesional y laboral del Dr. Dongo. Para ello se utiliza lo siguiente: <code><div>.....</div></code> Cada elemento de la lista tendrá la información siguiente: - Abogado, Magister en Derecho del Trabajo y Seguridad Social. - Doctor en Derecho - Ex Juez Superior, Presidente de la Sala Laboral de la Corte Suprema de Arequipa. - Juez Supremo (Sala Transitoria de Derecho Constitucional y Social de la Corte Suprema de Justicia de la República. - Docente Universitario. - Conferencista.	
Importancia	Vital.	
Estado	En construcción.	

ERS-009	División para la ubicación del calendario.	
----------------	---	---

Versión	2.0
M/C	C: Medida B
Autor	Grupo de Trabajo.
Fuente	EL-004
Descripción	Se debe considerar una división que contenga un include a la clase con el nombre calendario.php. La clase calendario, está conformada por código php y muestra la información correspondiente a la fecha en una tabla. Para incluir la clase calendario dentro de la división se puede considerar la siguiente sentencia: <code><div><?php include ("calendario.php");?></div></code>
Importancia	Vital.
Estado	En construcción.

ERS-010	División para la ubicación de las preguntas realizadas con mayor frecuencia al Dr. Dongo.
Versión	2.0
M/C	C: Medida B
Autor	Grupo de Trabajo.
Fuente	EL-005
Descripción	Se debe considerar una división que contenga una imagen y texto; la imagen debe tener una referencia a una nueva ventana de diálogo donde se encontrará las preguntas realizadas con mayor frecuencia. La estructura será: <code><div><a> imagen<p>Título</p></div></code>
Importancia	Vital.
Estado	En construcción.

ERS-011	Ventana de preguntas frecuentes.
Versión	1.2
M/C	C: Medida B
Autor	Grupo de Trabajo.
Fuente	EL-005
Descripción	En la ventana de preguntas frecuentes se mostrará una tabla que tendrá en cada fila dos columnas. La primera columna indica el número de la pregunta y la segunda el contenido de dicha pregunta.
Importancia	Vital.
Estado	En construcción.

ERS-012	División para consignar noticias de ámbito jurídico.
Versión	2.0
M/C	C: Medida B
Autor	Grupo de Trabajo.
Fuente	EL-006
Descripción	Se debe considerar una división que contenga una imagen y texto; la imagen debe tener una referencia a una nueva ventana de diálogo donde se encontrará las noticias que el Dr. Dongo crea por conveniente.

	La estructura será:
	<code><div><a> imagen<p>Título</p></div></code>
Importancia	Vital.
Estado	En construcción.

ERS-013	Ventana de noticias.	
Versión	2.0	
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	EL-006	
Descripción	En la ventana de noticias se mostrará una tabla que tendrá en cada fila dos columnas. La primera columna indica el título de la noticia y la segunda el contenido de dicha noticia.	
Importancia	Vital.	
Estado	En construcción.	

ERS-0014	Características del formulario de Reserva de citas.	
Versión	1.01	
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	EL-007	
Descripción	Para crear los formularios de consulta se utilizará código html; empleando las etiquetas <code><form></code> y <code><table></code> para el diseño. Todos los campos del formulario a excepción del cuerpo de la consulta se realizarán en cajas de texto de una línea, las cuales son creadas dentro de la tabla del formulario con la etiqueta <code><input type="text"></code>. El cuerpo de la consulta se hará usando una caja de texto multilínea mediante la etiqueta <code><textarea></textarea></code>. Se podrá adjuntar un documento del disco duro y enviarlo utilizando el control <code><input type="file" /></code> Se contará con un botón "Reservar cita" para enviar los datos del formulario, para ello se usa la etiqueta <code><input type="submit" value="Reservar cita"></code>. El formulario debe definirse de la siguiente forma: <code>form<method="get" action="reserva.php"></code>, para que los valores de los campos de formulario se vean por la barra de direcciones y sean enviados al archivo reserva.php.	
Importancia	Vital.	
Estado	En construcción.	

ERS-015	Formulario para una reservación de cita de un usuario natural.	
----------------	---	---

Versión	2.0
M/C	C: Medida B
Autor	Grupo de Trabajo.
Fuente	EL-007
Descripción	El formulario para que un usuario natural pueda realizar una reservación de cita consta de la siguiente estructura: Debe estar contenido en la estructura: <form> </form> Etiquetas con los nombres siguientes: • Nombres • Apellidos • DNI • Correo electrónico • Teléfono • Dirección • Asunto de la reservación. • Fecha de reservación (sugerida por el usuario). • Hora de reservación (sugerida por el usuario). Cada etiqueta deberá estar asociado a un elemento de entrada o caja de texto; los cuales deberán ser llenados por el usuario obligatoriamente. Al final del formulario debe haber un botón con el nombre “Reservar cita” que al ser seleccionado envíe la petición de reserva solo si se ha llenado los campos requeridos obligatoriamente.
Importancia	Vital.
Estado	En construcción.

ERS-015	Formulario para una reservación de cita de un usuario bajo un contrato
Versión	2.0
M/C	C: Medida B
Autor	Grupo de Trabajo.
Fuente	EL-007
Descripción	El formulario para que un usuario bajo un contrato pueda realizar una reservación de cita consta de la siguiente estructura: Debe estar contenido en la estructura: <form> </form> Etiquetas con los nombres siguientes: • Nombres • Apellidos • DNI • Correo electrónico • Teléfono • Dirección • Asunto de la reservación. • Fecha de reservación (sugerida por el usuario). • Hora de reservación (sugerida por el usuario).

C

Cada etiqueta deberá estar asociado a un elemento de entrada o caja de texto. Los elementos de entrada que deben ser campos obligatorios a llenar son los siguientes (Los restantes se autogeneran):

- Asunto de la reservación
- Fecha de reservación (sugerida por el usuario).
- Hora de reservación (sugerida por el usuario).

Al final del formulario debe haber un botón con el nombre “Reservar cita” que al ser seleccionado envíe la petición de reserva solo si se ha llenado los campos requeridos obligatoriamente.

Importancia	Vital.
Estado	En construcción.

ERS-016	Enviar notificación de la reservación de una cita al correo electrónico del Dr. Dongo.
Versión	1.02
M/C	C: Medida B
Autor	Grupo de Trabajo.
Fuente	EL-007
Descripción	Después de guardar la reservación de una cita en la base de datos se enviará un correo electrónico al Dr. Dongo mandándole la información correspondiente del usuario. Para ello se utilizará la función mail: • mail(\$para,\$asunto,\$mensaje) La variable <i>\$para</i> indica el correo del Dr. Dongo; la variable <i>\$asunto</i> indicará: Nueva reservación de cita del Estudio Jurídico Virtual; y el <i>\$mensaje</i> indicará la información del usuario, fecha y hora de la reserva solicitada por el usuario.
Importancia	Vital.
Estado	En construcción.

ERS – 017	Mensaje de confirmación del envío de la reservación.
Versión	2.0
M/C	C: Medida B
Autor	Grupo de Trabajo.
Fuente	EL-007
Descripción	El sistema debe mostrar un mensaje que confirme el envío de una reserva de cita dependiendo sea el caso; siguiendo la siguiente estructura: • Envío satisfactorio: <i><h3>“Su reservación fue enviada satisfactoriamente; recibirá una respuesta tan pronto como sea</i>

posible mediante el envío de un mensaje a su correo electrónico”</h3>

- <h3>“Su reservación no fue enviada. Por favor revise sus datos e intente nuevamente. Disculpe los inconvenientes.” </h3>

Importancia Vital.

Estado En construcción.

ERS-018 Recoger datos de un determinado formulario.

Versión 1.02

M/C C: Medida B

Autor Grupo de Trabajo.

Fuente EL-008

Descripción

Después de llenar un respectivo formulario, se presiona el botón correspondiente a dicho formulario, los datos del formulario se enviarán a otra página y se almacenarán utilizando variables como por ejemplo: \$variable1 = \$_GET['nom_variable'];

Importancia Vital.

Estado En construcción.

ERS-019 Guardar información.

Versión 1.02

M/C C: Medida B

Autor Grupo de Trabajo.

Fuente EL-008

Descripción

Después de haber llenado un formulario, se toman los datos del formulario y se guardan en una determinada tabla que se encuentra dentro de la Base de Datos “Estudio_Juridico_Virtual”.

Para ello se hará la consulta de inserción en la tabla, por ejemplo:
\$sql=mysql_query(“INSERT INTO `Consultas` VALUES (\$nombres
,\$apellidos,\$dni,\$telefono,\$direccion,\$email,\$asunto,\$cuerpo)”);

Importancia Vital.

Estado En construcción.

ERS-020 Almacenar la información en una base de datos en MySQL

Versión 1.02

M/C C: Medida B

Autor Grupo de Trabajo.

Fuente EL-008

Descripción

Se creará una base de datos “Estudio_Juridico_Virtual” utilizando MySQL con las tablas correspondientes.

La conexión con la base de datos se hará usando:
@mysql_connect(“localhost”, “root”, “”)

@mysql_select_db("Estudio_Juridico_Virtual")	
Importancia	Vital.
Estado	En construcción.
ERS-021	Colores de página web.
Versión	102
M/C	C: Medida B
Autor	Grupo de Trabajo.
Fuente	EL-009
Descripción	La página web tendrá los siguientes colores de acuerdo a la estructura: • Cabecera: color celeste degradado. • Menú: color negro degradado. • Contenido: color blanco.
Importancia	Vital.
Estado	En construcción.
ERS-022	Menú
Versión	102
M/C	C: Medida B
Autor	Grupo de Trabajo.
Fuente	EL-010
Descripción	El sistema contara con un menú con los siguientes nombres, los cuales intentan describir de la mejor manera su contenido, el menú será desarrollado con una lista, la estructura y nombres serán los siguientes: • Inicio Realizar Consulta Reservar Cita Ayuda Cada lista hará una referencia a una página correspondiente a cada sección.
Importancia	Vital.
Estado	En construcción.
ERS-023	Secciones para cada menú
Versión	102
M/C	C: Medida B
Autor	Grupo de Trabajo.
Fuente	EL-010
Descripción	El proyecto de la aplicación estará dividido por carpetas para cada menú, a excepción del inicio, el cual será nuestro index principal de la página. La estructura se muestra de la siguiente forma • Proyecto -Ayuda (Carpeta) -RealizarConsulta (Carpeta)

	-ReservarCita (Carpeta) -index.php (Archivo) Cada carpeta tendrá dentro un archivo index.php referente a la información de su carpeta correspondiente.
Importancia	Vital.
Estado	En construcción.

ERS-024	Estructura de la página
Versión	1.02
M/C	C: Medida B
Autor	Grupo de Trabajo.
Fuente	EL-011
Descripción	La aplicación estará formada por la siguiente estructura utilizando divisiones: • División que contiene a toda la aplicación. ○ División para la cabecera. ▪ División para el logo ▪ División para el título de la aplicación ○ División para el menú. ○ División para el contenido. ○ División para el pie de pagina
	Para las divisiones se utilizara la etiqueta , como se ve lo siguiente: <pre><div id=pagina> <div id= "divCabecera" > <div id= "divLogo" ></div> <div id= "divTitulo" ></div> </div> <div id= "divMenu" ></div> <div id= "divContenido" ></div> <div id= "divPie" ></div> </div></pre>
Importancia	Vital.
Estado	En construcción.

ERS-25	Sección ayuda	C
Versión	1.02	
M/C	C: Medida B	
Autor	Grupo de Trabajo.	
Fuente	EL-012	
Descripción	En esta sección se mostrará la ayuda en un pdf incrustado en la página; para ello se utiliza la siguiente estructura:	

<object type="application/pdf" data="../../documentos/MANUAL DE USUARIO.pdf" style="width: 100%; height: 350px;"> ERROR (no puede mostrarse el objeto) </object>

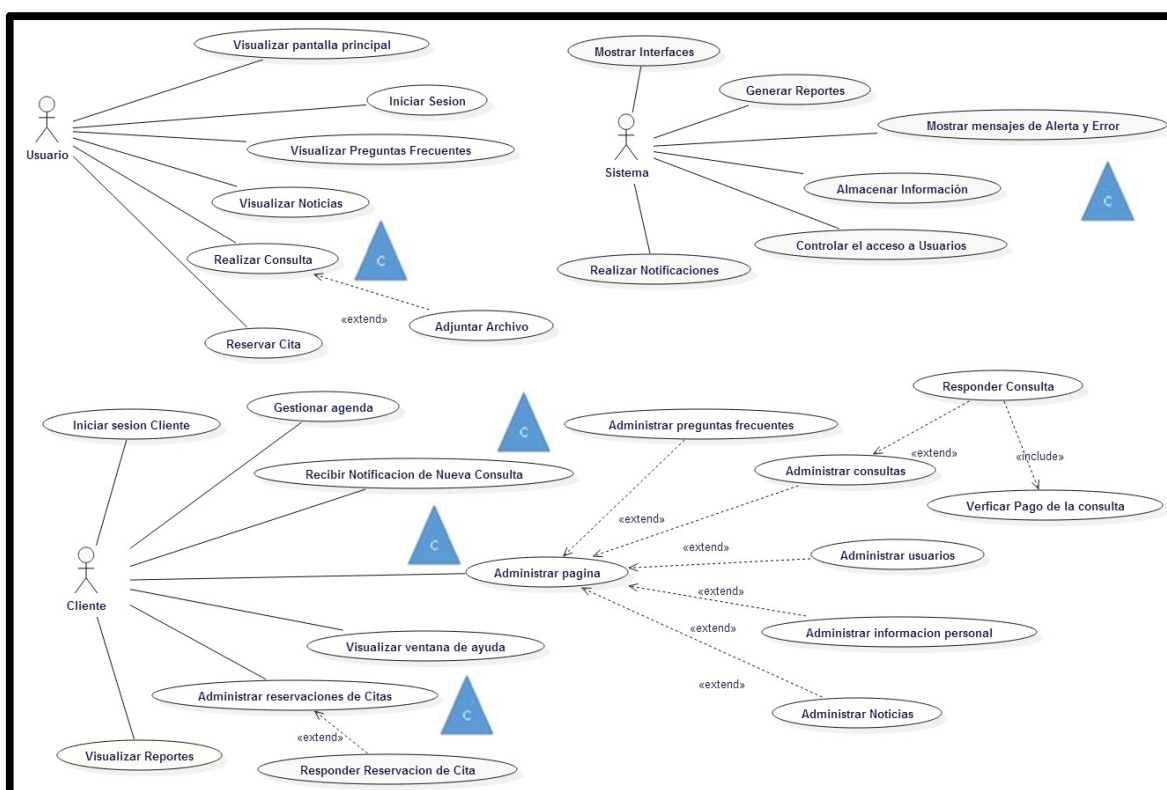
Importancia

Vital.

Estado

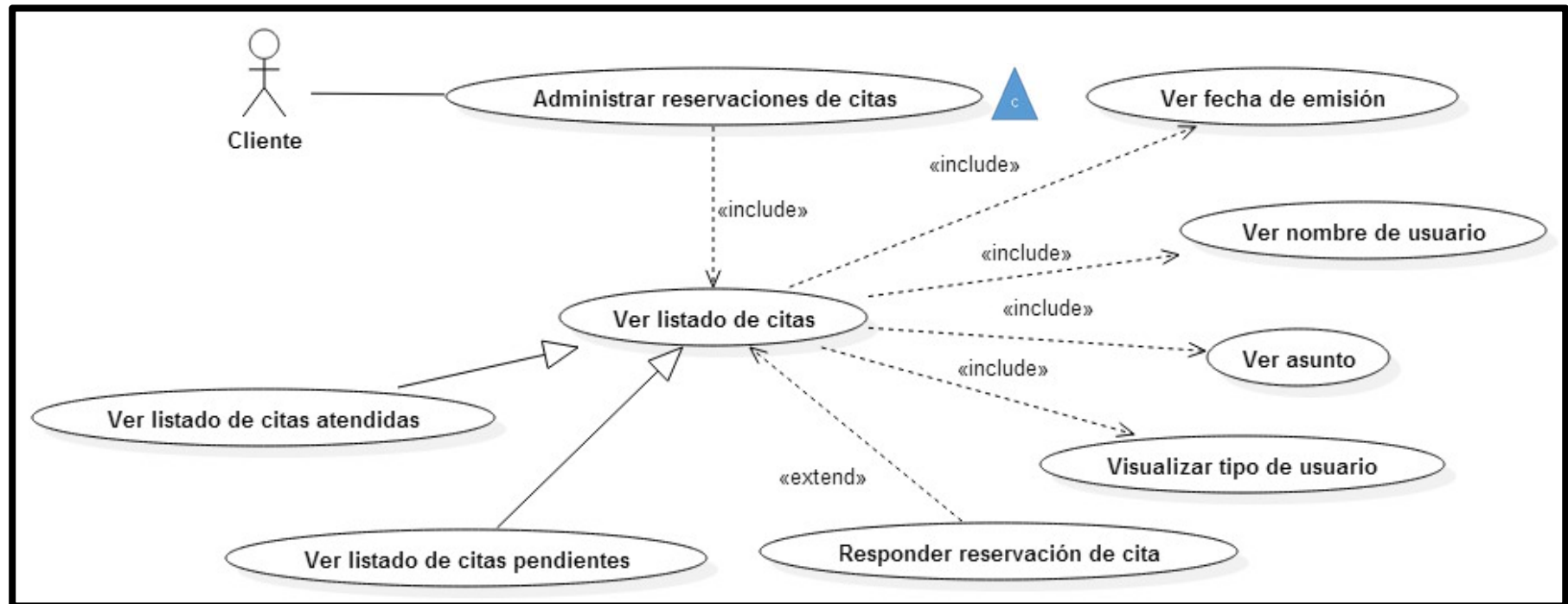
En construcción.

4.2.4 Fase de diseño del diagrama de casos de uso



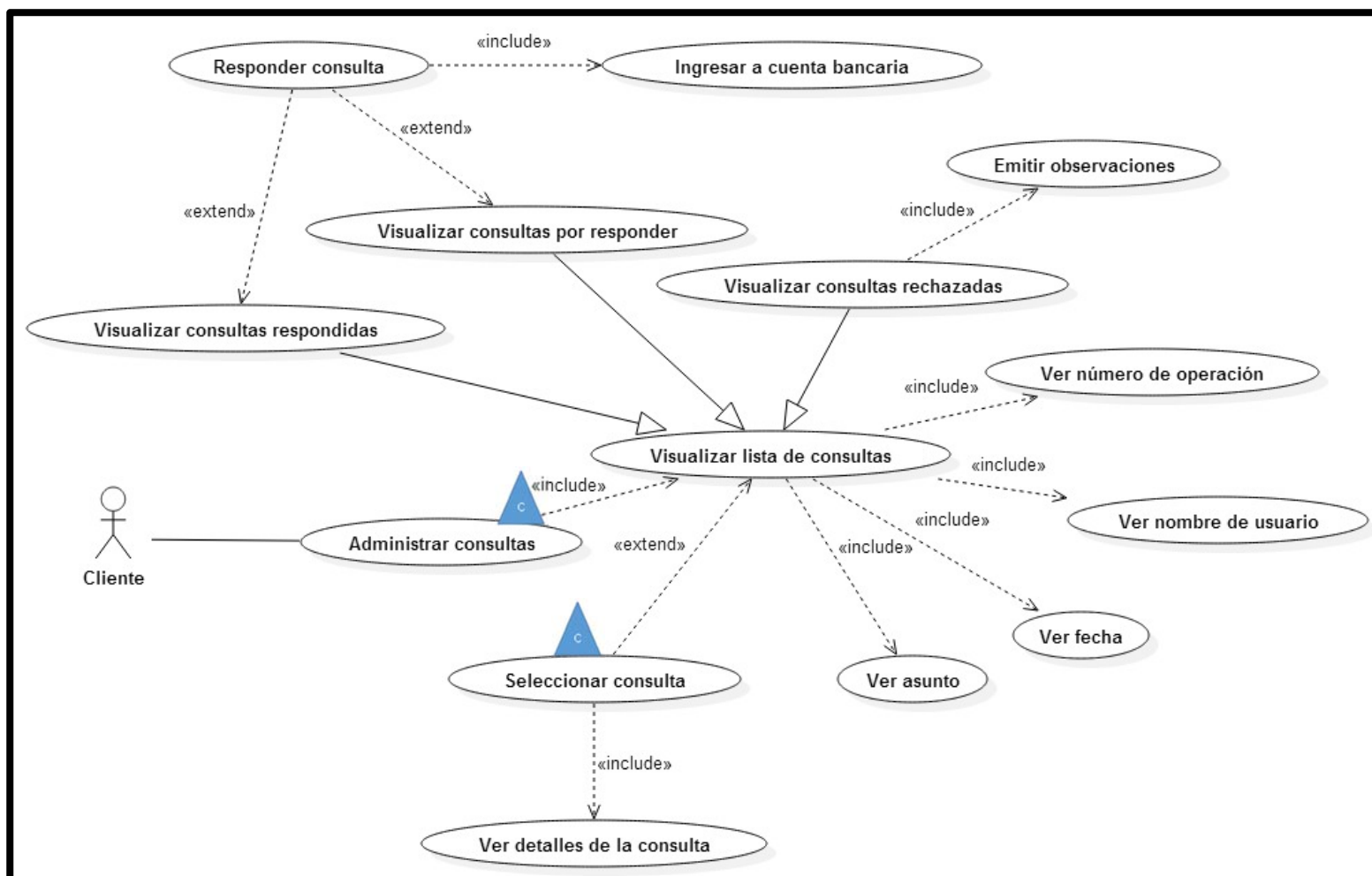
Leyenda:

CASO DE USO	M/C	DESCRIPCIÓN
Realizar consulta	C	Controlar que la consulta esté bien hecha
Recibir notificación de nueva consulta	C	Controlar las notificaciones de consulta
Administrar página	C	Controlar la administración de páginas
Administrar reservación de cita	C	Controlar que la reservación de citas esté bien hecha
Almacenar información	C	Controlar que la información se almacene correctamente



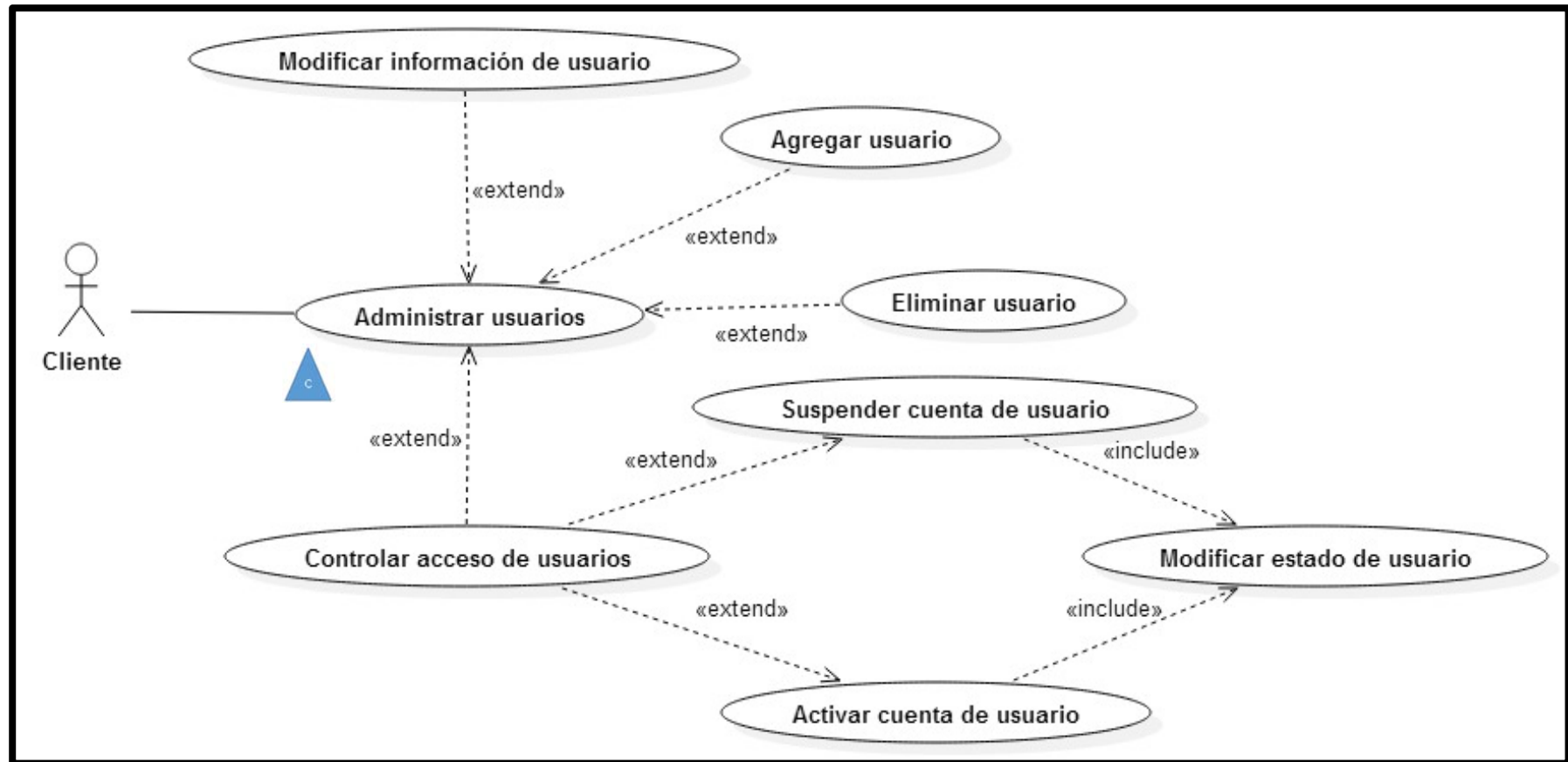
Leyenda:

CASO DE USO	M/C	DESCRIPCIÓN
Administrar reservaciones de citas	C	Controlar que las reservaciones se lleven a cabo en forma correcta



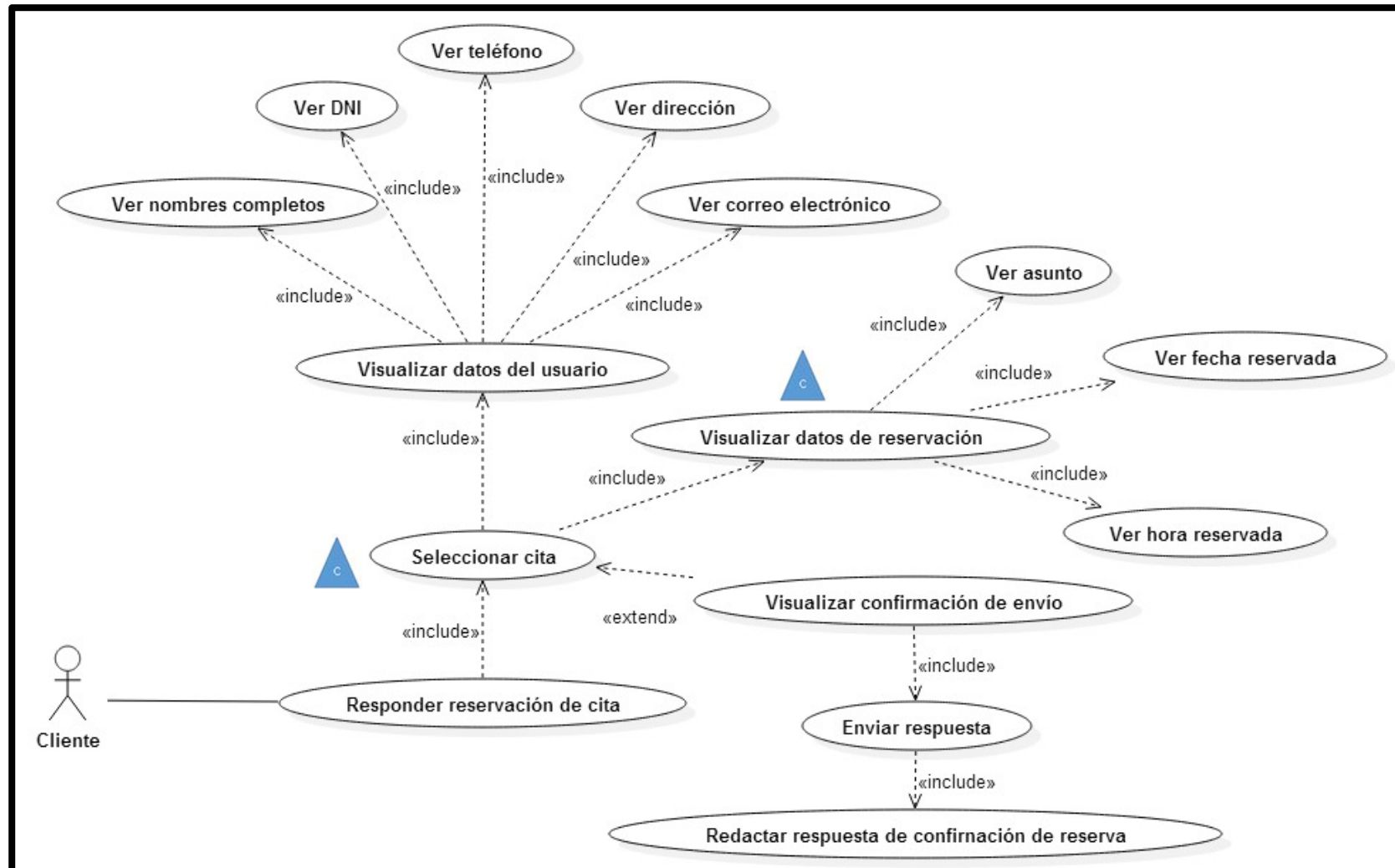
Leyenda:

CASO DE USO	M/C	DESCRIPCIÓN
Administrar consultas	C	Controlar la administración de las consultas
Seleccionar consulta	C	Controlar la selección de consultas



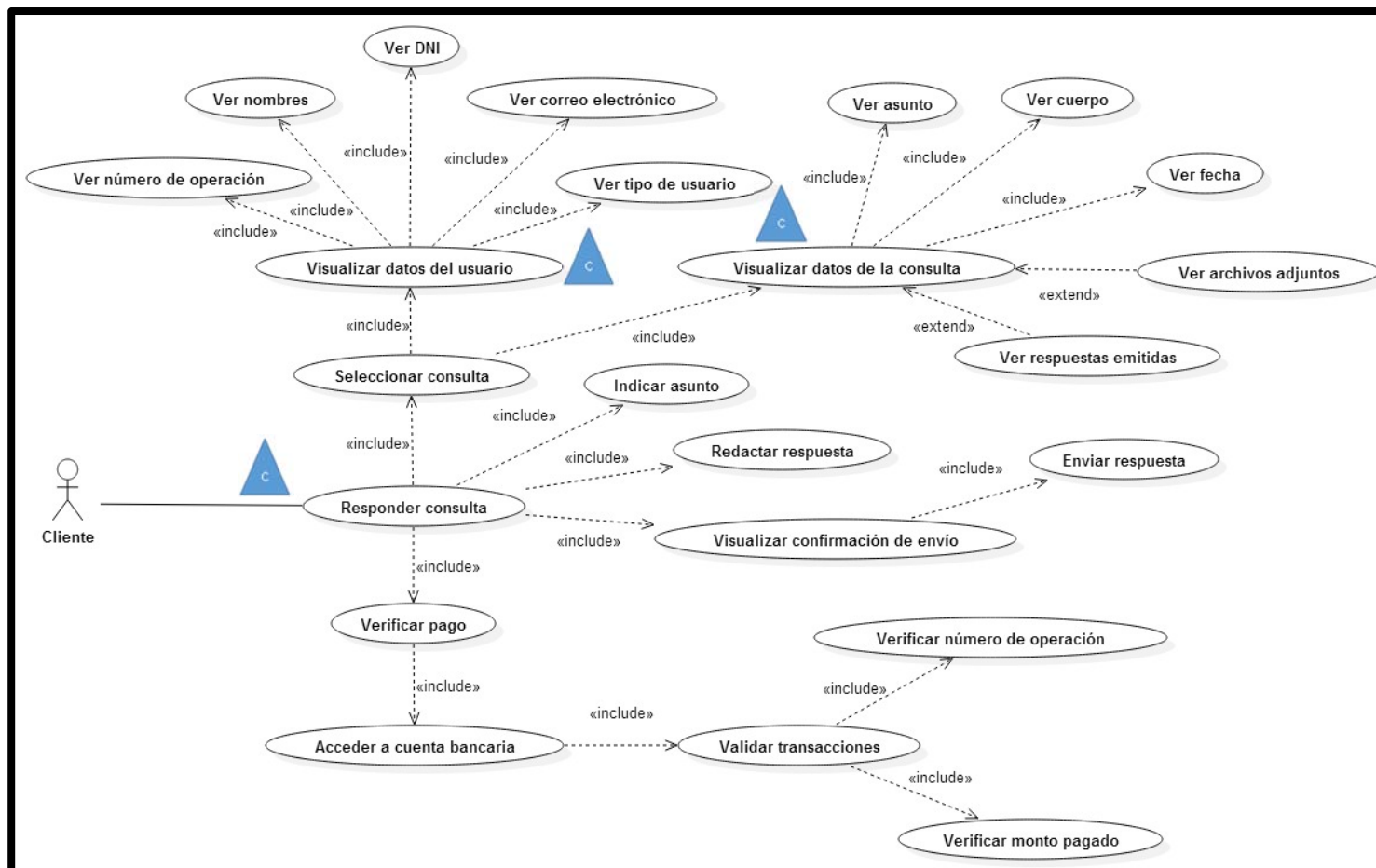
Leyenda:

CASO DE USO	M/C	DESCRIPCIÓN
Administrar usuarios	C	Controlar que la administración de usuarios se haga de manera correcta



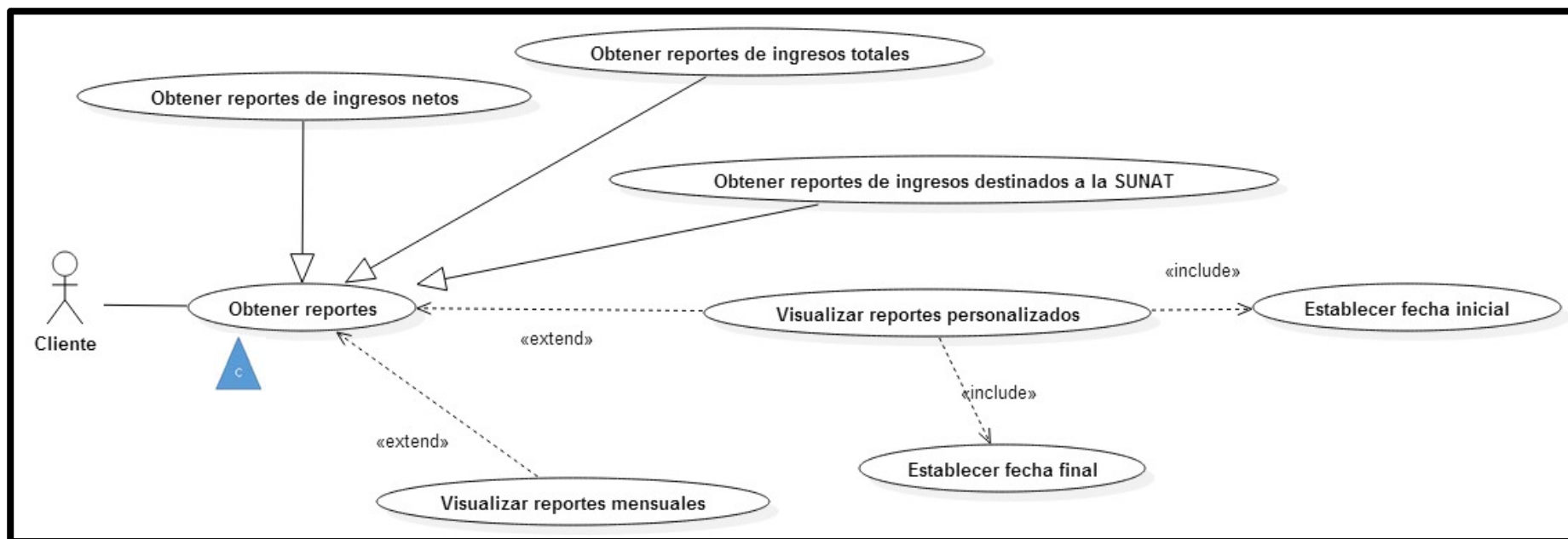
Leyenda:

CASO DE USO	M/C	DESCRIPCIÓN
Seleccionar cita	C	Controlar la selección de citas
Visualizar datos de reservación	C	Controlar que los datos correspondan al elemento seleccionado



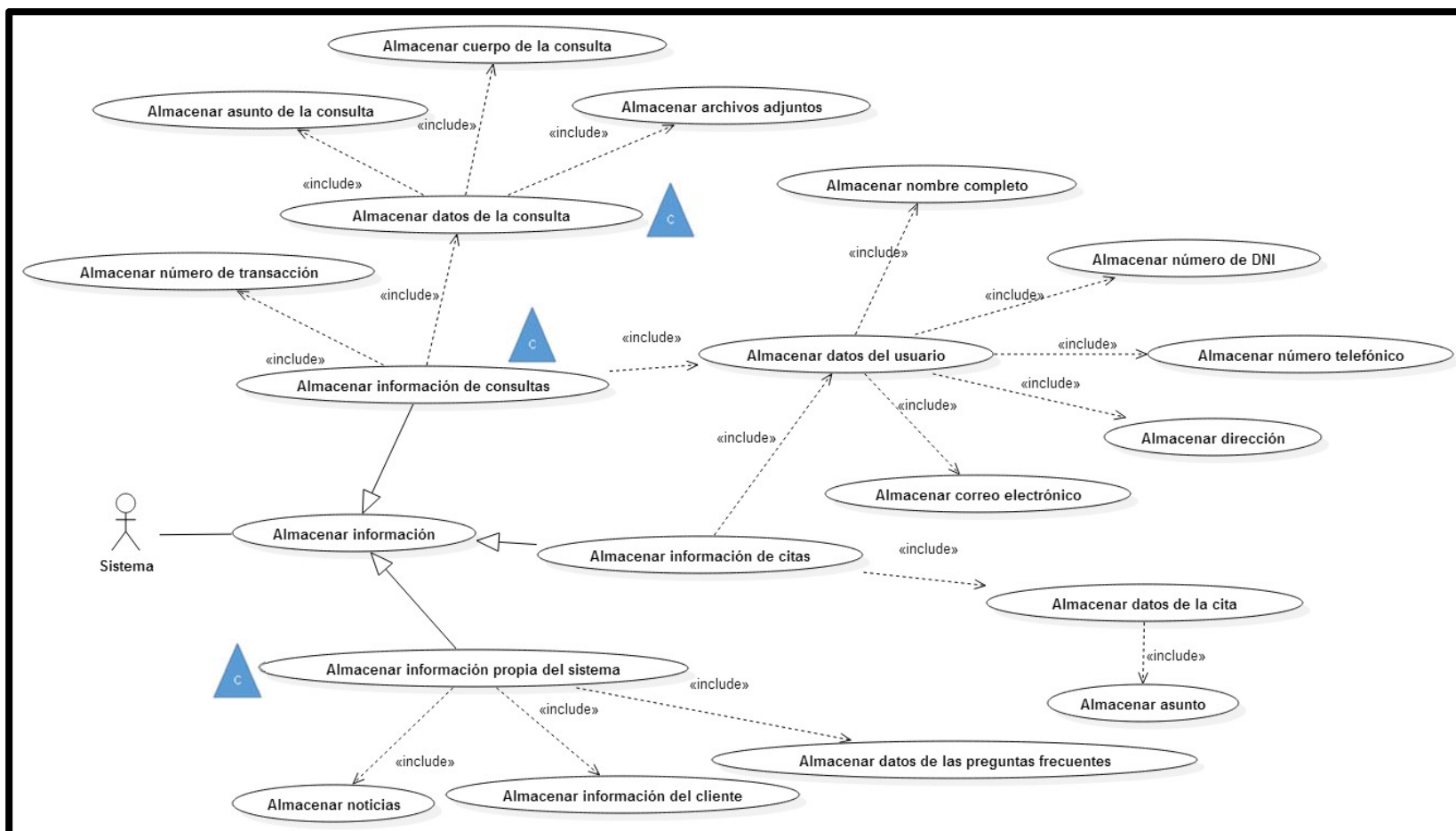
Leyenda:

CASO DE USO	M/C	DESCRIPCIÓN
Responder consulta	C	Controlar que la respuesta sea correcta
Visualizar datos del usuario	C	Controlar que los datos correspondan al usuario
Visualizar datos de la consulta	C	Controlar que los datos correspondan a la consulta realizada



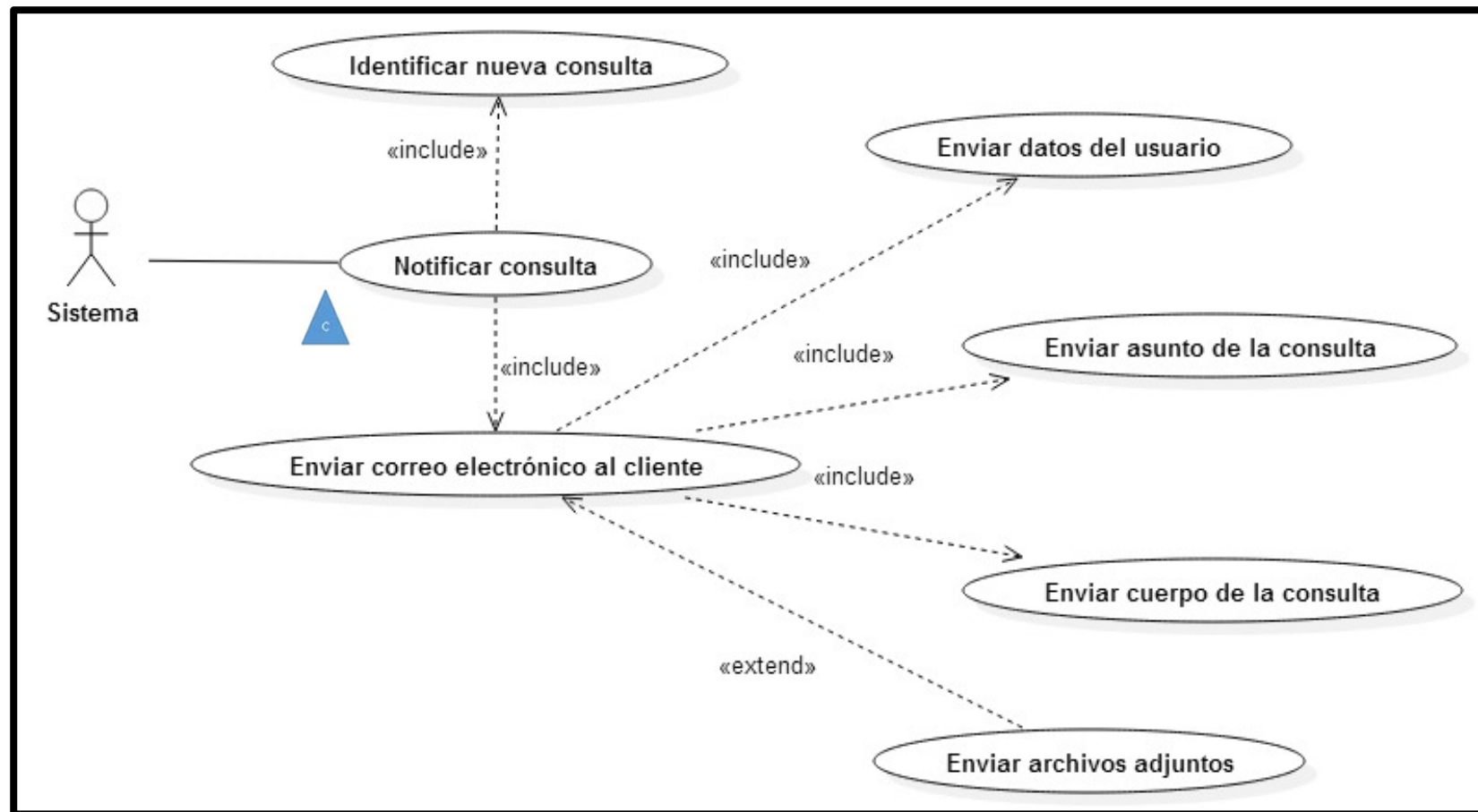
Leyenda:

CASO DE USO	M/C	DESCRIPCIÓN
Obtener reportes	C	Controlar que los reportes correspondan a los solicitados



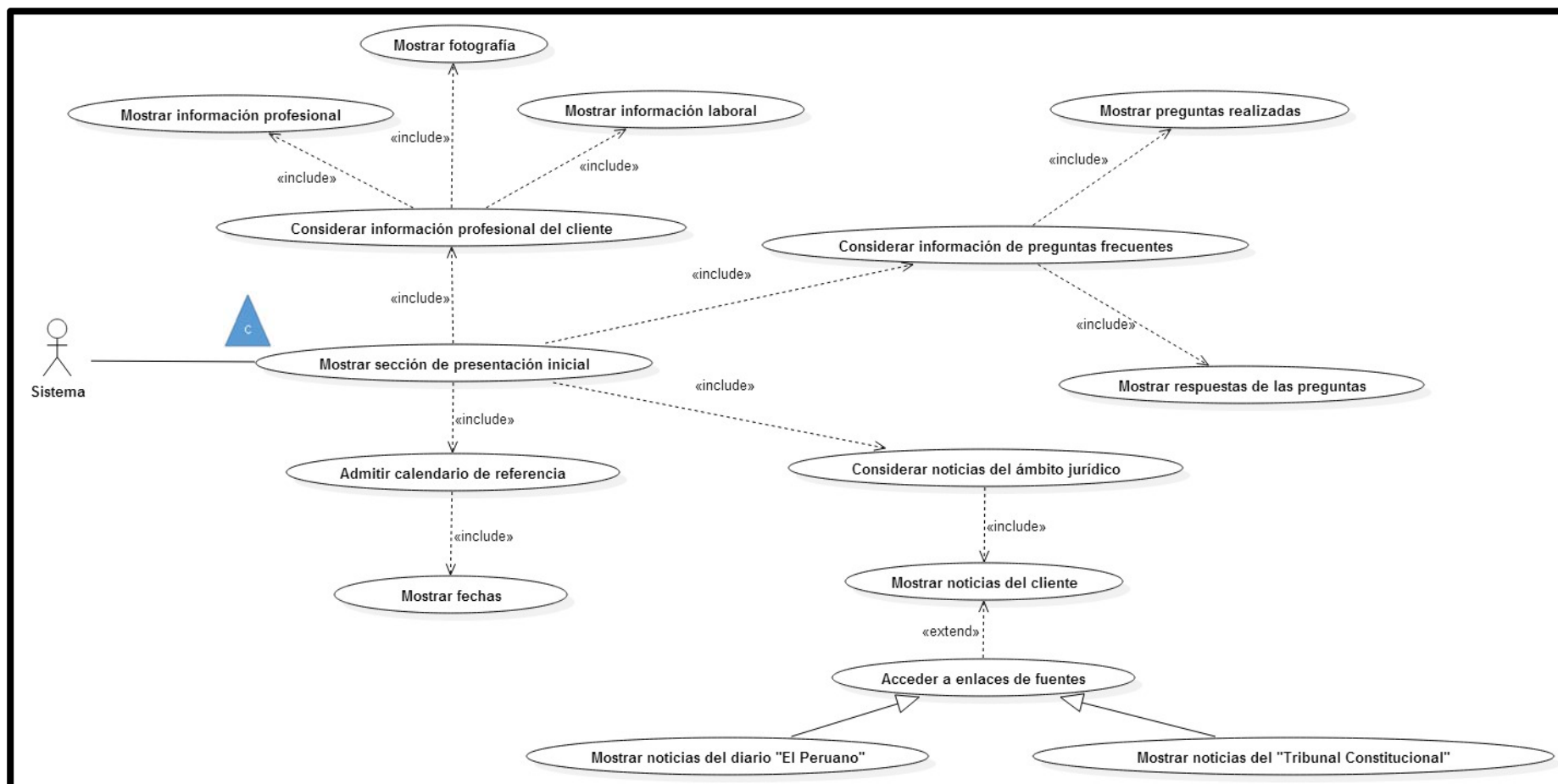
Legenda:

CASO DE USO	M/C	DESCRIPCIÓN
Almacenar datos de la consulta	C	Controlar que los datos de la consulta se almacenen correctamente
Almacenar información de consultas	C	Controlar que la información de las consultas se almacenen de manera adecuada
Almacenar información propia del sistema	C	Controlar la información que arroja el sistema



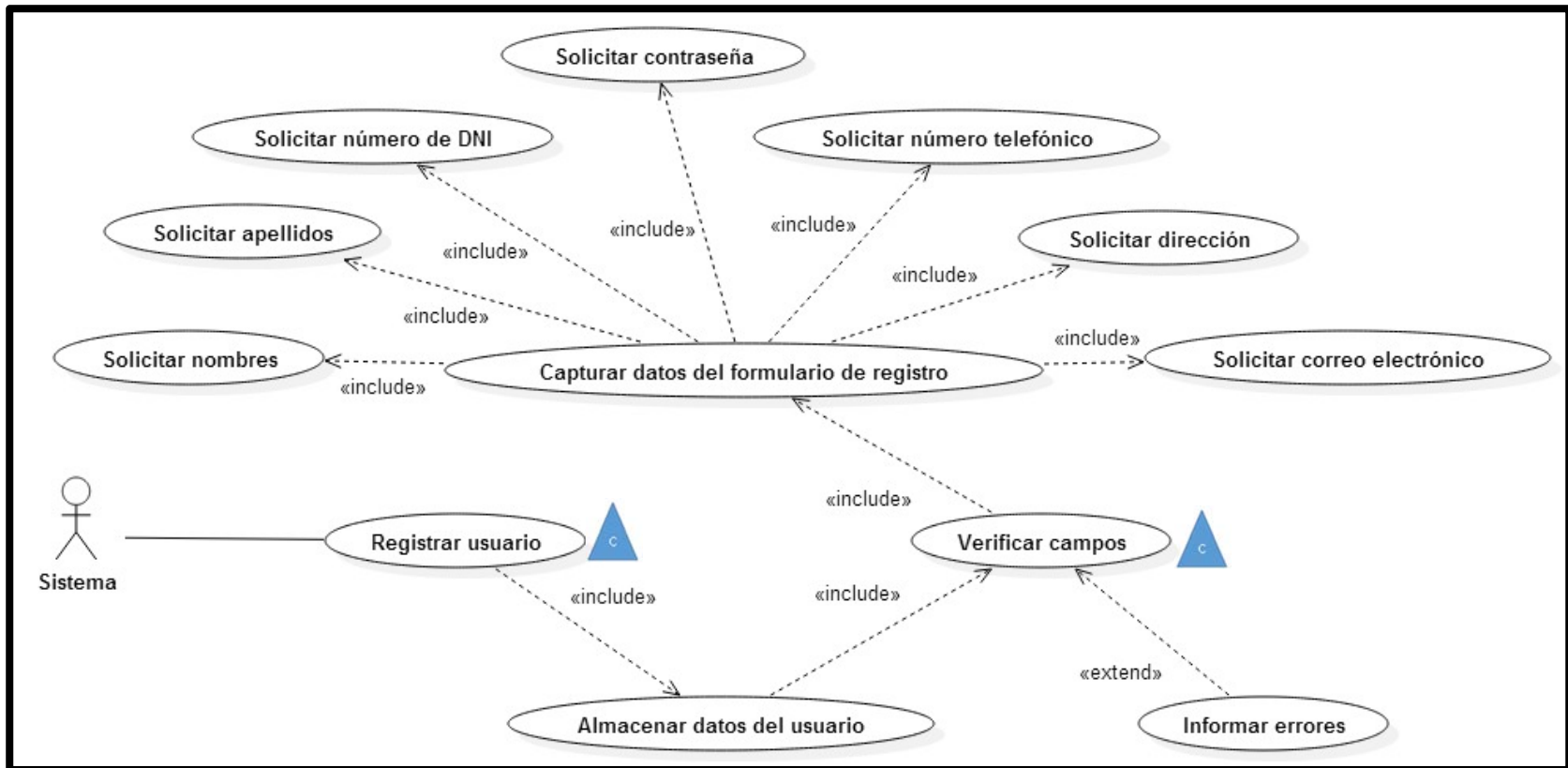
Leyenda:

CASO DE USO	M/C	DESCRIPCIÓN
Notificar consulta	C	Controlar que las consultas sean notificadas de manera correcta



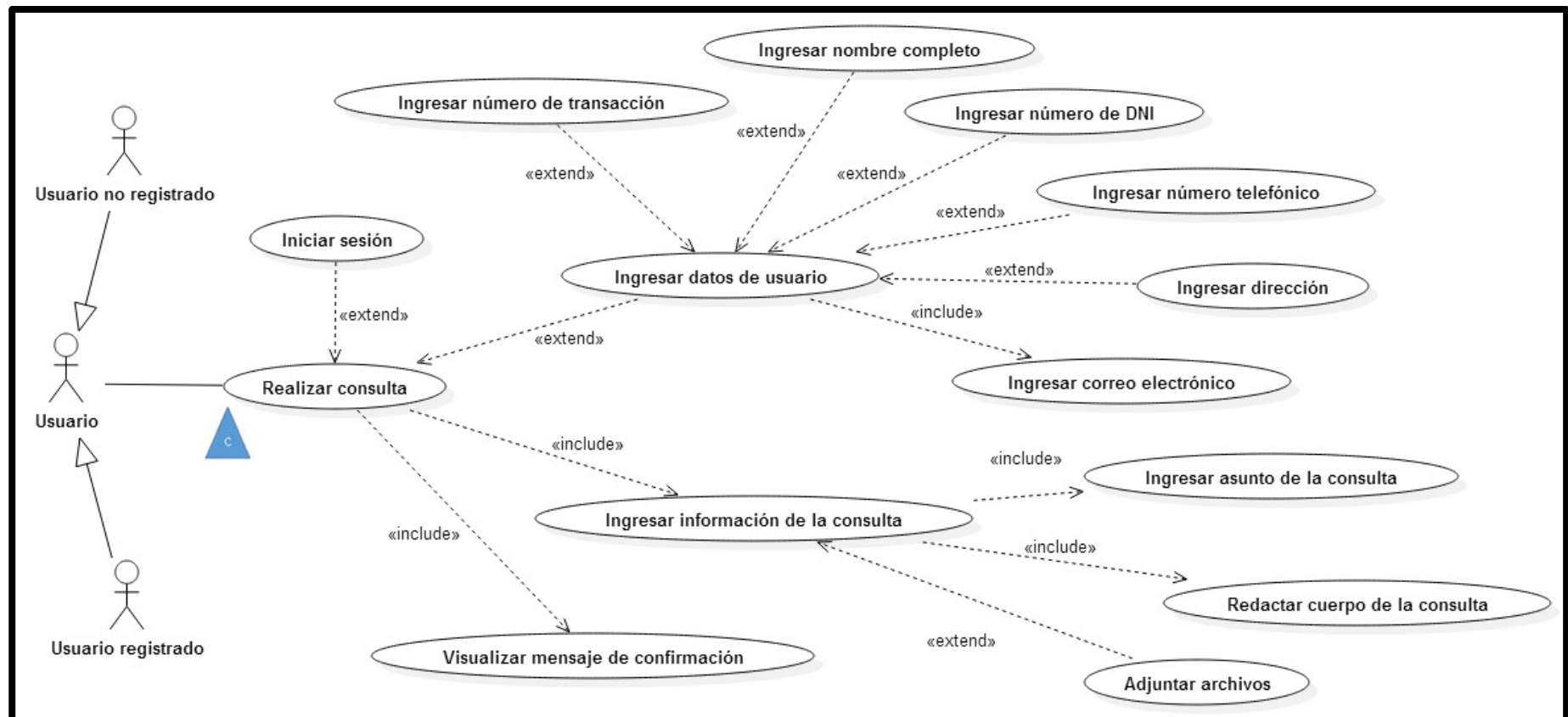
Leyenda:

CASO DE USO	M/C	DESCRIPCIÓN
Mostrar sección de presentación inicial	C	Controlar que la sección se presente de manera adecuada



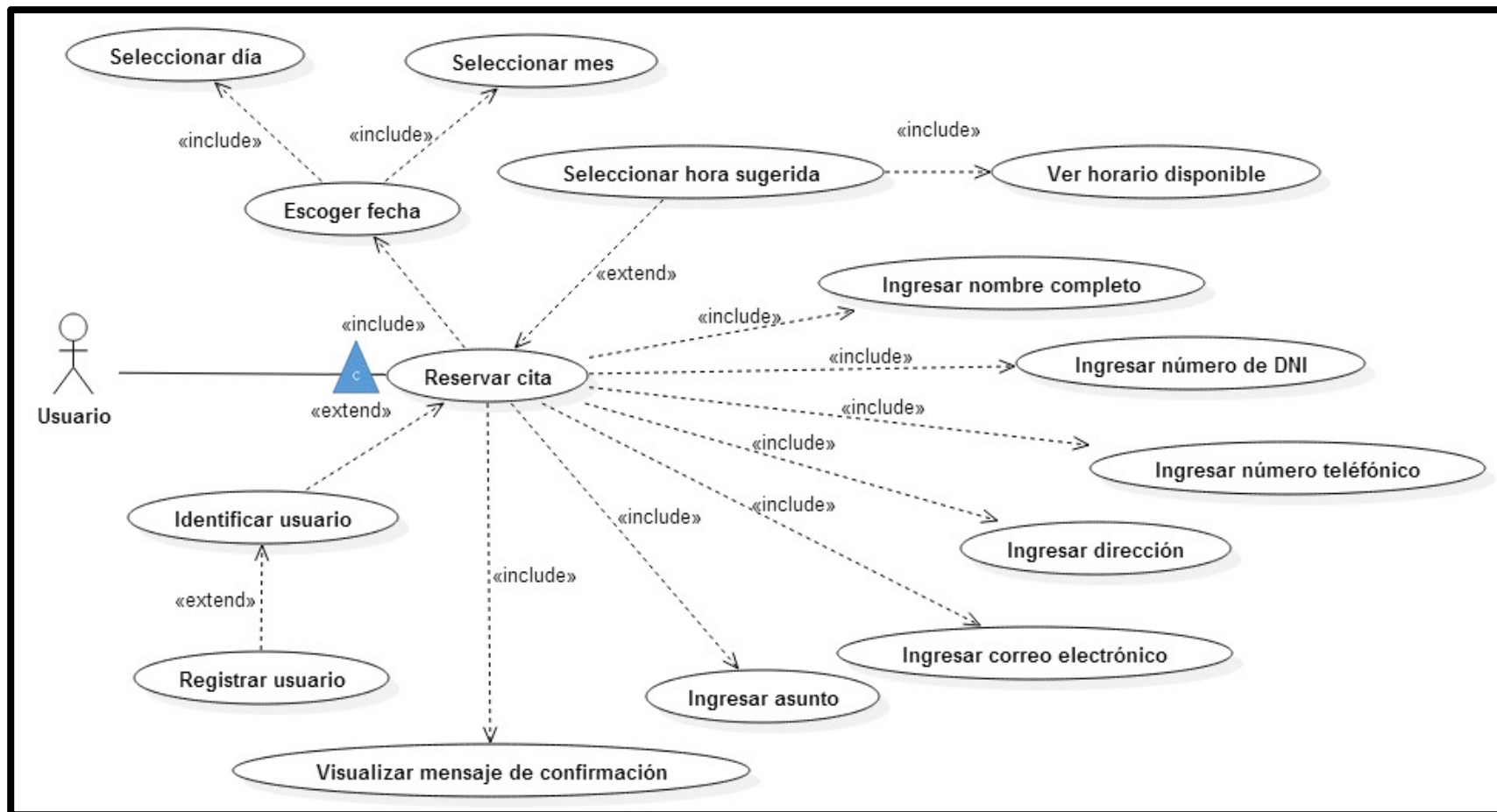
Leyenda:

CASO DE USO	M/C	DESCRIPCIÓN
Registrar usuario	C	Controlar que el usuario sea registrado de manera correcta
Verificar campos	C	Controlar la verificación de campos



Leyenda:

CASO DE USO	M/C	DESCRIPCIÓN
Realizar consulta	C	Controlar que la consulta se lleve a cabo de manera adecuada



Leyenda:

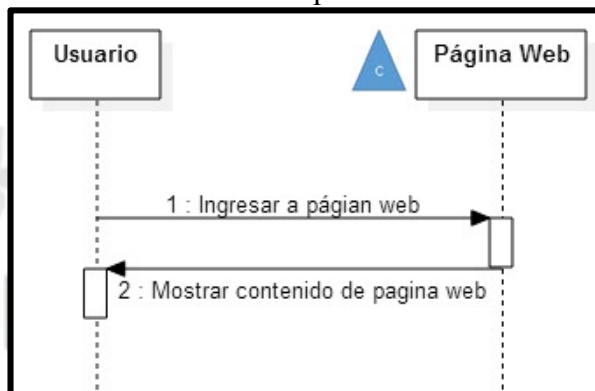
CASO DE USO	M/C	DESCRIPCIÓN
Reservar cita	C	Controlar que la reserva de citas se haga correctamente

4.2.5 Etapa de diseño del modelo conceptual

La construcción del producto orientado a la web necesitó una especificación clara del diagrama de clases, el cual permitió una clara concepción del producto a construir.

4.2.6 Etapa de diseño de los diagramas de secuencias

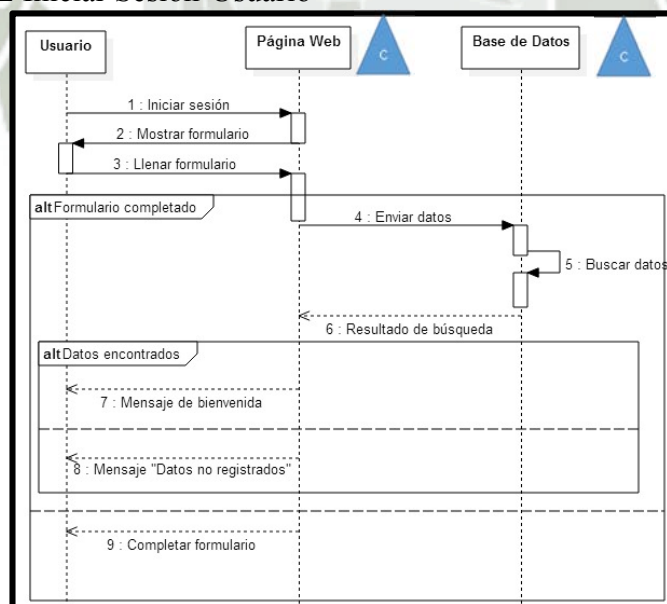
UC-0001 Visualizar interfaz de la aplicación.



Leyenda:

OBJETO	M/C	DESCRIPCIÓN
Página Web	C	Controlar la correcta visualización de la página web

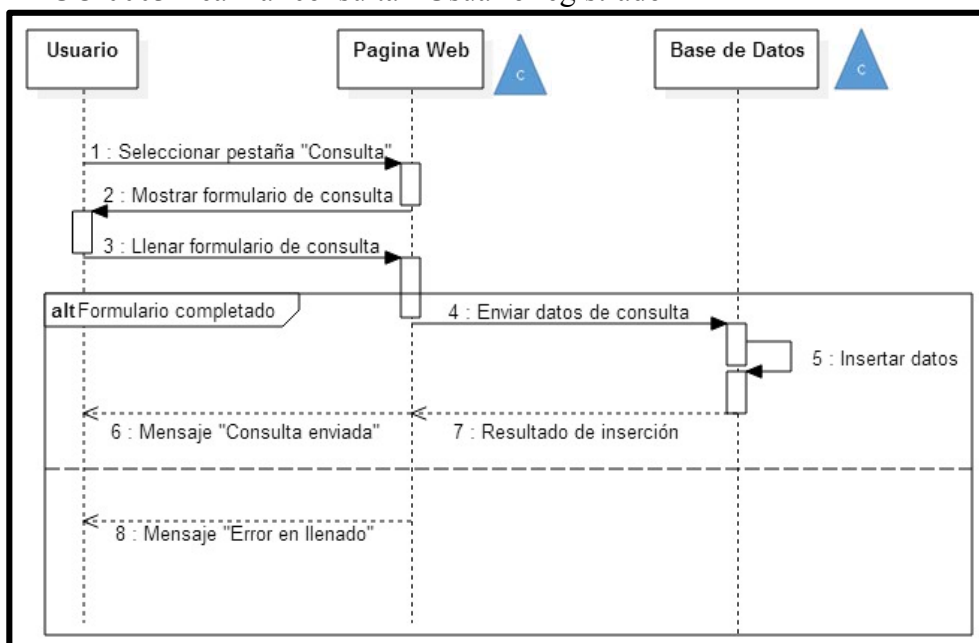
UC-0002 Iniciar Sesión-Usuario



Leyenda:

OBJETO	M/C	DESCRIPCIÓN
Página Web	C	Controlar la correcta visualización de la página web
Base de Datos	C	Controlar la correcta búsqueda de la información

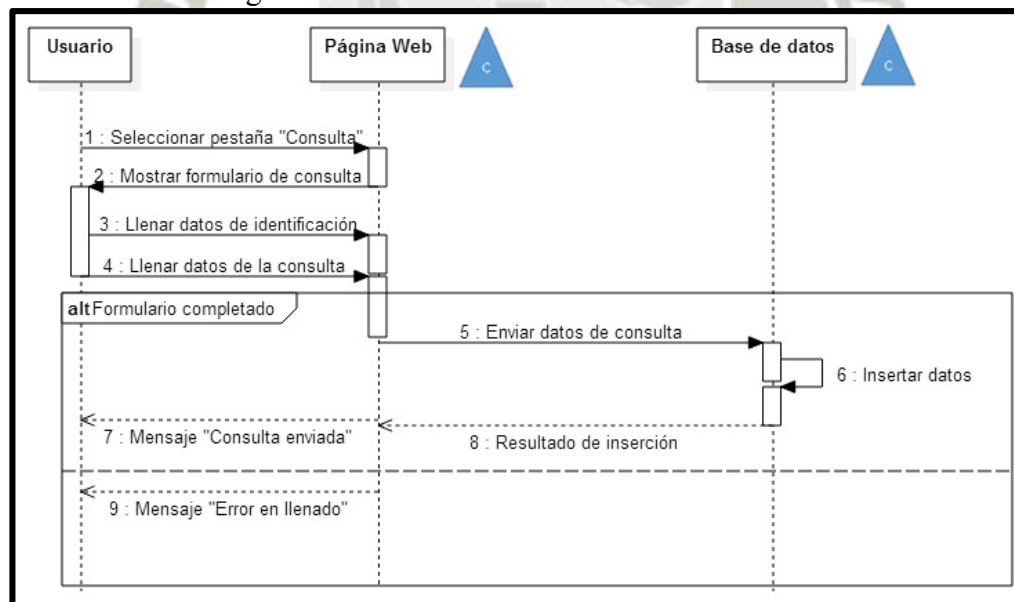
UC-0003 Realizar consulta - Usuario registrado



Leyenda:

OBJETO	M/C	DESCRIPCIÓN
Página Web	C	Controlar la correcta visualización de la página web
Base de Datos	C	Controlar la correcta búsqueda de la información

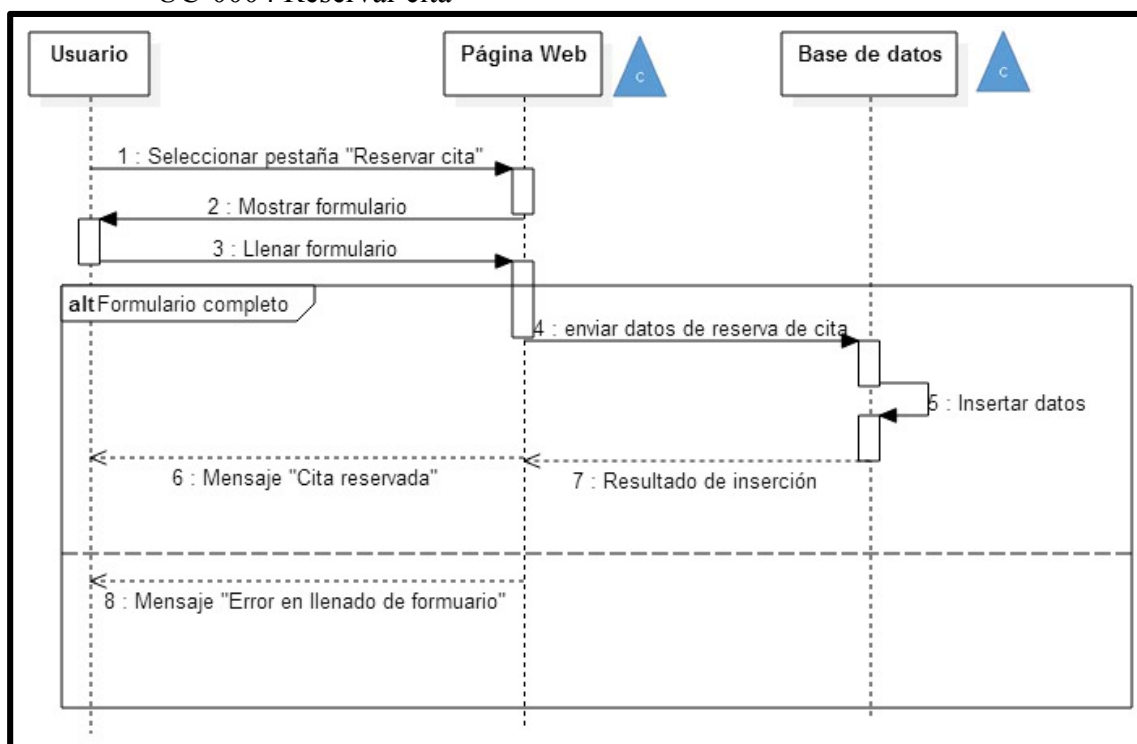
Usuario no registrado



Leyenda:

OBJETO	M/C	DESCRIPCIÓN
Página Web	C	Controlar la correcta visualización de la página web
Base de Datos	C	Controlar la correcta búsqueda de la información

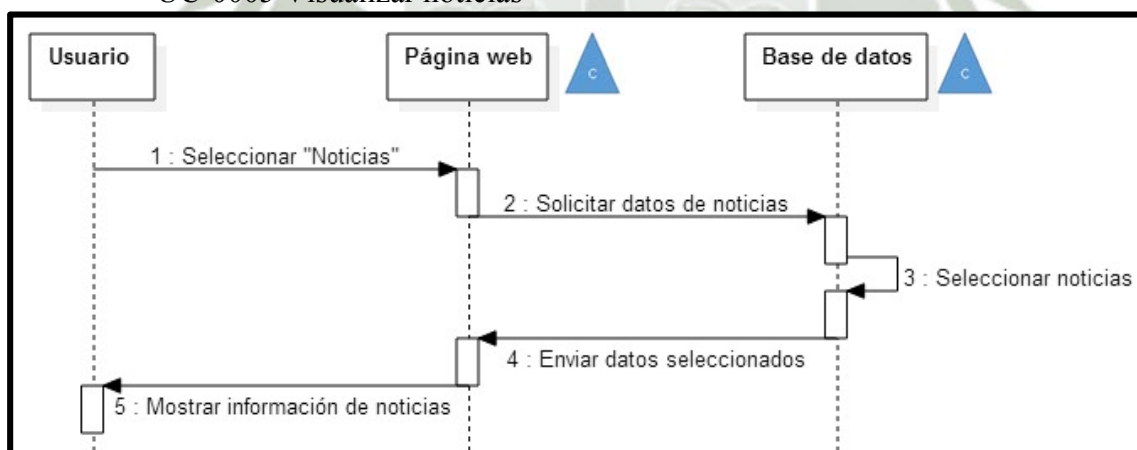
UC-0004 Reservar cita



Leyenda:

OBJETO	M/C	DESCRIPCIÓN
Página Web	C	Controlar la correcta visualización de la página web
Base de Datos	C	Controlar la correcta búsqueda de la información

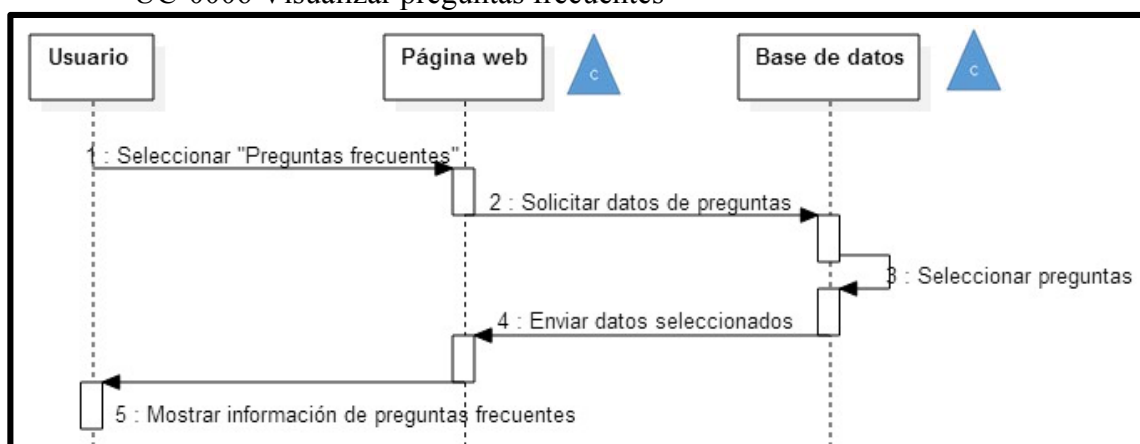
UC-0005 Visualizar noticias



Leyenda:

OBJETO	M/C	DESCRIPCIÓN
Página Web	C	Controlar la correcta visualización de la página web
Base de Datos	C	Controlar la correcta búsqueda de la información

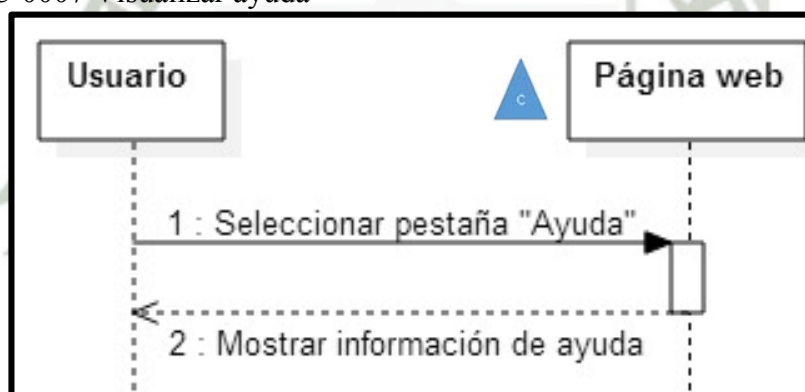
UC-0006 Visualizar preguntas frecuentes



Leyenda:

OBJETO	M/C	DESCRIPCIÓN
Página Web	C	Controlar la correcta visualización de la página web
Base de Datos	C	Controlar la correcta búsqueda de la información

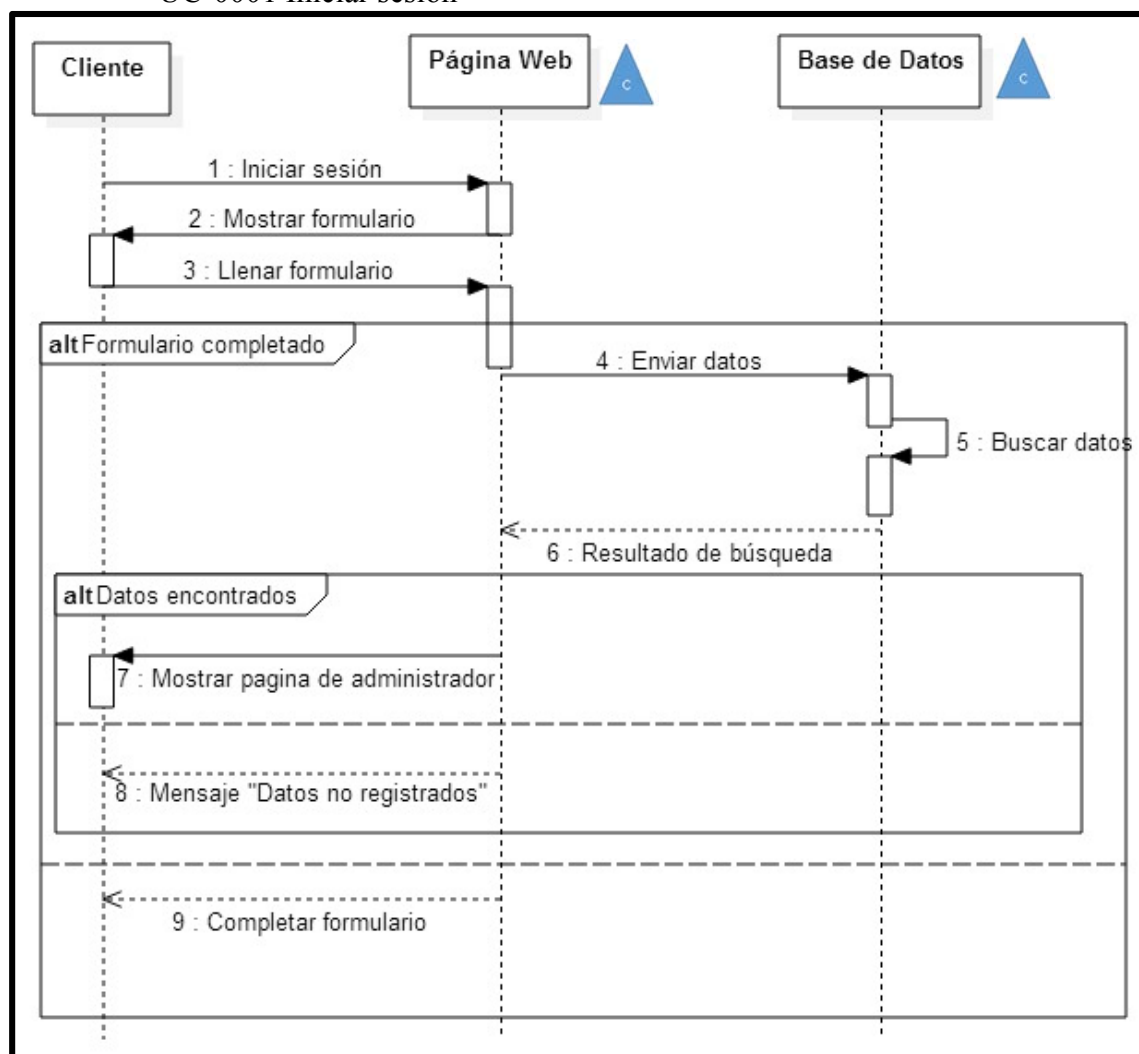
UC-0007 Visualizar ayuda



Leyenda:

OBJETO	M/C	DESCRIPCIÓN
Página Web	C	Controlar la correcta visualización de la página web

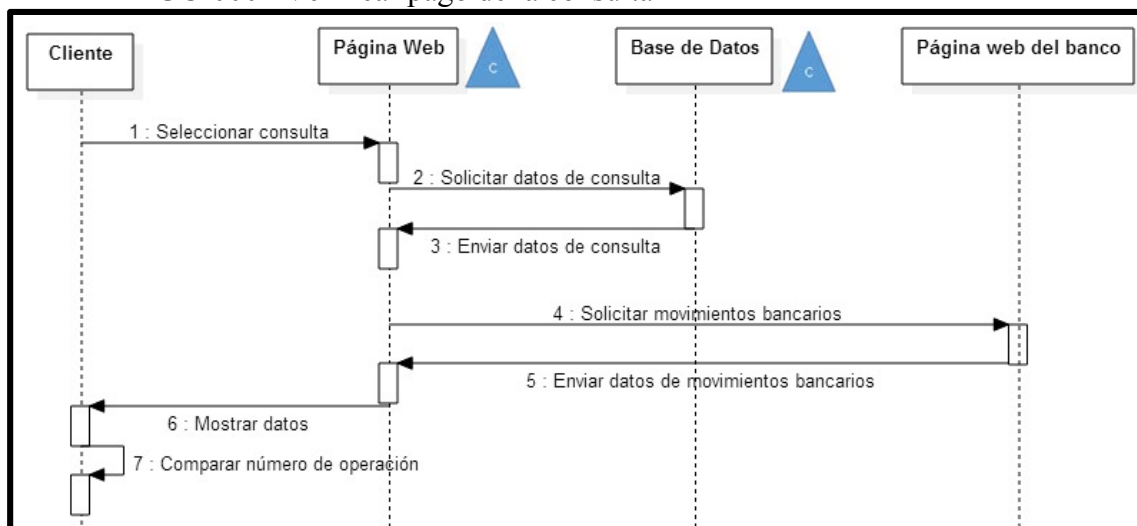
UC-0001 Iniciar sesión



Leyenda:

OBJETO	M/C	DESCRIPCIÓN
Página Web	C	Controlar la correcta visualización de la página web
Base de Datos	C	Controlar la correcta búsqueda de la información

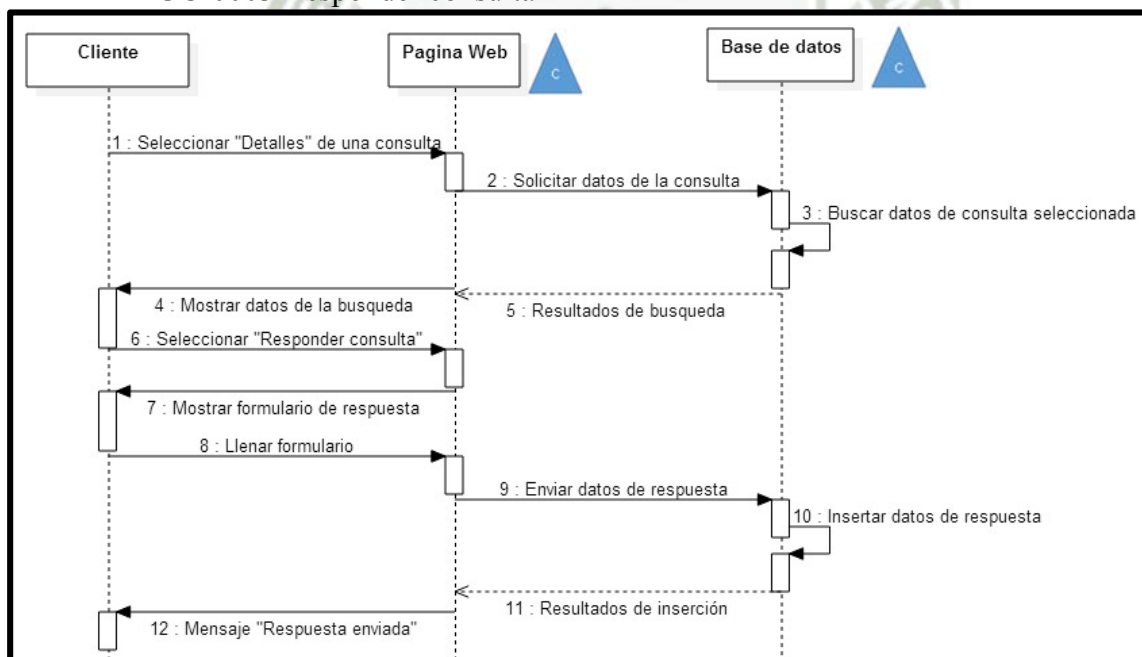
UC-0002 Verificar pago de la consulta



Leyenda:

OBJETO	M/C	DESCRIPCIÓN
Página Web	C	Controlar la correcta visualización de la página web
Base de Datos	C	Controlar la correcta búsqueda de la información

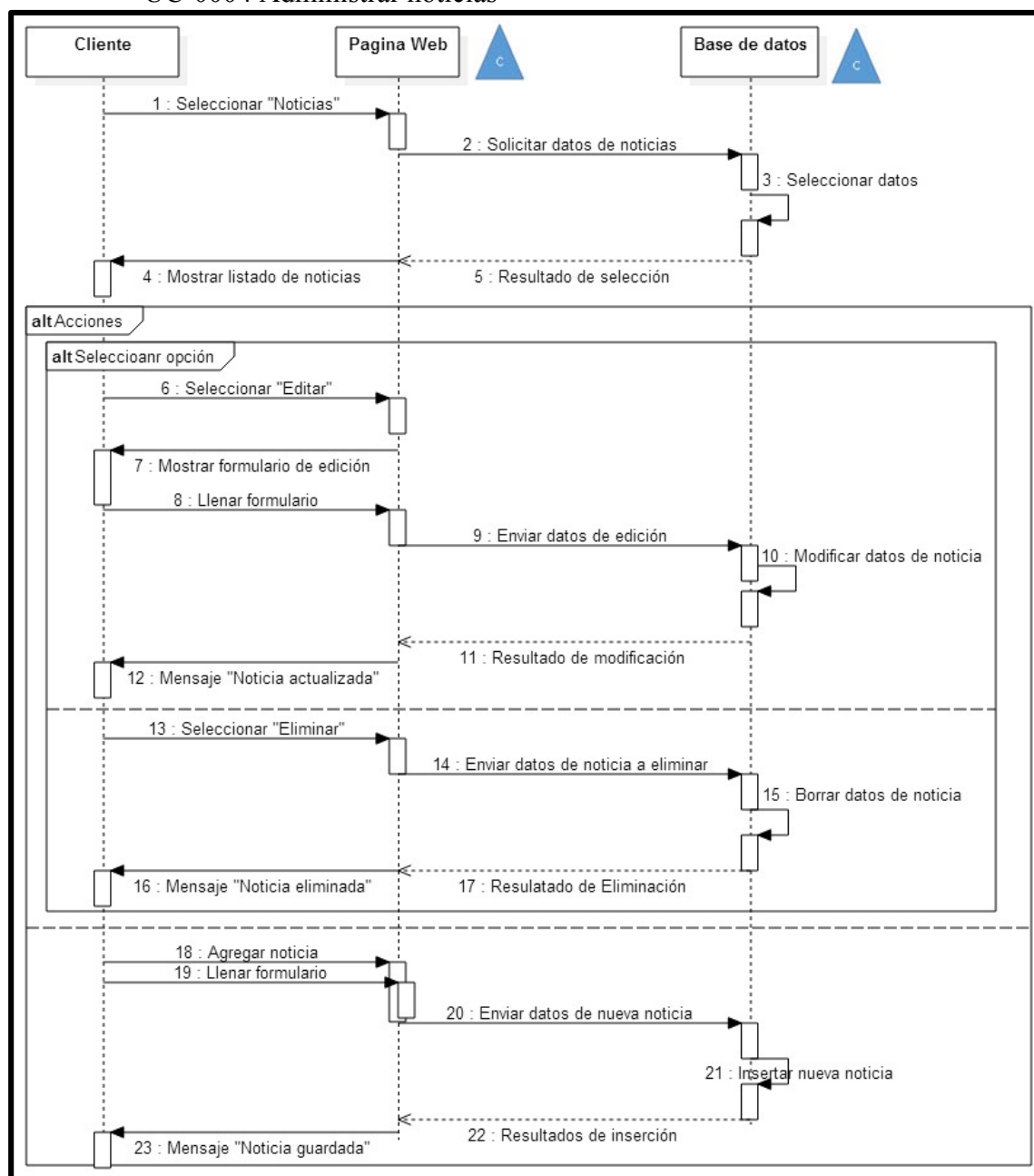
UC-0003 Responder consulta



Leyenda:

OBJETO	M/C	DESCRIPCIÓN
Página Web	C	Controlar la correcta visualización de la página web
Base de Datos	C	Controlar la correcta búsqueda de la información

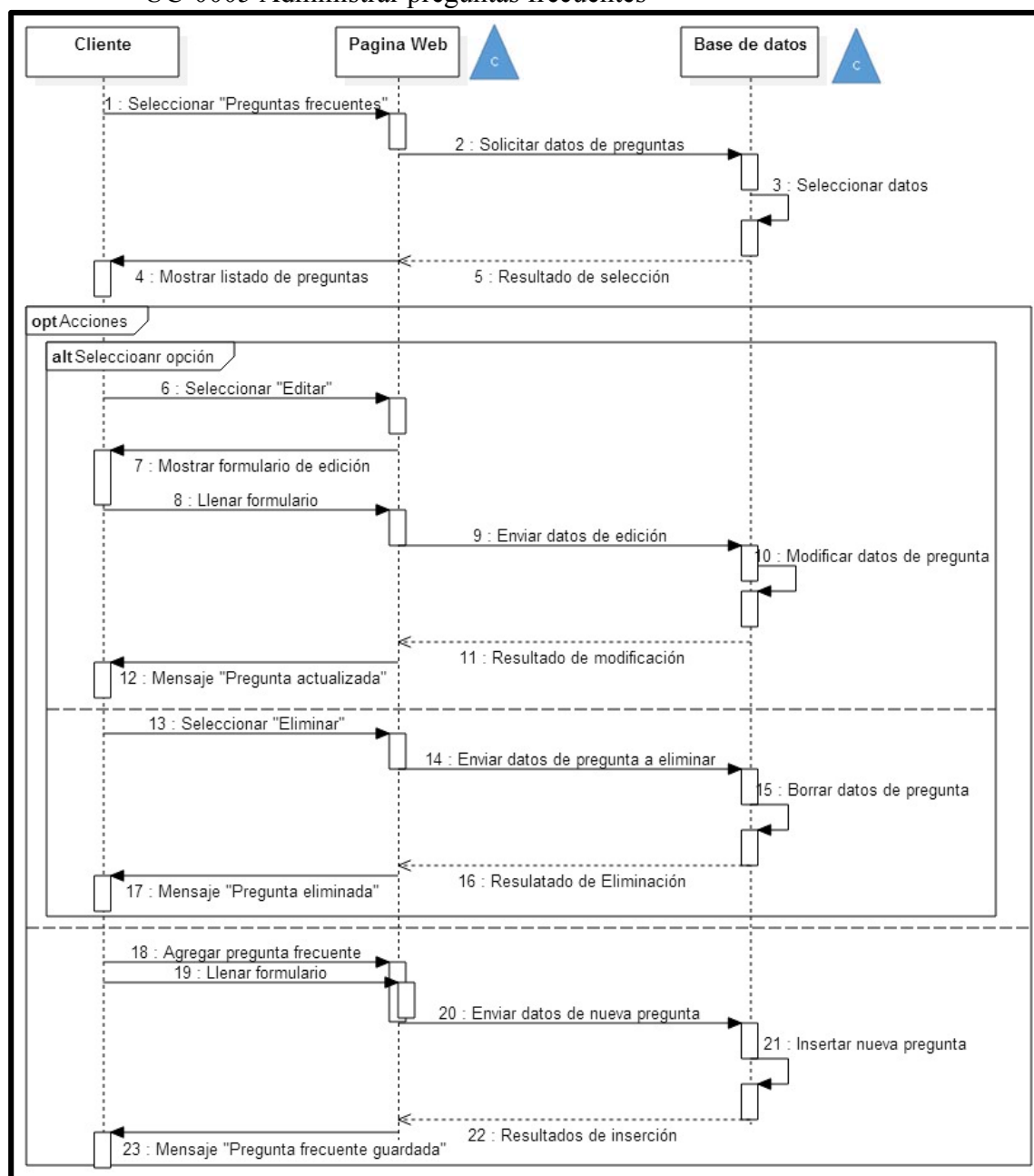
UC-0004 Administrar noticias



Leyenda:

OBJETO	M/C	DESCRIPCIÓN
Página Web	C	Controlar la correcta visualización de la página web
Base de Datos	C	Controlar la correcta búsqueda de la información

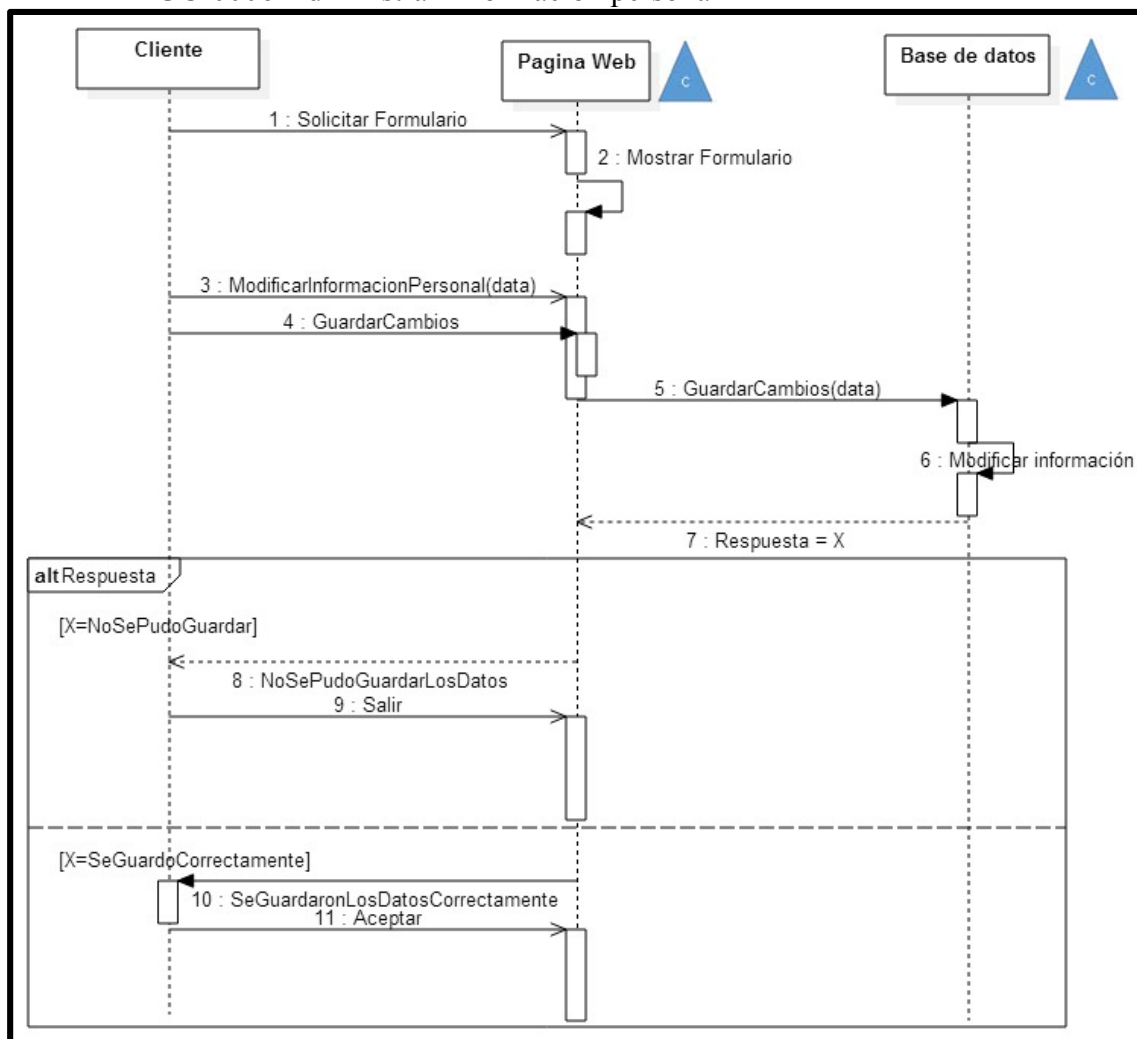
UC-0005 Administrar preguntas frecuentes



Legenda:

OBJETO	M/C	DESCRIPCIÓN
Página Web	C	Controlar la correcta visualización de la página web
Base de Datos	C	Controlar la correcta búsqueda de la información

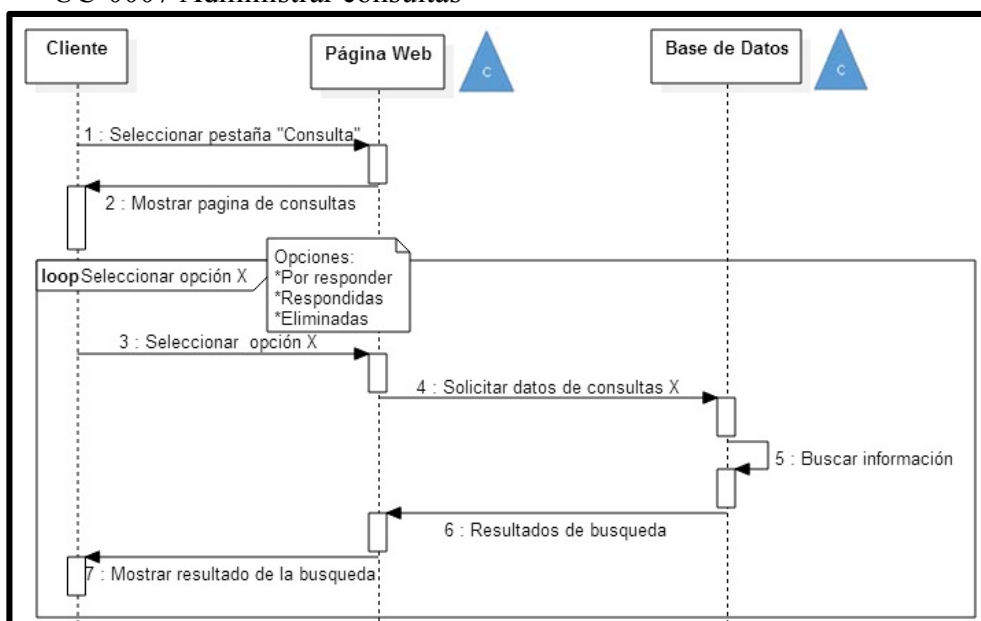
UC-0006 Administrar información personal



Legenda:

OBJETO	M/C	DESCRIPCIÓN
Página Web	C	Controlar la correcta visualización de la página web
Base de Datos	C	Controlar la correcta búsqueda de la información

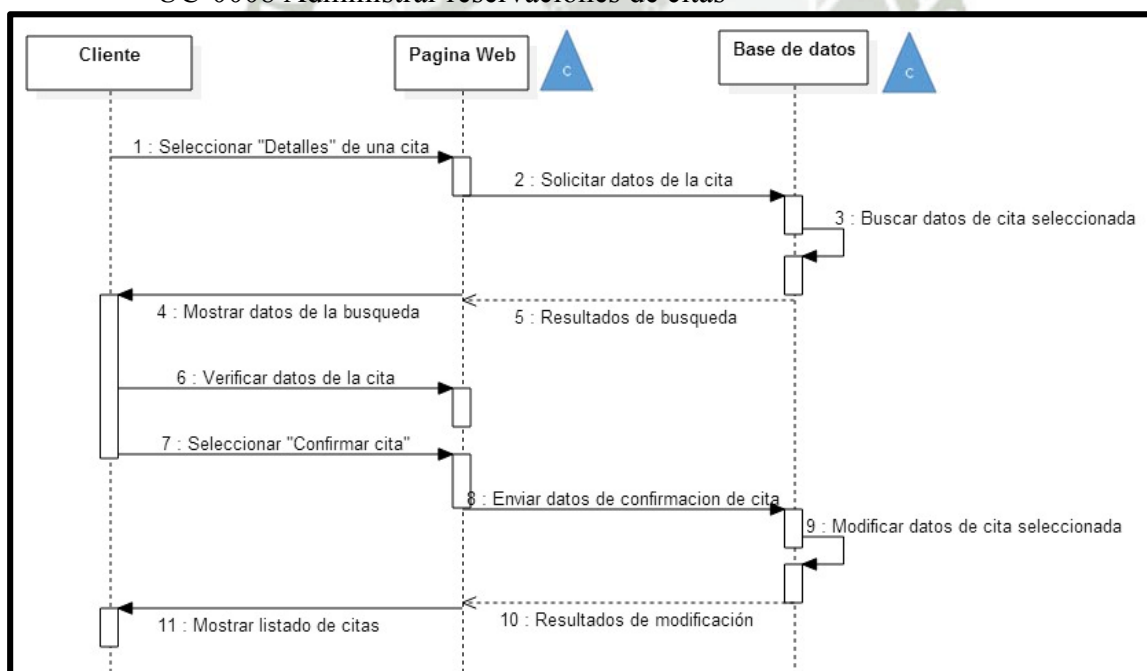
UC-0007 Administrar consultas



Leyenda:

OBJETO	M/C	DESCRIPCIÓN
Página Web	C	Controlar la correcta visualización de la página web
Base de Datos	C	Controlar la correcta búsqueda de la información

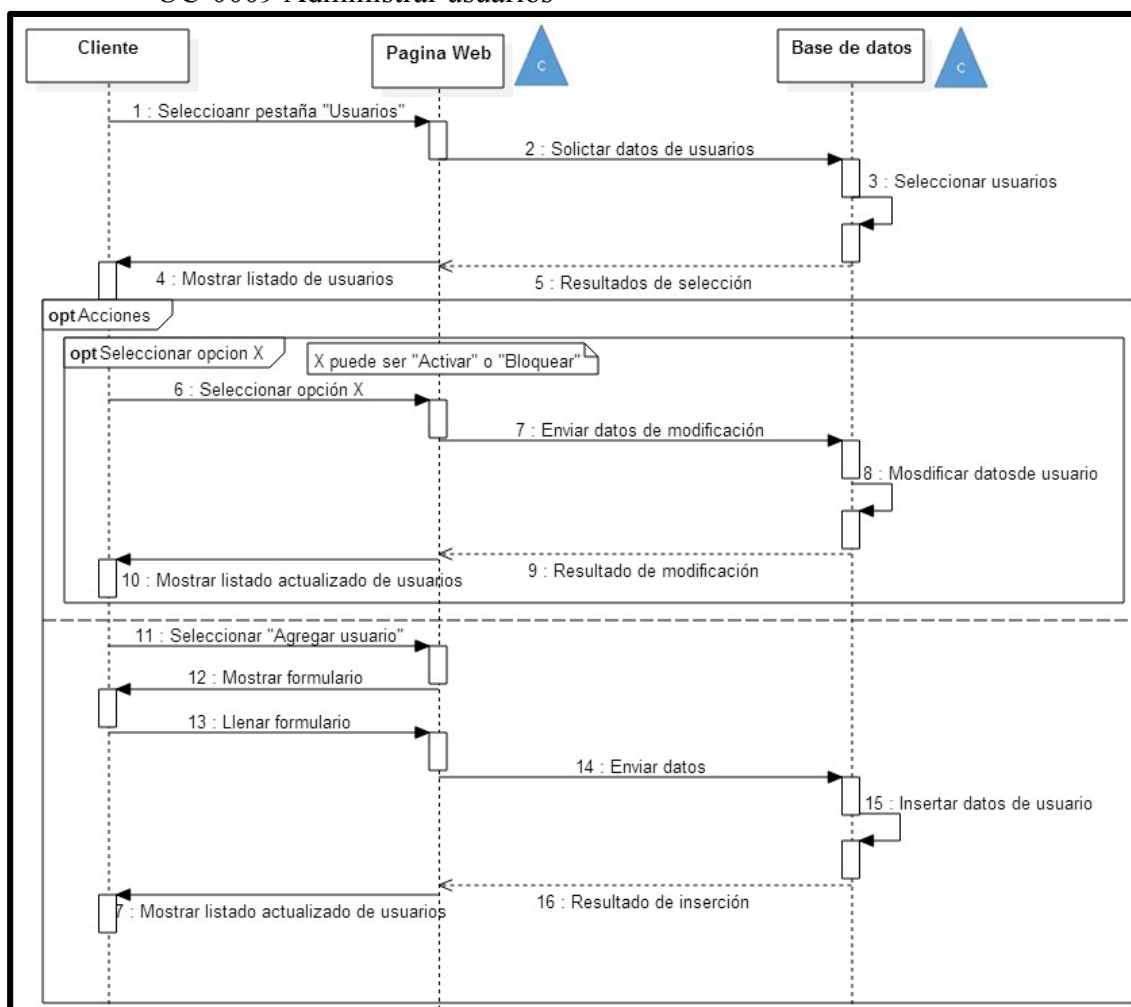
UC-0008 Administrar reservaciones de citas



Leyenda:

OBJETO	M/C	DESCRIPCIÓN
Página Web	C	Controlar la correcta visualización de la página web
Base de Datos	C	Controlar la correcta búsqueda de la información

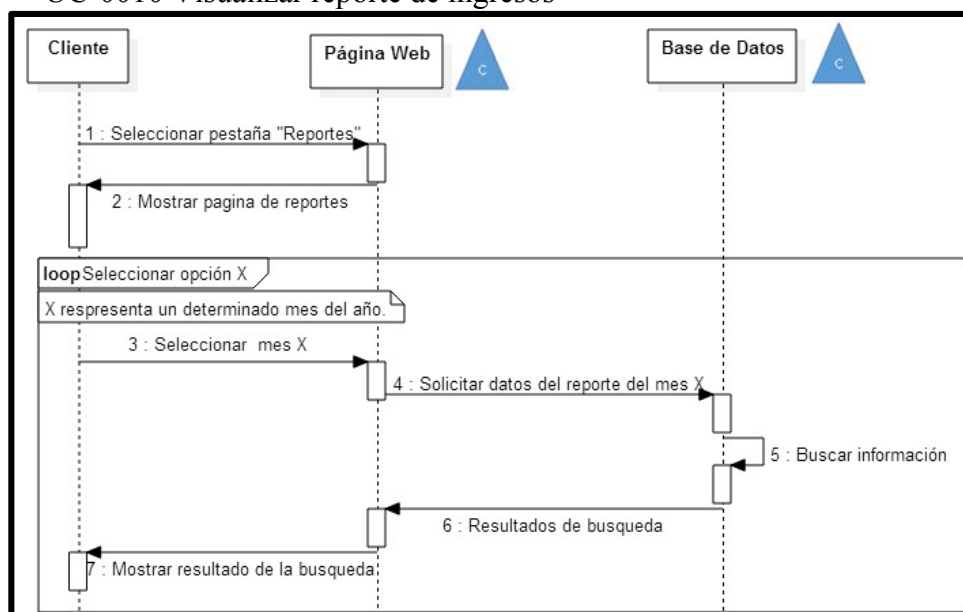
UC-0009 Administrar usuarios



Legenda:

OBJETO	M/C	DESCRIPCIÓN
Página Web	C	Controlar la correcta visualización de la página web
Base de Datos	C	Controlar la correcta búsqueda de la información

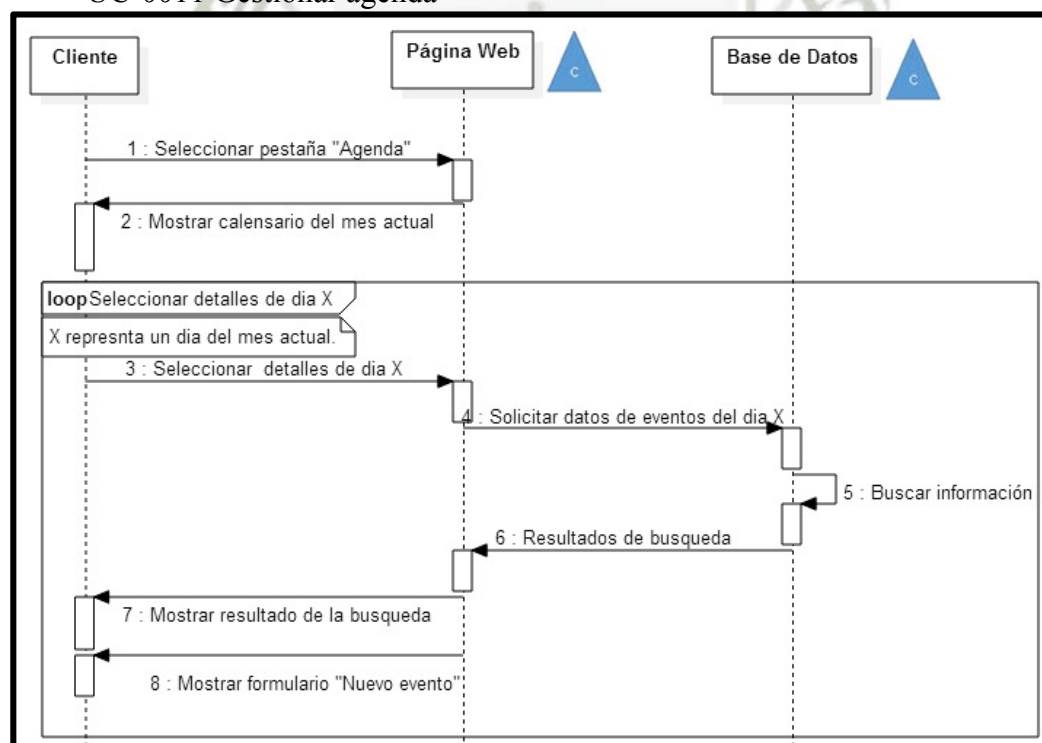
UC-0010 Visualizar reporte de ingresos



Leyenda:

OBJETO	M/C	DESCRIPCIÓN
Página Web	C	Controlar la correcta visualización de la página web
Base de Datos	C	Controlar la correcta búsqueda de la información

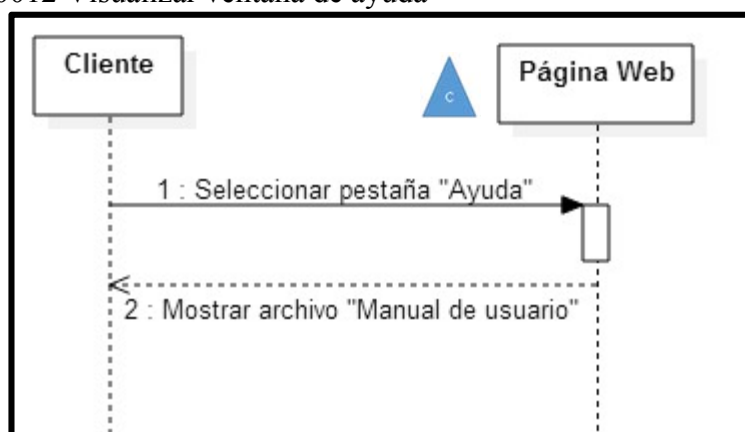
UC-0011 Gestionar agenda



Leyenda:

OBJETO	M/C	DESCRIPCIÓN
Página Web	C	Controlar la correcta visualización de la página web
Base de Datos	C	Controlar la correcta búsqueda de la información

UC-0012 Visualizar ventana de ayuda



Leyenda:

OBJETO	M/C	DESCRIPCIÓN
Página Web	C	Controlar la correcta visualización de la página web

4.2.7 Etapa de diseño de interfaces gráficas de usuario



Leyenda:

ITEM	M/C	DESCRIPCIÓN
Consultas	C	Controlar que el diálogo CONSULTAS sea disparado correctamente
Usuarios	C	Controlar que el diálogo USUARIOS sea disparado correctamente



118 x 88

ESTUDIO JURIDICO VIRTUAL

DAVID DONGO

INICIO
CONSULTAS
CITAS
USUARIOS
REPORTES
AGENDA
AYUDA

CONSUL

Nro Operacion	fecha	Nombre Usuario	Asunto
Content 1	Content 2		
Content 3	Content 4		



RESPONDIDAS

POR RESPONDER



CALENDARIO

✓ Iniciar Sesión

COLOR BEIGE

PREGUNTAS FRECUENTES

NOTICIAS

Leyenda:

ITEM	M/C	DESCRIPCIÓN
Consul	C	Controlar que la información presentada en CONSUL sea correcta



118 x 88

ESTUDIO JURIDICO VIRTUAL

DAVID DONGO

INICIO
CONSULTAS
CITAS
USUARIOS
REPORTES
AGENDA
AYUDA

CONSUL

Nro Operacion	fecha	Nombre Usuario	Asunto
Content 1	Content 2		
Content 3	Content 4		



RESPONDIDAS

POR RESPONDER



CALENDARIO

✓ Iniciar Sesión

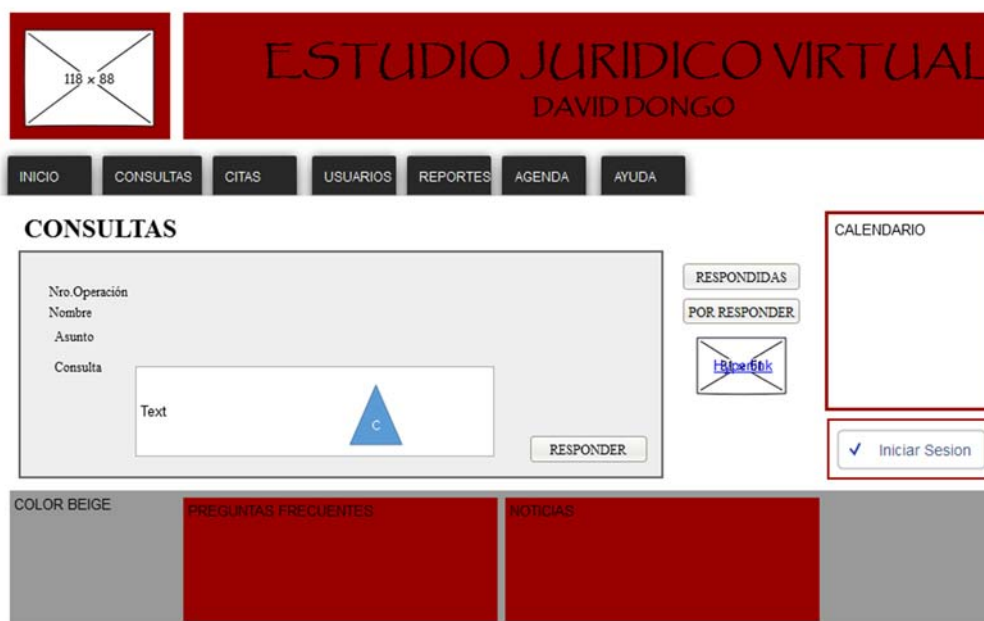
COLOR BEIGE

PREGUNTAS FRECUENTES

NOTICIAS

Leyenda:

ITEM	M/C	DESCRIPCIÓN
Respondidas	C	Controlar que la información respondida concuerde con las solicitudes



Leyenda:

ITEM	M/C	DESCRIPCIÓN
Text	C	Controlar que la información almacenada en Text corresponda a la de las condiciones



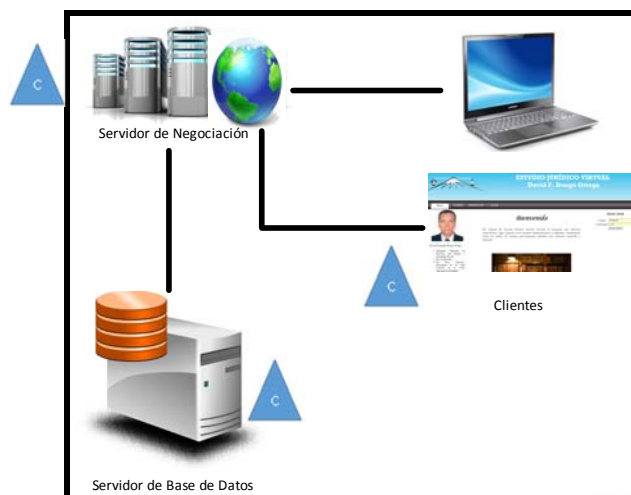
Leyenda:

ITEM	M/C	DESCRIPCIÓN
CITAS	C	Controlar que la información de las citas sean correctas

4.2.8 Etapa de diseño de la arquitectura del software

Considerando que la arquitectura de una aplicación es la vista conceptual de la estructura de esta y que toda aplicación contiene código de presentación, código de procesamiento de datos y código de almacenamiento de datos; la arquitectura de las aplicaciones difieren según como está distribuido este código. Por lo tanto, la arquitectura utilizada para el desarrollo del software es la “Arquitectura en Tres Capas o Tres Niveles”. A continuación se describen las tres capas principales que conforman esta arquitectura:

- **Capa de Presentación:** la que ve el usuario (también se la denomina "capa de usuario"), presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). También es conocida como interfaz gráfica y debe tener la característica de ser "amigable" (entendible y fácil de usar) para el usuario. Esta capa se comunica únicamente con la capa de negocio.
- **Capa de Negocio:** es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos almacenar o recuperar datos de él. También se consideran aquí los programas de aplicación.
- **Capa de Datos:** es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.



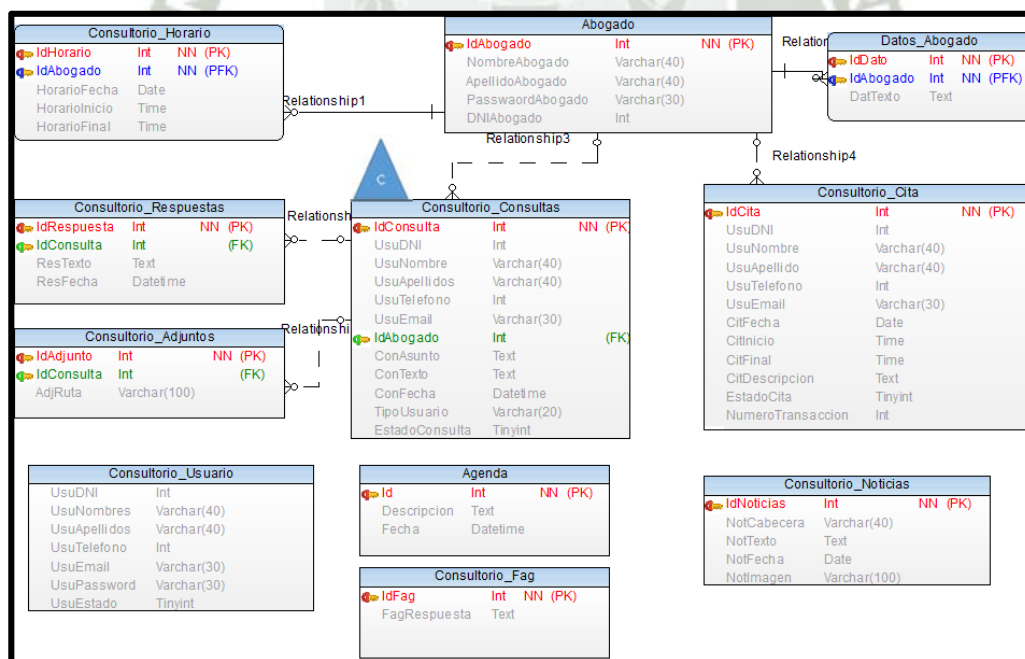
Leyenda:

ITEM	M/C	DESCRIPCIÓN
Servidor de Base de Datos	C	Controlar que el servidor de base de datos funciones de manera adecuada
Clientes	C	Controlar la efectiva comunicación con los clientes

4.2.9 Etapa de diseño de los diagramas de colaboración

En vista de que la implementación del sistema se encuentra orientado a la web, sólo bastó trabajar con los diagramas de secuencias.

4.2.10 Etapa de diseño del diagrama de clases



Leyenda:

ITEM	M/C	DESCRIPCIÓN
Consultorio consultas	C	Controlar que el correcto llenado de la tabla

4.2.11 Medidas e indicadores

Las medidas e indicadores obtenidos se muestran en la siguiente tabla:

Tabla 8

Medidas e indicadores del caso de estudio

Conglomerado	Etapas	Medidas	Indicadores	Parámetro (correcto si)
Ingeniería de Requisitos	Educción de requisitos (métricas, control)	• A = 13 • B = 7	IEd = 53.8%	< 10%
	Elicitación de requisitos (métricas, control)	• A = 37 • B = 6	IEl = 16.2%	< 10%
	Especificación de requisitos de software (métricas, control)	• A = 63 • B = 28	IEs = 44.4%	< 10%
Diseño	Diagrama de casos de uso (métricas, control)	• $\sum A_i = 13$ • $\sum A_i B_i = 188$ • C = 8.3	Iin = 14.5 Icu = 57%	< 15%
	Modelo conceptual (métricas, control)	• A = 7 • B = 3	Imc = 42.8%	< 10%
	Diagrama de secuencias (métricas, control)	• $\sum A_i = 21$ • $\sum A_i B_i = 58$ • C = 0.9	Iin = 2.8 Icu = 32.1%	< 15%
	Interfaces gráficas de usuario (métricas, control)	• A = 5 • B = 3	Iig = 60%	< 10%
	Arquitectura de software (métricas, control)	• A = 4 • B = 2	Iar = 50%	< 10%
	Diagrama de colaboración (métricas, control)	• A = 0 • B = 0 • C = 0	Icu = 0%	< 15%
	Diagrama de clases (métricas, control)	• A = 11 • B = 4	Icl = 33.6%	< 10%

Fuente: Elaborado por el autor

CONCLUSIONES

- PRIMERA:** La propuesta sobre la ampliación de UML para representar métricas y control en la fase de diseño ha permitido identificar, de manera coherente, aquellos componentes que necesiten de un control o de alguna métrica, permitiendo hacer correcciones en esta fase y sin permitir que estos errores continúen en la siguiente fase de construcción del producto de software.
- SEGUNDA:** El marco de trabajo propuesto no degenera los artefactos propuestos por UML, más aún, estos quedan enriquecidos por las necesidades de control de cada uno de los componentes. Las funcionalidades de los artefactos originales son respetadas en su integridad.
- TERCERA:** La simbología agregada a los artefactos propuestos por UML agrega claridad a las necesidades del que desea un mayor control en el proceso de software. Esto permite, que consecuentemente se preste una mayor atención a aquellos componentes marcados.
- CUARTA:** La ampliación de los elementos del Lenguaje de Modelado Unificado permiten incluir indicadores que nos permiten llevar a cabo un mejor control de la calidad del producto a construir en función de las medidas previamente definidas para el control del proyecto de desarrollo de software.
- QUINTA:** El caso de estudio muestra que la obtención de medidas y control es mucho más real logrando un entendimiento definitivo de las necesidades de control del producto a construir

RECOMENDACIONES

PRIMERA: Se recomienda insertar estos cambios en los artefactos del proceso de desarrollo para lograr observar el comportamiento de la construcción del producto de software en sus siguientes fases.

SEGUNDA: Se recomienda automatizar la ampliación UML para poder ser usado desde este ámbito de trabajo, permitiendo tener los artefactos reales y alternativos en un mismo ambiente.



BIBLIOGRAFÍA

- (ARIAS, 2013) Arias Barragán Luís & Fernando García Diego. Modelo de gestión de redes de distribución mediante UML y BPL. Revista Tecnura Volumen 17 Numero 37 paginas 109 - 120 Julio - Septiembre de 2013.
- (BOOCH, 2005) Grady Booch, James Rumbaugh, Ivar Jacobson. El Lenguaje Unificado de Modelado, Addison Wesley. Segunda edición. 2005.
- (GACITUA, 2003) Gacitúa Bustos Ricardo. Métodos de desarrollo de software: el desafío pendiente de la estandarización. Revista Theoria, Vol. 12: 23-42, 2003 ISSN 0717-196X. 2003.
- (JACOBSON, 2000) Jacobson, I., Booch, G., Rumbaugh J., El Proceso Unificado de Desarrollo de Software, Addison Wesley 2000.
- (MORENO, 2006) Moreno Espino Mailyn & Rosete Suárez Alejandro. Ingeniería de software orientada a agentes: roles y metodologías. Revista de Informática. Vol. XXVII No. 2-3. Facultad de Ingeniería. CUJAE. Cuba. 2006
- (PEREZ, 2006) Pérez Lavalle Sonia & Orejas Valdés Fernando. Extensión del metamodelo de UML para una arquitectura de componentes. Revista Informática. Vol. XXVII. N° 2. Facultad de Ingeniería Industrial. CUJAE. Cuba. 2006.
- (PRESSMAN, 2002) Pressman, Richard. Ingeniería del Software: Un enfoque práctico. Sétima edición. McGraw Hill 2014.
- (RODRIGUEZ, 2009) Rodríguez Rosseline y Goncalves Marlene. Perfil UML para el modelado visual de requisitos difusos. Enl@ce: Revista Venezolana de Información, Tecnología y Conocimiento ISSN: 1690-7515 Depósito legal pp 200402ZU1624 Año 6: No. 3, Septiembre-Diciembre 2009, pp. 29-46. 2009.
- (RUELAS, 2013) Ruelas Llerena Mary Victoria. Propuesta de un Modelo de Evaluación del Proceso de Desarrollo de Software para la pequeña empresa de Arequipa. Tesis para optar el grado académico de Magister en Ingeniería de Sistemas. Universidad Católica Santa María. 2013.
- (SOMMERVILLE, 2005) Sommerville, Ian. Ingeniería de Software. Editorial Pearson Educación. Sétima edición. 2005.

- (TABARES, 2008) Tabares Martha & Pineda Juan. Un patrón de interacción entre diagramas de actividades UML y sistemas workflow. Revista EIA, ISSN 1794-1237 Número 10, p. 105-120. Escuela de Ingeniería de Antioquia, Medellín (Colombia). 2008.
- (ZAPATA, 2008) Zapata Carlos & Gonzáles Guillermo. Especificación formal en OCL de reglas de consistencia entre los diagramas de clases y casos de uso de UML y el modelo de interfaces. Revista Ingenierías. Universidad de Medellín. 2008.
- (ZAPATA, 2009) Zapata Carlos & Giraldo Gloria. Aproximación a una ontología para lenguajes de modelado gráfico. Revista de Ingeniería. Universidad de los Andes. Bogotá, Colombia. rev. ing. ISSN. 0121-4993. mayo de 2009.



ANEXO N° 1

ESPECIFICACION DE LOS CASOS DE USO DEL CASO DE ESTUDIO

- Especificaciones de Usuario

UC-0001	Visualizar interfaz de la aplicación.	
Versión	1.0 (20/11/2014)	
Autores	Grupo de trabajo	
Fuentes		
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario ingrese a la página web, podrá visualizar las diferentes interfaces de la aplicación; <i>tales como: interfaz de inicio, noticias, preguntas frecuentes; entre otras.</i>	
Precondición	Ingresar a la página web	
Secuencia normal	Paso	Acción
	~	~
Postcondición	Mostrar la pantalla principal.	
Importancia	Vital	
Urgencia	Inmediatamente	
Estado	en construcción	
Estabilidad	Alta	
Comentarios	Ninguno	

UC-0002	Iniciar Sesión-Usuario	
Versión	1.0 (29/10/2014)	
Autores	Grupo de trabajo	
Fuentes		
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario acceda a la página web y desee iniciar sesión, l deberá llenar un formulario de manera correcta.	
Precondición	El usuario debe haber sido registrado satisfactoriamente.	
Secuencia normal	Paso	Acción
	~	~
Postcondición	El usuario podrá acceder a todos los servicios brindados por la página web como usuario registrado.	
Excepciones	Paso	Acción
	~	~
Importancia	Importante	
Urgencia	Inmediatamente	
Estado	en construcción	
Estabilidad	Media	

Comentarios	Ninguno
-------------	---------

UC-0003	Realizar consulta	
Versión	1.0 (28/10/2014)	
Autores	Grupo de trabajo	
Fuentes		
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario registrado o no registrado podrá realizar consultas.</i> <i>En el caso del usuario registrado: Sólo deberá iniciar sesión de tal manera que cuando realice una consulta; este solo deba llenar los datos de la consulta.</i> <i>En el caso del usuario no registrado: Deberá llenar campos obligatorios, los cuales serán: número de operación de pago, nombre y su correo electrónico.</i>	
Precondición	Haber realizado el pago correspondiente a la consulta	
Secuencia normal	Paso	Acción
	~	~
Postcondición	Recibir mensaje de confirmación de envío de consulta.	
Excepciones	Paso	Acción
	~	~
Importancia	Vital	
Urgencia	Inmediatamente	
Estado	en construcción	
Estabilidad	Media	
Comentarios	Si el usuario lo desea; puede adjuntar archivos.	

UC-0004	Reservar cita	
Versión	1.0 (28/10/2014)	
Autores	Grupo de trabajo	
Fuentes		
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario desee contactarse directamente con el cliente podrá realizar una reservación de una cita; llenando un formulario de identificación con los siguientes campos obligatorios: DNI, nombres, apellidos, teléfono, dirección, e-mail, asunto de reservación, fecha de la reservación y hora sugerida.	
Precondición	Acceder a la sección de reservas de citas.	
Secuencia normal	Paso	Acción
	~	~
Postcondición	Recibir un mensaje de confirmación de reservación.	

Excepciones	Paso	Acción
	~	~
Importancia	Vital	
Urgencia	Inmediatamente	
Estado	En construcción	
Estabilidad	Alta	
Comentarios	Ninguno	

UC-0005	Visualizar noticias	
Versión	1.0 (28/10/2014)	
Autores	Grupo 06	
Fuentes		
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>El usuario al momento de ingresar a la página web, podrá visualizar las noticias actuales con respecto al ámbito jurídico, las cuales serán extraídas de fuentes de información como: Diario "El Peruano", "El tribunal constitucional".</i>	
Precondición	Ingresar al enlace de noticias.	
Secuencia normal	Paso	Acción
	~	~
Excepciones	Paso	Acción
	~	~
Importancia	Quedaría bien	
Urgencia	Puede esperar	
Estado	en construcción	
Estabilidad	Media	
Comentarios	Ninguno	

UC-0006	Visualizar preguntas frecuentes	
Versión	1.0 (29/10/2014)	
Autores	Grupo de trabajo	
Fuentes		
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>El usuario ingrese a la página web, podrá visualizar la sección de "Preguntas frecuentes", ésta sección presentará un grupo de preguntas que serán seleccionadas por el Dr. David Dongo.</i>	
Precondición	Ingresar al enlace de preguntas frecuentes.	
Secuencia normal	Paso	Acción

	~	~
Excepciones	Paso	Acción
	~	~
Importancia	Quedaría bien	
Urgencia	Puede esperar	
Estado	En construcción	
Estabilidad	Media	
Comentarios	Ninguno	

UC-007	Visualizar ayuda	
Versión	10 (28/10/2014)	
Autores	Grupo de trabajo	
Fuentes		
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario acceda a la sección de ayuda; se debe mostrar pautas necesarias para el uso correcto de la página web; de tal manera que oriente al usuario.</i>	
Precondición	Acceder a la sección de ayuda.	
Secuencia normal	Paso	Acción
	~	~
Excepciones	Paso	Acción
	~	~
Frecuencia esperada	10 veces por día(s)	
Importancia	Vital	
Urgencia	Inmediatamente	
Estado	en construcción	
Estabilidad	Alta	
Comentarios	Ninguno	

• Especificaciones de Administrador o Cliente

UC-001	Iniciar sesión	
Versión	10 (20/11/2014)	
Autores	Grupo de trabajo	
Fuentes		
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>El cliente, en este caso el Dr. David Dongo, deberá iniciar sesión, para lograr validar su entrada</i>	

	<i>a la página web. De tal forma que pueda visualizar las diferentes opciones para la gestión como administrador de la aplicación.</i>	
Precondición	Ingresar a la página web y seleccionar la opción de inicio de sesión.	
Secuencia normal	Paso	Acción
	~	~
Postcondición	Visualizar las diferentes a administrar.	
Excepciones	Paso	Acción
	~	~
Importancia	Vital	
Urgencia	Inmediatamente	
Estado	en construcción	
Estabilidad	Muy alta	
Comentarios	Ninguno	

UC-002	Verificar pago de la consulta	
Versión	10 (28/10/2014)	
Autores	Grupo de trabajo	
Fuentes		
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>se realice una consulta, el sistema en colaboración con el cliente verificarán que los datos, por ejemplo, el número de operación, sean correctos para proceder a emitir una respuesta de la consulta.</i>	
Precondición	Llenar correctamente los datos solicitados previos al envío de una nueva consulta y haber realizado el pago correspondiente.	
Secuencia normal	Paso	Acción
	~	~
Postcondición	Responder a la consulta recibida.	
Excepciones	Paso	Acción
	~	~
Frecuencia esperada	10 veces por día(s)	
Importancia	Vital	
Urgencia	Inmediatamente	
Estado	en construcción	
Estabilidad	Alta	
Comentarios	Ninguno	

UC-003	Responder consulta	
Versión	1.0 (28/10/2014)	
Autores	Grupo de trabajo	
Fuentes		
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>se reciba una nueva consulta y esta cumpla con los requisitos necesarios; se deberá emitir una respuesta a dicha consulta al correo electrónico del usuario.</i>	
Precondición	Haber recibido, leído y verificado el pago de la consulta.	
Secuencia normal	Paso	Acción
	~	~
Postcondición	Emitir mensaje de confirmación de envío de respuesta.	
Excepciones	Paso	Acción
	~	~
Frecuencia esperada	10 veces por día(s)	
Importancia	Vital	
Urgencia	Inmediatamente	
Estado	en construcción	
Estabilidad	Alta	
Comentarios	Ninguno	

UC-004	Administrar noticias	
Versión	1.0 (19/11/2014)	
Autores	Grupo de trabajo	
Fuentes		
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el cliente ingrese a la página web, podrá modificar las noticias que se muestran en dicha página, lo que implicaría añadir o eliminar algunas noticias existentes.</i>	
Precondición	Ingresar a la opción de noticias como administrador.	
Secuencia normal	Paso	Acción
	~	~
Postcondición	Actualizar datos si es que se ha realizado algún cambio.	
Excepciones	Paso	Acción
	~	~
Importancia	Vital	
Urgencia	Inmediatamente	
Estado	en construcción	

Estabilidad	Media	
Comentarios	Ninguno	
UC-005	Administrar preguntas frecuentes	
Versión	10 (19/11/2014)	
Autores	Grupo de trabajo	
Fuentes		
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el cliente ingrese a la página web a la sección preguntas frecuentes, podrá modificar las preguntas frecuentes que se muestran en dicha página, lo que implicaría añadir o eliminar algunas preguntas existentes.</i>	
Precondición	Ingresar a la opción de preguntas frecuentes como administrador.	
Secuencia normal	Paso	Acción
	~	~
Postcondición	Actualizar datos si es que se ha realizado algún cambio.	
Excepciones	Paso	Acción
	~	~
Importancia	Vital	
Urgencia	Inmediatamente	
Estado	en construcción	
Estabilidad	Media	
Comentarios	Ninguno	

UC-006	Administrar información personal	
Versión	10 (19/11/2014)	
Autores	Grupo de trabajo	
Fuentes		
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el cliente ingrese a la sección correspondiente a la información personal, podrá modificar su información, de acuerdo a lo que él crea por conveniente.</i>	
Precondición	Ingresar a la opción de información personal como administrador.	
Secuencia normal	Paso	Acción
	~	~
Postcondición	Actualizar datos si es que se ha realizado algún cambio.	
Excepciones	Paso	Acción
	~	~
Importancia	Vital	
Urgencia	inmediatamente	

Estado	en construcción
Estabilidad	Alta
Comentarios	No se puede eliminar la fotografía; solo modificarla.

UC-007	Administrar consultas	
Versión	1.0 (20/11/2014)	
Autores	Grupo de trabajo	
Fuentes		
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el cliente ingrese a la sección de reservaciones de citas; podrá visualizar las mismas en diferentes categorías; tales como: respondidas y por responder. Asimismo, se contará con una categoría en la que se encontrarán las consultas con alguna observación (Consultas que no han sido pagadas completamente u otros aspectos).</i>	
Precondición	Acceder a la sección de consultas como administrador.	
Secuencia normal	Paso	Acción
	~	~
Postcondición	Actualizar estados de la consulta.	
Excepciones	Paso	Acción
	~	~
Importancia	Importante	
Urgencia	Inmediatamente	
Estado	en construcción	
Estabilidad	Media	
Comentarios	Ninguno	

UC-008	Administrar reservaciones de citas	
Versión	1.0 (19/11/2014)	
Autores	Grupo de trabajo	
Fuentes		
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el cliente ingrese a la sección de reservaciones de citas; podrá visualizar las mismas en diferentes categorías; tales como: respondidas y por responder.</i>	
Precondición	Acceder a la sección de reservaciones de citas.	
Secuencia normal	Paso	Acción
	~	~
Postcondición	Actualizar estados de la consulta.	
Excepciones	Paso	Acción

	~	~
Importancia	Importante	
Urgencia	inmediatamente	
Estado	en construcción	
Estabilidad	Alta	
Comentarios	Ninguno	

UC-009	Administrar usuarios	
Versión	1.0 (19/11/2014)	
Autores	Grupo de trabajo	
Fuentes		
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>El cliente acceda a la sección de usuarios, podrá agregar, actualizar información un usuario. Asimismo, se podrá suspender o activar la cuenta de un usuario.</i>	
Precondición	Acceder a la sección de administración de usuarios.	
Secuencia normal	Paso	Acción
	~	~
Postcondición	Actualizar datos si es que se ha realizado algún cambio.	
Excepciones	Paso	Acción
	~	~
Importancia	Vital	
Urgencia	Inmediatamente	
Estado	en construcción	
Estabilidad	Alta	
Comentarios	La suspensión o activación de la cuenta de un usuario registrado será de acuerdo a las reglas de negocio del cliente.	

UC-0010	Visualizar reporte de ingresos	
Versión	1.0 (20/11/2014)	
Autores	Grupo de trabajo	
Fuentes		
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el cliente ingrese a la opción de reportes; se deberá mostrar diferentes tipos de reportes.</i>	
Secuencia normal	Paso	Acción

	-	-
Importancia	Vital	
Urgencia	Inmediatamente	
Estado	en construcción	
Estabilidad	Alta	
Comentarios	Ninguno	

UC-0011	Gestionar agenda	
Versión	10 (20/11/2014)	
Autores	Grupo de trabajo	
Fuentes		
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el cliente acceda a la opción agenda, podrá distribuir su tiempo de tal manera que pueda ver un calendario en el cual pueda agregar eventos en una determinada fecha; así como también actualizarlos o eliminarlos. Asimismo podrá visualizar las citas reservadas que tenga programadas.</i>	
Precondición	Acceder a la sección de la agenda.	
Secuencia normal	Paso	Acción
	-	-
Postcondición	Actualizar los datos a ser mostrados en el calendario.	
Excepciones	Paso	Acción
	-	-
Importancia	Vital	
Urgencia	Inmediatamente	
Estado	en construcción	
Estabilidad	Media	
Comentarios	Ninguno	

UC-0012	Visualizar ventana de ayuda	
Versión	10 (20/11/2014)	
Autores	Grupo de trabajo	
Fuentes		
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el cliente ingrese a la sección ayuda, podrá ver información de ayuda que le permitirá tener una mejor guía en lo que respecta al uso de la página web, y así evitar algunas posibles confusiones.</i>	
Precondición	Ingresar a la opción "Ayuda".	

Secuencia normal	Paso	Acción
	~	~
Excepciones	Paso	Acción
	~	~
Importancia	quedaría bien	
Urgencia	puede esperar	
Estado	Validado	
Estabilidad	Baja	
Comentarios	Ninguno	



ANEXO N° 2

EMPRESAS DE DESARRO DE SOFTWARE EN REQUIPA

RELACIÓN DE EMPRESAS DE DESARROLLO DE SOFTWARE EN AREQUIPA¹

N°	Empresa	Nombre	Cargo
1	Infinito	Johnny Moscoso Rossel	Consultor SMP
2	Colca Lodge	Mateo Castro García	Jefe de Sistemas
3	EBS Solutions SA	Seef Alexander Sheen Cáceres	Desarrollador Senior
4	Autoridad Administrativa del agua caplina Ocoña	Jair Chacón Villegas	Jefe de desarrollo SW
5	Atix Plus SAC	Javier Gutiérrez Mamani	Sub gerente
6	Caja IncaSur	René Nieto Valencia	Analista de sistemas
7	Cooperativa de ahorro y crédito del sur	Jean Carlo Zamudio La Rosa	Analista de sistemas
8	Adia Soft	Efraín Gonzáles Huarhua	Gerente General
9	Fondesurco		
10	Compartamos	Alex Rony Cruz Manchego	Especialista GD
11	Apoyo total		
12	Incalpaca	Mateo Castro García	Jefe de Sistemas
13	MiBanco	Gregory León Ballón	Developer
14	Tisur		
15	NextSoft	Mario Rodriguez	Gerente
16	LP analistas y consultores		
17	Rico pollo	Diosalina Salas Pumacayo	Analista desarrollador
18	De Larrobla y asociados	Luis Carlos Mejía García	Analista Programador
19	LSCV Systems EIRL	Guztavo Benavides Hernández	Gerente de TI
20	Infonova Perú EIRL	Aldo Zegarra Gago	Ingeniero de soluciones informáticas

¹ (RUELAS, 2013) Ruelas Llerena Mary Victoria. Propuesta de un Modelo de Evaluación del Proceso de Desarrollo de Software para la pequeña empresa de Arequipa