



Facultad 1

Instrumento para medir la satisfacción de los clientes de la Universidad de las Ciencias Informáticas

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autor:

Cristhiam D. López Nuñez

Tutores:

Ing. Mayleidis López Fernández

Lic. Raynel Batista Tellez

La Habana, junio de 2021

“Año 63 de la Revolución”

Agradecimientos

A todos mis profesores y docentes, y en especial a mis tutores por su ayuda, paciencia y dedicación que brindaron en todo momento. Agradecerle también a toda mi familia por darme ánimo durante este proceso y apoyarme sin descanso. A todas las personas que me apoyaron e hicieron posible que este trabajo se realizara con éxito, y en especial a todos mis amigos que me acompañaron estos últimos cinco años y que dieron cada paso a mi lado y se encuentran junto a mí en este momento.

Dedicatoria

El esfuerzo y la dedicación en una carrera universitaria es un ejemplo y consecuencia de las personas que están detrás. El esfuerzo realizado en este trabajo de investigación va principalmente dedicado a mis padres, por el apoyo a mi educación y por ser los que siempre estuvieron conmigo. A mis tutores, compañeros de estudio, profesores y personal educativo que fueron los que me acompañaron en cada paso que di. A todos los amigos que hoy considero familia, a todos los que están aún a mi lado y a los que por diversas razones no están presentes, pero los llevo siempre conmigo.

Declaración de Autoría

El autor del trabajo de diploma con título ***“Instrumento para medir de la satisfacción de los clientes de la Universidad de las Ciencias Informáticas”*** concede a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declara como único autor de su contenido. Para que así conste firmamos la presente a los 7 días del mes de diciembre del año 2021.

Cristhiam D. López Nuñez



Firma del Autor

Mayleidis López Fernández



Firma del Tutor

Raynel Batista Tellez



Firma del Tutor

Resumen

En la era actual la informática ha avanzado a pasos agigantados, diariamente se generan un sinnúmero de productos y servicios informáticos destinados a satisfacer las necesidades de clientes específicos. La Universidad de las Ciencias Informáticas trabaja constantemente en la informatización de la sociedad cubana y es un eslabón clave en el desarrollo del país. Como parte de su ideología, la conformidad del cliente con el producto no es una opción, sino, un requisito, por lo que se generan innumerables encuestas de satisfacción para mejorar la calidad de los servicios y productos a desarrollar. La universidad no cuenta con un sistema para la medición de dicha satisfacción, por tanto, el objetivo principal de este trabajo se centra en desarrollar una aplicación informática que cubra dicha necesidad. Esta aplicación cubrirá el almacenamiento, gestión y procesamiento de encuestas realizadas a clientes sobre los productos que ofrece la institución. Se realizará una investigación del marco teórico referencial y se seleccionará la metodología de desarrollo de software más acorde a la aplicación a desarrollar. Se desarrollará el análisis, diseño y levantamiento de requisitos, así como las respectivas pruebas de software para validar la calidad del producto final.

Palabras Clave: gestión, procesamiento, recopilación, encuesta, cliente, sistema.

Abstract

In the current era, computing has advanced by leaps and bounds, daily countless computer products and services are generated to meet the needs of specific customers. The University of Informatics Sciences is constantly working on the computerization of Cuban society and is a key link in the development of the country. As part of its ideology, customer compliance with the product is not an option, but a requirement, which is why countless satisfaction polls are generated to improve the quality of services and products to be developed. The university does not have a system for measuring this satisfaction, therefore, the main objective of this work is focused on developing a computer application that meets this need. This application will cover the storage, management and processing of customer polls on the products offered by the institution. An investigation of the theoretical framework will be carried out and the software development methodology most in line with the application to be developed will be selected. The analysis, design and requirements survey will be developed, as well as the respective software tests to validate the quality of the final product.

Keywords: management, processing, collection, poll, customer, system.

Índice

Introducción	1
Situación Problemática	1
Problema de Investigación.....	1
Objeto de estudio	1
Campo de acción	1
Objetivo General	1
Hipótesis	2
Tareas de Investigación	2
Métodos de Investigación	2
Resultados Esperados	2
Capítulo 1. Fundamentos y referentes teórico-metodológicos sobre el objeto de estudio.	3
1.1 Conceptos fundamentales asociados al dominio del problema	3
1.1.1 Encuesta	3
1.1.2 Satisfacción del Cliente.....	3
1.1.3 Calidad	4
1.1.4 Cuestionario	4
1.2 Soluciones Informáticas Similares	4
1.2.1 SoftExpert Encuesta	4
1.2.2 CRR Marketing – Consultoría y formación.....	5
1.2.3 Orchestra 7	6
1.2.4 Especificación de Soluciones Similares.....	6
1.3 Tecnologías de Desarrollo	7
1.3.1 Frontend.....	7
1.3.2 Backend	8
1.4.3 Frameworks.....	9
1.4 Herramientas de Desarrollo	10
Visual Paradigm 15.2	10
PyCharm 2021.1.1.....	11
WebStorm 2021.1.1.....	11
Navicat Premium 15	11
SQLite3.....	12
Postman 7.36	12
Microsoft Edge, Google Chrome, Mozilla Firefox, Opera	12

1.5 Metodología de Desarrollo de Software	13
XP (Extreme Programming)	14
Conclusiones parciales del Capítulo	15
Capítulo 2. Diseño de la solución propuesta al problema científico.	17
2.1 Modelado del proceso de negocio.	17
2.1.1 Diagrama de Casos de Uso del Sistema.....	18
2.2 Requisitos, análisis y diseño de la aplicación	24
2.2.1 Requisitos funcionales	24
2.2.2 Requisitos no funcionales.....	25
2.3 Historias de Usuario.....	26
HU Listar Usuario	26
HU Adicionar Usuario	26
HU Modificar usuario	27
HU Eliminar Usuario	27
HU Listar Cliente	28
HU Adicionar Cliente	28
HU Modificar Cliente.....	29
HU Eliminar Cliente	29
2.3.1 Estimación de historias de usuarios	30
2.4 Diagrama Entidad-Relación	30
2.5 Diseño Arquitectónico de la aplicación	31
2.5.1 Estándares de Codificación.....	31
2.5.2 Patrón de arquitectura	32
2.5.3 Patrones generales de asignación de responsabilidades (GRASP)	32
2.6 Diagramas de Clases de Diseño con Estereotipos Web	32
2.6.1 CU Gestionar Usuario	33
2.6.2 CU Rellenar Encuesta	34
2.6.3 CU Autenticar Usuario.....	34
2.7 Diagramas de Secuencia	34
2.7.1 CU Gestionar Usuario	35
2.7.2 CU Rellenar Encuesta	37
2.7.3 CU Autenticar Usuario.....	38
Conclusiones parciales del Capítulo	38
Capítulo 3. Implementación y Pruebas del Sistema.	39
3.1 Implementación de la Propuesta de Solución.....	39

3.2 Diagrama de Despliegue.....	40
3.3 Pruebas de Software para la evaluación de la Propuesta de Solución	41
3.3.1 Pruebas de Caja Negra	42
3.3.2 Pruebas de Regresión	44
3.3.3 Pruebas de Usabilidad	44
3.3.2 Pruebas de Aceptación	48
Conclusiones parciales del Capítulo	49
Conclusiones	50
Referencias.....	51
Anexos	54

Índice de Tablas

Tabla 1: Especificación de Soluciones Similares.	6
Tabla 2: Ventajas y Desventajas de la Metodología XP.....	14
Tabla 3: Especificación del Caso de Uso Autenticar Usuario.....	18
Tabla 4: Especificación del Caso de Uso Gestionar Usuario.	19
Tabla 5: Especificación del Caso de Uso Rellenar Encuesta.....	22
Tabla 6: Especificación del Caso de Uso Generar Reporte de Encuestas.	23
Tabla 7: Descripción de Requisitos Funcionales.....	24
Tabla 8: Descripción de HU Listar Usuario.....	26
Tabla 9: Descripción de HU Adicionar Usuario.	26
Tabla 10: Descripción de HU Modificar Usuario.....	27
Tabla 11: Descripción de HU Eliminar Usuario.	27
Tabla 12: Descripción de HU Listar Cliente.....	28
Tabla 13: Descripción de HU Adicionar Cliente.....	28
Tabla 14: Descripción de HU Modificar Cliente.....	29
Tabla 15: Descripción de HU Eliminar Cliente.....	29
Tabla 16: Estimación de Historias de Usuario.....	30
Tabla 17: Iteraciones por Historia de Usuario.	39
Tabla 18: Especificación de Pruebas de Regresión.....	44
Tabla 19: Lista de Chequeo de Pruebas de Usabilidad.	45

Índice de Figuras

Figura 1: Modelo del Negocio.....	17
Figura 2: Diagrama de Casos de Uso del Sistema.....	18
Figura 3: Diagrama Entidad-Relación.....	31
Figura 4: Diagrama de Clases del Caso de Uso Gestionar Usuario.	33
Figura 5: Diagrama de Clases del Caso de Uso Rellenar Encuesta.	34
Figura 6: Diagrama de Clases del Caso de Uso Autenticar Usuario.....	34
Figura 7: Diagrama de Clases del Caso de Uso Autenticar Usuario.....	35
Figura 8: Diagrama de Clases del Caso de Uso Autenticar Usuario.....	36
Figura 9: Diagrama de Clases del Caso de Uso Autenticar Usuario.....	36
Figura 10: Diagrama de Clases del Caso de Uso Autenticar Usuario.....	37
Figura 11: Diagrama de Clases del Caso de Uso Autenticar Usuario.....	37
Figura 12: Diagrama de Clases del Caso de Uso Autenticar Usuario.....	38
Figura 13: Diagrama de Despliegue.....	41
Figura 14: Diagrama de Caso de Prueba RF Autenticar Usuario.....	43
Figura 15: Diagrama de Caso de Prueba RF Adicionar Usuario.....	43
Figura 16: Diagrama de Caso de Prueba RF Modificar Usuario.	43
Figura 17: Diagrama de Caso de Prueba RF Listar Usuario.	44

Introducción

Desde su surgimiento, la Universidad de las Ciencias Informáticas (UCI) ha tenido entre sus objetivos principales formar ingenieros capaces, comprometidos con la tarea de la Revolución, e informatizar la Sociedad Cubana.

Desde hace varios años la entidad viene desarrollando diversas acciones para informatizar la sociedad. Esta tarea se ha llevado a cabo realizando proyectos: aplicaciones web, de escritorio, aplicaciones móviles, y brindando servicios informáticos y de soporte para varias ramas del país.

Situación Problemática

Uno de los aspectos más importante a la hora de desplegar y poner en marcha algún proyecto o brindar algún servicio, ya sea de soporte u otra función, es la satisfacción del cliente a la hora de usar dicho sistema, una evaluación por parte del usuario facilitaría las cosas a la hora de desarrollar nuevas actualizaciones de algún producto o a la hora de prestar soporte.

Uno de los métodos más usados hoy en día son las encuestas, largos cuestionarios de preguntas en los que el cliente emite su valoración, y esta, al combinarla con la de otros usuarios, devuelven un resultado, dándole a los desarrolladores un porcentaje o puntaje de calidad.

Se puede afirmar entonces que las encuestas son un método positivo, pero a la vez tediosos y complicados a la hora de procesar la información, debido a la gran cantidad de datos que requieren y que generan.

Problema de Investigación

Las encuestas realizadas por la Universidad de las Ciencias Informáticas son efectivas, pero no existe un proceso para automatizarlas por lo que se hace muy difícil recoger toda la información y analizarla para mostrar el resultado final.

Objeto de estudio

Encuestas, técnicas de recopilación y clasificación de datos.

Campo de acción

Sistemas de Gestión de Encuestas para productos o servicios informáticos.

Objetivo General

Desarrollar un Sistema de Gestión de Encuestas para la Universidad de las Ciencias Informáticas, para automatizar y acelerar el proceso de recopilación de información y procesamiento de las encuestas sobre los productos de la Universidad.

Hipótesis

Si se desarrolla un sistema de gestión para las encuestas, sería más fácil para los usuarios emitir su criterio sobre los productos desarrollados por la universidad y a su vez se automatizaría el proceso de valoración de dichos productos.

Tareas de Investigación

1. Recopilar información de los principales Sistemas de Gestión de Encuestas (SGE) del mercado y analizar los puntos fuertes de cada uno.
2. Desarrollar un SGE para la Universidad de las Ciencias Informáticas.
3. Realizar las pruebas al SGE para comprobar que se hayan cumplido los objetivos.

Métodos de Investigación

- Analítico-Sintético: posibilitó la extracción de los elementos más importantes de otros métodos para la medición de satisfacción, para conformar la estructura y funcionalidades del proyecto.
- Análisis Histórico-Lógico: permitió analizar la evolución de las encuestas a lo largo del tiempo para determinar la forma de hacer llegar la información al cliente de una forma dinámica y rápida, sin preguntas largas ni tediosas.
- Modelado: se desarrolló una propuesta de solución que fue mostrada a varios clientes para determinar la estructura básica del sistema.
- Observación: se sometió el sistema a un número de personas y se constató el logro del objetivo trazado.

Resultados Esperados

Como resultado de esta investigación se espera desarrollar un Sistema de Gestión de Encuestas (SGE) que permita informatizar el proceso de medir la satisfacción de los clientes.

El desarrollo de dicho sistema facilitará el proceso de calificación de los productos de la universidad y permitirá a los clientes la evaluación de una forma más rápida, aumentando así la velocidad en la que se muestran los resultados, siendo este el principal aporte de esta investigación.

Capítulo 1. Fundamentos y referentes teórico-metodológicos sobre el objeto de estudio.

Mediante el desarrollo de este capítulo se expondrán los temas teóricos y conceptuales de la investigación. Para ello se ofrecerán fundamentos teóricos y conceptos relacionados con el tema, analizando resultados obtenidos por otros investigadores, así como publicaciones de diversas fuentes actualizadas que permitan crear el marco teórico que sustenta la investigación. Se planteará la propuesta de solución, los métodos y medios utilizados para cumplir el objetivo general y se determinará la metodología a utilizar.

1.1 Conceptos fundamentales asociados al dominio del problema

1.1.1 Encuesta

El Diccionario de la Real Academia Española plantea que una encuesta es un conjunto de preguntas tipificadas dirigidas a una muestra representativa de grupos sociales, para averiguar estados de opinión o conocer otras cuestiones que les afectan. (1)

Las encuestas son una herramienta para conocer las características de un grupo de personas. Puede tratarse de variables económicas, como el nivel de ingresos, o de otro tipo, como las preferencias políticas.

Las encuestas son entonces un método de investigación y recopilación de datos utilizadas para obtener información de personas sobre diversos temas, tienen una variedad de propósitos y se pueden llevar a cabo de muchas maneras dependiendo de la metodología elegida y los objetivos que se deseen alcanzar. (2)

1.1.2 Satisfacción del Cliente

Según el Diccionario de la Real Academia Española, la satisfacción es una razón, acción o modo con que se sosiega y responde enteramente a una queja o sentimiento. (1)

La satisfacción del cliente representa entonces el grado de cumplimiento de las expectativas de un cliente tras recibir algún producto o servicio. (3)

La satisfacción del cliente se puede definir como el sentimiento o la actitud del cliente hacia un producto, una empresa o un servicio prestado por una empresa, esta aparece cuando las necesidades o expectativas del cliente se han cumplido y puede basarse en factores tangibles como la calidad de un producto o factores intangibles como el comportamiento de su personal hacia el cliente. (4)

1.1.3 Calidad

El Diccionario de la Real Academia Española plantea que la calidad es una propiedad o conjunto de propiedades inherentes a algo, que permiten juzgar su valor. (1)

También la calidad se refiere a la capacidad que posee un objeto para satisfacer necesidades implícitas o explícitas según un parámetro, un cumplimiento de requisitos de cualidad. (5)

El término calidad entonces hace alusión a que un objeto o producto tiene la capacidad de satisfacer las necesidades básicas, explícitas o implícitas según los parámetros de las empresas y los consumidores, de manera que cumpla con los requisitos de la cualidad del objeto o producto. En las empresas y compañías existen herramientas de calidad o indicadores de calidad, en las cuales se establece si los productos son buenos para las personas o no. (6)

1.1.4 Cuestionario

Un cuestionario es una herramienta de recopilación de información, es decir, un tipo de encuesta, que consiste en una serie sucesiva y organizada de preguntas. Su nombre proviene del latín “*quaestionarius*”, que significa “lista de preguntas”. (7)

Un cuestionario es aquel que plantea una serie de preguntas para extraer determinada información de un grupo de personas. El cuestionario permite recolectar información y datos para su tabulación, clasificación, descripción y análisis en un estudio o investigación. (8)

1.2 Soluciones Informáticas Similares

En este epígrafe se analizarán algunos de los Sistemas de Gestión de Encuestas del mercado, enmarcando las ventajas y desventajas de cada uno para ayudar a la realización de la futura aplicación.

1.2.1 SoftExpert Encuesta

SoftExpert Encuesta permite que los usuarios creen y publiquen un cuestionario en minutos, desde incluir preguntas, organizar en páginas a editar el cuestionario de la forma deseada. La herramienta de análisis del SoftExpert Encuesta además posibilita análisis más profundos de los resultados y mejores decisiones a través de agrupamientos, filtros y cruce de informaciones. (9)

Entre sus principales funcionalidades brinda:

- Creación y modelaje de cuestionarios dinámicos y flexibles
- Uso y aplicación de modelos de cuestionarios.
- Duplicación de cuestionarios existentes.

- Cuestiones de múltiples estilos, incluyendo tablas, respuesta única, respuestas múltiples y preguntas abiertas.
- Determinación de la respuesta correcta.
- Personalización de estilos.
- Uso de imágenes en lugar de preguntas y alternativas de respuestas.
- Uso de imágenes de fondo.
- Importación de lista de encuestados a través de planillas electrónicas.
- Control de inicio y fin de cuestionarios y respuestas.
- Almacenamiento de respuestas parciales.
- Corrección automática o manual.
- Automación y programación de cuestionarios e invitación por e-mail.
- Posibilidad de uso de cuestiones aleatorias.
- Análisis crítico sobre los resultados.

1.2.2 CRR Marketing – Consultoría y formación

CRR Marketing facilita herramientas y servicios para conocer las necesidades, gustos y preferencias de los consumidores y adaptar sus productos, servicios o estrategias de promoción a éstos. (10)

Entre sus características se encuentran:

- Implantación de sistema de encuestas multi-idioma.
- Diseño de Encuestas.
- Tratamiento Estadístico de Datos.
- Informes periódicos de Resultados y Evolución.
- Benchmarking y tendencias del sector.
- Formación y soporte a su personal en el uso del sistema de encuestas.
- Estudios de satisfacción de clientes y calidad de servicio.

Sus estudios de satisfacción y calidad de servicio permiten conocer los puntos fuertes y las áreas de mejora de la empresa, las preferencias de los clientes y los aspectos del servicio más importantes para ellos. Permite acceder en tiempo real a los resultados de la encuesta, consultar gráficas y resultados generados de forma automática, envío de encuestas por e-mail a clientes y recordatorios masivos sólo a los que no han rellenado la encuesta. (10)

Al ser un sitio web “responsive” permite la visualización del contenido en todos los dispositivos y posee la facilidad de compartir la encuesta en distintas Redes Sociales como Facebook. Posee un hosting de sistema de encuestas que puede instalarse en cualquier sitio web de la empresa y brindan servicios de hosting en caso de que la empresa no lo posea. (10)

1.2.3 Orchestra 7

Este módulo permite diseñar una encuesta en línea fácil de usar y enviar una invitación de respuesta al cliente. Las preguntas pueden ser de diferentes tipos (clasificación por estrellas, opción múltiple, texto libre). El cuestionario también puede contener preguntas de seguimiento condicionales que pueden diferir según la respuesta anterior. (11)

Algunas de sus funcionalidades:

- Mantiene informados a los clientes a través de avisos de voz con información efectiva y SMS.
- Propone un diseño de interacciones más personales y apropiadas gracias al conocimiento de las necesidades de los clientes.
- Admite varios tipos de preguntas y preguntas de seguimiento condicionales.
- Registra, controla e informa sobre todas las actividades de los usuarios y cambios en la configuración producidos en el sistema.
- Permite informes automáticos e informes personalizados que proporcionan la información y los conocimientos explícitos sobre el negocio.

1.2.4 Especificación de Soluciones Similares

Tabla 1: Especificación de Soluciones Similares.

Funcionalidades	SoftExpert	CRR Marketing	Orchestra
Generar Reporte	✓	✓	✓
Personalización de Encuestas	✓	✗	✗
Multiplicidad de Datos por Producto	✓	✓	✗
Integración con Redes Sociales	✗	✗	✓
Recomendación de Productos	✓	✗	✗
Resúmenes Generales del Sistema	✗	✓	✓
Gestión de Encuestas y sus Relaciones	✗	✗	✗

1.3 Tecnologías de Desarrollo

En este epígrafe se especificarán algunas las tecnologías usadas en la realización de soluciones informáticas, y se definirá el *framework* y los lenguajes de programación y maquetado a usar en el desarrollo de la aplicación.

1.3.1 Frontend

Frontend es la parte de un programa o dispositivo a la que un usuario puede acceder directamente. Son todas las tecnologías de diseño y desarrollo web que corren en el navegador y que se encargan de la interactividad con los usuarios.

El desarrollo frontend está basado en **HTML** y **CSS**, los lenguajes de maquetación que permiten definir la estructura y estilos de una página web, las funcionalidades, animaciones y otros elementos del frontend son programados con **JavaScript**, un lenguaje de programación para definir la lógica de nuestra aplicación, recibir las solicitudes de los usuarios y enviárselos al backend. (12)

HTML: Lenguaje de etiquetas de hipertexto

HTML (Lenguaje de Marcas de Hipertexto, del inglés *HyperText Markup Language*) es el componente más básico de la Web. Define el significado y la estructura del contenido web. Además de HTML, generalmente se utilizan otras tecnologías para describir la apariencia/presentación de una página web (CSS) o la funcionalidad/comportamiento (JavaScript). (13)

"Hipertexto" hace referencia a los enlaces que conectan páginas web entre sí, ya sea dentro de un único sitio web o entre sitios web. Los enlaces son un aspecto fundamental de la Web. Al subir contenido a Internet y vincularlo a las páginas creadas por otras personas, te conviertes en un participante activo en la «*World Wide Web*» (Red Informática Mundial). (14)

HTML utiliza "marcas" para etiquetar texto, imágenes y otro contenido para mostrarlo en un navegador Web. Un elemento HTML se distingue de otro texto en un documento mediante "etiquetas", que consisten en el nombre del elemento rodeado por "<" y ">". El nombre de un elemento dentro de una etiqueta no distingue entre mayúsculas y minúsculas. Es decir, se puede escribir en mayúsculas, minúsculas o una mezcla. (14)

HTML es el lenguaje donde se define la información o el contenido del documento, el formato de los archivos es *.html*.

CSS: Hojas de Estilo en Cascada

Hojas de Estilo en Cascada (del inglés *Cascading Style Sheets*) o CSS es el lenguaje de estilos utilizado para describir la presentación de documentos HTML o XML (en-US) (incluyendo varios lenguajes basados en XML como SVG, MathML o XHTML). CSS describe como debe ser renderizado el elemento estructurado en la pantalla, en papel, en el habla o en otros medios.

CSS es utilizado para diseñar y dar estilo a las páginas web, por ejemplo, alterando la fuente, color, tamaño y espaciado del contenido, dividirlo en múltiples columnas o agregar animaciones y otras características decorativas. Este módulo proporciona un suave comienzo hacia el dominio de CSS con los conceptos básicos acerca de su funcionamiento, la sintaxis y la manera en que puedes comenzar a utilizarlo para agregar estilos al HTML. (15)

CSS es el lenguaje donde se especifica el diseño del documento, maneja todo lo relacionado con la parte visual, el formato de los archivos es `.css`. (13)

JavaScript

JavaScript es un lenguaje de programación o de secuencias de comandos que permite implementar funciones complejas en páginas web, muestra oportunas actualizaciones de contenido, mapas interactivos, animación de Gráficos 2D/3D, desplazamiento de máquinas reproductoras de vídeo. Es la tercera capa de las tecnologías web estándar, junto a HTML y CSS, es realmente el lenguaje de programación que nos permite crear sitios web, el formato de los archivos es `.js`. (13) (16)

1.3.2 Backend

El backend es la capa de acceso a datos de un software o cualquier dispositivo, que no es directamente accesible por los usuarios, además contiene la lógica de la aplicación que maneja dichos datos. El backend también accede al servidor, que es una aplicación especializada que entiende la forma como el navegador solicita cosas. (12)

Algunos lenguajes de programación para Backend son Python, Node.js, PHP y C#. Y así como en el frontend, todos estos lenguajes tienen diferentes *frameworks* que permiten trabajar mejor según el proyecto que se desarrolle, como Django, Laravel, Symphony y ASP.Net. (12)

Python

Python es un lenguaje de scripting independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web. Es un lenguaje interpretado, lo que significa que no se necesita compilar

el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad. (17)

Se utilizará Python en su versión 3.10.0, esto se debe a que es la versión estable más actualizada hasta la fecha.

1.4.3 Frameworks

Además de los lenguajes de programación que definen cómo estará escrito el código, es importante destacar la importancia de los *frameworks* de desarrollo. Herramientas y librerías de código pre-escrito que facilitan a los desarrolladores las tareas cotidianas o de uso común para evitar ejecutarlas desde cero.

Bootstrap

Bootstrap es un framework CSS utilizado en aplicaciones *frontend* para desarrollar aplicaciones que se adaptan a cualquier dispositivo.

Por esta razón, tiene varios recursos para configurar los estilos de los elementos de la página de una manera simple y eficiente, además de facilitar la construcción de páginas que, al mismo tiempo, están adaptadas para la web y para dispositivos móviles.

Desarrollado por Twitter en 2010, inicialmente se llamó *Twitter Blueprint* y, un poco más tarde, en 2011, se transformó en código abierto y su nombre cambió a Bootstrap.

El framework combina CSS y JavaScript para estilizar los elementos de una página HTML, por lo que ofrece una serie de componentes que facilitan la comunicación con el usuario, como menús de navegación, controles de página, barras de progreso y más.

Además de todas las características que ofrece el framework, su principal objetivo es permitir la construcción de sitios web responsive para dispositivos móviles, esto significa que las páginas están diseñadas para funcionar en *desktop*, *tablets* y *smartphones*, de una manera muy simple y organizada. (18)

Material Design Bootstrap (MDB)

Material Design es un lenguaje de diseño elaborado por Google y presentado en junio de 2014, que se integró con el lanzamiento de Android 5.0 (*Lollipop*) y se ha seguido utilizando en su sistema operativo desde entonces. Además, dada su buena recepción, su uso se ha ido extendiendo a otros productos de Google. (19)

Bootstrap es un *framework* de maquetación, mientras que Material Design no. Realmente no hay un *framework* que aplique sus estilos, sino que se creó como una guía de estilos y a partir de ahí surgieron proyectos que crearon el *framework* (por ejemplo, Materialize y Material Design Bootstrap). (20)

Django

Django fue desarrollado inicialmente entre 2003 y 2005 por un equipo que era responsable de crear y mantener sitios web de periódicos. Después de crear varios sitios, el equipo empezó a tener en cuenta y reutilizar muchos códigos y patrones de diseño comunes. Este código común se convirtió en un *framework* web genérico, que fue de código abierto, conocido como proyecto "Django" en julio de 2005.

Es un *framework* web de alto nivel escrito en Python que permite el desarrollo rápido de sitios web seguros y mantenibles. Es gratuito y de código abierto, tiene una comunidad próspera y activa, una gran documentación y muchas opciones de soporte gratuito y de pago. Provee casi todo lo que los desarrolladores quisieran que tenga por defecto, sigue principios de diseño consistentes y tiene una amplia y actualizada documentación.

Django usa un componente basado en la arquitectura "*shared-nothing*" (cada parte de la arquitectura es independiente de las otras, y por lo tanto puede ser reemplazado o cambiado si es necesario). Utiliza el principio No te repitas, "*Don't Repeat Yourself*" (DRY) para que no exista una duplicación innecesaria, reduciendo la cantidad de código. Agrupa código relacionado en módulos y siguiendo el patrón MVT (Modelo-Vista-Template) basado en el patrón Modelo Vista Controlador. (21)

En el desarrollo de la aplicación se usará la versión 3.2.8, la última versión con soporte a largo plazo del *framework*.

1.4 Herramientas de Desarrollo

Visual Paradigm 15.2

Visual Paradigm es una herramienta CASE: Ingeniería de Software Asistida por Computación. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación.

Ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Fue diseñado para una amplia gama de

usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un enfoque Orientado a Objetos. (22)

PyCharm 2021.1.1

PyCharm es un IDE multiplataforma dirigido principalmente a Python y al desarrollo web. Proporciona una finalización del código inteligente, inspecciones del código, indicación de errores sobre la marcha y arreglos rápidos, así como refactorización de código automática y completas funcionalidades de navegación.

El editor de código inteligente de PyCharm ofrece compatibilidad de primer nivel con Python, JavaScript, CoffeeScript, TypeScript, CSS, lenguajes de plantilla populares y más. ¡Aproveche la finalización de código que tiene en cuenta el lenguaje, detección de errores y corrección de código sobre la marcha!

Incluye un depurador y un ejecutor de pruebas integrados, perfilador Python, un terminal integrado y herramientas de base de datos integradas, capacidades de desarrollo remoto con intérpretes remotos y un terminal ssh. (23)

WebStorm 2021.1.1

WebStorm es un IDE multiplataforma principalmente para desarrollo web frontend. Está creado en una plataforma IntelliJ Platform de código abierto desarrollada por JetBrains y ofrece numerosos ajustes que hacen de la programación una experiencia más agradable y productiva.

Es directamente compatible con JavaScript, TypeScript, React, React Native, Vue, Angular, Node.js, HTML, hojas de estilo y muchas otras tecnologías. Comprende perfectamente la estructura del proyecto y ayuda en todos los aspectos de la escritura de código, facilita la detección de errores tipográficos y ortográficos en el código. Incluye cientos de inspecciones para todos los lenguajes compatibles y un corrector de ortografía y gramática. (24)

Navicat Premium 15

Navicat Premium es una herramienta de desarrollo de bases de datos que permite conectarse simultáneamente a bases de datos MySQL, MariaDB, MongoDB, SQL Server, Oracle, PostgreSQL y SQLite desde una sola aplicación.

La transferencia, la sincronización de datos y la sincronización de estructuras ayudan a migrar datos de una forma más fácil y rápida y con menos sobrecarga. Presenta un potente asistente de importación para transferir datos a una base de datos desde diversos formatos, o desde ODBC después de configurar una conexión de origen de datos.

Visual SQL / Query Builder ayuda a crear, editar y ejecutar sentencias / consultas SQL. Permite crear, modificar y administrar todos los objetos de la base de datos utilizando diseñadores de objetos profesionales y convierte las bases de datos en representaciones gráficas utilizando una sofisticada herramienta de diseño y modelado de bases de datos. (25)

SQLite3

SQLite es una biblioteca de C que provee una base de datos ligera basada en disco que no requiere un proceso de servidor separado y permite acceder a la base de datos usando una variación no estándar del lenguaje de consulta SQL. Algunas aplicaciones pueden usar SQLite para almacenamiento interno. (26)

El módulo `sqlite3` implementa una interfaz compatible a Python DB-API 2.0 a SQLite, una base de datos relacional en proceso. SQLite está diseñada para integrarse en aplicaciones, en lugar de utilizar un programa de servidor de base de datos como MySQL, PostgreSQL u Oracle. Es rápida, rigurosamente probada y flexible, lo que la hace adecuada para crear prototipos y despliegue de producción para algunas aplicaciones. Una base de datos SQLite se almacena como un solo archivo en el sistema de archivos. La biblioteca gestiona el acceso al archivo, incluido el bloqueo para evitar corrupción cuando varios escritores lo utilizan. La base de datos se crea la primera vez que se accede al archivo, pero la aplicación es responsable de gestionar las definiciones de tabla, o esquema, dentro de la base de datos. (27)

Postman 7.36

Postman es una herramienta que se utiliza, sobre todo, para el *testing* de API REST, aunque también admite otras funcionalidades que se salen de lo que engloba el *testing* de este tipo de sistemas. Gracias a esta herramienta, además de testear, consumir y depurar API REST, podremos monitorizarlas, escribir pruebas automatizadas para ellas, documentarlas y simularlas. (28)

Microsoft Edge, Google Chrome, Mozilla Firefox, Opera

Al trabajar con HTML5 surge un tema que genera dudas y es la compatibilidad de HTML5 en los navegadores actuales. En términos generales ningún navegador es 100% compatible con todas las características de HTML5 y CSS3 es por eso que el sistema debe ser ejecutado en la gran mayoría de los navegadores actuales para comprobar su funcionalidad, de ahí el hecho de usar los navegadores más comunes que existen hoy en día. (29)

Microsoft Edge 90.0.818.62

Edge es el navegador propio de Microsoft que sustituyó al longevo Internet Explorer en Windows 10 y Xbox One en el año 2015. Edge viene instalado en todos los dispositivos Windows y, en comparación con su predecesor, presenta grandes avances en consonancia con los navegadores modernos.

Las versiones actuales de Edge y Chrome se basan en Chromium, el código abierto de Google, incluye además varias funciones de privacidad que Chrome no tiene, como una herramienta anti seguimiento integrada. (30)

Google Chrome 91.0.4472.124

Chrome es, con diferencia, el navegador más popular, pues se ha hecho con casi dos tercios de la cuota global de mercado (cifra de verano de 2020) en todos los dispositivos. Se incluyen ordenadores, tanto portátiles como de escritorio, *smartphones* y tabletas. (30)

Mozilla Firefox 89.0.2

Mozilla Firefox es un navegador de código abierto y la solución de terceros más popular. Eso quiere decir que no lo ha creado ningún desarrollador de sistemas operativos como Google (Android), Microsoft (Windows) o Apple (macOS y iOS). En todos los tipos de dispositivos, en torno al 4 % de las personas de todo el mundo elige Firefox para acceder a la red. (30)

Opera 79.0.4143.66

Opera es un navegador multiplataforma que, pese a no atraer mucha atención entre el público general, es pionero en el desarrollo de diversas funciones básicas de un navegador que ahora consideramos lo más normal del mundo. Las pestañas de navegador, el bloqueo incorporado de las ventanas emergentes y la posibilidad de volver a abrir las últimas páginas cerradas son prestaciones que se estrenaron en Opera. (30)

1.5 Metodología de Desarrollo de Software

Una metodología de desarrollo de software tiene como principal objetivo aumentar la calidad del software que se produce en todas y cada una de sus fases de desarrollo. No existe una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así que el proceso sea configurable. Las metodologías de desarrollo se pueden dividir en dos grupos de acuerdo con sus características y los objetivos que persiguen: ágiles y robustas. (31)

Las metodologías ágiles

Se caracterizan por hacer énfasis en la comunicación cara a cara, es decir, se basan en una fuerte y constante interacción, donde clientes desarrolladores y desarrolladores trabajan constantemente juntos, estableciéndose así una estrecha comunicación. Estas metodologías están orientadas al resultado del producto y no a la documentación; exige que el proceso sea adaptable, permitiendo realizar cambios de último momento. Se puede hacer mención dentro de las metodologías ágiles a: XP (por sus siglas en inglés Extreme Programming), Scrum y Crystal Methodologies. (31)

XP (Extreme Programming)

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre. Kent Beck, el padre de XP, describe la filosofía de XP sin cubrir los detalles técnicos y de implantación de las prácticas. Posteriormente, otras publicaciones de experiencias se han encargado de dicha tarea. (31)

El principio de la sencillez se corresponde con la base del valor con el mismo nombre, pero incluye instrucciones concretas de aplicación. En este sentido se usan dos métodos:

- ***You ain't gonna need it (YAGNI)***: mientras una función no se solicite expresamente, no se debe implementar para no realizar trabajo en vano.
- ***Don't repeat yourself (DRY)***: debe evitarse repetir tareas y diseñar el código de manera que los cambios no tengan que aplicarse en varios puntos, sino una sola vez, en la medida de lo posible.

Tabla 2: Ventajas y Desventajas de la Metodología XP.

Ventajas	Inconvenientes
Relación estrecha con el cliente	Mayor esfuerzo de trabajo
Ausencia de trabajos de programación innecesarios	El cliente se implica en el proceso

Software estable debido a continuas pruebas	Requiere mucho tiempo
Menos errores gracias a la programación en pareja	Relativamente caro
Ausencia de horas extra, gestión propia del tiempo	Requiere control de versiones
Aplicación rápida de cambios	Requiere autodisciplina en la aplicación
Código de comprensión sencilla en todo momento	

Conclusiones parciales del Capítulo

Al haber culminado este capítulo, se han esclarecido los conceptos más importantes sobre la temática y se ha alcanzado un nivel más alto de comprensión en el ámbito de las encuestas y los cuestionarios al cliente. Se analizaron las soluciones similares, señalando sus ventajas y desventajas y se ha concretado una idea más clara sobre la estructura del sistema.

Se decidió usar un *framework* para el desarrollo de la solución informática, en ambas partes del sistema, tanto *frontend* como *backend*. Del lado del cliente se utilizará el *framework* Bootstrap 4, en su versión Material Design 3.3.0, basado en HTML5, CSS y JavaScript, implementando jQuery en su versión 3.5.1 para las validaciones y animaciones requeridas, y *Fontawesome Free* 5.8.1 para el trabajo con iconos y fuentes en formato *svg* para mayor funcionalidad del diseño. Por parte del servidor se utilizará el *framework* Django en su versión 3.2.8, basado en Python en su versión 3.10, como Sistema Gestor de Base de Datos se usará *SQLite3*, el gestor de base de datos implementado por el propio Django, debido a su alta velocidad y portabilidad.

Se utilizará PyCharm 2021.1.1 como entorno de desarrollo del lado del servidor y WebStorm 2021.1.1 del lado del cliente. La interfaz para acceder al Sistema Gestor de Base de Datos será Navicat Premium 15, en su versión 15.0.23. Como símbolo del sistema, se decidió remplazar la consola de comandos de Windows por la aplicación Cmder Console Emulator debido a su alta fiabilidad y compatibilidad con las versiones más actuales de los *frameworks* a utilizar.

Se ha decidido usar la metodología de desarrollo de software Extreme Programming pues las continuas pruebas generan sistemas muy estables y el procedimiento iterativo en colaboración con el enfoque minimalista garantizan que solo se creen funciones que realmente son importantes para el proyecto. Se realizan las modificaciones en pasos pequeños, en lugar de

implementar grandes actualizaciones para contrarrestar varias fuentes de errores a la vez, solo se trata un problema cada vez, así el equipo reacciona más rápido y es más fácil entender la causa de las modificaciones.

Capítulo 2. Diseño de la solución propuesta al problema científico.

Al diseñar una arquitectura de software se crean y representan componentes que interactúan entre sí, con responsabilidades específicas que se organizan de forma tal que se logren los requerimientos establecidos.

En este capítulo se definirán las características del sistema mediante una especificación de requisitos divididos en requisitos funcionales y no funcionales. Se presentarán los actores y las especificaciones de los casos de uso del negocio, así como los distintos diagramas de clases y de secuencia del mismo, los patrones de diseño y de codificación generales para una mayor funcionalidad y estabilidad del software. Por último, se mostrará el modelo de datos de la solución y su respectivo diagrama de despliegue y de componentes.

2.1 Modelado del proceso de negocio.

El modelado del proceso del negocio resume la interacción entre los actores y los casos de uso, las cuales representan las funcionalidades del sistema. Se representan además las interacciones existentes entre los casos de uso, siendo el actor el que ejecuta o genera un caso de uso al interactuar con el sistema para obtener o modificar una información.

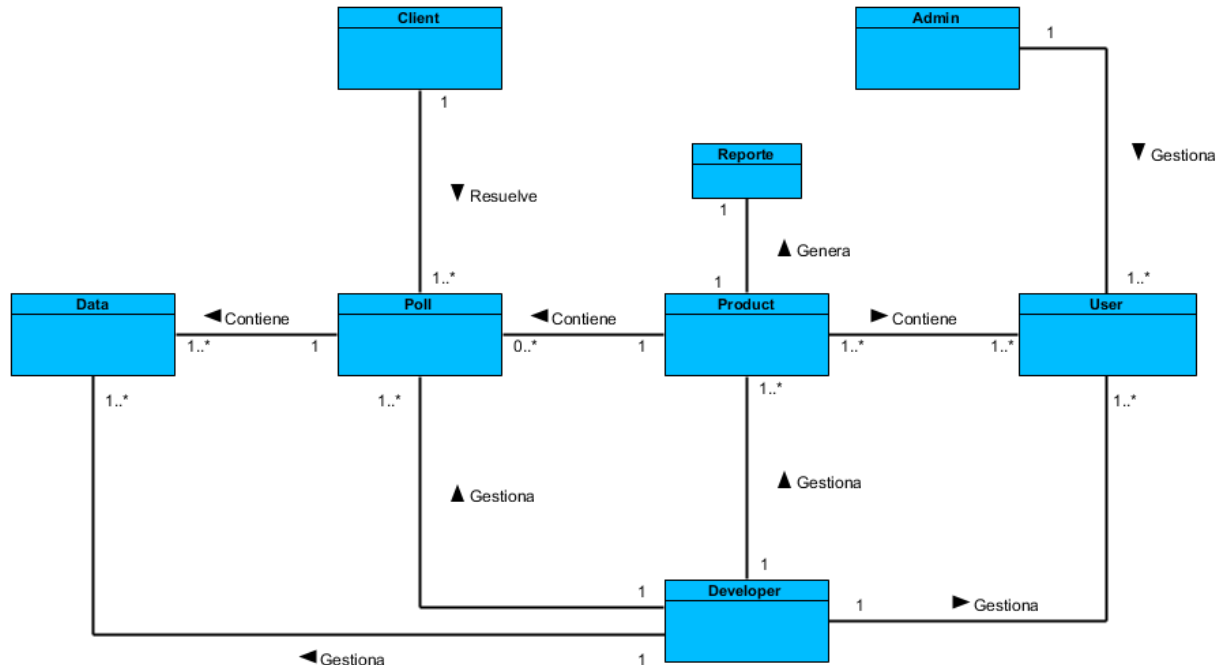


Figura 1: Modelo del Negocio.

2.1.1 Diagrama de Casos de Uso del Sistema

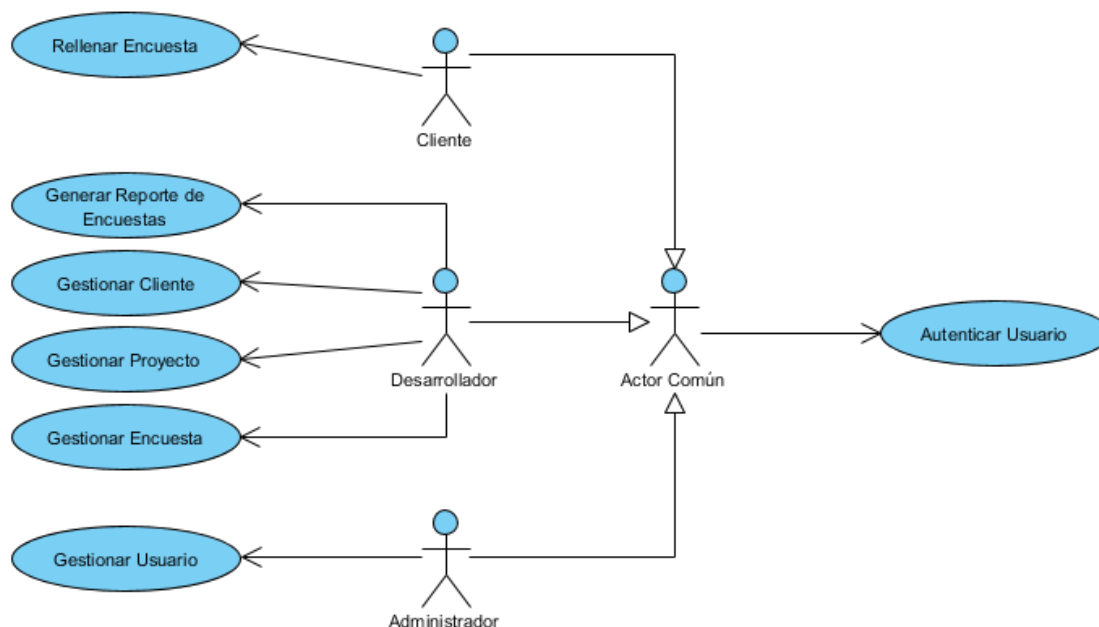


Figura 2: Diagrama de Casos de Uso del Sistema.

En el diagrama se recogen un total de siete casos de uso, de los cuales se explicarán algunos de los que resultan indispensables para el funcionamiento del sistema.

Caso de Uso <Autenticar Usuario>

Tabla 3: Especificación del Caso de Uso Autenticar Usuario.

Objetivo	Permitir el acceso del usuario al sistema	
Actores	Actor Común	
Resumen	El caso de uso se inicia cuando el Actor Común decide acceder al sistema	
Complejidad	Media	
Prioridad	Crítica	
Precondiciones	Encontrarse en la página de inicio del sistema	
Postcondiciones	Se accede al sistema	
Flujo de eventos		
Flujo básico <Autenticar Usuario>		
Actor	Sistema	
1	Desea acceder al sistema.	1.1 Muestra un campo para introducir el usuario o correo electrónico y otro para la contraseña y el botón "Enviar".

2	Inserta usuario y contraseña y presiona el botón Enviar.	2.1 Verifica que los campos estén llenos.
3		3.1 Verifica que los campos sean correctos.
4		4.1 Accede al sistema, finalizando así el caso de uso.
Flujo Alternativo de Eventos 2a. “Campos Vacíos y/o incorrectos”		
Actor		Sistema
2		2.1 Muestra el mensaje “Debe rellenar todos los campos”. 2.2 Muestra el mensaje “Usuario o contraseña incorrecta”.
Flujo Alternativo de Eventos 3a. “Usuario inexistente”		
Actor		Sistema
3		3.1 Muestra un mensaje indicando que el usuario no existe.

Caso de Uso <Gestionar Usuario>

Tabla 4: Especificación del Caso de Uso Gestionar Usuario.

Objetivo	Permitir adicionar, modificar y eliminar datos acerca de un usuario.	
Actores	Administrador	
Resumen	El caso de uso se inicia cuando el Administrador decide adicionar, modificar o eliminar datos acerca de un usuario.	
Complejidad	Alta	
Prioridad	Alta	
Precondiciones	Administrador ya autenticado.	
Postcondiciones	Se adiciona un usuario, se modifican los datos o se elimina el mismo.	
Flujo de eventos		
Flujo básico <Gestionar Usuario>		
Actor		Sistema
1	Desea adicionar, modificar o eliminar los datos de un usuario.	1.1 Genera automáticamente una lista con todos los usuarios .

2		<p>2.1 Da la posibilidad de alguna de las siguientes acciones:</p> <p>Si decide adicionar un usuario, ir a la sección “Adicionar Usuario”.</p> <p>Si decide modificar los datos de un usuario, ir a la sección “Modificar Usuario”.</p> <p>Si decide eliminar uno o varios usuarios, ir a la sección “Eliminar Usuario”.</p>
Sección No. 1: “Adicionar Usuario”		
Actor		Sistema
2	Selecciona la opción “Adicionar Usuario”	<p>2.1 Muestra una ventana con los siguientes campos a introducir:</p> <ul style="list-style-type: none"> - <i>username</i> - <i>first_name</i> - <i>last_name</i> - <i>email</i> - <i>password</i> - <i>role</i> <p>Y el botón “Aceptar”.</p>
3	Introduce los datos y presiona el botón “Aceptar”.	3.1 Verifica que todos los campos estén llenos.
4		4.1 Verifica que los datos introducidos sean correctos.
5		5.1 Verifica que el usuario no exista en el sistema.
6		6.1 Almacena los datos del usuario y muestra el mensaje “Usuario registrado correctamente”. Finalizando así el caso de uso.
Flujo Alternativo de Eventos 3a. “Campos Vacíos”		
Actor		Sistema
3		3.1 Muestra el mensaje “Debe rellenar todos los campos”.

Flujo Alternativo de Eventos 4a. “Datos Incorrectos”	
Actor	Sistema
4	4.1 Muestra el mensaje “Datos incorrectos”
Flujo Alternativo de Eventos 5a. “Usuario Existente”	
Actor	Sistema
5	5.1 Muestra el mensaje “El usuario ya existe en el sistema”.
Sección No. 2: “Modificar Usuario”	
Actor	Sistema
2	<p>Selecciona la opción “Modificar Usuario”</p> <p>2.1 Muestra una ventana con los siguientes campos a modificar:</p> <ul style="list-style-type: none"> - <i>username</i> - <i>first_name</i> - <i>last_name</i> - <i>email</i> - <i>password</i> - <i>role</i> <p>Y el botón “Aceptar”.</p>
3	Introduce los datos y presiona el botón “Aceptar”.
4	3.1 Verifica que todos los campos estén llenos.
4	4.1 Verifica que los datos introducidos sean correctos.
5	5.1 Verifica que el usuario no exista en el sistema.
Sección No. 3: “Eliminar Usuario”	
Actor	Sistema
2	<p>Marca la opción “Eliminar” de un usuario y presiona la opción “Aceptar”</p> <p>2.1 Muestra el mensaje de confirmación “¿Está seguro que desea eliminar el usuario seleccionado?” y los botones “Aceptar” y “Cancelar”.</p>

3	Presiona el botón “Aceptar”.	3.1 Elimina el usuario seleccionado y emite el mensaje “Se ha eliminado el usuario”.
4	Presiona el botón “Cancelar”.	4.1 Se cancela la operación y vuelve al paso 2.1 del flujo básico de Gestionar Usuario.

Caso de Uso <Rellenar Encuesta>

Tabla 5: Especificación del Caso de Uso Rellenar Encuesta.

Objetivo	Permite al cliente rellenar una encuesta y enviarla	
Actores	Cliente	
Resumen	El caso de uso se inicia cuando el Cliente selecciona una encuesta del listado de encuestas disponibles según el proyecto/servicio asignado a este.	
Complejidad	Media	
Prioridad	Crítica	
Precondiciones	Haberse autenticado como cliente en el sistema y tener un proyecto/servicio asignado.	
Postcondiciones	Se guardan los resultados de la encuesta en el sistema.	
Flujo de eventos		
Flujo básico <Rellenar Encuesta>		
Actor	Sistema	
1	Selecciona la encuesta a rellenar.	1.1 Muestra los campos para completar la encuesta.
2	Rellena la encuesta y presiona el botón “Enviar”.	2.1 Verifica que los campos estén llenos.
3		3.1 Se guardan los resultados, muestra el mensaje “Encuesta enviada” y vuelve a la página de inicio del sistema, finalizando así el caso de uso.
Flujo Alternativo de Eventos 2a. “Campos Vacíos”		
Actor	Sistema	

2	2.1 Muestra el mensaje “Debe rellenar todos los campos”.
---	--

Caso de Uso <Generar Reporte de Encuestas>

Tabla 6: Especificación del Caso de Uso Generar Reporte de Encuestas.

Objetivo	Permite al desarrollador mostrar un reporte con los resultados de las encuestas relacionadas a sus proyectos o servicios.
Actores	Desarrollador
Resumen	El caso de uso se inicia cuando el desarrollador presiona el botón “Generar Reporte”.
Complejidad	Alta
Prioridad	Crítica
Precondiciones	Haberse autenticado como desarrollador en el sistema y tener un proyecto/servicio asignado con al menos una encuesta creada.
Postcondiciones	Se genera el reporte con las valoraciones y resultados de las encuestas.
Flujo de eventos	
Flujo básico <Generar Reporte de Encuestas>	
Actor	Sistema
1	Presiona el botón “Generar Reporte”.
2	Selecciona un proyecto y presiona el botón “Generar”
3	
	1.1 Muestra el listado de los proyectos disponibles para generar el reporte
	2.1 Verifica que el proyecto tenga al menos una valoración(encuesta).
	3.1 Muestra los resultados de la(s) valoración(es) de forma individual por cada cliente y de forma general a modo de puntuación.
Flujo Alternativo de Eventos 2a. “Valoración Vacía”	
Actor	Sistema
2	
	2.1 Muestra el mensaje “Aún no existe una valoración para el producto seleccionado”.

2.2 Requisitos, análisis y diseño de la aplicación

Es esencial y de suma importancia conocer los requisitos funcionales que debe cumplir el sistema para poder identificar y entender su funcionamiento. Durante este epígrafe se conocerán las capacidades, así como las funciones que dicho sistema debe cumplir y las cualidades que el producto debe tener para un mejor rendimiento.

2.2.1 Requisitos funcionales

Tabla 7: Descripción de Requisitos Funcionales.

Nombre	Descripción	P	C
Listar Usuario	El sistema debe permitir mostrar un listado con los usuarios.	B	B
Adicionar Usuario	El sistema debe permitir adicionar un usuario.	A	A
Modificar Usuario	El sistema debe permitir modificar un usuario.	M	A
Eliminar Usuario	El sistema debe permitir eliminar un usuario.	B	A
Listar Cliente	El sistema debe permitir mostrar un listado con los usuarios que tengan el rol de clientes.	B	B
Adicionar Cliente	El sistema debe permitir adicionar un cliente.	A	A
Modificar Cliente	El sistema debe permitir modificar un cliente.	M	A
Eliminar Cliente	El sistema debe permitir eliminar un cliente.	B	A
Listar Encuesta	El sistema debe permitir mostrar un listado con las encuestas.	B	B
Adicionar Encuesta	El sistema debe permitir adicionar una encuesta.	A	A
Modificar Encuesta	El sistema debe permitir modificar una encuesta.	M	A
Eliminar Encuesta	El sistema debe permitir eliminar una encuesta.	B	A
Listar Proyecto	El sistema debe permitir mostrar un listado con los proyectos.	B	B
Adicionar Proyecto	El sistema debe permitir adicionar un proyecto.	A	A
Modificar Proyecto	El sistema debe permitir modificar un proyecto.	M	A
Eliminar Proyecto	El sistema debe permitir eliminar un proyecto.	B	A
Autenticar Usuario	El sistema debe permitir restringir el acceso a determinadas funcionalidades a usuarios.	A	M

Generar Reporte	El sistema debe permitir generar un reporte con los resultados de todas las encuestas por proyecto.	A	A
Rellenar Encuesta	El sistema debe permitir rellenar una encuesta a un cliente.	A	M

2.2.2 Requisitos no funcionales

RNF 1 Software:

RNF 1.1 EL Sistema operativo debe ser Linux o Windows 10.

RNF 1.2 El navegador web a usar debe ser Mozilla Firefox superior a la versión 80.0, Google Chrome superior a la versión 70.0, Microsoft Edge superior a la versión 60.5 u Opera superior a la versión 50.0.

RNF 2 Hardware:

RNF 2.1 Se requiere de 1GB de memoria RAM para acceder al sistema.

RNF 2.2 El servidor requiere 50 GB de disco duro para alojar los datos generados por el sistema.

RNF 3 Restricciones en el diseño e implementación:

RNF 3.1 Debe utilizar el framework de desarrollo Django.

RNF 3.2 Debe usar el lenguaje de programación Python.

RNF 3.3 Herramienta de modelado Visual Paradigm 15.2.

RNF 4 Seguridad:

RNF 4.1 Debe permitir autenticar un usuario, este dispondrá de distintos permisos según el rol asignado.

RNF 4.2 La contraseña de los usuarios debe poseer un mínimo de 8 caracteres, conteniendo letras y números o caracteres alfanuméricos.

RNF 5 Rendimiento:

RNF 5.1 El tiempo de respuesta no debe de excederse de 5 segundos.

RNF 5.2 Permitir el acceso de hasta 100 personas simultáneamente.

RNF 6 Apariencia o Interfaz Externa:

RNF 6.1 Diseño gráfico con colores en la gama de colores azul y blanco.

RNF 6.2 El diseño del sistema debe contar con una interfaz responsive adaptado a dispositivos móviles.

RNF 6.3 La interfaz debe ser intuitiva y de fácil comprensión para los usuarios.

2.3 Historias de Usuario**HU Listar Usuario**

Tabla 8: Descripción de HU Listar Usuario.

Historia de Usuario	
Número: 1	Usuario: Administrador
Nombre: Listar Usuario	
Prioridad en negocio: Baja	Riesgo en desarrollo: Bajo
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Cristhiam D. López Nuñez	
Descripción: El administrador podrá listar los usuarios.	
Validación: El listado solo es accesible por un usuario con el rol de administrador.	

HU Adicionar Usuario

Tabla 9: Descripción de HU Adicionar Usuario.

Historia de Usuario	
Número: 2	Usuario: Administrador
Nombre: Adicionar Usuario	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Cristhiam D. López Nuñez	
Descripción: El administrador podrá adicionar un usuario.	

Validación:

El formulario solo es accesible por un usuario con el rol de administrador, no puede contener campos vacíos.

HU Modificar usuario

Tabla 10: Descripción de HU Modificar Usuario.

Historia de Usuario	
Número: 3	Usuario: Administrador
Nombre: Modificar usuario	
Prioridad en negocio: Media	Riesgo en desarrollo: Alto
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Cristhiam D. López Nuñez	
Descripción: El administrador podrá modificar un usuario.	
Validación: El formulario solo es accesible por un usuario con el rol de administrador, puede contener campos vacíos.	

HU Eliminar Usuario

Tabla 11: Descripción de HU Eliminar Usuario.

Historia de Usuario	
Número: 4	Usuario: Administrador
Nombre: Modificar Usuario	
Prioridad en negocio: Baja	Riesgo en desarrollo: Alto
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Cristhiam D. López Nuñez	
Descripción: El administrador podrá adicionar, eliminar, listar o modificar uno de sus proyectos.	
Validación: El botón solo es accesible por un usuario con el rol de administrador.	

HU Listar Cliente*Tabla 12: Descripción de HU Listar Cliente.*

Historia de Usuario	
Número: 5	Usuario: Desarrollador
Nombre: Listar Cliente	
Prioridad en negocio: Baja	Riesgo en desarrollo: Baja
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Cristhiam D. López Nuñez	
Descripción: El desarrollador podrá listar los usuarios con el rol de cliente.	
Validación: El listado solo es accesible por un usuario con el rol de administrador o desarrollador.	

HU Adicionar Cliente*Tabla 13: Descripción de HU Adicionar Cliente.*

Historia de Usuario	
Número: 6	Usuario: Desarrollador
Nombre: Adicionar Cliente	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Cristhiam D. López Nuñez	
Descripción: El desarrollador podrá adicionar un usuario con el rol de cliente.	
Validación: El formulario solo es accesible por un usuario con el rol de administrador o desarrollador, no puede contener campos vacíos.	

HU Modificar Cliente*Tabla 14: Descripción de HU Modificar Cliente.*

Historia de Usuario	
Número: 7	Usuario: Desarrollador
Nombre: Modificar Cliente	
Prioridad en negocio: Media	Riesgo en desarrollo: Alto
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Cristhiam D. López Nuñez	
Descripción: El desarrollador podrá modificar un usuario con el rol de cliente.	
Validación: El formulario solo es accesible por un usuario con el rol de administrador o desarrollador, puede contener campos vacíos.	

HU Eliminar Cliente*Tabla 15: Descripción de HU Eliminar Cliente*

Historia de Usuario	
Número: 8	Usuario: Desarrollador
Nombre: Eliminar Cliente	
Prioridad en negocio: Baja	Riesgo en desarrollo: Alto
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Cristhiam D. López Nuñez	
Descripción: El desarrollador podrá eliminar un usuario con el rol de cliente.	
Validación: El botón solo es accesible por un usuario con el rol de administrador o desarrollador.	

2.3.1 Estimación de historias de usuarios

Tabla 16: Estimación de Historias de Usuario.

Historia de Usuario	Mejor Caso	Peor Caso	Resultado
Listar Usuario	2 horas	4 horas	2 horas
Adicionar Usuario	3 horas	6 horas	4 horas
Modificar Usuario	4 horas	6 horas	4 horas
Eliminar Usuario	2 horas	6 horas	3 horas
Listar Cliente	2 horas	4 horas	2 horas
Adicionar Cliente	3 horas	6 horas	4 horas
Modificar Cliente	4 horas	6 horas	4 horas
Eliminar Cliente	2 horas	6 horas	3 horas
Listar Encuesta	3 horas	4 horas	2 horas
Adicionar Encuesta	4 horas	6 horas	4 horas
Modificar Encuesta	5 horas	7 horas	5 horas
Eliminar Encuesta	3 horas	6 horas	4 horas
Listar Proyecto	2 horas	4 horas	2 horas
Adicionar Proyecto	3 horas	6 horas	4 horas
Modificar Proyecto	4 horas	7 horas	4 horas
Eliminar Proyecto	2 horas	4 horas	3 horas
Autenticar Usuario	2 horas	4 horas	3 horas
Generar Reporte	2 horas	4 horas	3 horas
Rellenar Encuesta	3 horas	5 horas	3 horas

2.4 Diagrama Entidad-Relación

Un diagrama entidad-relación permite representar las entidades más relevantes presentes en la base de datos de un sistema, detallando sus campos, propiedades y relaciones.

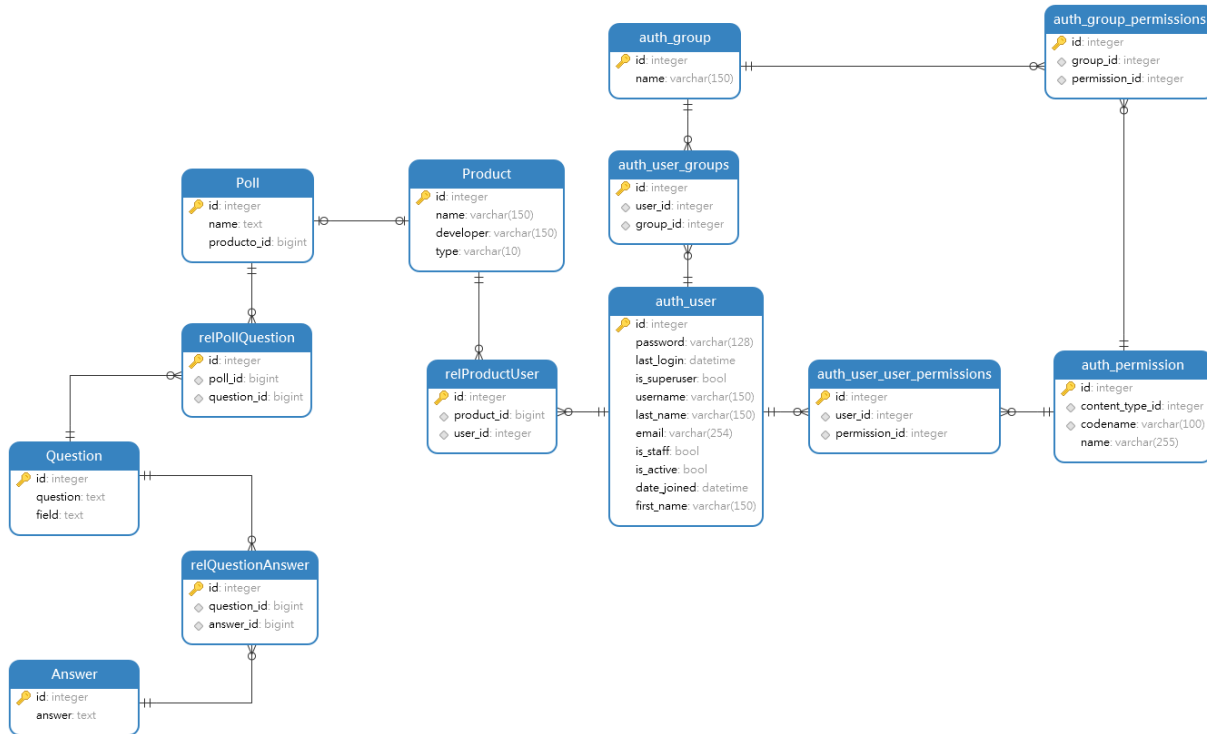


Figura 3: Diagrama Entidad-Relación

2.5 Diseño Arquitectónico de la aplicación

El diseño arquitectónico nos permite definir cómo debe organizarse un sistema y cómo tiene que diseñarse la estructura global de este. Es el enlace entre el diseño y la ingeniería de requisitos, que identifica los principales componentes estructurales en un sistema y la relación entre ellos, localizando problemas en el sistema y proporcionando soluciones generales para cada uno.

2.5.1 Estándares de Codificación

Los estándares de codificación o programación son términos que describen convenciones para escribir código fuente en los lenguajes de programación.

- Utilizar 4 espacios para la sangría.
- Cada línea de código no debe exceder los 80 caracteres en la medida de lo posible.
- La declaración de importación debe escribirse en líneas separadas.
- Poner sufijo *Exception* a todas las excepciones.
- Poner sufijo *Template* a las vistas.
- Poner sufijo *Model* a los modelos.
- Poner sufijo *View* a las clases controladoras.
- Las funciones y nombres de clases usaran la codificación *CamelCase*.

- Las variables usaran las letras en minúscula y separando las palabras con un guion bajo.
- Usar verbos para los métodos y sustantivos para atributos.
- Escribir comentarios para encontrar código más fácilmente.
- Los estilos van en archivos separados del código HTML.
- Usar nombres de variable significativos como *cuenta* o *valor_previo* en lugar de *c* o *vp*.

2.5.2 Patrón de arquitectura

El patrón de arquitectura Modelo Vista Template (MVT) tiene como principio que cada uno de los componentes esté separado en diferentes objetos, quiere decir que los componentes no se pueden combinar dentro de una misma clase.

El patrón de diseño MVT es constituido por tres componentes:

- **Modelo:** contiene sólo los datos de aplicación más puros, como entidades u objetos de la base de datos, no contiene ninguna lógica o método para presentar dichos datos al usuario.
- **Vista:** existe entre el template y el modelo, es la encargada de tomar los datos del modelo, procesarlos y enviarlos al template.
- **Template:** presenta los datos del modelo al usuario. El template accede a los datos del modelo, pero no interpreta o manipula la información.

Las tres partes de MVT están interconectadas, el template muestra los datos generados por la vista, la cual los extrae y modifica del modelo.

2.5.3 Patrones generales de asignación de responsabilidades (GRASP)

- **Experto:** básico para asignación de responsabilidades. La clase encargada de crear un objeto es la que conoce todos sus métodos y atributos. De este modo obtendremos mayor cohesión, seguridad y encapsulamiento.
- **Bajo Acoplamiento:** usar la menor interacción o ligamiento entre clases, para crear independencia, así, en caso de modificar alguna clase, esto genera la menor afectación posible en la estructura del sistema.
- **Alta Cohesión:** la información almacenada por cada clase debe ser la correspondiente a esta.

2.6 Diagramas de Clases de Diseño con Estereotipos Web

Se han representado los diagramas de clases del diseño en los cuales se encuentran las clases, relaciones y formularios usados para cada caso de uso en específico. En lugar de usar los

estereotipos básicos, se han usado los estereotipos para diagramas UML basados en la web, para una mayor comprensión de los mismos y de la estructura y el funcionamiento de cada caso de uso.

2.6.1 CU Gestionar Usuario

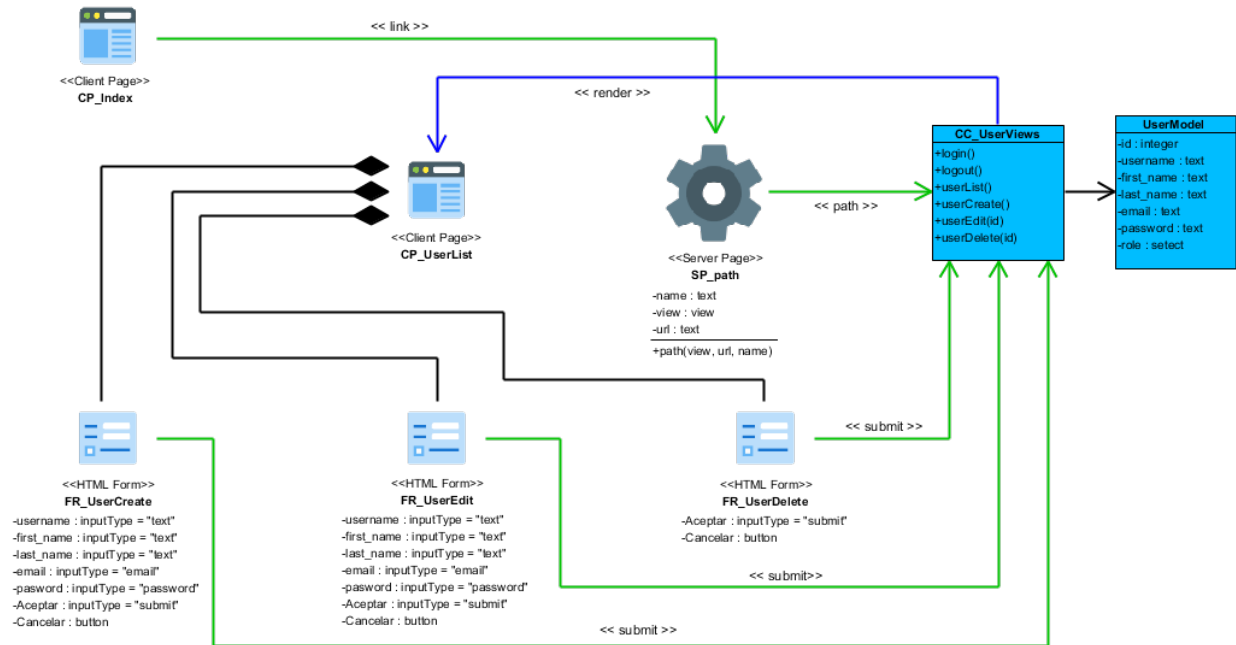


Figura 4: Diagrama de Clases del Caso de Uso Gestionar Usuario.

2.6.2 CU Rellenar Encuesta

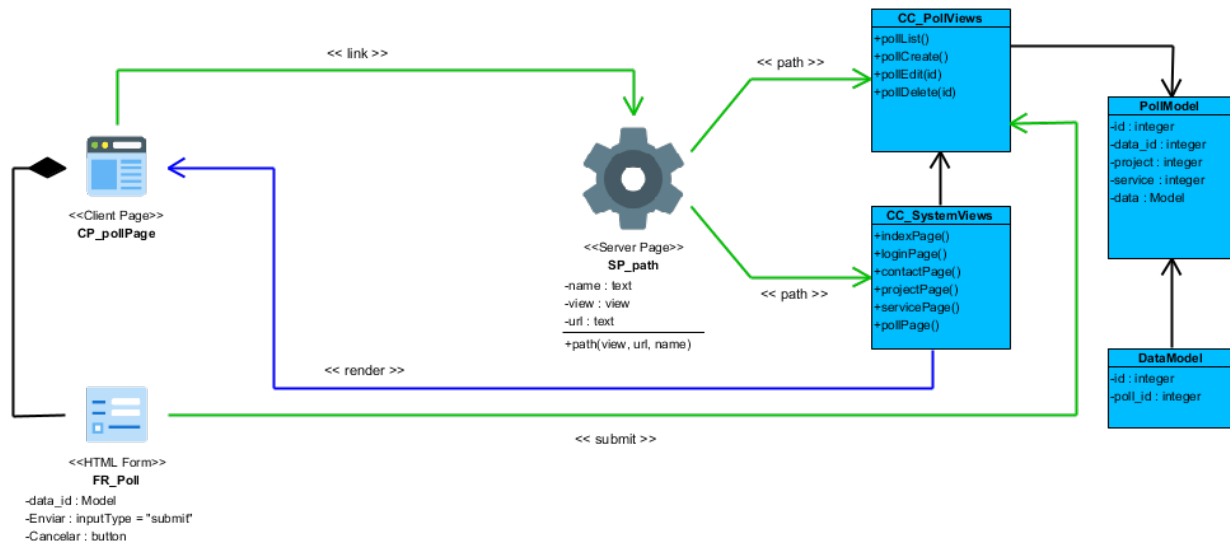


Figura 5: Diagrama de Clases del Caso de Uso Rellenar Encuesta.

2.6.3 CU Autenticar Usuario

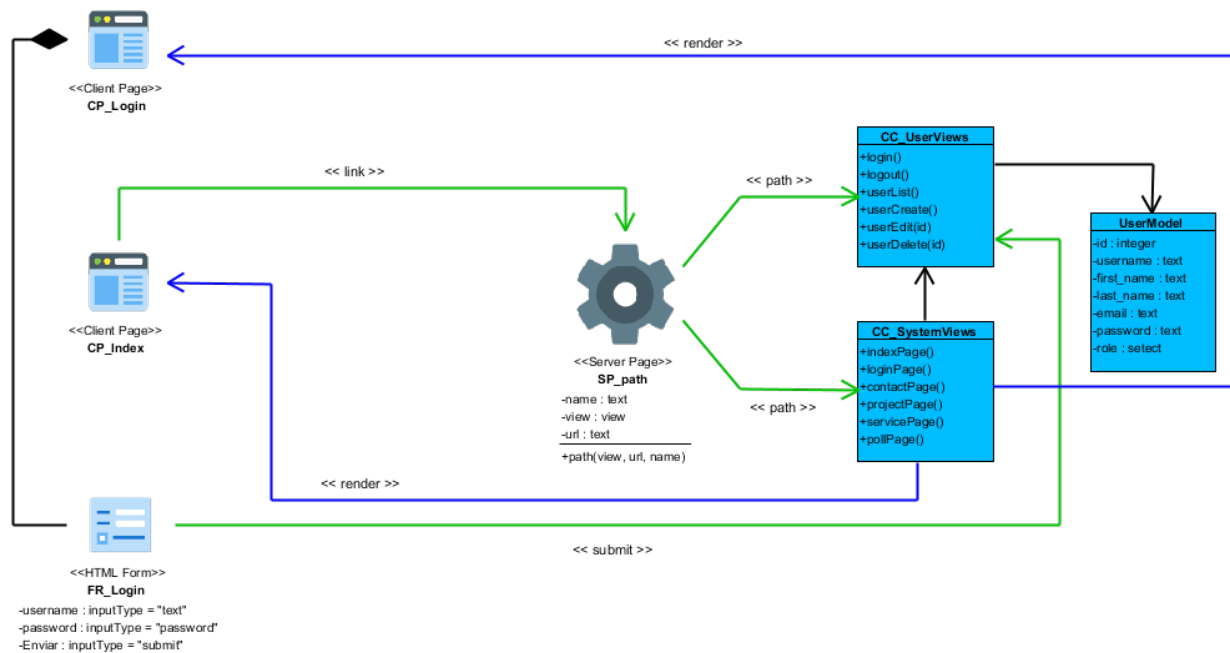


Figura 6: Diagrama de Clases del Caso de Uso Autenticar Usuario.

2.7 Diagramas de Secuencia

Los diagramas de secuencia se centran específicamente en líneas de vida de los procesos y objetos del sistema, mostrando los mensajes intercambiados entre ellos para ejecutar una

función antes de que la línea de vida termine. Cada diagrama representa los detalles de un caso de uso, modelando la lógica de una operación, o una función, para ver con claridad como los objetos y los componentes interactúan entre sí y planificar y comprender la funcionalidad detallada de un escenario. Se han representado los diagramas correspondientes a los Casos de Uso descritos anteriormente en el epígrafe 2.3 y 2.5.

2.7.1 CU Gestionar Usuario

Para una mayor comprensión del Caso de Uso se ha separado el diagrama de secuencia para cada escenario, mostrando detalladamente cada interacción entre las distintas clases que lo componen.

2.7.1.1 CU Gestionar Usuario: Crear

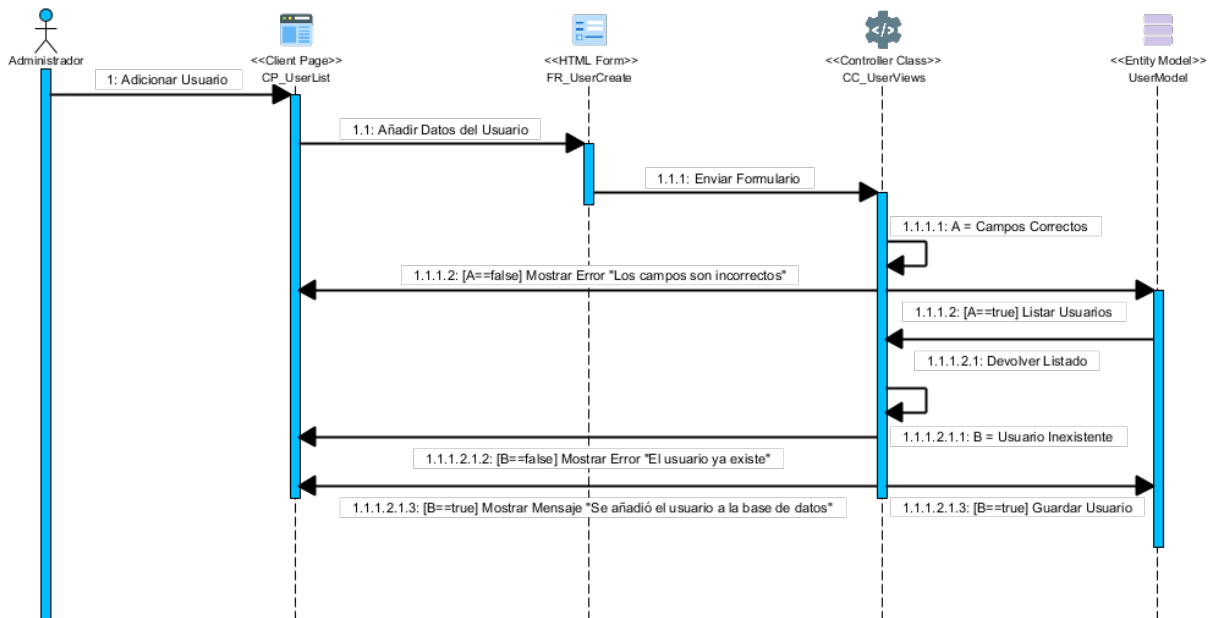


Figura 7: Diagrama de Clases del Caso de Uso Autenticar Usuario.

2.7.1.2 CU Gestionar Usuario: Listar

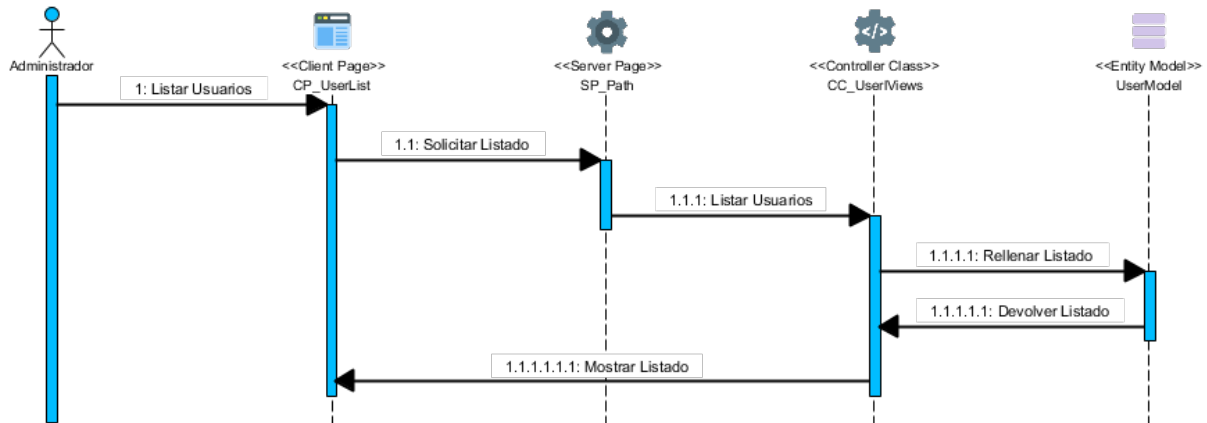


Figura 8: Diagrama de Clases del Caso de Uso Autenticar Usuario.

2.7.1.3 CU Gestionar Usuario: Editar

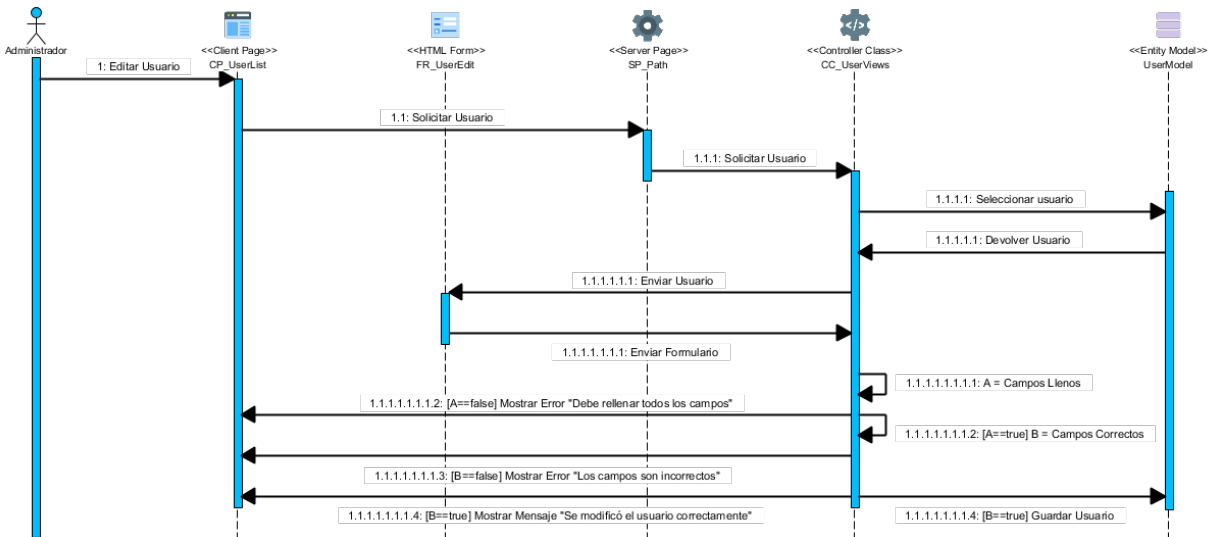


Figura 9: Diagrama de Clases del Caso de Uso Autenticar Usuario.

2.7.1.4 CU Gestionar Usuario: Eliminar

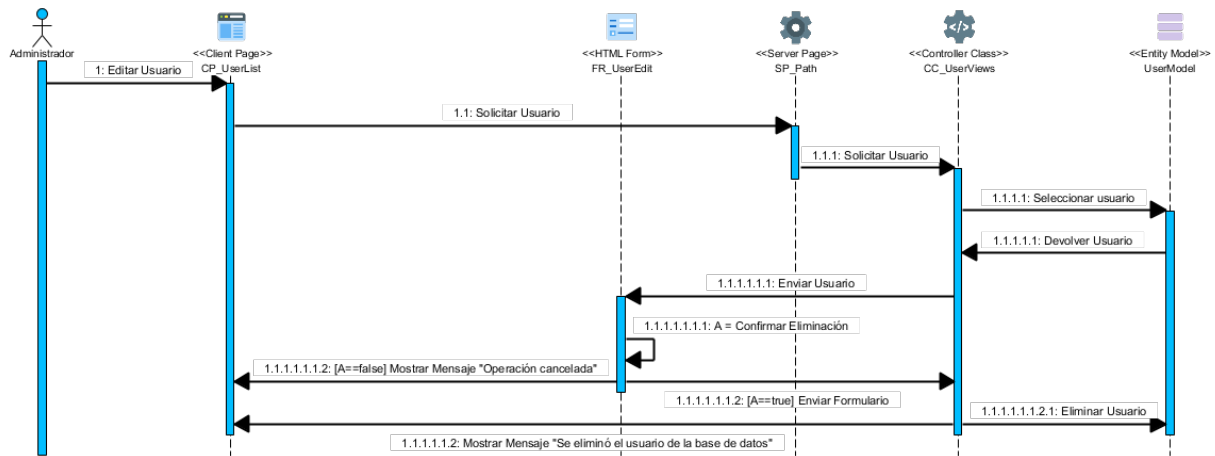


Figura 10: Diagrama de Clases del Caso de Uso Autenticar Usuario.

2.7.2 CU Rellenar Encuesta

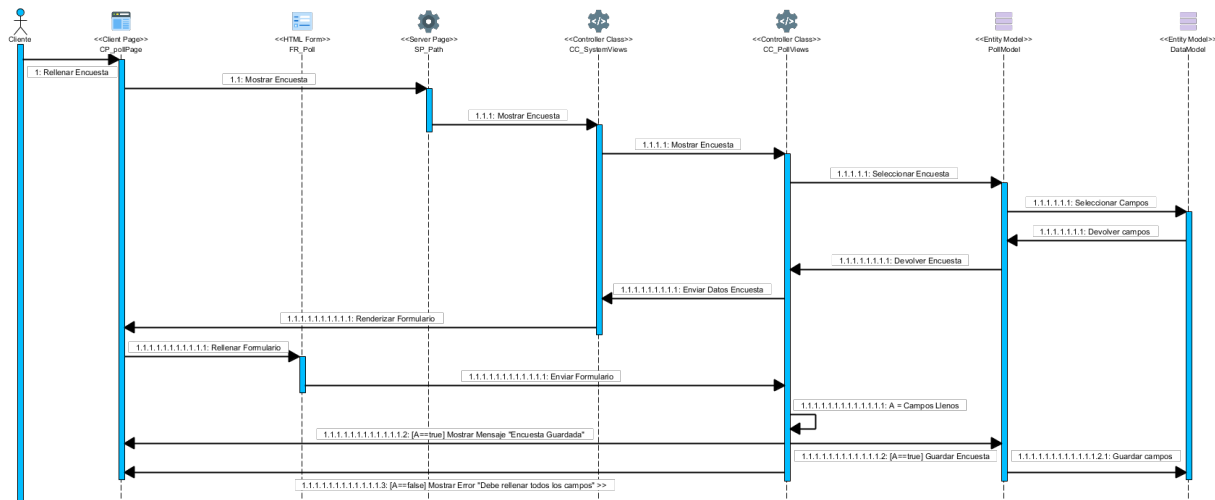


Figura 11: Diagrama de Clases del Caso de Uso Autenticar Usuario.

2.7.3 CU Autenticar Usuario

□

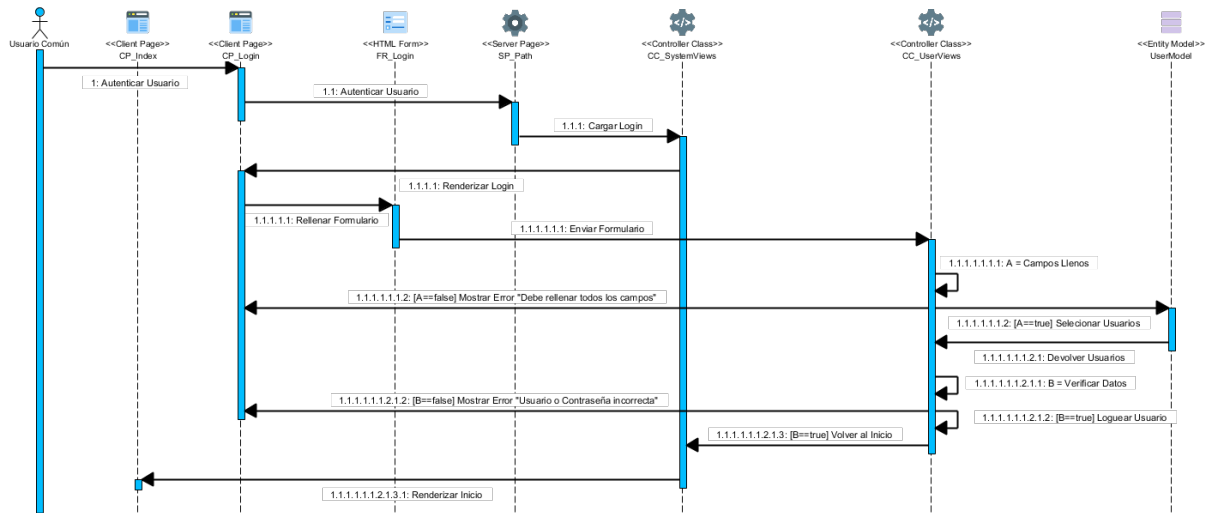


Figura 12: Diagrama de Clases del Caso de Uso Autenticar Usuario.

Conclusiones parciales del Capítulo

Con el desarrollo del capítulo se ha diseñado un sistema específicamente para las necesidades empresariales concretas que se puede adaptar a través de la programación de software personalizado para adaptarse a las necesidades operativas de una organización.

El sistema es más fácil e intuitivo de implementar y no contiene requisitos o implementaciones innecesarias. Se ha garantizado la flexibilidad en comparación con los paquetes de software debido a que el producto se puede cambiar y modificarse con el tiempo según los cambios en los requerimientos del negocio.

Capítulo 3. Implementación y Pruebas del Sistema.

La calidad en sentido general, tanto de software como de otros tipos de productos, es un elemento que cada día se tiene más en cuenta a nivel mundial y su logro se relaciona directamente con el proceso que se emplee para obtenerla.

Los sistemas informáticos, programas y aplicaciones han crecido a niveles inimaginables en complejidad e interoperabilidad, con lo cual también se han incrementado las posibilidades de defectos (bugs), a simple vista insignificantes, pero que pudieran adquirir proporciones catastróficas.

3.1 Implementación de la Propuesta de Solución

Se implementaron las Historias de Usuario en tres iteraciones, repasando y corrigiendo los detalles con cada una. Al culminar el proceso se realizó la revisión del plan de iteraciones y se corrigieron los cambios identificados. Específicamente las pruebas de software permiten evaluar las soluciones y determinar el nivel de calidad que poseen, por lo que se debe definir un proceso que se pueda emplear en el entorno de desarrollo de aplicaciones informáticas en la universidad.

- **Iteración No.1:** Se implementó la HU Gestionar Usuario y parte de la HU Gestionar Cliente, quedando pendiente el Requisito Funcional Modificar Cliente.
- **Iteración No.2:** Se culminó la implementación de la HU Gestionar Cliente y se implementó, además, la HU Gestionar Proyecto y parte de la HU Gestionar Encuesta, quedando pendiente el Requisito Funcional Eliminar Encuesta.
- **Iteración No.3:** Se finalizó la HU Gestionar Encuesta y se implementaron las HU Autenticar Usuario, HU Rellenar Encuesta y HU Generar Reporte.

Como resultado de las iteraciones anteriores, se realizó la siguiente tabla que muestra el tiempo estimado y real de cada Historia de usuario y su respectiva iteración.

Tabla 17: Iteraciones por Historia de Usuario.

Historia de Usuario	Tiempo Estimado	Tiempo Real	Iteración
Listar Usuario	4 horas	2 horas	1
Adicionar Usuario	6 horas	4 horas	1
Modificar Usuario	6 horas	4 horas	1
Eliminar Usuario	6 horas	3 horas	1

Listar Cliente	4 horas	2 horas	1
Adicionar Cliente	6 horas	4 horas	1
Modificar Cliente	6 horas	4 horas	1
Eliminar Cliente	6 horas	3 horas	1 y 2
Listar Encuesta	4 horas	2 horas	2
Adicionar Encuesta	6 horas	4 horas	2
Modificar Encuesta	7 horas	5 horas	2
Eliminar Encuesta	6 horas	4 horas	2 y 3
Listar Proyecto	4 horas	2 horas	3
Adicionar Proyecto	6 horas	4 horas	3
Modificar Proyecto	7 horas	4 horas	3
Eliminar Proyecto	4 horas	3 horas	3
Autenticar Usuario	4 horas	3 horas	3
Generar Reporte	4 horas	3 horas	3
Rellenar Encuesta	5 horas	3 horas	3

3.2 Diagrama de Despliegue

Para proporcionar una base para comprender la distribución física del sistema en un conjunto de nodos de proceso, se utiliza una vista de la arquitectura llamada vista de despliegue en el flujo de trabajo de análisis y diseño. La vista de despliegue ilustra la distribución de procesos en un conjunto de nodos del sistema, incluida la distribución física de procesos.

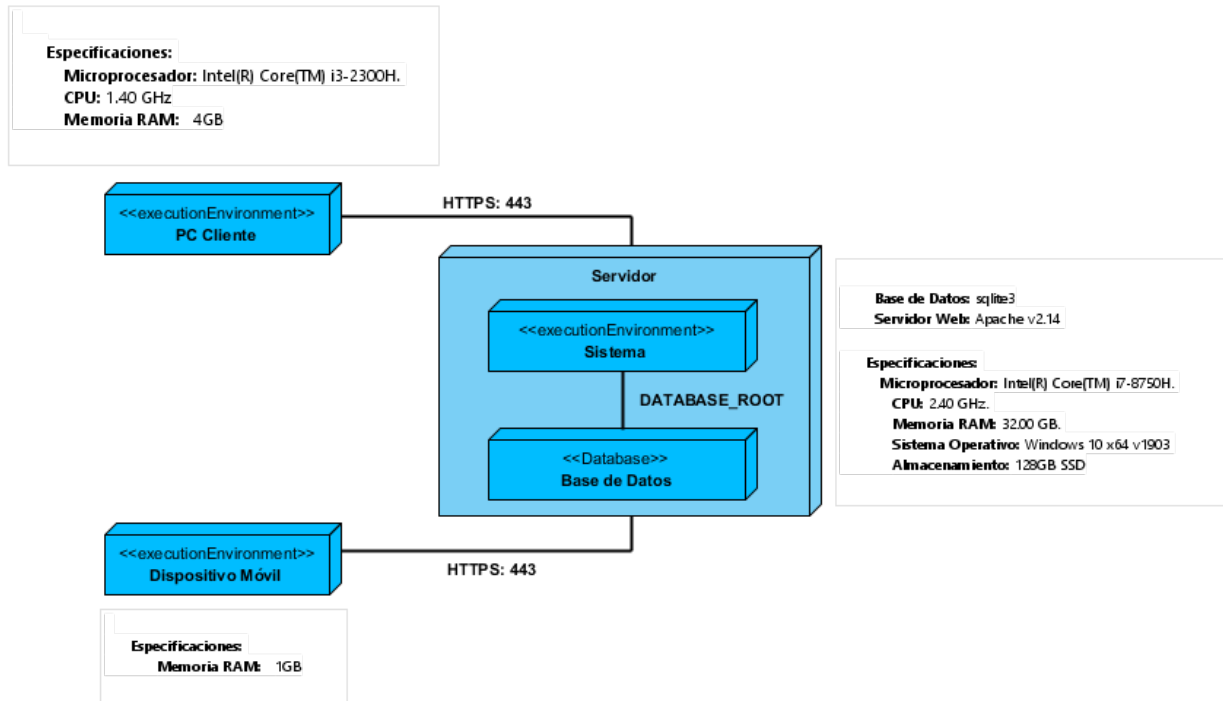


Figura 13: Diagrama de Despliegue.

PC Cliente - Dispositivo Móvil

Es el dispositivo electrónico por el cual el usuario interactúa con la aplicación. El cliente es un ordenador o dispositivo que consume un servicio remoto en otro ordenador conocido como servidor normalmente a través de una red.

Servidor

Es el host virtual donde se encuentra el servidor de la aplicación o sistema el cual gestiona la mayor parte de las funciones de acceso a los datos de la aplicación.

HTTPS - HTTP

Para distinguir una comunicación o página web segura, la *url* debe comenzar con *"https://"* (empleando el puerto 443 por defecto. HTTPS fue adoptado como estándar web y opera en la capa más alta del modelo TCP/IP, la capa de Aplicación.

3.3 Pruebas de Software para la evaluación de la Propuesta de Solución

La prueba es un proceso que se enfoca sobre la lógica interna del software y las funciones externas. Es un proceso de ejecución de un programa con la intención de descubrir un error, no puede asegurar la ausencia de defectos; sólo puede demostrar que existen defectos en el software.

Las pruebas de software comprenden el conjunto de actividades que se realizan para identificar posibles fallos de funcionamiento, configuración o usabilidad de un programa o aplicación, por medio de pruebas sobre el comportamiento del mismo.

3.3.1 Pruebas de Caja Negra

La técnica de Caja Negra puede utilizarse para lograr objetivos de cobertura de entrada y salida, con entradas humanas, vía interfaces a un sistema, o parámetros de interfaz de las pruebas de integración". En esta técnica es importante identificar las clases de equivalencia, por ejemplo, rango de valores entre 1 y 10 serán las clases de equivalencia, es decir que todo valor menor a 1 y todo valor mayor a 10 serán valores inválidos. Luego se generan los casos de prueba con diferentes valores para asegurar que la aplicación solo acepte valores entre 1 y 10 (32).

Se emplearon las pruebas de caja negra con el objetivo de encontrar errores en las siguientes categorías:

- Funciones incorrectas o faltantes.
- Errores de interfaz.
- Errores en las estructuras de datos.
- Errores de comportamiento o rendimiento.

Para desarrollar el método de caja negra se utilizan las técnicas (33):

- **Partición de equivalencia:** divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.
- **Análisis de valor de frontera:** prueba la habilidad del programa para manejar datos que se encuentran en los límites o fronteras aceptables.

Para poner en práctica este método de pruebas, según las técnicas que se describieron anteriormente, se toma en cuenta los Diseños de Casos de Pruebas (DCP). Una clase de equivalencia representa un conjunto de estados válidos o inválidos para condiciones de entrada. Por su parte, el análisis de valor de frontera es una técnica de diseño de casos de prueba que complementan la partición de equivalencia. En lugar de enfocarse exclusivamente en las condiciones de entrada, también deriva casos de prueba a partir del dominio de salida (33).

Luego de realizar el diseño de casos de prueba se completó la tabla de control de documento y control de cambios, se tomó para esta prueba el Caso de Uso Gestionar Usuario y el Requisito Funcional Autenticar Usuario.

Descripción general

El sistema permite autenticar usuario

Condiciones de ejecución

El usuario debe estar registrado en el sistema con anterioridad.

RF Autenticar Usuario

Escenario	Descripción	Nombre de usuario	Contraseña	Respuesta del sistema	Flujo central
Autenticar usuario de forma correcta	El sistema autentica un usuario de forma correcta.	usuario	Contraseña123*	El sistema autentica al usuario y le permite el acceso a las funcionalidades del sistema, según su rol.	1-El usuario accede a la url del portal web y selecciona la opción "Iniciar Sesión" al final de la misma. 2. El sistema muestra una interfaz con el formulario de autenticación. 3- El usuario introduce la información y presiona el botón: "Enviar".
Autenticar usuario de forma incorrecta	El sistema no autentica un usuario de forma incorrecta.	usuario	123	El sistema no autentica al usuario y emite el emsaje "Campos Inválidos"	
		-usuario	Contraseña123*		
		.123	123		
	El sistema no autentica un usuario dejando campos obligatorios vacíos.	usuario			
			Contraseña123*		
Cerrar Sesión en el Sistema	El sistema desloguea al usuario de forma correcta.			Al usuario cerrar la sesión, se desloguea del sistema y este lo dirige a la portada del mismo.	El usuario autenticado selecciona el icono de usuario, y luego, del menú correspondiente en la barra de administración la opción "Cerrar sesión"

Figura 14: Diagrama de Caso de Prueba RF Autenticar Usuario.

Descripción general

El sistema permite crear un usuario

Condiciones de ejecución

El usuario debe estar logueado y tener el rol administrador

RF Adicionar usuario

Escenario	Descripción	Nombre de usuario	Correo electrónico	Contraseña	Confirmar contraseña	Nombre	Apellidos	Respuesta del sistema
Crear usuario de forma correcta	El sistema inserta un usuario de forma correcta.	usuario	usuario@subdominios.dominio	contraseña123*	contraseña123*	Nombre	Apellido Apellido	El sistema adiciona el usuario
Crear usuario de forma incorrecta	El sistema no inserta un usuario de forma incorrecta.	usuario	usuario.subdominios.dominio	1234	1234	Nombre	Apellido Apellido	El sistema no adiciona al usuario y emite el mensaje "No se pudo añadir el usuario: Campos Incorrectos"
		usuario	usuario@subdominios.dominio	contraseña123*	adssdsdf	Nombre	Apellido Apellido	El sistema no adiciona al usuario y emite el mensaje "No se pudo añadir el usuario: Campos Incorrectos"
Crear usuario dejando campos vacíos.	El sistema no inserta un usuario dejando campos vacíos.					Nombre	Apellido Apellido	El sistema no adiciona al usuario y emite el mensaje "No se pudo añadir el usuario: Campos Vacíos"
Crear usuario con valores existentes.	El sistema no inserta un usuario con valores existentes.	valor existente	valor existente					El sistema no adiciona al usuario y emite el mensaje "No se pudo añadir el usuario: Usuario o correo existente"
Cancelar operación	El sistema no realiza ninguna operación.							No realiza ninguna operación y regresa a la página anterior.

Figura 15: Diagrama de Caso de Prueba RF Adicionar Usuario.

Descripción general

El sistema permite modificar un usuario

Condiciones de ejecución

El usuario debe estar logueado y tener el rol administrador

RF Modificar Usuario

Escenario	Descripción	Nombre de usuario	Correo electrónico	Contraseña	Nombre	Apellidos	Respuesta del sistema
Modificar usuario de forma correcta	El sistema modifica un usuario de forma correcta.	usuario	usuario@subdominios.dominio	contraseña123*	Nombre	Apellido Apellido	El sistema modifica el usuario
Modificar usuario de forma incorrecta	El sistema no modifica un usuario de forma incorrecta.	usuario	usuario.subdominios.dominio	1234	Nombre	Apellido Apellido	El sistema no adiciona al usuario y emite el mensaje "No se pudo añadir el usuario: Campos Incorrectos"
		usuario	usuario@subdominios.dominio	contraseña123*	Nombre	Apellido Apellido	El sistema no modifica al usuario y emite el mensaje "No se pudo modificar el usuario: Campos Incorrectos"
Modificar usuario dejando campos vacíos.	El sistema modifica los campos que contengan datos.				Nombre	Apellido Apellido	El sistema modifica el usuario
Modificar usuario con valores existentes.	El sistema no modifica un usuario con valores existentes.	valor existente	valor existente				El sistema no modifica al usuario y emite el mensaje "No se pudo modificar el usuario: Usuario o correo existente"
Cancelar operación	El sistema no realiza ninguna operación.						No realiza ninguna operación y regresa a la página anterior.

Figura 16: Diagrama de Caso de Prueba RF Modificar Usuario.

Descripción general

El sistema muestra el listado de usuarios.

Condiciones de ejecución

El usuario debe estar logueado y tener el rol administrador

RF Listar Usuarios

Escenario	Descripción	Respuesta del sistema	Flujo central
Mostrar listado de usuarios.	El sistema muestra el listado de usuarios.	El sistema muestra el listado de los usuarios existentes con su id, nombre de usuario, nombre, apellidos, correo electrónico, el último acceso, fecha de creación y las operaciones que se pueden realizar.	El usuario se autentica en el sistema y accede al módulo Usuarios ubicado en la barra de administración y se muestra una página con el listado de todos los usuarios existentes en el sistema.

Figura 17: Diagrama de Caso de Prueba RF Listar Usuario.

3.3.2 Pruebas de Regresión

Las pruebas de regresión son pruebas de un programa previamente probado que ha sufrido modificaciones, para asegurarse que no se han introducido o descubierto defectos en áreas del software que no han sido modificadas como resultado de los cambios realizados. Se realiza cuando el software o su entorno han sido modificados.

En el caso de la aplicación se realizaron las pruebas de regresión descritas al CRUD Gestionar Usuario y se arrojaron distintos resultados en tres distintas iteraciones:

Tabla 18: Especificación de Pruebas de Regresión.

Iteración	Error a Resolver	Problemas Encontrados
Iteración 1	No era posible añadir un usuario	Al Eliminar Usuario no se eliminaba el usuario correspondiente al identificador enviado
Iteración 2	Se corrigió el error de la Iteración 1	La interfaz de eliminación de usuario mostraba errores en el <i>responsive design</i>
Iteración 3	Se corrigió el error de la Iteración 2	No se encontraron problemas

3.3.3 Pruebas de Usabilidad

Las pruebas de usabilidad son un servicio de aseguramiento de calidad que consiste en probar un producto y detectar no conformidades sobre su facilidad de uso. El objetivo principal de estas pruebas es identificar los problemas de usabilidad, para mejorar el producto y la satisfacción de sus clientes. Proporcionar retroalimentación para mejorar el diseño. Valorar en qué medida se cumplen los objetivos de los usuarios y la organización y poder exponer cuantitativamente el grado de usabilidad que presenta el producto probado. (34)

La subcaracterísticas de usabilidad que define la ISO/IEC 25000 son (34):

- Reconocimiento apropiado: Capacidad del producto que permite al usuario entender si el software es adecuado para sus necesidades.
- Aprendibilidad: Capacidad del producto que permite al usuario aprender su aplicación.
- Operatividad: Capacidad del producto que permite al usuario operarlo y controlarlo con facilidad.
- Protección contra errores de usuario: Capacidad del sistema para proteger a los usuarios de cometer errores.
- Estética de la interfaz de usuario: Capacidad de la interfaz de usuario de agradar y satisfacer la interacción con el usuario.
- Accesibilidad: Capacidad del producto que permite que sea utilizado por usuarios con determinadas características y discapacidades.

Tabla 19: Lista de Chequeo de Pruebas de Usabilidad.

Elementos definidos por la metodología				
No	Indicador a evaluar	Evaluación	NP	Observación
Visibilidad del sistema				
1	¿Cada pantalla empieza con un título que describe su contenido?	Si		
2	¿Los iconos que aparecen se identifican claramente con lo que representan?	Si		
3	¿El menú de navegación aparece en un lugar destacado?	Si		
4	¿No utiliza más de siete opciones principales en el menú de navegación?	Si		

Lenguaje común entre sistema y usuario				
5	¿El lenguaje es simple, con un tono adecuado?	Si		
6	¿La información que se presenta en la aplicación es fácil de entender y memorizar?	Si		
7	¿Utiliza los conceptos establecidos para las funciones estándar? (“buscar” para las búsquedas, etc.)	No		Algunas funcionalidades presentan nombres especificados por el lenguaje.
8	¿Se utiliza siempre la misma nomenclatura para las mismas funciones?	No		Algunas funciones son predefinidas por el <i>framework</i> .
Libertad y control por parte del usuario				
9	¿Existe una manera lógica de acceder a páginas relacionadas o a otras secciones?	Si		
10	¿Tras una acción relevante hay una opción de vuelta atrás?	No		La opción de eliminar o modificar no se puede deshacer.
Consistencia y estándares				
11	¿Para tareas similares, los diálogos, formularios son similares?	Si		

12	¿Puede utilizarse en cualquier navegador?	Si		
13	¿El sitio tiene una URL correcta, clara y fácil de recordar?	No		Existen URLs compuestas que presentan una estructura compleja.
Estética y diseño minimalista				
14	¿Cumple el sitio con el principio de usabilidad de realizar las operaciones con un máximo de tres <i>click</i> ?	Si		
15	¿Existe suficiente contraste entre el color del fondo y el del texto?	Si		
16	¿Los tipos y tamaños de letra son legibles y distinguibles?	Si		
Prevención de errores				
17	¿Los botones de acción, (tales como “Enviar”) siempre son invocados por el usuario y no automáticamente invocados por el sistema cuando el último campo de un formulario ha sido lleno?	No		Algunas funciones son invocadas automáticamente sin presionar el botón.
Ayuda y documentación				
18	¿Información necesaria en página de contacto? (Atención al cliente,	Si		

	Consultas, Soporte técnico, Solicitudes de empleo.)			
19	¿Funcionan correctamente los formularios de contacto?	No		No implementado.
Flexibilidad y eficiencia de uso				
29	¿Las partes o secciones más importantes de los sitios son accesibles desde la página de inicio?	No		El panel de administrador solo es accesible desde el login.
21	¿El cursor se desplaza adecuadamente en un formulario al presionar "tabulador"?	Si		

3.3.2 Pruebas de Aceptación

Las pruebas de aceptación de usuario (*User Acceptance Testing* o Pruebas UAT) se enfocan en verificar si el sistema es "apto para el uso". Se diseñan principalmente a partir de las especificaciones de requerimientos, casos de uso y de los procesos de negocio definidos.

De forma general las pruebas de aceptación pueden afrontarse mediante los siguientes tipos (33):

- Pruebas de Aceptación del Usuario (UAT).
- Pruebas de Aceptación Operativa (OAT).
- Pruebas de Aceptación de la Normativa.
- Pruebas de Aceptación Alfa y Beta.

Para el caso de la propuesta de solución se sometió su funcionamiento al proceso de pruebas de aceptación número cuatro. Se tomó una representación de 10 usuarios que probaron el sistema durante 5 días consecutivos y se demostró que la aplicación está lista para ser desplegada en un entorno de producción.

Conclusiones parciales del Capítulo

Con el desarrollo del presente capítulo se ha demostrado que se requieren pruebas de software para señalar los errores que ocurren durante las fases de desarrollo. Mediante estas se asegura de que las actuaciones de la aplicación sean adecuadas y de que los clientes estén satisfechos con ella. Cuando el producto entregado es de calidad, ayuda a ganar la confianza de los clientes. Las pruebas de software con una ejecución de prueba estricta aseguran un menor costo de mantenimiento.

Conclusiones

Con el desarrollo del presente documento se pudo arribar a la conclusión de que la elaboración del marco teórico referencial es de suma importancia para cualquier trabajo escrito ya que permite realizar el análisis teórico del problema. Amplía el horizonte del estudio y guía al investigador para que se centre en su problema evitando desviaciones del planteamiento original.

El trabajo con una metodología de desarrollo de software permitió reducir el nivel de dificultad, organizar las tareas, agilizar el proceso y mejorar el resultado final de la aplicación a desarrollar. Enfrascándose en un proceso ágil y realizando varias iteraciones para corregir errores de implementación.

Se demostró la vital importancia que tiene la ingeniería de requerimientos para generar una adecuada especificación que contemplara claramente y sin ambigüedades los requerimientos del sistema. El proceso de la Ingeniería de Requerimientos sirvió para recopilar la información necesaria para establecer la funcionalidad que se quería alcanzar con el sistema.

Se confirmó que la validación de pruebas de software es una de las actividades más importantes y fundamentales en el desarrollo de un proyecto, ya que posibilita los procesos, métodos de trabajo y herramientas necesarias para garantizar la calidad de cualquier desarrollo.

La definición de las tecnologías y herramientas de desarrollo maximizaron la calidad en todo el ciclo de vida al desarrollar la aplicación informática mejorando la productividad y automatizando los procesos de creación, normalización y generación de la base de datos, generación de documentación, diagramas y del propio código.

Referencias

1. *Diccionario de la lengua española*. [En línea] 2021. [Citado el: 28 de 06 de 2021.] <https://dle.rae.es/conocimiento>.
2. ¿Qué es una encuesta? | QuestionPro. [En línea] [Citado el: 28 de 06 de 2021.] <https://www.questionpro.com/es/una-encuesta.html>.
3. Satisfacción del cliente. ¿Qué es y cómo medirla? [En línea] [Citado el: 29 de 06 de 2021.] <https://elviajedelcliente.com/satisfaccion-del-cliente/>.
4. Satisfacción del cliente: ¿qué es? ¿cómo gestionarla? [En línea] [Citado el: 04 de 07 de 2021.] <https://www.efficacy.com/es/satisfaccion-del-cliente-definicion-historia-retos-y-estrategia/>.
5. Significado de Calidad (Qué es, Concepto y Definición) - Significados. [En línea] [Citado el: 06 de 07 de 2021.] <https://www.significados.com/calidad/>.
6. Pérez, Mariana. Definición de Calidad. [En línea] 27 de 01 de 2021. [Citado el: 06 de 07 de 2021.] <https://conceptodefinicion.de/calidad/>.
7. Raffino, María Estela. "Cuestionario". Concepto.de. [En línea] 11 de 02 de 2021. [Citado el: 06 de 07 de 2021.] <https://concepto.de/cuestinario/>.
8. "Cuestionario". En: Significados.com. [En línea] 06 de 07 de 2021. <https://www.significados.com/cuestionario/>.
9. SoftExpert Encuesta. [En línea] [Citado el: 06 de 07 de 2021.] <https://www.softexpert.com/es/produo/gestion-encuestas/>.
10. Gestión de encuestas - CRR Marketing. [En línea] [Citado el: 06 de 07 de 2021.] <https://crrmarketing.es/gestion-encuestas/>.
11. Sistemas de encuestas a clientes — Qmatic. [En línea] [Citado el: 06 de 07 de 2021.] <https://www.qmatic.com/es-es/soluciones/opiniones-de-los-clientes/>.
12. Qué es Frontend y Backend, diferencias y características – Platzi. [En línea] [Citado el: 06 de 07 de 2021.] <https://platzi.com/blog/que-es-frontend-y-backend/>.
13. ¿Qué es HTML/CSS/JS? (platzi.com). [En línea] [Citado el: 06 de 07 de 2021.] <https://platzi.com/clases/1050-programacion-basica/5104-que-es-htmlcssjs/>.
14. HTML: Lenguaje de etiquetas de hipertexto. [En línea] 14 de 11 de 2021. [Citado el: 15 de 11 de 2021.] <https://developer.mozilla.org/es/docs/Web/HTML>.
15. CSS: Hojas de Estilo en Cascada. [En línea] 14 de 11 de 2021. [Citado el: 15 de 11 de 2021.] <https://developer.mozilla.org/es/docs/Web/CSS>.
16. ¿Qué es JavaScript? [En línea] 14 de 11 de 2021. [Citado el: 15 de 11 de 2021.] https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/What_is_JavaScript.
17. Qué es Python (desarrolloweb.com). [En línea] [Citado el: 06 de 07 de 2021.] <https://desarrolloweb.com/articulos/1325.php>.

18. Bootstrap: guía para principiantes de qué es, por qué y cómo usarlo. [En línea] 12 de 04 de 2020. [Citado el: 15 de 11 de 2021.] <https://rockcontent.com/es/blog/bootstrap/>.
19. Frameworks con Material Design. [En línea] [Citado el: 15 de 11 de 2021.] <https://www.paradigmadigital.com/dev/frameworks-material-design-agilizar-desarrollo-aplicaciones/>.
20. Bootstrap vs. Material Design. [En línea] [Citado el: 15 de 11 de 2021.] <https://ifgeekthen.everis.com/es/bootstrap-vs-material-design>.
21. Introducción a Django - Aprende sobre desarrollo web | MDN (mozilla.org). [En línea] [Citado el: 06 de 07 de 2021.] <https://developer.mozilla.org/es/docs/Learn/Server-side/Django/Introduction>.
22. Visual Paradigm - EcuRed. [En línea] [Citado el: 06 de 07 de 2021.] https://www.ecured.cu/Visual_Paradigm.
23. Funcionalidades - PyCharm (jetbrains.com). [En línea] [Citado el: 06 de 07 de 2021.] <https://www.jetbrains.com/es-es/pycharm/features/>.
24. WebStorm: funcionalidades (jetbrains.com). [En línea] [Citado el: 06 de 07 de 2021.] <https://www.jetbrains.com/es-es/webstorm/features/>.
25. Navicat Premium. [En línea] [Citado el: 06 de 07 de 2021.] <https://navicat.com/es/products/navicat-premium>.
26. DB-API 2.0 interfaz para bases de datos SQLite. [En línea] [Citado el: 15 de 11 de 2021.] <https://docs.python.org/es/3.10/library/sqlite3.html>.
27. sqlite3 — Base de datos relacional integrada. [En línea] [Citado el: 15 de 11 de 2021.] <https://rico-schmidt.name/pymotw-3/sqlite3/index.html>.
28. Qué es Postman y para qué sirve | OpenWebinars. [En línea] [Citado el: 06 de 07 de 2021.] <https://openwebinars.net/blog/que-es-postman/>.
29. HTML5: los secretos para lograr compatibilidad con los navegadores – RedUSERS. [En línea] [Citado el: 06 de 07 de 2021.] <https://www.redusers.com/noticias/html5-los-secretos-para-lograr-compatibilidad-con-los-navegadores/>.
30. Los navegadores más rápidos de 2021: ¿Cuál es el navegador web más rápido? | Avast. [En línea] [Citado el: 06 de 07 de 2021.] <https://www.avast.com/es-es/c-fastest-web-browsers>.
31. Metodologías para La Gestión y Desarrollo de Software. [En línea] [Citado el: 15 de 11 de 2021.] <https://www.scribd.com/doc/8255409/Metodologias-para-la-gestion-y-desarrollo-de-Software>.
32. Mera, Julián y Paz, Andres. *Análisis del proceso de pruebas de calidad de software*. 2016.
33. Pressman, Roger S. *Ingeniería del Software. Un enfoque práctico*. México : The McGraw-Hill Companies : s.n., 2021. ISBN: 978-607-15-0314-5.

34. Ferré Grau, Xavier. *Principios Básicos de Usabilidad para Ingenieros Software*. Madrid : Facultad de Informática, Universidad Politécnica de Madrid, 2021.
35. *Economipedia.com*. [En línea] Guillermo Westreicher, 23 de 02 de 2021. [Citado el: 29 de 06 de 2021.] <https://economipedia.com/definiciones/encuesta.html>.
36. Sánchez, Tamara Rodríguez. *Metodología de desarrollo para la Actividad productiva de la UCI*.
37. ESTRATEGIAS DE PRUEBA PARA SOFTWARE | SELECCIÓN DE TÉCNICAS DE INGENIERÍA DE SOFTWARE. [En línea] [Citado el: 12 de 11 de 2021.] Disponible en: https://virtual.itca.edu.sv/Mediadores/stis/42estrategias_de_prueba_para_software.html.
38. Pruebas de software. [En línea] [Citado el: 15 de 11 de 2021.] https://www.ecured.cu/Pruebas_de_software.
39. Ocampo Acosta, Alejandro y Correa Tapasco, Luisa Marcela. *Impacto de las pruebas no funcionales en la medición de la calidad del producto software desarrollado*. Colombia : Universidad Tecnológica de Pereira, 2011.
40. Echeverria, Delvis. *Tiempo de Respuestas y Experiencia de Usuario*. Montevideo, Uruguay : GeneXus Consulting, 2016.

Anexos

Tabla 20: Descripción de HU Listar Encuesta

Historia de Usuario	
Número: 9	Usuario: Desarrollador
Nombre: Listar Encuesta	
Prioridad en negocio: Baja	Riesgo en desarrollo: Baja
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Cristhiam D. López Nuñez	
Descripción: El desarrollador podrá listar las encuestas.	
Validación: El listado solo es accesible por un usuario con el rol de desarrollador	

Tabla 21: Descripción de HU Adicionar Encuesta

Historia de Usuario	
Número: 10	Usuario: Desarrollador
Nombre: Adicionar Encuesta	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Cristhiam D. López Nuñez	
Descripción: El desarrollador podrá adicionar una encuesta y relacionarla a un producto y a usuarios.	
Validación: El formulario solo es accesible por un usuario con el rol de desarrollador, no puede contener campos vacíos.	

Tabla 22: Descripción de HU Modificar Encuesta

Historia de Usuario	
Número: 11	Usuario: Desarrollador

Nombre: Modificar Encuesta	
Prioridad en negocio: Media	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Cristhiam D. López Nuñez	
Descripción: El desarrollador podrá modificar una encuesta.	
Validación: El formulario solo es accesible por un usuario con el rol de desarrollador, puede contener campos vacíos.	

Tabla 23: Descripción de HU Eliminar Encuesta

Historia de Usuario	
Número: 12	Usuario: Desarrollador
Nombre: Eliminar Encuesta	
Prioridad en negocio: Baja	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 3
Programador responsable: Cristhiam D. López Nuñez	
Descripción: El desarrollador podrá eliminar una encuesta.	
Validación: El botón solo es accesible por un usuario con el rol de desarrollador.	

Tabla 24: Descripción de HU Listar Proyecto

Historia de Usuario	
Número: 13	Usuario: Desarrollador
Nombre: Listar Proyecto	
Prioridad en negocio: Baja	Riesgo en desarrollo: Baja
Puntos estimados: 1	Iteración asignada: 3
Programador responsable: Cristhiam D. López Nuñez	

Descripción: El desarrollador podrá listar los proyectos.
Validación: El listado solo es accesible por un usuario con el rol de desarrollador

Tabla 25: Descripción de HU Adicionar Proyecto

Historia de Usuario	
Número: 14	Usuario: Desarrollador
Nombre: Adicionar Proyecto	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 3
Programador responsable: Cristhiam D. López Nuñez	
Descripción: El desarrollador podrá adicionar un proyecto y vincularlo a uno o varios usuarios.	
Validación: El formulario solo es accesible por un usuario con el rol de desarrollador, no puede contener campos vacíos.	

Tabla 26: Descripción de HU Modificar Proyecto

Historia de Usuario	
Número: 15	Usuario: Desarrollador
Nombre: Modificar Proyecto	
Prioridad en negocio: Media	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 3
Programador responsable: Cristhiam D. López Nuñez	
Descripción: El desarrollador podrá modificar un proyecto.	
Validación: El formulario solo es accesible por un usuario con el rol de desarrollador, puede contener campos vacíos.	

Tabla 27: Descripción de HU Eliminar Proyecto

Historia de Usuario	
Número: 16	Usuario: Desarrollador
Nombre: Eliminar Proyecto	
Prioridad en negocio: Baja	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 3
Programador responsable: Cristhiam D. López Nuñez	
Descripción: El desarrollador podrá eliminar un proyecto.	
Validación: El botón solo es accesible por un usuario con el rol de desarrollador.	

Tabla 28: Descripción de HU Autenticar Usuario

Historia de Usuario	
Número: 17	Usuario: --
Nombre: Autenticar Usuario	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 3
Programador responsable: Cristhiam D. López Nuñez	
Descripción: El usuario podrá autenticarse en el sistema.	
Validación: El formulario es accesible por todos los usuarios, no puede contener campos vacíos.	

Tabla 29: Descripción de HU Generar Reporte

Historia de Usuario	
Número: 18	Usuario: Desarrollador
Nombre: Generar Reporte	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 3

Programador responsable: Cristhiam D. López Nuñez
Descripción: El desarrollador podrá generar un reporte con los resultados de las encuestas de su producto.
Validación: El listado solo es accesible por un usuario con el rol de desarrollador.

Tabla 30: Descripción de HU Rellenar Encuesta

Historia de Usuario	
Número: 19	Usuario: Cliente
Nombre: Rellenar Encuesta	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 3
Programador responsable: Cristhiam D. López Nuñez	
Descripción: El cliente podrá resolver las encuestas relacionadas de su producto.	
Validación: El formulario solo es accesible por un usuario con el rol de cliente, no puede contener campos vacíos.	