



**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

CARRERA DE COMPUTACIÓN

**DESARROLLO DE UNA APLICACIÓN WEB QUE GESTIONE Y ADMINISTRE
COBROS Y USUARIOS PARA LOS ENTRENADORES DE LA ACADEMIA DE
ECUAVOLEY EN QUITO**

Trabajo de titulación previo a la obtención del título de:
Ingenieros en Ciencias de la Computación

AUTORES: OSCAR ANIBAL JIMÉNEZ LOGACHO
ERICK BOLIVAR TAPIA RAMIREZ

TUTOR: MANUEL RAFAEL JAYA DUCHE

Quito - Ecuador

2022

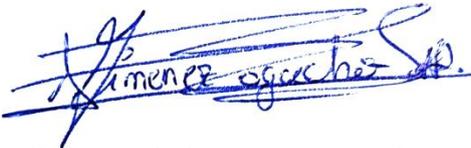
CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Oscar Anibal Jiménez Logacho con documento de identificación N° 1718856170 y Erick Bolivar Tapia Ramirez con documento de identificación N° 1725069908; manifestamos que:

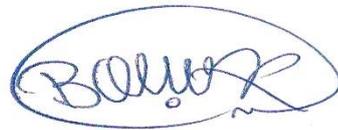
Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Quito, 15 de marzo del año 2022

Atentamente,



Oscar Anibal Jiménez Logacho
1718856170



Erick Bolivar Tapia Ramirez
1725069908

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE
TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Nosotros, Oscar Anibal Jiménez Logacho con documento de identificación N° 1718856170 y Erick Bolivar Tapia Ramirez con documento de identificación N° 1725069908, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Proyecto Técnico: “Desarrollo de una aplicación web que gestione y administre cobros y usuarios para los entrenadores de la Academia de Ecuavoley en Quito”, el cual ha sido desarrollado para optar por el título de: Ingenieros en Ciencias de la Computación, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

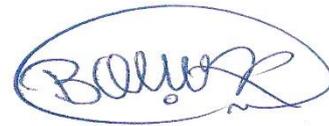
Quito, 15 de marzo del año 2022

Atentamente,



Oscar Anibal Jiménez Logacho

1718856170



Erick Bolivar Tapia Ramirez

1725069908

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Manuel Rafael Jaya Duche con documento de identificación N° 1710631035, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DESARROLLO DE UNA APLICACIÓN WEB QUE GESTIONE Y ADMINISTRE COBROS Y USUARIOS PARA LOS ENTRENADORES DE LA ACADEMIA DE ECUAVOLEY EN QUITO, realizado por Oscar Anibal Jiménez Logacho con documento de identificación N° 1718856170 y por Erick Bolivar Tapia Ramirez con documento de identificación N° 1725069908, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 15 de marzo del año 2022

Atentamente,



Ing. Manuel Rafael Jaya Duche, MSc

1710631035

DEDICATORIA

De Oscar Jiménez

Esta tesis está dedicada a: mi madre Patricia y a Edwin, por ser el apoyo incondicional en esta etapa de mi vida. Mis abuelitos, Josefina y Anibal que han estado conmigo desde siempre en todos los momentos, buenos o malos. Mi hermana Miley, que ha llegado para cambiar mi vida y ser el motor de mi inspiración. Por último, mis tíos Anibal y Javier, mi tía Lili y mis primas Karen, Samanta, Eva y Ainara, que a pesar de la distancia, son parte de mi día a día. A todos ustedes, siempre, los llevaré en el corazón.

De Erick Tapia

Dedico esta tesis: A mi madre Ismenia que desde el cielo jamás me ha abandonado. A mi padre Bolívar por todo su esfuerzo para culminar este anhelo. Mis hermanos Fabián, Jerry, Karla, Alan, Navelyn, Stefano y André por ser mi inspiración y mantenerme cuerdo. A mis padres de vida Patricia y Carlos por haberme cuidado y educado. A Iván, Georgina, Narciza y Brenda por haber sido un apoyo incondicional. Mis amigas Jessica y Karla por brindarme su amistad. Finalmente, a Karen por hacer y ser todo.

AGRADECIMIENTO

Nuestro más sincero agradecimiento a la Universidad Politécnica Salesiana, por abrirnos sus puertas, brindándonos la oportunidad de seguir adelante en nuestros estudios. Así mismo a todos los docentes que formaron parte de nuestro desarrollo, tanto académico como personal, inculcándonos e impartiéndonos su conocimiento de la mejor manera posible.

De igual forma a nuestros familiares y grupo de amigos, los cuales nos apoyaron e incentivaron en todo momento a lo largo de este trayecto.

Finalmente, agradecemos de manera especial a nuestro tutor, el ingeniero Rafael Jaya. por su ayuda y tiempo para la culminación del presente trabajo de titulación.

ÍNDICE

RESUMEN.....	1
ABSTRACT	2
INTRODUCCIÓN	3
ANTECEDENTES.....	3
PROBLEMA	4
JUSTIFICACIÓN.....	5
OBJETIVOS GENERALES Y ESPECÍFICOS.....	6
OBJETIVO GENERAL	6
OBJETIVOS ESPECÍFICOS	6
CAPÍTULO 1	7
1. REVISIÓN DE LA LITERATURA O FUNDAMENTOS TEÓRICOS.....	7
<i>1.1 Arquitectura del aplicativo web.</i>	7
<i>1.1.1 Arquitectura de dos capas</i>	7
<i>1.1.2 Arquitectura de tres capas</i>	7
<i>1.2 Back-end</i>	9
<i>1.3 Front-end</i>	9
<i>1.4 Bases de Datos Mongo DB</i>	9
<i>1.5 JavaScript</i>	10
<i>1.6 NodeJs</i>	10
<i>1.7 Nodemailer</i>	10

1.8 jQuery.....	10
1.9 Data Tables	11
1.10 Ajax.....	11
1.11 Aplicación Web	11
1.12 Sistema Gestor de cobros.....	12
CAPÍTULO 2	13
2. MARCO METODOLÓGICO	13
2.1 Inicio del diseño de la aplicación	13
2.2 Planificación y Estimación.....	19
2.3 Implementación	46
2.3.1 Levantamiento del servidor.....	46
2.3.2 Conexión a la base	46
2.3.3 Registro de Usuarios.....	46
2.3.4 Encriptación	47
2.3.5 Inicio de sesión.....	48
2.3.6 Sesiones.....	49
2.3.7 Función buscar usuario.....	50
2.3.8 Función editar usuario.....	50
2.3.9 Función eliminar usuario.....	50
2.3.10 Función editar cobro.....	50
2.3.11 Registrar cobro.....	50
2.3.12 Recuperar contraseña.....	51

2.3.13 Recordatorio.....	51
2.3.14 Inventario.	52
2.3.15 Cerrar sesión.....	53
2.3.16 Mensajes de acciones.	53
2.4 Despliegue.....	53
2.5 Pruebas.....	53
2.5.1 Pruebas de Caja Negra.	53
2.5.2 Pruebas de carga.	63
2.5.3 Pruebas de Stress	68
2.5.4 Pruebas de diseño e interacción	69
CAPÍTULO 3	80
3. RESULTADOS	80
3.1 Análisis de pruebas caja negra	80
3.2 Análisis pruebas de carga	80
3.3 Análisis pruebas de stress	81
3.4 Análisis pruebas de diseño e implementación.....	82
CRONOGRAMA	82
CONCLUSIONES	86
RECOMENDACIONES	87
REFERENCIAS BIBLIOGRÁFICAS	88

ÍNDICE DE TABLAS

Tabla 1.....	14
Tabla 2.....	14
Tabla 3.....	14
Tabla 4.....	35
Tabla 5.....	35
Tabla 6.....	36
Tabla 7.....	37
Tabla 8.....	39
Tabla 9.....	40
Tabla 10.....	41
Tabla 11.....	42
Tabla 12.....	44
Tabla 13.....	44
Tabla 14.....	54
Tabla 15.....	56
Tabla 16.....	57
Tabla 17.....	58
Tabla 18.....	59
Tabla 19.....	60
Tabla 20.....	61
Tabla 21.....	62
Tabla 22.....	82

ÍNDICE DE FIGURAS

Figura 1	13
Figura 2	14
Figura 3	15
Figura 4	16
Figura 5	16
Figura 6	18
Figura 7	19
Figura 8	20
Figura 9	21
Figura 10	22
Figura 11	22
Figura 12	23
Figura 13	24
Figura 14	25
Figura 15	26
Figura 16	27
Figura 17	28
Figura 18	29
Figura 19	30
Figura 20	30
Figura 21	31
Figura 22	31
Figura 23	32
Figura 24	44

Figura 25	47
Figura 26	62
Figura 27	62
Figura 28	63
Figura 29	63
Figura 30	64
Figura 31	64
Figura 32	65
Figura 33	66
Figura 34	67
Figura 35	68
Figura 36	68
Figura 37	69
Figura 38	70
Figura 39	71
Figura 40	72
Figura 41	72
Figura 42	73
Figura 43	73
Figura 44	74
Figura 45	75
Figura 46	75
Figura 47	76
Figura 48	77

RESUMEN

El presente proyecto técnico tiene como objetivo desarrollar una aplicación web, mediante el uso de herramientas de programación, que gestione y administre cobros a los usuarios de la academia de ecuavoley en Quito. Para lo cual se hizo uso del lenguaje de programación NodeJs y el motor de base de datos Mongo DB. En la parte del front-end se utilizó la librería Express con su plantilla handlebars donde se introdujo el paquete de diseño Bootstrap. Para la gestión de datos se implementó Data Tables con AJAX, todo esto bajo la metodología ágil Scrum. El producto final es una aplicación web que permite registrar cobros de los estudiantes, además cuenta con un inventario, en el cual los entrenadores podrán gestionar los datos. Cuenta con mensajería para notificar a los estudiantes los pagos realizados o atrasados mediante correo electrónico o WhatsApp. Para terminar, se realizaron pruebas de caja negra los cuales verificaron el cumplimiento de los requerimientos, así mismo se realizaron pruebas de carga simulando una ejecución con 200 peticiones al servidor en un 1 segundo, obteniendo un 100% de éxito. También se realizó pruebas de stress donde se simuló el tráfico mínimo de 500 usuarios, arrojando un 0% de errores, y un tráfico máximo de 1000 usuarios mostrando inconsistencias en ciertas cargas. Al finalizar, se concluyó con una aplicación web óptima, que aporta innovación en la academia, generando aceptación y confianza en otras empresas motivando en la adquisición de softwares que faciliten el trabajo en múltiples actividades aportando mejores resultados.

Palabras clave aplicación web, gestión de cobros, pymes, Node Js

ABSTRACT

The objective of this technical project is to develop a web application, through the use of programming tools, that manages and administers charges to users of the ecuavoley academy in Quito. For which the NodeJs programming language and the Mongo DB database engine were used. In the front-end part, the Express library was used with its handlebars template where the Bootstrap design package was introduced. For data management, Data Tables with AJAX were implemented, all this under the agile Scrum methodology. The final product is a web application that allows students to record payments, it also has an inventory, in which the coaches can manage the data. It has messaging to notify students of payments made or late by email or WhatsApp. To finish, black box tests were carried out which verified compliance with the requirements, likewise load tests were carried out simulating an execution with 200 requests to the server in 1 second, obtaining 100% success. Stress tests were also carried out where the minimum traffic of 500 users was simulated, showing 0% errors, and a maximum traffic of 1000 users showing inconsistencies in certain loads. At the end, it was concluded with an optimal web application, which provides innovation in the academy, generating acceptance and trust in other companies, motivating the acquisition of software that facilitates work in multiple activities, providing better results.

Keywords web application, collection management, SMEs, Node Js

INTRODUCCIÓN

ANTECEDENTES

A medida que pasa el tiempo la tecnología cada vez se posiciona de mejor manera y está generando en la sociedad, una necesidad, y no solo un simple deseo de tenerla. En su artículo sobre la evaluación constructiva de la tecnología menciona Cardoso (2021):

La ingeniería, entendida como la reunión de conocimientos científicos y tecnológicos para la innovación, creación, avance y optimización de técnicas, así como un conjunto de herramientas útiles para suplir las necesidades sociales y resolver problemas técnicos tanto de las personas como de la comunidad. (p. 3)

Las pymes es la denominación que se les da a las pequeñas y grandes empresas, las cuales surgen de gente que tiene una economía no tan elevada, pero útil para empezar a construir la idea de negocio que tiene en mente y desean emprender. En el Ecuador, las pymes son consideradas empresas de suma importancia para la sostenibilidad económica del país. “La fuente del desarrollo social en cuanto a producción, demanda y compra de productos o simplemente por valor agregado, lo que significa que se ha convertido en un factor indispensable para generar riqueza y empleo” (Yance Carvajal, 2017, p.10). Por otro lado, en la actualidad, estamos pasando por épocas difíciles debido a la aparición del nuevo virus COVID-19. Este es el motivo por el cual la interacción de objetos entre humanos es un riesgo alto de contagios, ya que genera un medio de transmisión para el virus.

La Academia de ecuavoley ubicada en la ciudad de Quito realiza entrenamientos para las personas que gustan y practican el deporte nacional llamado “ecuavoley”. Estas clases las realizan profesores expertos en el deporte, impartíéndolas diariamente. La empresa tiene alrededor de 200 deportistas inscritos, que van en horarios diferentes dispuestos por la propia academia. El modelo de negocio que maneja la empresa es mediante el cobro de una matrícula,

que acredita al cliente como alumno en la academia, para posteriormente ir cobrando una mensualidad a los estudiantes. Una vez realizado todo este proceso se le entrega al estudiante una ficha, la cual sirve para ir registrando sus pagos mes a mes.

Con toda esta información se ha podido identificar dos variables las cuales son: “Registro de cobros y gestión de datos”, en el primer, se identifica que las compañías deberían trabajar con la tecnología orientadas a mejorar la competitividad de las empresas de servicios, con énfasis en CRM (gestión de la relación con el cliente en las empresas de servicios), donde los sistemas de comunicaciones juegan un papel importante, especialmente si la comunicación con un cliente es integrada y de fácil acceso (Domazet, 2016).

Para ODK arquitectos de implementación, la pérdida de datos se refiere a que los datos recopilados se vuelven permanentes no disponible (Camille Cobb, 2017). La necesidad de una gestión de datos responsable se intensifica con el creciente impacto de los datos en la sociedad. Un lugar central del impacto social de los datos son los sistemas de decisión automatizados (ADS), sistemas sociolegales-técnicos que se utilizan ampliamente en la industria, las organizaciones sin fines de lucro y el gobierno. (Julia Stoyanovich, 2020)

La importancia del proyecto nace al querer digitalizar el proceso de registro de cobros que se hace de forma manual para cada cliente. Esto hace que el proyecto proporcione a la empresa un avance tanto en organización como innovación por medio de la tecnología.

PROBLEMA

Como se dijo anteriormente, la academia de ecuavoley registra sus cobros manualmente por medio de la ficha que se le otorga a cada estudiante cuando se matricula, es aquí donde se origina el problema de gestión de datos, ya que los afiliados deben llevar consigo la ficha donde consta el pago de su mensualidad. Este documento sirve tanto para verificar su pago, como para registrar uno nuevo. Como se puede observar esta práctica de registro de cobros es anticuada y

a su vez arcaica siendo también, en estos tiempos, un posible transmisor del virus COVID-19 para las personas que interactúan con dicho objeto.

JUSTIFICACIÓN

Según Onofa (2017), “Ecuador solo el 30% de las pymes utiliza las ventajas tecnológicas de información y comunicación”. (p. 154). Es por ello que la finalidad del trabajo es tanto de innovación como de motivación hacia otras empresas que no necesariamente tienen una cartera amplia de clientes, pero si considerada, para que puedan manejar softwares de gestión de datos, útiles y prácticos. De igual forma se busca contribuir con el almacenamiento digital de datos, ya que es bien conocido en la actualidad que, con las nuevas tecnologías, minería de datos y los recientes desarrollos de Big Data, han motivado la aparición de tecnologías novedosas para el procesamiento distribuido de datos a gran escala en grupos de computadoras, lo que ha dado lugar a sistemas de gestión de datos espaciales distribuidos (García, 2020). Gracias a esto se pueden realizar mejores análisis para la toma de decisiones y aprovechar al máximo la información generada a lo largo del tiempo.

Con este proyecto se desea aportar un sistema que pueda ser capaz de administrar los cobros, mediante el uso de una aplicación web, sin la necesidad de una ficha física. Con esto, tanto los entrenadores de la academia como los clientes tendrán los registros de pagos en un sitio web, y podrán acceder a estos cuando lo requieran.

Las causas de no tener un sistema que permita el registro de cobros son las siguientes: i) pérdida de datos, ii) carecimiento de registros históricos, iii) desactualización tecnológica. Por ello el propósito es crear una aplicación web que permita realizar este proceso y así tener datos almacenados para garantizar la disponibilidad.

Lo que se realizará para solventar este problema es diseñar una aplicación web que permita a los entrenadores de la academia de ecuavoley registrar usuarios, y sus respectivos cobros, para posteriormente visualizar esta información en un navegador web, ya sea Mozilla o Chrome, cuando lo necesiten. De igual manera se podrá editar los datos personales y pagos de los clientes, o a su vez eliminarlos si es el caso. Esta aplicación será desarrollada con el uso de NodeJs, Mongo DB, HTML y Bootstrap, los cuales ayudarán a la creación tanto de la interfaz de usuario, como a la administración de la información en la base de datos respectiva. Para que la visualización de los datos este organizada se utilizará Data Tables, lo que permitirá una mejor interacción entre el usuario y los datos.

OBJETIVOS GENERALES Y ESPECÍFICOS

OBJETIVO GENERAL

Desarrollar una aplicación web, mediante el uso de herramientas de programación, que gestione y administre cobros a los usuarios de la academia de ecuavoley en Quito.

OBJETIVOS ESPECÍFICOS

- Diseñar una página web que permita registrar el cobro de cada usuario.
- Desarrollar funciones que permita realizar un inventario de usuarios.
- Implementar métodos que ayuden a guardar, visualizar, actualizar y eliminar la información de los clientes.
- Verificar validaciones para el correcto funcionamiento de la aplicación web.

CAPÍTULO 1

1. REVISIÓN DE LA LITERATURA O FUNDAMENTOS TEÓRICOS

En el trabajo “Desarrollo e implementación de un aplicativo web, utilizando la metodología Scrum, para mejorar el proceso de atención al cliente en la empresa Z Aditivos S.A.” realizado por Diaz Jimmy y Romero Mitcheli en el año 2017, una aplicación web es una aplicación informática que se utiliza mediante el uso de un servidor web a través de Internet para ser desplegada en un navegador.

Las aplicaciones web ayudan a la interacción de datos los cuales pasan a ser accesibles a través de una conexión a internet, sin la necesidad de descargar y proceder a instalar un software. Algunos ejemplos son los web mails, web log o tiendas en línea.

1.1 Arquitectura del aplicativo web.

1.1.1 Arquitectura de dos capas. Es la arquitectura más conocida, la cual tiene cliente/servidor. Es necesario una interfaz de usuario que se instala y ejecuta en una PC enviando peticiones a un servidor el cual ejecuta las tareas requeridas.

1.1.2 Arquitectura de tres capas. Esta arquitectura es actual ya que presenta una capa intermedia para el proceso. Estas capas son:

- El primer nivel (Navegador Web), consiste en la capa de presentación que contiene además del navegador, el servidor web que es el responsable de imprimir los datos de forma correcta.
- El segundo nivel (Servidor de Aplicaciones), está concentrado básicamente a algún tipo de script o programa.
- El tercer nivel (Servidor de Datos), entrega al anterior nivel todos los datos que este requiere para su adecuada ejecución. Jimmy Jhonon Diaz Ortiz (2017)

Mark Raasveldt (2020) en su investigación “Data Management for Data Science” El auge de la ciencia de datos ha provocado una afluencia de nuevos usuarios en necesidad de

soluciones de gestión de datos. Sin embargo, en cambio de utilizar las soluciones RDBMS existentes por las que están optando utilizar una pila de soluciones independientes para el almacenamiento de datos y procesamiento pegado por lenguajes de secuencias de comandos. Esto no es porque no necesitan la funcionalidad que un integrado RDBMS proporciona, sino más bien porque las implementaciones de RDBMS existentes no se adaptan a su caso de uso. Para resolver estos problemas, se propuso una nueva clase de sistemas de gestión de datos: sistemas analíticos integrados. Estos sistemas están estrechamente integrados con herramientas analíticas y proporcionan un acceso rápido y eficiente a los datos almacenados en ellas. En este trabajo, describimos los desafíos y oportunidades únicos w.r.t cargas de trabajo, resiliencia y cooperación a las que se enfrenta esta nueva clase de sistemas y los pasos que se ha dado hacia abordarlos en el sistema DuckDB.

Muhammad Yahya (2020), en su trabajo realizado sobre un sistema de información para estudiantes de SMP Negeri 9 Tapung, utiliza lenguajes de programación basados en texto como HTML, JavaScript, CSS y PHP. Este sistema gestiona los datos de los estudiantes relacionados con el carácter de actitud. Evaluación para evaluar el comportamiento de los estudiantes mientras están en la escuela para facilitar la realización de la rutina del estudiante. asuntos y mejorar el carácter en el proceso de enseñanza y aprendizaje. Las evaluaciones de carácter que son procesados son honestidad, disciplina, responsabilidad, tolerancia, cooperación, autoconfianza y tolerancia. Un sistema computarizado simplificará las tareas del maestro en la ejecución de la aplicación disciplinaria. acciones y aumentar el carácter de los estudiantes en la escuela. Este sistema de información es la etapa inicial en Manejo de datos sobre las actitudes y valores de carácter de los estudiantes. Por esta razón, el desarrollo, mejora, y adición de características y sugerencias y críticas de todas las partes que esperan crear un sistema de información mejor y estructurado para la mejora de la calidad y el carácter de estudiantes que son buenos en la escuela

Según Moreno Javier (2016) en su artículo sobre “Una comparación de rendimiento entre Oracle y MongoDB”, el objetivo fue comparar una base relacional y otra NoSql. La metodología fue cualitativa por el hecho realizar esta investigación por medio de almacenamiento de grandes cantidades de datos. Dictaminaron que la Inserción, Actualización y Borrado de datos es mucho más rápida cuando se trabaja con Mongo DB, dejando en claro que este motor base de datos, es más favorable por su velocidad. De igual forma se concluye que hoy en día poseer información almacenada en una base de datos es de sumo valor, ya que con esto se puede realizar diferentes tipos de análisis, útiles para la toma de decisiones de una empresa. La razón es, por que los datos contienen información de los diversos comportamientos que tiene los usuarios y esto permite comprender el porqué de las acciones realizadas.

Algunos conceptos básicos que se deben saber previo a la realización del proyecto y ofrecer un mejor entendimiento tanto en la programación como en el diseño de la aplicación web serán explicados a continuación.

1.2 Back-end

La estructura de back-end es la que permite almacenar, buscar, alterar y eliminar objetos en la base de datos como servicio (Kohl,/2014).

1.3 Front-end

Es la parte que el usuario puede ver e interactuar en un software. Para diseñar dicha interfaz web, se utilizan ciertas herramientas, que generalmente son una combinación de HTML, CSS y JavaScript, todos controlados por el navegador. (Hanin M. Abdullah, 2014).

1.4 Bases de Datos Mongo DB

Mongo DB es una base de datos NoSql, es decir que no existen tablas ni relaciones en su mecanismo. Utiliza el concepto de colecciones para almacenar datos con su respectivo ID.

Esto ha hecho que esta arquitectura sea eficaz, fácil de ser implementada y con una escalabilidad virtuosa. (Mongo DB, 2015).

1.5 JavaScript

Creado por Netscape, JavaScript es muy reconocido y a su vez utilizado como eje de programación en HTML. Este lenguaje es útil para el manejo de funciones en una página web, generando dinamismo e interactividad en la misma. (Navarrete, 2006).

1.6 NodeJs

Un entorno de ejecución de eventos asíncronos es lo que define a NodeJs. Para trabajar con él se hace su uso del lenguaje de programación JavaScript. La particularidad es que los procesos son llamados únicamente cuando se los necesita (Edward Haro, 2018).

1.7 Nodemailer

Es una librería de Node.js que se utiliza para enviar correos electrónicos de forma sencilla. La idea empezó en el año 2010 cuando no exista una herramienta fácil de usar, actualmente se ha convertido en una opción inmediata para la mayoría de los usuarios de Node.js. (Nodemailer, 2021).

1.8 jQuery

Es una biblioteca de JavaScript eficaz, no tan grande y aglomera múltiples funcionalidades. Hace actividades como el recorrido y la interacción de documentos HTML, la implementación de eventos, la ilustración y Ajax sean mucho más sencillos con el uso de una API fácil de usar. (jQuery, 2021).

1.9 Data Tables

Este plugin que ofrece la librería de jQuery para JavaScript es una herramienta útil y rápida al momento de implementarla, además que sus funcionalidades hacen que el consumo de datos sea estructurado y fácil de manipular. (SpryMedia, 2021).

1.10 Ajax

La biblioteca jQuery contiene una amplia variedad de capacidades Ajax. Los métodos que trae consigo nos permiten cargar datos a partir del servidor sin la necesidad de recargar la página del navegador. (jQuery, 2021).

1.11 Aplicación Web

El diseño de sitios web se ha convertido, hoy en día, en un eje fundamental para el marketing digital de las empresas. Sin embargo, los sitios web ya no funcionan solo como portadas o estanterías digitales, si no que han dado un paso más convirtiéndose en páginas completamente funcionales par el provecho del público. A este tipo de entornos se los conocen como aplicaciones web.

En la práctica desarrollada por Edward Haro (2018) con de desarrollo de aplicaciones web con servicios Restful, mediante el uso y comparación de los frameworks Node.js y Spring Boot, se realizaron pruebas de robustez, integración con tecnologías, seguridad, soporte, mantenimiento, estabilidad y ejecución, donde se observó que ambas tecnologías son fáciles al momento de ser implementada, y que además son herramientas útiles para la creación de aplicaciones empresariales por su buena capacidad para consumir datos.

1.12 Sistema Gestor de cobros

La investigación enfocada al “Análisis y mejora de procesos en la gestión de cobranzas para una concesionaria en la ciudad de Guayaquil” tiene como objetivo acudir a los saberes del análisis de procesos y tecnología para aprovechar los métodos que presenta y así dar una mejora en las actividades en la fábrica, permitiendo facilitar el trabajo en ciertas actividades mediante la automatización que pueden solventar los sistemas informáticos.

El estudio de los criterios de cobranzas realizadas al personal de gestión de cobros, permitieron identificar causas en la problemática planteada y a su vez describir el proceso que se desea mejorar.

El motivo fue disminuir el tiempo en el que se validaban de los cobros que se hacían a los clientes en el aplicativo. El diseño del modelo logró definir las características básicas de la aplicación (Villegas, 2018).

CAPÍTULO 2

2. MARCO METODOLÓGICO

Scrum es un marco ligero que ayuda a personas, equipos y organizaciones a generar alternativas y soluciones en sus proyectos, capaces de ser adaptables para solventar problemas de alta dificultad. Esta metodología emplea un enfoque iterativo e incremental para optimizar pre-visualización disminuyendo riesgos futuros. Scrum intenta que todas las personas se encuentren involucradas, ya que colectivamente, todas tienen habilidades y experiencia suficiente para hacer su trabajo y poder compartir sus ideas según sea necesario.

Scrum es una metodología sencilla para una colaboración en equipo capaz de sobrellevar proyectos de software complicados (Ken Schwaber, 2020). El póster de Scrum Framework ofrece una vista global gráfica de cómo se debe trabajar con Scrum a nivel de organización dentro de un equipo de trabajo. El Marco se basa en The Scrum Guide que fue redactado por los co-creadores de Scrum Ken Schwaber y Jeff Sutherland.

Scrum aparece en 1993 con Jeff Sutherland, el cual propone un modelo de desarrollo de Software en Ease/Corporation. Esta metodología es adecuada para el desarrollo de productos que se realiza en entornos de incertidumbre, auto organización, control moderado y transmisión del conocimiento. Es este el motivo por el cual se eligió la metodología Scrum, ya que se iniciará un proyecto de cero con un usuario con requerimientos previos para el desarrollo del sistema web.

2.1 Inicio del diseño de la aplicación

Para este proyecto se ha decidido prescindir de un Scrum Master, aunque la metodología lo propone, se decidió esto ya que, al ser un equipo de solo dos personas no será necesario un coordinador como tal. Lo que si se asignó fueron patrocinadores los cuales fueron el tutor encargado del proyecto técnico, y el jefe de la academia de ecuavoley.

Tabla 1*Patrocinador de la aplicación*

Nombres	Cargo	Función
Jaya Rafael	Patrocinador	Encargado en hacer cumplir los requerimientos y objetivos planteados

Nota. Patrocinador tutor por parte de la Carrera. Elaborado por: Jiménez Oscar y Tapia Erick

Tabla 2*Patrocinador de la empresa*

Nombres	Cargo	Función
Jaya Rafael	Patrocinador	Encargado en hacer cumplir los requerimientos y objetivos planteados

Nota. Patrocinador de la empresa que requiere la aplicación web Elaborado por: Jiménez Oscar y Tapia Erick

Por otro lado, existen cargos, en este caso ambas personas participaron en el desarrollo de la aplicación, así como en el modelo y diseño del mismo.

Tabla 3*Cargos*

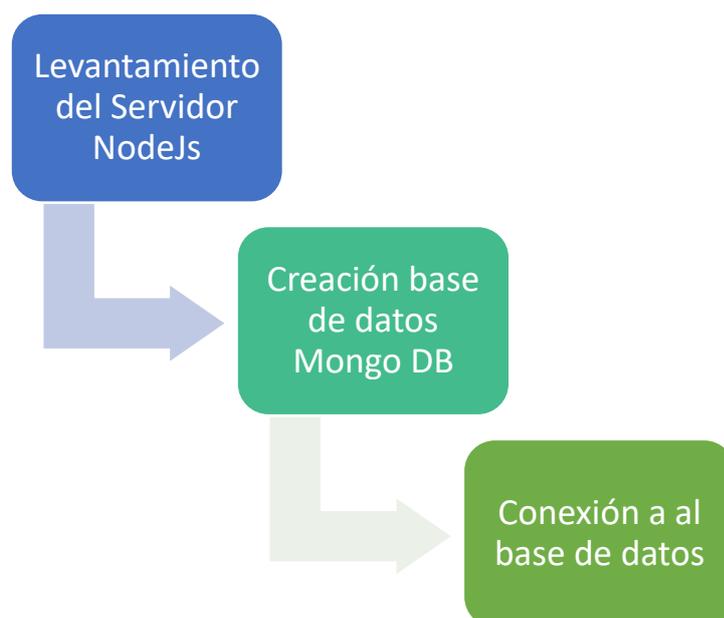
Nombres	Cargo	Función
Jiménez Oscar	Estudiante	Implementación del Software
Tapia Erick	Estudiante	Implementación del Software

Nota. Personas asignadas para realizar el proyecto Elaborado por: Jiménez Oscar y Tapia Erick

En cuanto a la organización y sprints del proyecto se decidió diseñar 5 bloques, los cuales estaban divididos en conjuntos iguales, lo que facilitó el reparto del proyecto sin ningún problema y generó la opción de trabajar en cualquier orden de sprints, siempre y cuando estos sean del mismo conjunto.

Figura 1

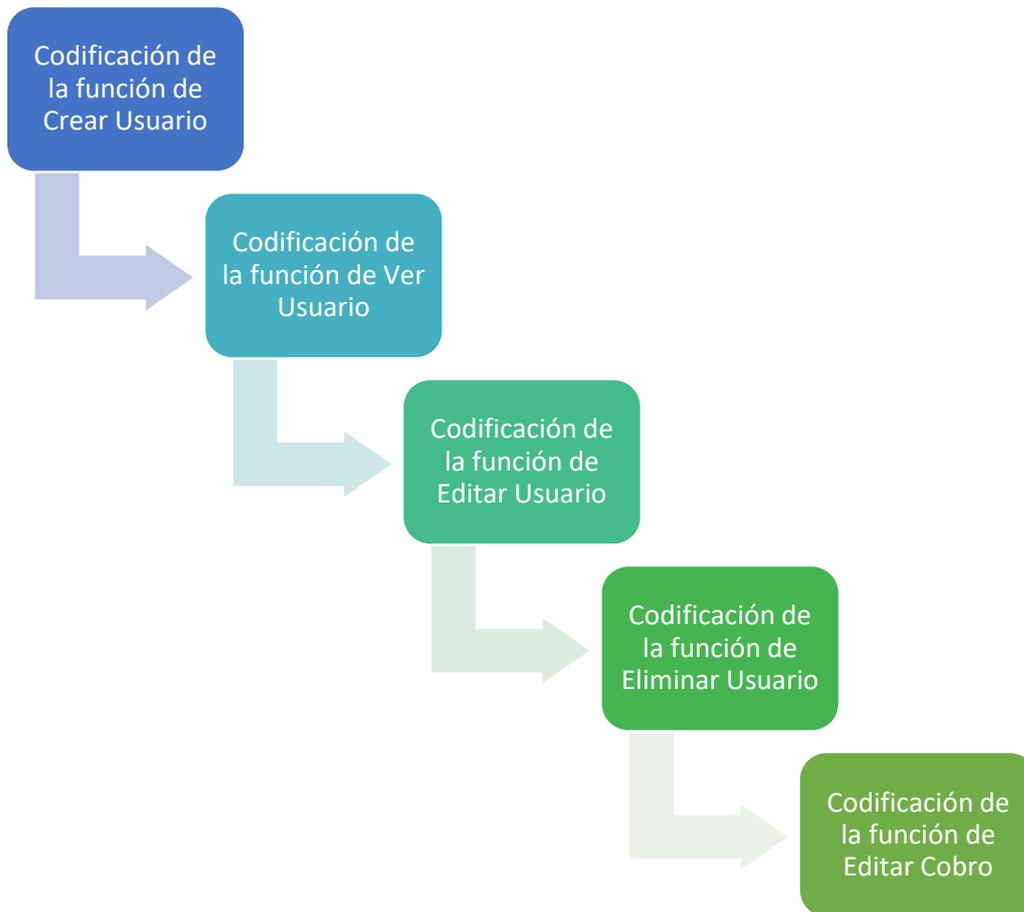
Primer bloque de sprints



Nota. Sprints para el inicio de la aplicación. Elaborado por: Jiménez Oscar y Tapia Erick

Figura 2

Segundo bloque de sprints

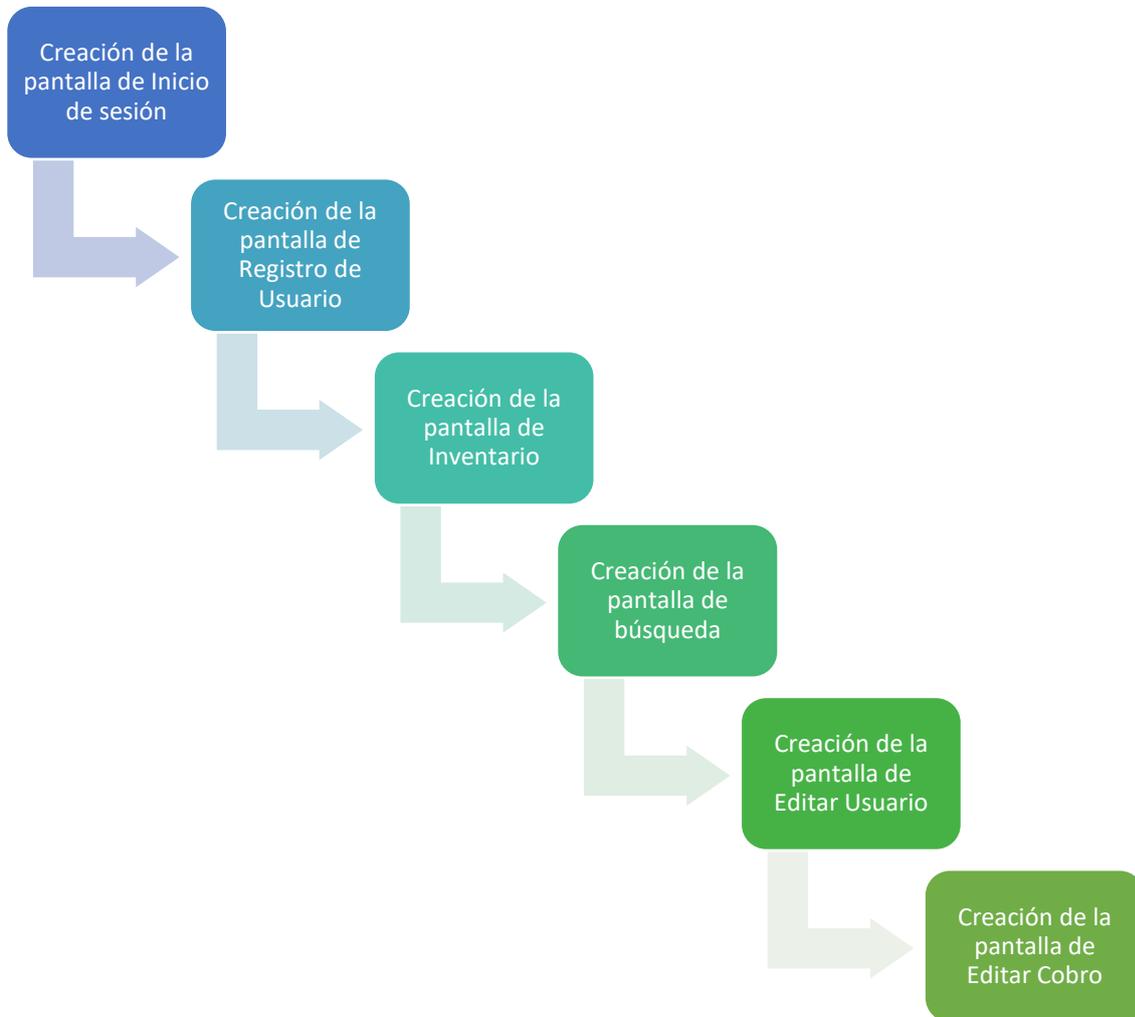


Nota. Sprints para la construcción de las funcionalidades, back-end. Elaborado por: Jiménez

Oscar y Tapia Erick

Figura 3

Tercer bloque de sprints

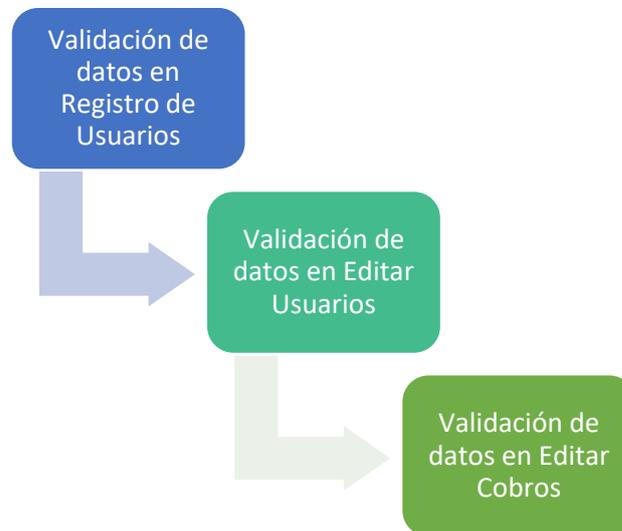


Nota. Sprints para la construcción del diseño de la aplicación web, front-end. Elaborado por:

Jiménez Oscar y Tapia Erick

Figura 4

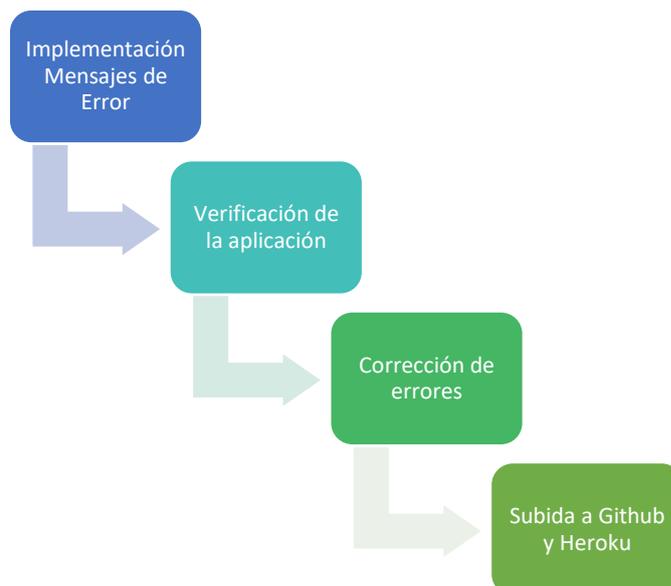
Cuarto bloque de sprints



Nota. Sprints para validaciones de la aplicación web. Elaborado por: Jiménez Oscar y Tapia Erick

Figura 5

Quinto bloque de sprints



Nota. Sprints pruebas y lanzamiento de la aplicación web. Elaborado por: Jiménez Oscar y Tapia Erick

2.2 Planificación y Estimación

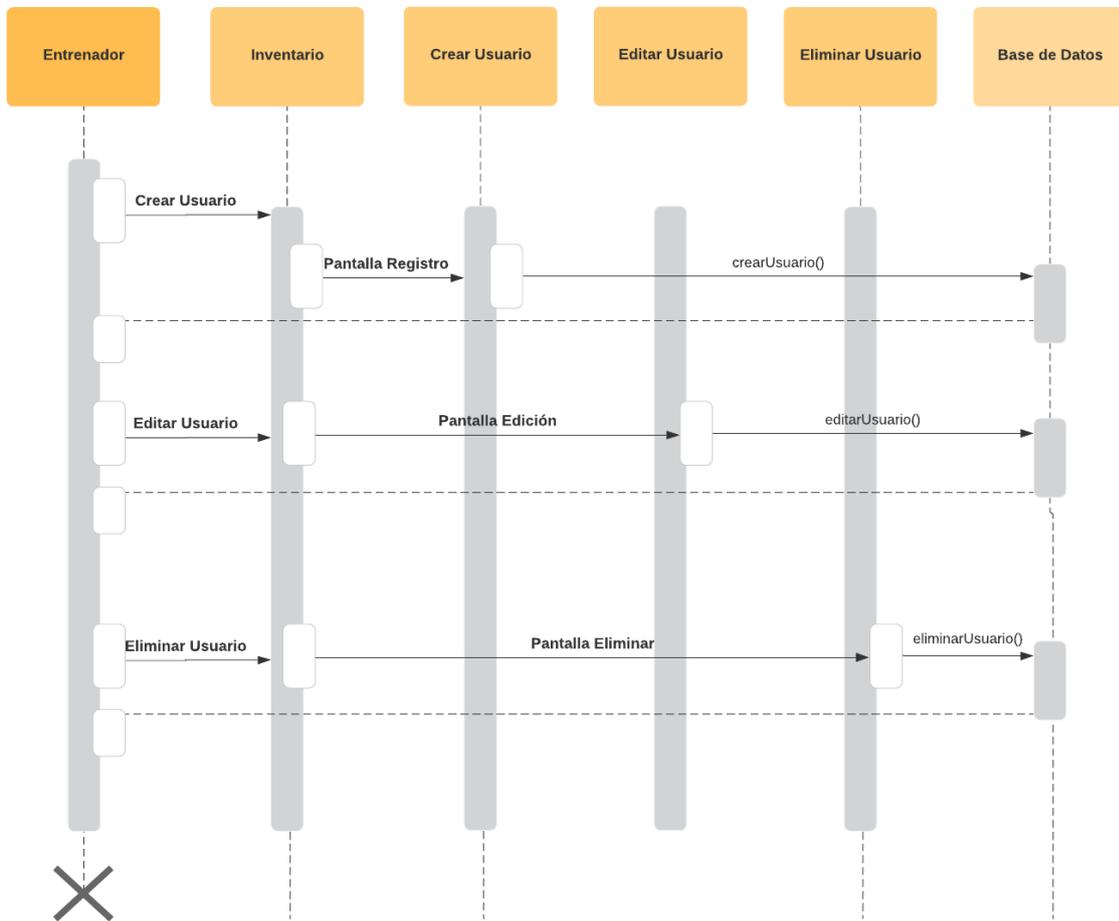
En la etapa de planificación se realizó el diseño funcional que tiene la aplicación web, haciendo uso de tres diagramas UML acorde con las necesidades del proyecto, los cuales fueron: diagrama de flujo, diagrama de actividades y diagrama de objetos.

Se dispuso, para esta aplicación web, realizar funciones realmente útiles y necesarias para el registro de cobros de cada usuario, evidentemente cumpliendo con los requerimientos del cliente, es por ello que la implementación del CRUD toma un papel fundamental en el programa.

En primera instancia están las funciones de crear, editar y eliminar usuarios Figura 6, las cuales son accesibles únicamente para los entrenadores, por lo que el flujo comienza en la acción de ellos. Si el entrenador se encuentra en la pantalla “inventario” y da clic en el botón “Crear Usuario” este le redirigirá a la pantalla respectiva, en este caso “Crear Usuario”, donde tendrá que llenar todos los campos del formulario. Al terminar, el entrenador deberá dar clic en “Guardar” dando paso al siguiente flujo el cual es, él envió de esta información a la base de datos, finalizando así el flujo. Por otra parte, si el entrenador da clic en “Editar Usuario” la aplicación realizará un proceso similar al de “Crear Usuario” ya que le redirigirá a la pantalla “Editar Usuario” donde tendrá un formulario con los campos respectivos para poder editar el dato que desee. Una vez realizado el cambio de datos, el entrenador deberá dar clic en el botón guardar enviando la información a la base de datos, concluyendo con éxito este flujo. Y por último está el botón eliminar, por lo que, si el entrenador acciona este botón, lo que hará el programa es obtener esta información enviándola nuevamente a la base de datos para buscarla y poder eliminarla finalizando con este proceso.

Figura 6

Diagrama de flujo 1



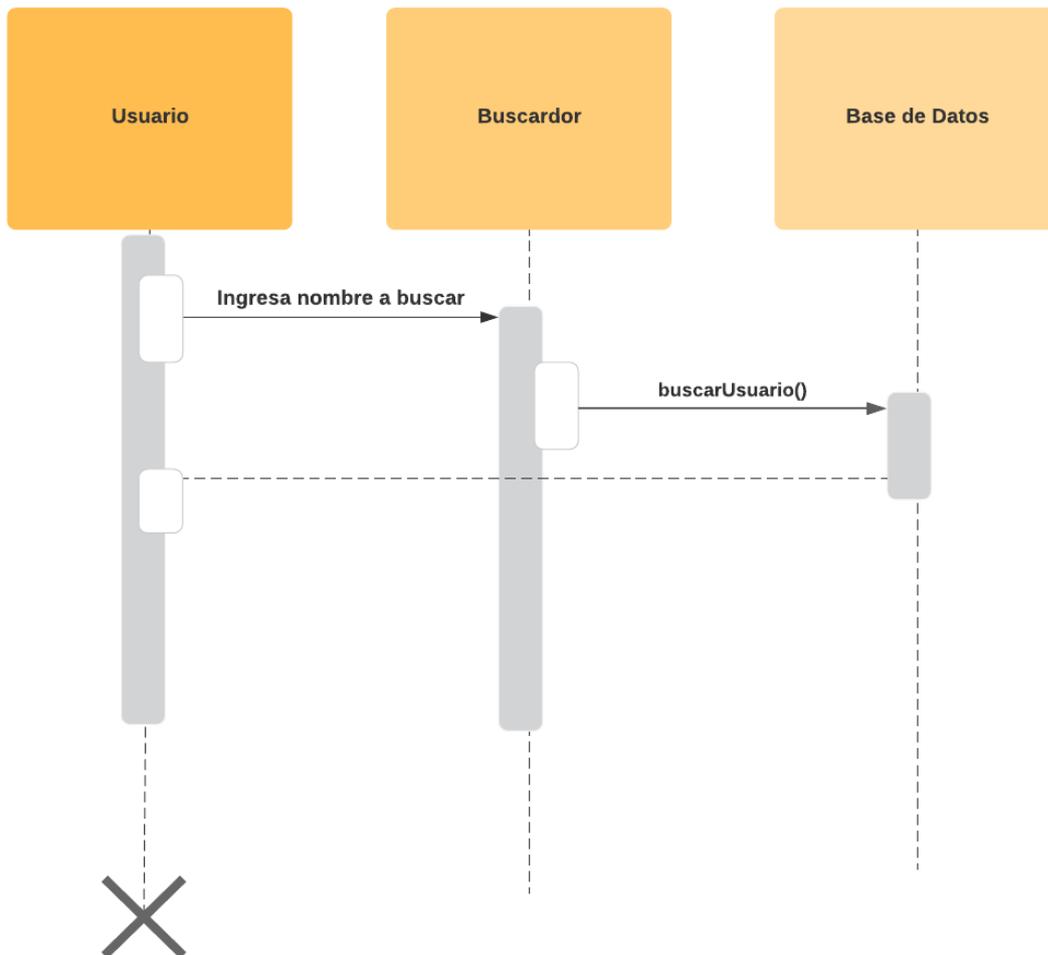
Nota. Diagrama de flujo de las funciones Crear Usuario, Editar Usuario y Eliminar Usuario.

Elaborado por: Jiménez Oscar y Tapia Erick

La Figura 7, muestra el flujo que tiene el proceso de “Buscar Usuario”, este proceso lo puede realizar cualquier tipo de usuario, sea entrenador o cliente. Lo que se debe hacer es, escribir el nombre del usuario, dando paso para que el programa obtenga el dato introducido por el cliente, y así la base de datos lo busque. Al encontrarlo devolver el dato a la pantalla de búsqueda mostrándole al usuario el resultado.

Figura 7

Diagrama de flujo 2

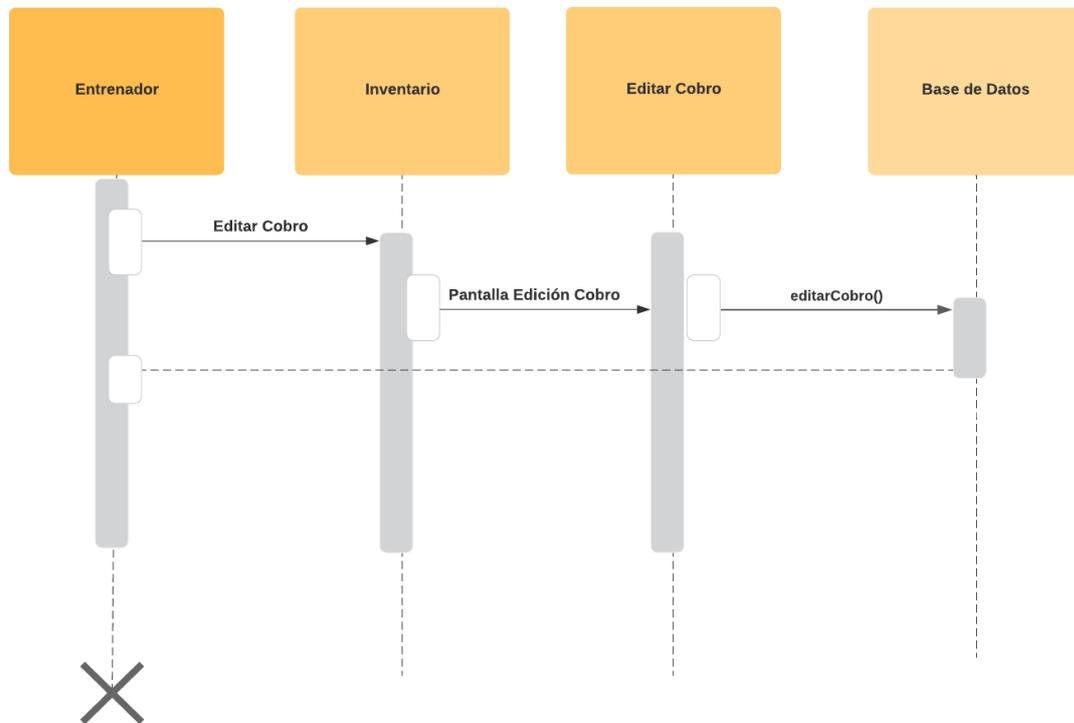


Nota. Diagrama de flujo de la función Buscar Usuario. Elaborado por: Jiménez Oscar y Tapia Erick

Al ser un programa que permite el registro de cobros es necesario la edición del mismo. Por ello existe la función de “Editar Cobro” Figura 8, el cual podrá utilizar solo el entrenador. Para ello lo que debe hacer es estar en la página de inventario y dar clic al botón “Editar Cobro”, lo que le llevará a la página editar cobro, donde podrá cambiar este valor y guardarlo, al dar clic en el botón “Guardar”. Esto hará que el programa obtenga el nuevo dato, enviándolo a la base de datos para que realice el respectivo cambio.

Figura 8

Diagrama de flujo 3



Nota. Diagrama de flujo de la función Editar Cobro. Elaborado por: Jiménez Oscar y Tapia Erick

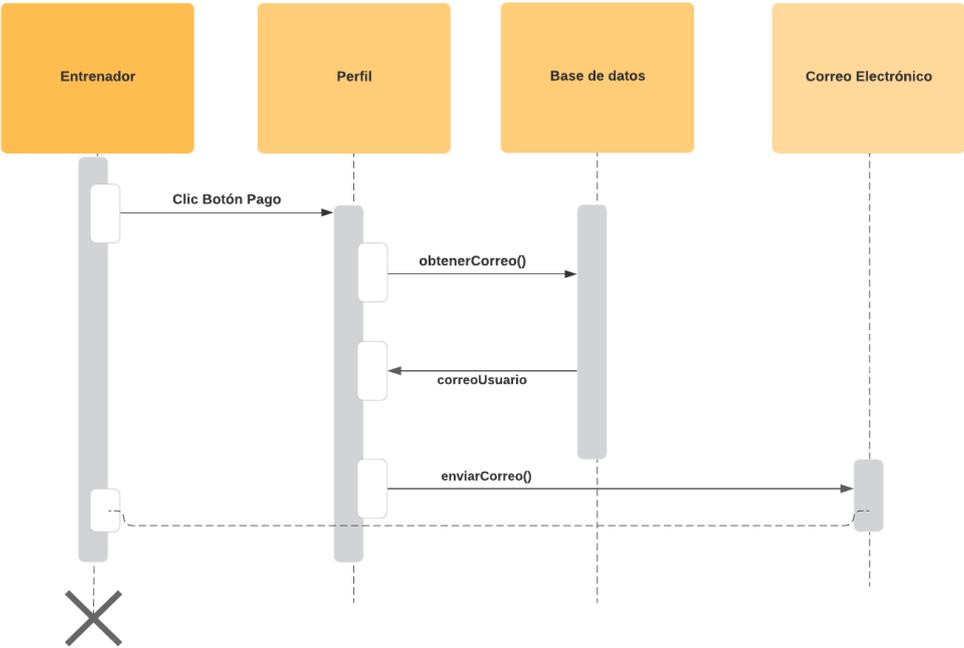
La aplicación cuenta con una funcionalidad de envío de mensajes vía WhatsApp Figura 11, y correo electrónico. Esta mensajería se realiza en varias ocasiones, una de ellas es cuando el entrenador registra un pago al alumno Figura 9, así como cuando necesita informarle que su suscripción está a punto de caducar Figura 10. Lo mismo hace al momento que el alumno o el entrenador olvida su contraseña Figura 12, básicamente envía un mensaje adjuntando un link al correo electrónico del usuario y al dar clic al link se abrirá la página de “crear nueva contraseña”. Es importante aclarar que no se necesita ingresar ningún dato previo al envío, ya que los mensajes fueron prescritos, por lo que el número de celular como el correo electrónico se obtienen de la base de datos automáticamente. La única vez que es

necesario un dato es cuando se quiere recurrir la contraseña, ya que se le pide al usuario que ingrese su correo, el cual será verificado en la base de datos y enviado al mismo.

Siguiendo con el proceso, para que todo funcione bien, el entrenador lo único que debe hacer es dar clic en los botones respectivos y todo el proceso se hará automáticamente.

Figura 9

Diagrama de flujo 4

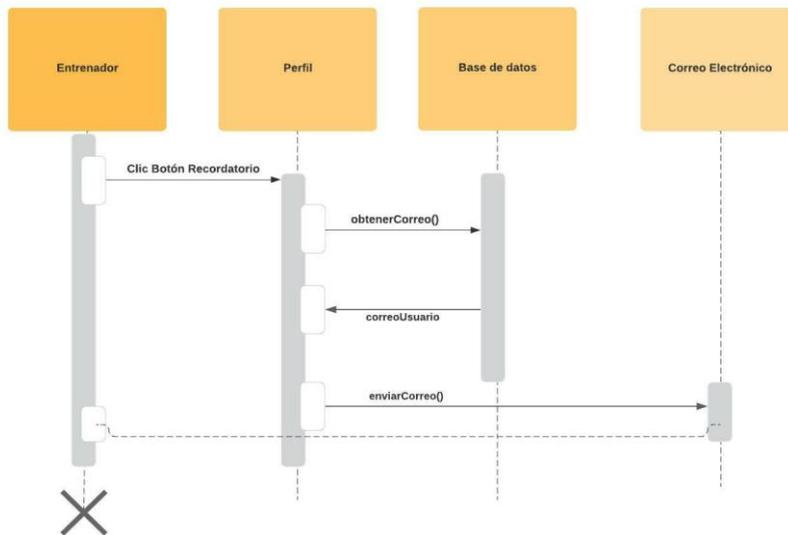


Nota. Diagrama de flujo de la función enviar corroboración de pago al correo electrónico.

Elaborado por: Jiménez Oscar y Tapia Erick

Figura 10

Diagrama de flujo 5

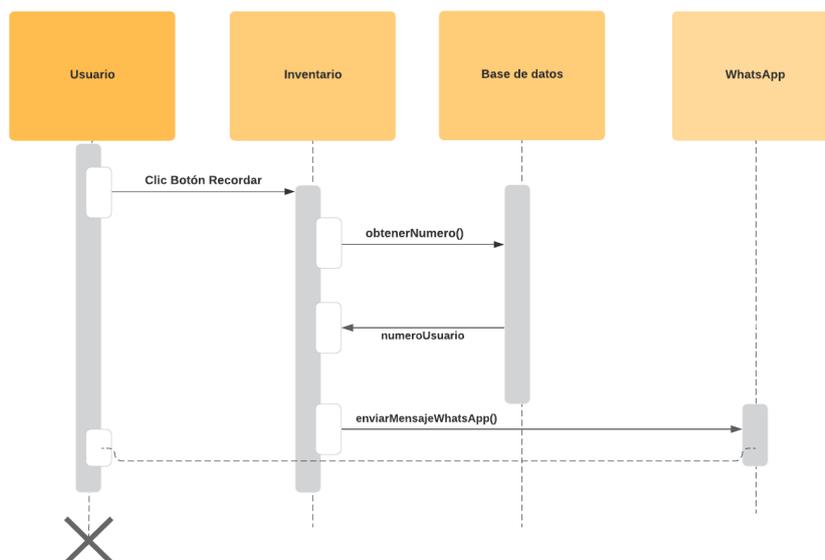


Nota. Diagrama de flujo de la función recordar pago al usuario vía correo electrónico.

Elaborado por: Jiménez Oscar y Tapia Erick

Figura 11

Diagrama de flujo 6

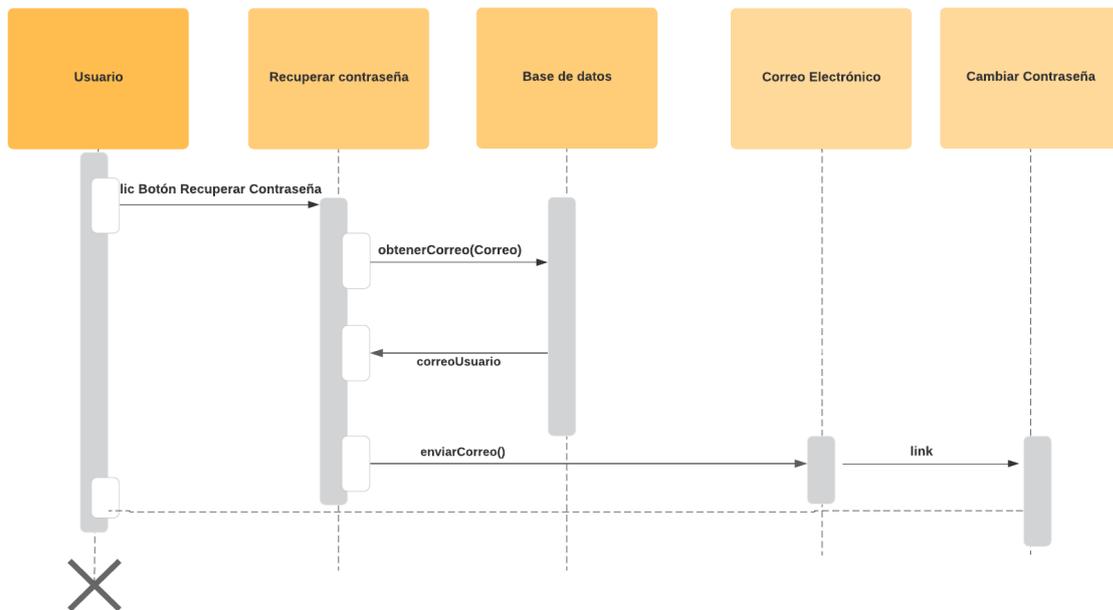


Nota. Diagrama de flujo de la función recordar pago al usuario vía WhatsApp. Elaborado

por: Jiménez Oscar y Tapia Erick

Figura 12

Diagrama de flujo 7



Nota. Diagrama de flujo de la función recordar pago al usuario vía correo electrónico.

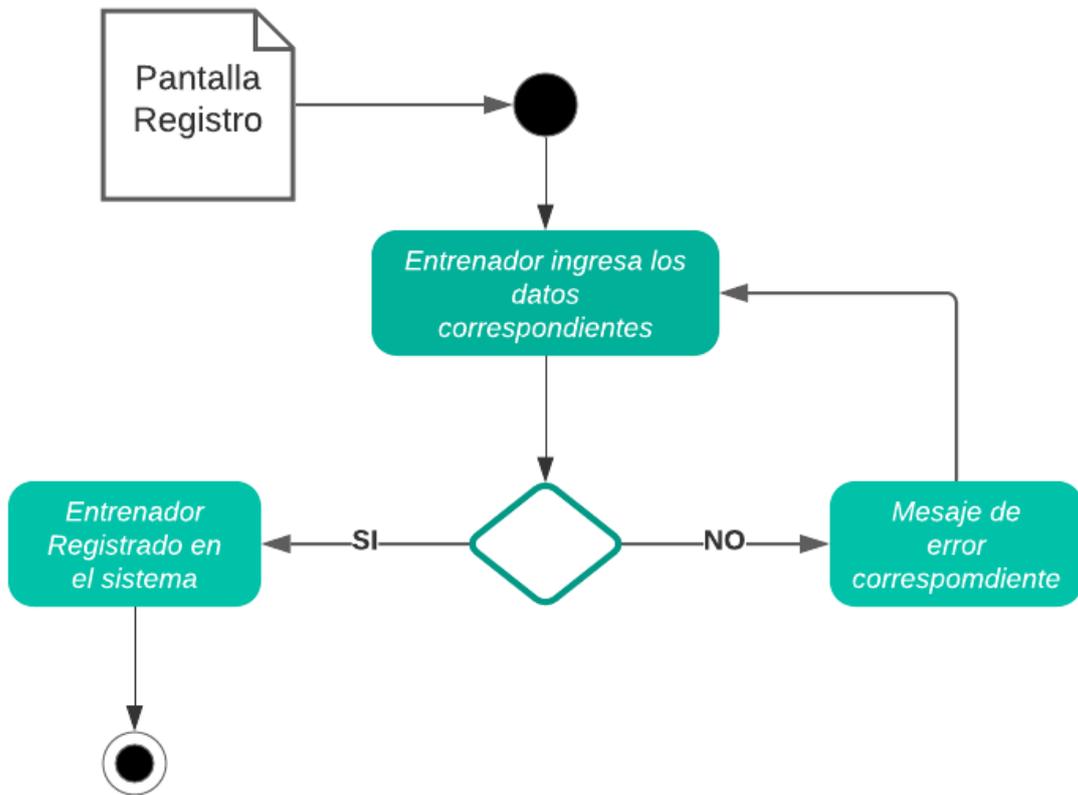
Elaborado por: Jiménez Oscar y Tapia Erick

Para verificar validaciones en los procesos se hizo uso del diagrama de actividades, y así tener en cuenta problemas que pueden generar errores en la aplicación.

Como primer proceso está el registro de usuarios, en la Figura 13, se observa que el entrenador deberá ingresar los datos, pero al momento de querer guardar esta información, el programa lo que hará es revisar si los campos obligatorios están llenos o si no existe algún error en el tipo de dato; por ejemplo, si el campo requiere un tipo de dato numérico, verificar que el entrenador haya introducido un número, mas no una letra u otro carácter. Una vez revisado todo esto el programa guardará los datos, o sino devolverá al entrenador el error y regresará a la misma pantalla.

Figura 13

Diagrama de actividades 1

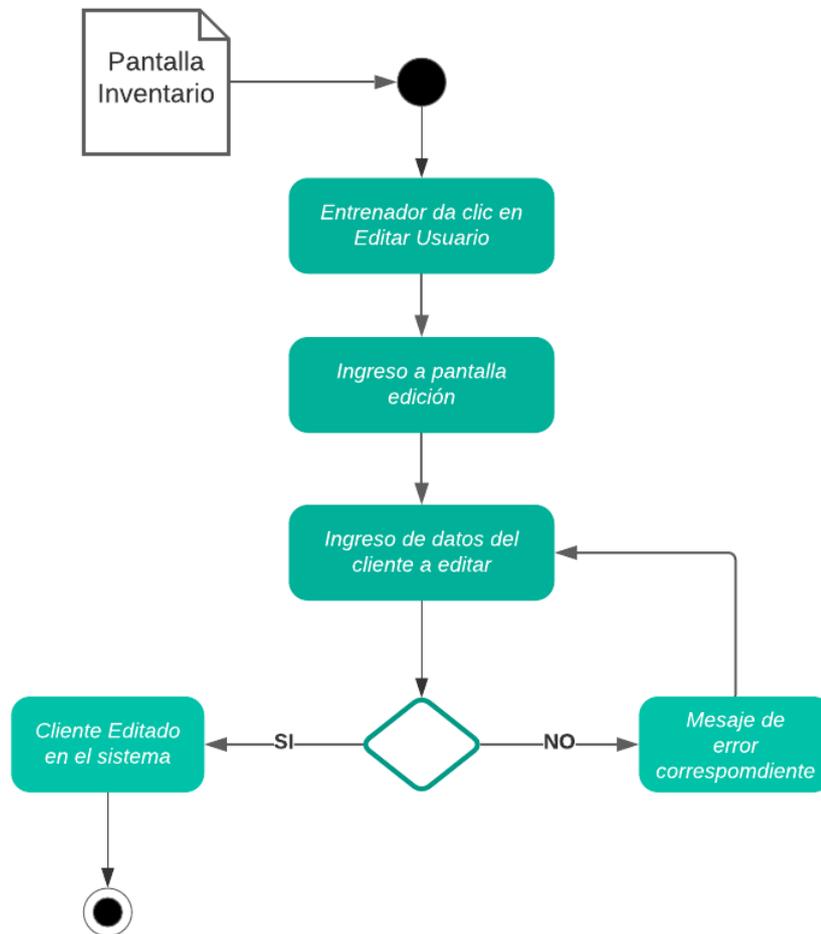


Nota. Diagrama de actividades del registro de un nuevo usuario. Elaborado por: Jiménez Oscar y Tapia Erick

Al momento de querer editar un usuario, como se muestra en la Figura 14, el proceso será el mismo que antes ya que, para guardar el dato cambiado, primero el programa deberá revisar si el tipo de dato está bien, y de igual forma si el campo está vacío o no. Si cumple todos los parámetros, el dato se guardará caso contrario se mantendrá en la misma página hasta que el entrenador introduzca bien el dato.

Figura 14

Diagrama de actividades 2

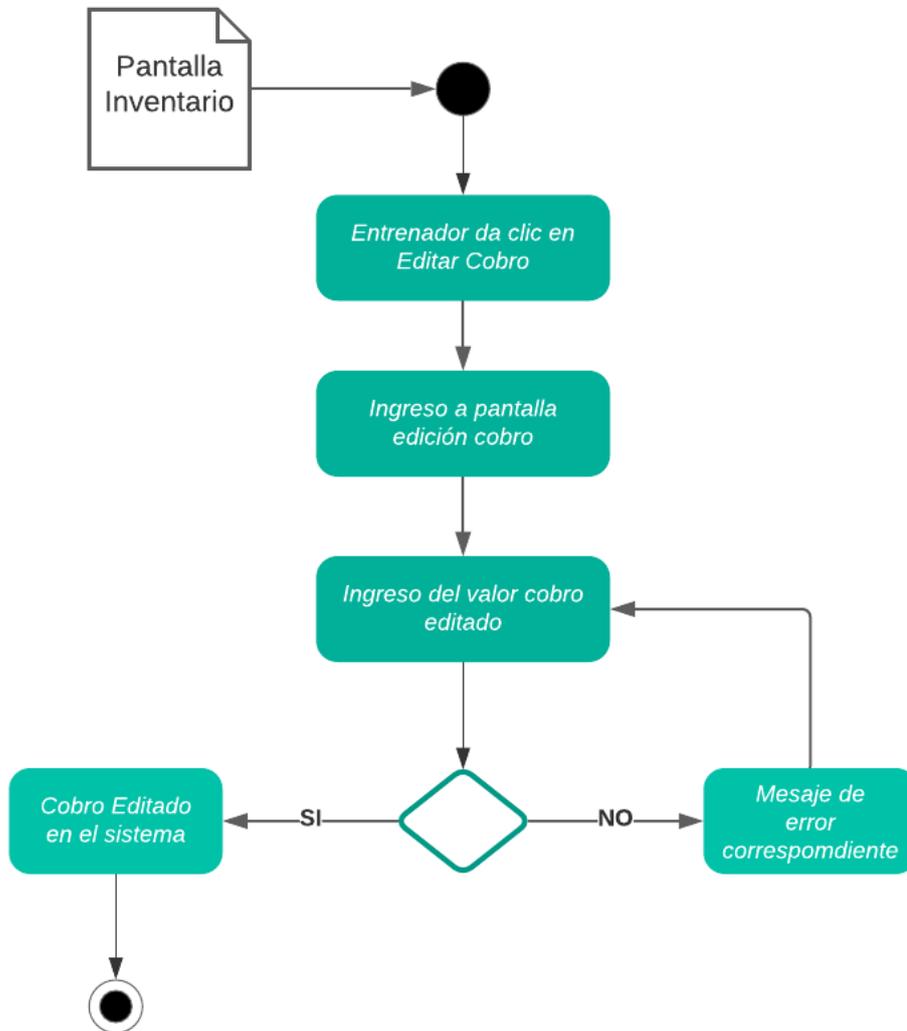


Nota. Diagrama de actividades para la edición de un usuario. Elaborado por: Jiménez Oscar y Tapia Erick

Si lo que se busca es editar el monto del cobro o pago Figura 15, el proceso es básicamente el mismo, se revisará el dato antes de ser editado y guardado en la base de datos, caso contrario el entrenador se mantendrá en la misma página hasta que cumpla correctamente la edición del dato.

Figura 15

Diagrama de actividades 3



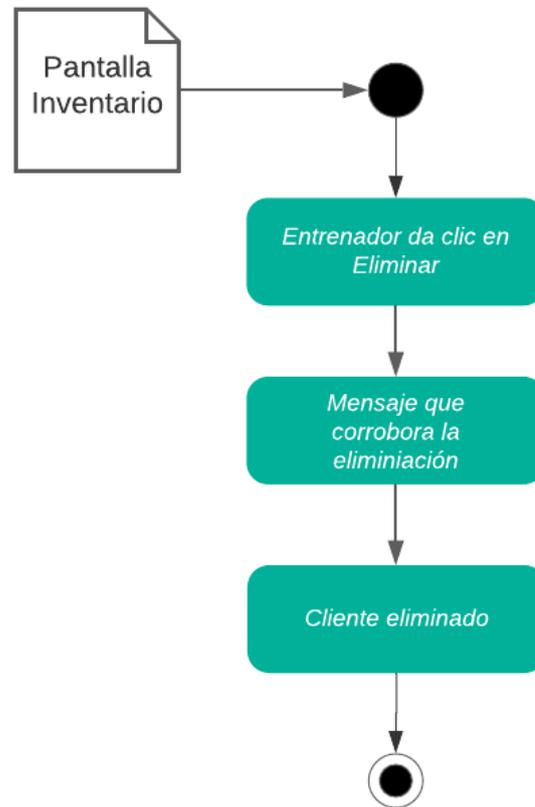
Nota. Diagrama de actividades para la edición del monto del cobro o pago del usuario

Elaborado por: Jiménez Oscar y Tapia Erick

Al momento de querer eliminar el entrenador algún usuario Figura 16, lo que aparecerá será un mensaje para corroborar que quiere eliminar el dato, si el entrenador da clic en aceptar el dato será eliminado, si no pues no se realizara ningún proceso.

Figura 16

Diagrama de actividades 4

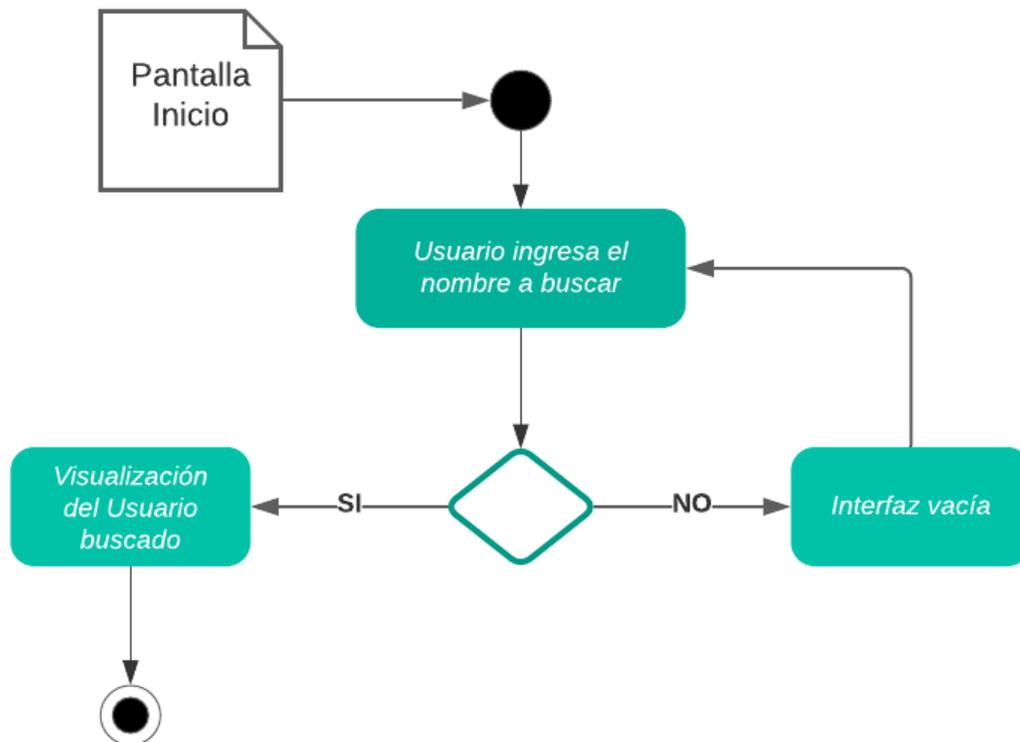


Nota. Diagrama de actividades para eliminación de algún cliente. Elaborado por: Jiménez Oscar y Tapia Erick

Para buscar un dato Figura 17, el usuario, sea cliente o entrenador, deberá ingresar el número de cedula de algún cliente registrado previamente en el programa, para que así el programa pueda buscar dicho usuario en la base de datos. Si no encuentra ninguna coincidencia, la interfaz estará vacía, de lo contrario mostrará los datos del usuario que se encontró.

Figura 17

Diagrama de actividades 5

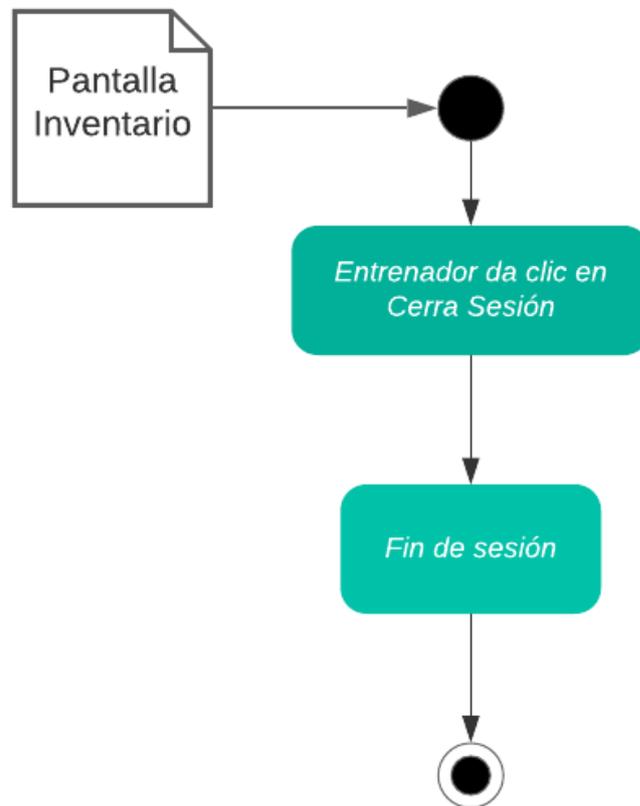


Nota. Diagrama de actividades la búsqueda de clientes. Elaborado por: Jiménez Oscar y Tapia Erick

Una actividad necesaria en este tipo de programas es la de cerrar sesión, esto permitirá que nadie que no esté autorizado pueda ingresar al sistema y alterar los datos de la empresa. Por ello el proceso es cerrar la sesión del entrenador y bloquear la entrada de cualquier otra persona Figura 18.

Figura 18

Diagrama de actividades 6

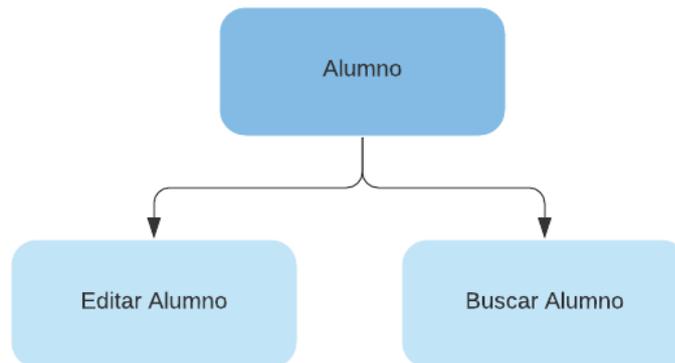


Nota. Diagrama de actividades la búsqueda de clientes. Elaborado por: Jiménez Oscar y Tapia Erick

La aplicación cuenta con tipos de usuarios, el primero es el entrenador que básicamente tiene acceso a todas las funciones del programa Figura 20. Y el segundo tipo de usuario es el cliente, el cual solo puede buscar y visualizar su propia tarjeta de cobros o la de algún otro cliente Figura 19.

Figura 19

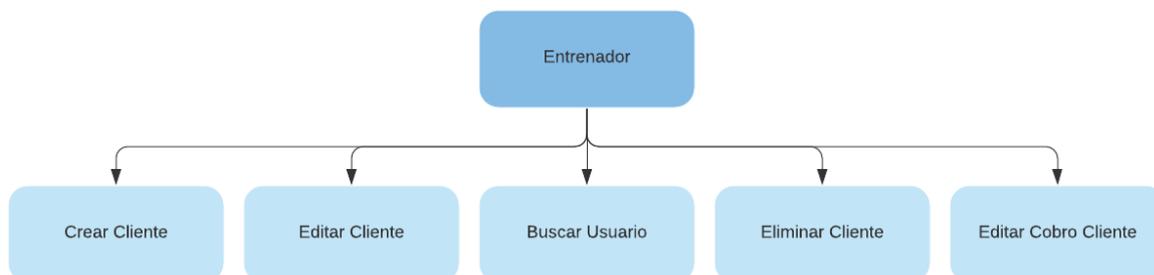
Diagrama de objetos 1



Nota. Tipo de usuario cliente y accesos permitidos. *Elaborado por:* Jiménez Oscar y Tapia Erick

Figura 20

Diagrama de objetos 2

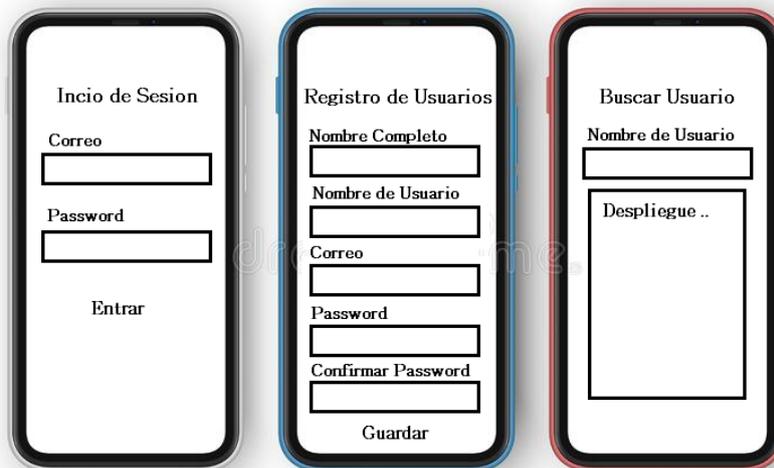


Nota. Tipo de usuario entrenador y accesos permitidos. *Elaborado por:* Jiménez Oscar y Tapia Erick

Se presenta un diseño previo a la aplicación final para tener una idea tanto de la estructura como de las pantallas que este tendrá.

Figura 21

Primer bloque de bocetos

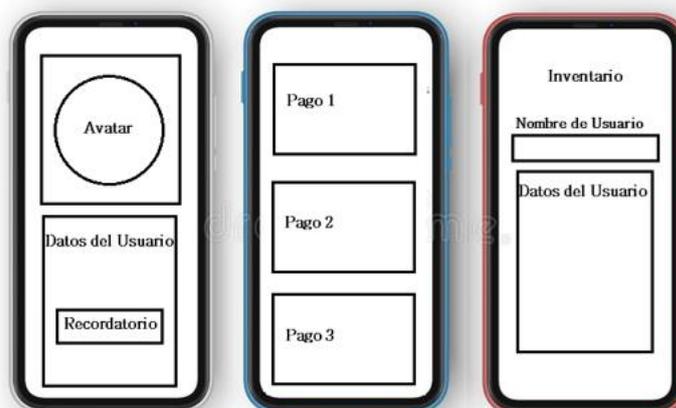


Nota. Boceto de las pantallas: inicio de sesión, registro de usuario y buscar usuario.

Elaborado por: Jiménez Oscar y Tapia Erick

Figura 22

Segundo bloque de bocetos



Nota. Boceto de las pantallas: perfil de usuario, pagos e inventario. *Elaborado por: Jiménez*

Oscar y Tapia Erick

Figura 23

Tercer bloque de bocetos



Nota. Boceto de las pantallas: registro de pagos, editar usuario y recuperar contraseña.

Elaborado por: Jiménez Oscar y Tapia Erick

Culminada la parte del diseño es indispensable obtener los requerimientos que la academia quiere, por ello se hizo uso de historias de usuario para el registro y constatación de cada uno de ellos. Existen los requerimientos funcionales los cuales tiene que ver con el proceso que el programa ejecuta, verificando los parámetros que necesita previamente para realizarlo. Por otro lado, existen los requerimientos no funcionales los cuales permiten visualizar aspectos que no tienen que ver con procesos y funciones pero que son necesarios para el uso y despliegue de la aplicación.

Tabla 4*Requerimientos funcionales*

CÓDIGO	REQUERIMIENTO
REQ-F1	Inicio de sesión para los usuarios
REQ-F2	Registro de usuarios
REQ-F3	Búsqueda de clientes registrados
REQ-F4	Edición de clientes registrados
REQ-F5	Eliminación de clientes registrados
REQ-F6	Registro de cobros de cada usuario por entrenador
REQ-F7	Visualización de pagos de los clientes por entrenador
REQ-F8	Edición de los cobros de los clientes por entrenador

Nota. Especificaciones de los requerimientos funcionales. Elaborado por: Jiménez Oscar y Tapia Erick

Tabla 5*Requerimientos no funcionales*

CÓDIGO	REQUERIMIENTO
REQ-NF1	Acciones primordiales intuitivas para el correcto uso del usuario
REQ-NF2	Conexión al motor de base de datos Mongo DB

REQ-NF3	Compatibilidad con los navegadores Mozilla y Chrome
REQ-NF4	Diseño responsive

Nota. Especificaciones de los requerimientos no funcionales. Elaborado por: Jiménez Oscar y Tapia Erick

Tabla 6

Historia de usuario para el requerimiento 1

HISTORIA DE USUARIO	
Requerimiento Funcional	REQ-F1
Prioridad en negocio	Alta
Prerrequisitos	Estar registrado en la aplicación
Descripción	<p>El usuario se encuentra en la página de Inicio de Sesión.</p> <p>En la sección de “Usuario”, ingresa su correo.</p> <p>En la sección de “Contraseña”, ingresa su contraseña.</p> <p>La aplicación obtiene los dos datos anteriormente introducidos, cuando el administrador de clic en el botón “Ingresar”.</p> <p>Revisa que ningún campo este vacío.</p> <p>Revisa que el correo tenga el símbolo @.</p>

	<p>Manda los datos al servidor para su verificación en la base de datos.</p> <p>La aplicación web envía un mensaje al usuario informándole que están correctos.</p> <p>La aplicación web permite ingresar al usuario a su pantalla de inicio personal.</p>
Flujo Alternativo	<p>Si los campos están vacíos el programa envía un mensaje.</p> <p>Si no existe el @ el programa envía un mensaje</p> <p>Si los datos no están correctos el programa envía un mensaje de error</p>

Nota. Especificaciones del primer requerimiento funcional. Elaborado por: Jiménez Oscar y

Tapia Erick

Tabla 7

Historia de usuario para el requerimiento 2

HISTORIA DE USUARIO	
Requerimiento Funcional	REQ-F2
Prioridad en negocio	Alta
Prerrequisitos	Ser administrador
Descripción	El usuario se encuentra en la página de Registro de Usuarios.

	<p>En la sección de “Nombre”, ingresa el nombre del cliente.</p> <p>En la sección de “Edad”, ingresa la edad del cliente.</p> <p>En la sección de “Usuario”, ingresa el correo del cliente.</p> <p>En la sección de “Contraseña”, ingresa la contraseña del cliente.</p> <p>La aplicación obtiene los dos datos anteriormente introducidos cuando el administrador da clic en el botón “Guardar”.</p> <p>Revisa que ningún campo este vacío.</p> <p>Revisa que el correo tenga el símbolo @.</p> <p>Manda los datos al servidor para almacenarlos en la base de datos.</p> <p>La aplicación web envía un mensaje al administrador informándole que están correctos.</p> <p>La aplicación web redirige al inventario de clientes mostrando al nuevo cliente con su respectiva información.</p> <p>La aplicación web permite editar o eliminar al cliente recién creado.</p>
--	--

Flujo Alternativo	<p>Si los campos están vacíos el programa envía un mensaje.</p> <p>Si no existe el @ el programa envía un mensaje</p>
-------------------	---

Nota. Especificaciones del segundo requerimiento funcional. Elaborado por: Jiménez Oscar y Tapia Erick

Tabla 8

Historia de usuario para el requerimiento 3

HISTORIA DE USUARIO	
Requerimiento Funcional	REQ-F3
Prioridad en negocio	Alta
Prerrequisitos	Ser administrador
Descripción	<p>El usuario se encuentra en la página de Inventario.</p> <p>En la sección de “Usuario”, ingresa el correo del cliente.</p> <p>La aplicación obtiene el dato anteriormente introducciones cuando el administrador da clic en el botón “Buscar”.</p> <p>Revisa que el campo no este vacío.</p> <p>Revisa que el correo tenga el símbolo @.</p>

	<p>Manda los datos al servidor para buscarlo en la base de datos.</p> <p>La aplicación web refresca la página Inventario mostrando el cliente.</p> <p>La aplicación web permite editar o eliminar al cliente.</p>
Flujo Alternativo	Si los campos están vacíos el programa no debe colapsar.

Nota. Especificaciones del tercer requerimiento funcional. Elaborado por: Jiménez Oscar y Tapia Erick

Tabla 9

Historia de usuario para el requerimiento 4

HISTORIA DE USUARIO	
Requerimiento Funcional	REQ-F4
Prioridad en negocio	Alta
Prerrequisitos	Ser administrador
Descripción	<p>El usuario se encuentra en la página de Inventario.</p> <p>En la tarjeta de cada usuario existe el botón editar.</p> <p>La aplicación obtiene el dato del cliente respectivo cuando el administrador da clic en el botón “Editar”.</p>

	<p>Se redirige a la página “Editar” donde podrá cambiar los datos del usuario.</p> <p>Revisa que ningún campo este vacío.</p> <p>Revisa que el correo tenga el símbolo @.</p> <p>Manda los datos al servidor para actualizarlos en la base de datos.</p> <p>La aplicación web redirige a la página Inventario mostrando el cliente actualizado.</p>
Flujo Alternativo	<p>Si los campos están vacíos el programa envía un mensaje.</p> <p>Si no existe el @ el programa envía un mensaje</p>

Nota. Especificaciones del cuarto requerimiento funcional. Elaborado por: Jiménez, Oscar y Tapia Erick

Tabla 10

Historia de usuario para el requerimiento 5

HISTORIA DE USUARIO	
Requerimiento Funcional	REQ-F5
Prioridad en negocio	Alta
Prerrequisitos	Ser administrador
Descripción	El usuario se encuentra en la página de Inventario.

	<p>En la tarjeta de cada usuario existe el botón eliminar.</p> <p>La aplicación obtiene el dato del cliente respectivo cuando el administrador da clic en el botón “Eliminar”.</p> <p>Manda los datos al servidor para eliminarlos de la base de datos.</p> <p>La aplicación web redirige a la página Inventario mostrando los clientes actuales.</p>
Flujo Alternativo	Si no se ha eliminado el cliente el programa redirige a la página inventario para que lo elimine nuevamente.

Nota. Especificaciones del quito requerimiento funcional. Elaborado por: Jiménez Oscar y Tapia Erick

Tabla 11

Historia de usuario para el requerimiento 6

HISTORIA DE USUARIO	
Requerimiento Funcional	REQ-F6
Prioridad en negocio	Alta
Prerrequisitos	Ser administrador
Descripción	El administrador se encuentra en la página de Inventario.

	<p>En la tarjeta de cada usuario existe el botón “cobro”.</p> <p>La aplicación obtiene el dato del cliente respectivo cuando el administrador da clic en el botón “cobro”.</p> <p>Se redirige a la página “Cobros” donde podrá ingresar el monto del cliente.</p> <p>Revisa que ningún campo este vacío.</p> <p>Revisa que el dato sea numérico.</p> <p>Manda los datos al servidor para actualizarlos en la base de datos.</p> <p>La aplicación web redirige a la página Inventario mostrando el monto del cliente actualizado.</p>
Flujo Alternativo	<p>Si los campos están vacíos el programa bloqueará el botón enviar.</p> <p>Si el dato no es numérico el programa no permitirá su escritura.</p>

Nota. Especificaciones del sexto requerimiento funcional. Elaborado por: Jiménez Oscar y

Tapia Erick

Tabla 12*Historia de usuario para el requerimiento 7*

HISTORIA DE USUARIO	
Requerimiento Funcional	REQ-F7
Prioridad en negocio	Alta
Prerrequisitos	Ser administrador
Descripción	El administrador se encuentra en la página de Inventario. En la página Inventario existen tarjetas con información de cada cliente, las cuales el administrador podrá ver.
Flujo Alternativo	Si no existen clientes la página estará vacía.

*Nota. Especificaciones del séptimo requerimiento funcional. Elaborado por: Jiménez Oscar y**Tapia Erick***Tabla 13***Historia de usuario para el requerimiento 8*

HISTORIA DE USUARIO	
Requerimiento Funcional	REQ-F8
Prioridad en negocio	Alta
Prerrequisitos	Ser administrador
Descripción	El administrador se encuentra en la página de Inventario.

	<p>En la tarjeta de cada usuario existe el botón “cobro”.</p> <p>La aplicación obtiene el dato del cliente respectivo cuando el administrador da clic en el botón “cobro”.</p> <p>Se redirige a la página “Cobros” donde podrá ingresar el nuevo monto del cliente.</p> <p>Revisa que ningún campo este vacío.</p> <p>Revisa que el dato sea numérico.</p> <p>Manda los datos al servidor para actualizarlos en la base de datos.</p> <p>La aplicación web redirige a la página Inventario mostrando el monto del cliente actualizado.</p>
Flujo Alternativo	<p>Si no existen datos la pantalla estará vacía.</p> <p>Si los campos están vacíos el programa bloqueará el botón actualizar.</p> <p>Si el dato no es numérico el programa no permitirá su escritura.</p>

Nota. Especificaciones del octavo requerimiento funcional. Elaborado por: Jiménez Oscar y

Tapia Erick

2.3 Implementación

2.3.1 Levantamiento del servidor. El servidor que se usó para el proyecto fue NodeJs, el cual es útil para realizar este tipo de aplicaciones ya que trabaja con el lenguaje de programación JavaScript y además tiene un sin número de librerías. Una de ellas es express Figura 24 que sirve para diseñar páginas web por medio de plantillas, por lo que se utilizó handlebars, el que permitió el uso de HTML CSS y Bootstrap.

Figura 24

Librerías

```
const express = require('express')
const mongoose = require("mongoose")
const hbs = require('hbs')
const session = require('express-session')
```

Nota. Librerías usadas para el levantamiento del servidor y base de datos. Elaborado por: Jiménez Oscar y Tapia Erick

2.3.2 Conexión a la base. La base de datos que se utilizó fue Mongo DB, esta es una base de datos no SQL, esto significa que no tiene estructura de tablas y tampoco de relaciones, simplemente hace uso de colecciones, las cuales tienen una estructura igual a JSON, lo que facilita la gestión de datos y cumple con todas las necesidades que requiere el programa.

2.3.3 Registro de Usuarios. Una vez realizada el levantamiento del servidor y la conexión a base de datos, se inició con la codificación de los respectivos procesos que conlleva el método CRUD. Las siglas antes mencionadas significan: Create, Read, Update, Delete, que en español vendría a ser: Crear, Leer, Actualiza/Editar, Eliminar. Estas funciones serán esenciales en la aplicación para el registro de cobros, ya que se necesita crear usuarios, buscar los usuarios y sus respectivos pagos, editar o actualizar, ya sea sus pagos o algún dato

en específico, y por último si el estudiante o profesor de la academia se da de baja habrá que eliminarlo de la aplicación.

Como se puede observar la aplicación web gira alrededor de estas funciones y por ello serán programadas de la mejor manera posible.

Para el registro de usuarios, se tomó como regencia las fichas asignadas a cada estudiante y así poder guardar la información en la base de datos en la única colección que existe en el aplicativo denominada “users”, el cual guarda el tipo de usuario, sea este administrador o estudiante, así como toda la información requerida en el inicio de sesión la cual es: nombre completo, nombre de usuario, tipo_usuario, correo electrónico, posición, horario, matrícula, pagos, edad y contraseña. El dato pago a su vez es un vector de objetos el cual guarda el registro de la fecha y el monto que pagó el estudiante. Por otra parte, el dato matricula es un booleano, asignando al estudiante que a pagado la matricula el valor de “true” o de lo contrario será “false”, cabe mencionar que aun principio todos los estudiantes serán guardados por defecto como “false” en este campo. Lo mismo sucede con la variable “tipo_usuario”, que también es de tipo booleano, donde “true” significa administrador y “false” estudiante. Esto se hace ya que se debe restringir el acceso al estudiante de las funciones de registrar pagos, así como de la pantalla de inventario.

Cabe recalcar que todos los datos han sido validados, por lo que, en Nombre Completo no puede introducir un número, de igual forma en el campo email deberá estar con el símbolo “@”, caso contrario no le permitirá guardar los datos. Y, por último, la contraseña debe constar de cuatro características, la primera es que debe tener al menos una letra mayúscula, la segunda es que debe haber al menos una letra minúscula, la tercera que debe existir por lo menos un número y la cuarta es que la contraseña en total debe tener más de 8 caracteres.

2.3.4 Encriptación. La contraseña consta con una encriptación, y para ello fue necesario acudir a la librería “bcrypt”. Esta contiene un algoritmo de hashing para contraseñas

y que fue desarrollada por Niels Provos y David Maxieres, el cual maneja el cifrado de Blowfish. Cabe mencionar que lleva incorporada una variable salt, el cual es un fragmento aleatorio que se utiliza para generar un hash vinculada a la contraseña, guardándose ambas en la base de datos, esto evita que se generen dos contraseñas similares y cubre aspectos de seguridad como, ataque por fuerza bruta a todas las contraseñas del sistema a la vez.

Finalizado las validaciones de los campos a llenar en el registro de usuarios, se procede a obtener estos datos y enviarlos a la base de datos, como se dijo antes a la colección “users”, la cual guarda esta información siempre y cuando todo este correcto, dando por terminado el proceso de registro de usuarios de la aplicación web.

2.3.5 Inicio se sesión. Con el registro de usuarios en la base de datos, se puede diseñar el inicio de sesión, para ello lo que se creó fue su respectiva pantalla con un formulario, en este caso para que el usuario pueda ingresar su email y contraseña, dando así al programa ambos datos para su respectiva búsqueda en la base de datos. Si ambos datos existen el programa redirigirá al usuario a la siguiente página caso contrario permanecerá en la misma mostrándole en pantalla el mensaje de error correspondiente.

Previamente se había mencionado que la contraseña para a ser encriptada antes de ser enviada a la base de datos, por lo que, si se desea obtener, esta vendrá encriptada.

Evidentemente la contraseña que está en la base de datos debe ser compara con la contraseña que ingresa el usuario, pero aquí surge un inconveniente, el usuario puede ingresar bien su contraseña, pero al no ser encriptada nunca será igual a la contraseña que esta guardada en la base de datos y no podrá ingresar a la siguiente página. Para ello Bcrypt tiene una función llamada “compare” que básicamente tiene dos parámetros esenciales, en este caso, el primero es la contraseña no encriptada y el segundo dato es la contraseña encriptada, así entonces esta función ayuda al proceso de comparación sin la necesidad de volver a encriptar la contraseña. Con todo este proceso realizado la pantalla de inicio de sesión queda lista para ser utilizada.

2.3.6 Sesiones. Es claro saber que cuando se trata de aplicaciones que llevan consigo un inicio de sesión, es necesario el uso de sesiones ya que estas permiten verificar si el usuario está dentro de su cuenta o no, así como cerrar la sesión para que no pueda ingresar nuevamente alguien desconocido sin su previa autenticación. La herramienta que permitió hacer esto fue “express” mediante su librería “session”. Lo que genera esto es un nuevo ID de sesión para el usuario y este dato se guardad en el lado del servidor. En la librería “session” existen varias funciones, una de ellas es “logged”, presentada en la Figura 25, el cual es una variable de tipo booleano que verifica si está un usuario logueado o no, siendo true el suceso de que el usuario este logueado y false si no lo está. Por otro lado, la función “role” contiene el rol que presenta el usuario, en este caso existen dos roles: “USER_ROLE” y “ADMIN_ROLE”. Siendo el primero el rol asignado para un cliente de la academia y el segundo para el entrenador de la misma. Otra de las funciones que se utilizo fue “username”, que en ella se guardara el nombre de usuario que actualmente está en sesión. Como última función utilizada fue “destroy”, que no necesita ningún parámetro simplemente con llamarlo funciona. Lo que hace es borrar toda la información de la sesión que se creó a un principio.

Figura 25

Sesiones

```
var islogged = function (req, res, next) {
  if (req.session.logged == undefined) {
    req.session.logged = false
  }
  if (req.session.role == undefined) {
    req.session.role = 'USER_ROLE'
  }
  next()
};
```

Nota. Función para verificar si un usuario se encuentra logeado. Elaborado por: Jiménez

Oscar y Tapia Erick

2.3.7 Función buscar usuario. Como siguiente función existe la de buscar usuario.

Para ello se debe obtener el valor que el propio usuario ingresa en el input denominado “Buscar”. Este valor será buscado en la base de datos, cabe resaltar que el valor a buscar es el nombre de usuario de la persona. Una vez identificado el nombre de usuario podrá dar clic en el mismo y este le redirigirá directo a su perfil.

2.3.8 Función editar usuario. En la pantalla del inventario existe un botón asignado para cada usuario el cual se llama “Modificar”. Lo que hace esta función es abrir una nueva pantalla donde aparecerá un formulario con los campos llenos del usuario a editar. Con el formulario en pantalla el profesor podrá editar la información que desea guardando todo esto al momento que da clic en el botón “Guardar”. En la pantalla del usuario también se introdujo un botón con el cual podrá editar sus datos personales, mas no el pago de matrícula, el cual si puede editar el profesor en la pantalla inventario antes mencionada. Para ello al dar clic en el botón, este redirigirá a la pantalla de editar usuario y allí podrá cambiar cualquier tipo de información y guardarla en la base de datos.

2.3.9 Función eliminar usuario. Para poder eliminar el usuario, lo que se hizo fue obtener el id del usuario, el cual busca este id en la base de datos y procede a eliminar toda la información que existe en ese objeto. Esta es una función muy sencilla, por no decir la más fácil de todas.

2.3.10 Función editar cobro. La función “editar cobro” es similar a la de “editar usuario”, aquí se puede dar énfasis al CRUD, gracias a estas funciones predeterminadas se puede reutilizar en varios aspectos que se necesite en el programa. En este caso el “editar cobro” es simplemente una réplica de “editar usuario”, simplemente se ha cambiado los que se necesitan editar en esta pantalla.

2.3.11 Registrar cobro. La raíz de todo el programa es esto, el poder registrar los pagos de los clientes de la academia de ecuavoley, por lo tanto, aquí es donde se obtiene el

valor que el estudiante a pagado en ese momento, así como la fecha del pago, todo esto se guarda en la base de datos de la aplicación. Se lo realizo con la función “registrar cobro”, el cual obtiene el dato y lo guarda en la base de datos, para luego mostrarla, editar o a su vez eliminarla. Al momento que se registra el nuevo pago, se le envía automáticamente al usuario un correo electrónico, tal cual como se realizó en la sección de recuperar contraseña, solo que en este apartado se envía un mensaje confirmando su mensualidad pagada. El entrenador podrá ver también el registro del cobro en la misma pantalla ya que aquí se visualizará la fecha en la que se ha guardado el pago y el nombre del cliente.

2.3.12 Recuperar contraseña. Con la librería Nodemailer, se pueden realizar envíos de correo electrónico con el uso del lenguaje de programación NodeJs. En la pantalla de recuperar contraseña lo que se hace es pedir al usuario que ingrese su correo electrónico con el cual se ha registrado en la aplicación, una vez ingresado el dato deberá dar clic al botón enviar. Una vez realizado esto, lo que se hace es buscar primeramente el correo electrónico para ver si existe en la base de datos, si no existe el proceso termina ahí, de lo contrario el siguiente proceso es enviar un link al correo electrónico del usuario que lo está solicitando. Este link es un acceso a la pantalla de cambiar contraseña. Una vez que el usuario este en la pantalla podrá registrar su nueva contraseña y así actualizarla, por lo que se hará uso de la función “actualizar contraseña”, la cual pedirá como entrada el nuevo dato, para después encriptarlo y así proceder a sustituir la nueva contraseña por la antigua.

2.3.13 Recordatorio. El botón “recordatorio”, sirve para enviar un email al usuario recordando el pago de su mensualidad en la academia. Lo que se realiza es, obtener el email del usuario mediante su id, el cual se encuentra generado ya que previamente el entrenador ha debido iniciar sesión para realizar este proceso. Se utiliza la misma librería para el envío de email, y únicamente se cambia los valores a enviar, que en este caso es un mensaje recordaron el pago al estudiante.

2.3.14 Inventario. Una pantalla sumamente importante para la gestión de cobros de la academia es la denominada “Inventario”, ya aquí tendrán acceso únicamente los administradores, en este caso los entrenadores, para poder visualizar todos los datos de sus estudiantes, así como también editar dicha información como pagos realizados, o a su vez eliminarlos de la plataforma. Es aquí donde todo el conjunto de funciones que se realizaron antes, y que ya fueron explicadas, serán implantadas en este módulo para tener un gestor completo.

A su vez, en la misma pantalla están las tablas de “pagos por cobrar”, “pagos vencidos” y “próximos pagos”. Estas tres funciones se adecúan para que los entrenadores obtengan una organización especial en la gestión de cobros, ya que básicamente lo que permiten hacer es jerarquizar los cobros que se deben realizar cada mes. Para un mejor entendimiento; “pagos por cobrar” es una tabla donde se observan a los estudiantes que deben cancelar su mensualidad en el plazo de dos días, es decir que, si un estudiante debe pagar su suscripción en la academia todos los días 14 de cada mes, entonces este aparecerá en dicha tabla a partir del 13, y así informar al entrenador que este estudiante ya debe realizar su pago, es importante aclarar que el estudiante estará en esta tabla hasta el 15, ya que la academia da un día de gracia a los alumnos. En esta tabla se presenta una función parecida, que fue explicada anteriormente, la cual es la de “recordatorio”, solo que al ser un recordatorio para un pago que se debe realizar en ese preciso momento, esta será notificada vía WhatsApp, lo que permite una interacción más directa con el alumno. Este envío de mensajes directos a celular fue realizado gracias a la API de WhatsApp, la cual permite realizar este tipo de funciones y así implementarlas en aplicaciones que lo necesiten.

Por otro lado, está la tabla de pagos vencidos, aquí se podrán observar todos los alumnos que no han realizado su pago en el tiempo dado, y de igual manera se les podrá notificar su atraso mediante mensaje vía WhatsApp. Y por último existe la tabla de “próximos

pagos”, en donde estarán todos los estudiantes que llevan al día con sus cuotas y simplemente se poder ver que fechas próximas son las que deben pagar por si se necesita consultar alguna de ellas.

2.3.15 Cerrar sesión. Este es el último proceso de la aplicación y lo único que hace es terminar la sesión que se generó a un principio con el uso de la librería “express”, con su función “`expres.session.destroy()`”, el cual destruye el id de la sesión, impidiendo así una redirección a cualquier URL sin previo inicio de sesión.

2.3.16 Mensajes de acciones. Los mensajes que se utilizador para el aviso de errores o de confirmaciones de acciones se las realizó con el uso de tarjetas que tiene Bootstrap. Están fueron integradas al programa para que el usuario tenga conocimiento de lo que ha hecho en las acciones más relevantes de la aplicación.

2.4 Despliegue

Por último, esta aplicación web se la subió a GitHub, como repositorio privado por decisiones de la empresa, y posteriormente fue subida a los servidores de Heroku para pasar a la fase de producción, en este caso una versión beta. Estas dos plataformas son completamente gratuitas y tiene funciones que ayudan a los programadores a publicar sus proyectos.

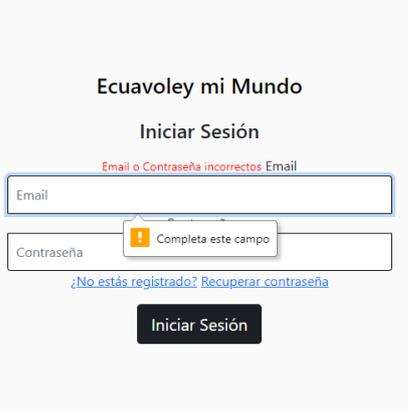
2.5 Pruebas

Se rindieron pruebas de caja negra, carga, stress y de diseño para obtener resultados sobre la aplicación web en general.

2.5.1 Pruebas de Caja Negra. Básicamente, se miden los requerimientos que fueron dispuestos a un principio, para verificar que se cumple con cada uno de ellos.

Tabla 14

Resultados de la prueba de caja negra I

Requerimiento Funcional	Acción a analizar	Resultado	Descripción
REQ-F1	Inicio de sesión para los usuarios	 <p>The screenshot shows the login interface for 'Ecuavoley mi Mundo'. At the top, it says 'Ecuavoley mi Mundo' and 'Iniciar Sesión'. Below this, there is a red error message: 'Email o Contraseña incorrectos'. There are two input fields: 'Email' and 'Contraseña'. The 'Contraseña' field has a tooltip that says 'Completa este campo'. Below the fields, there is a link: '?No estás registrado? Recuperar contraseña' and a black button labeled 'Iniciar Sesión'.</p>	Acceso a la aplicación por medio del inicio de sesión sin ningún dato ingresado. Mensaje: Completar este campo.
		 <p>This screenshot is similar to the previous one, but the 'Email' field is now filled with the text 'danigarck@hotmail.com'. The 'Contraseña' field remains empty, and the tooltip 'Completa este campo' is still present. The rest of the interface, including the error message and the 'Iniciar Sesión' button, remains the same.</p>	Acceso a la aplicación por medio del inicio de sesión solo con el correo electrónico y sin ingresar la contraseña.

			<p>Mensaje: Completar este campo.</p>
			<p>Acceso a la aplicación por medio del inicio de sesión con el ingreso del correo sin el “@” Mensaje: incluir el signo “@”.</p>
			<p>Acceso a la aplicación por medio del inicio de sesión con un correo electrónico o contraseña falsa. Mensaje: Email o</p>

			contraseña incorrectos.
--	--	--	-------------------------

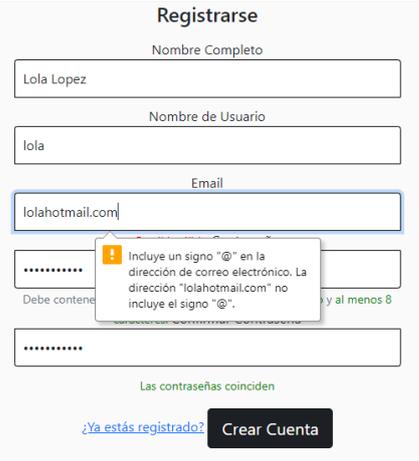
Nota. Resultados según la prueba de caja negra para el primer requerimiento funcional.

Elaborado por: Jiménez Oscar y Tapia Erick

Tabla 15

Resultados de la prueba de caja negra 2

Requerimiento Funcional	Acción a analizar	Resultado	Descripción
REQ-F2	Registro de Usuario		Envío del formulario para el registro de usuarios con los campos vacíos. Mensaje: Completar campos.

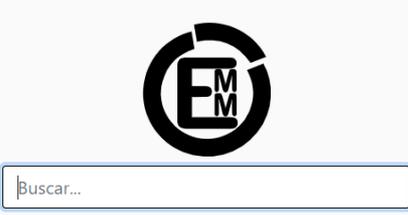
			<p>Envío del formulario para el registro de usuarios con la contraseña sin “@”. Mensaje: incluir el signo “@”.</p>
--	--	---	--

Nota. Resultados según la prueba de caja negra para el segundo requerimiento funcional.

Elaborado por: Jiménez Oscar y Tapia Erick

Tabla 16

Resultados de la prueba de caja negra 3

Requerimiento Funcional	Acción a analizar	Resultado	Descripción
REQ-F3	Búsqueda de clientes registrados		<p>Búsqueda de un cliente con los campos vacíos. Acción: el programa sigue funcionando</p>

			ya que no se buscará el usuario a medida que va ingresando el nombre.
--	--	--	---

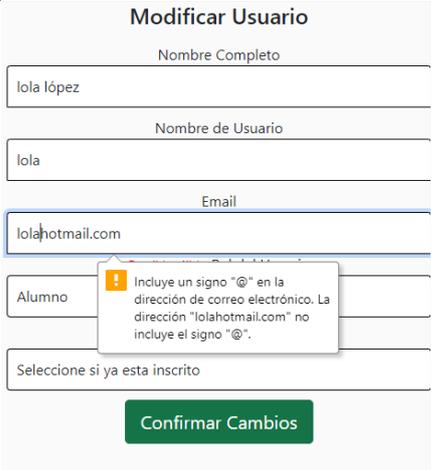
Nota. Resultados según la prueba de caja negra para el tercer requerimiento funcional.

Elaborado por: Jiménez Oscar y Tapia Erick

Tabla 17

Resultados de la prueba de caja negra 4

Requerimiento Funcional	Acción a analizar	Resultado	Descripción
REQ-F4	Edición de clientes registrados		Envío del formulario para la edición de usuarios con los campos vacíos. Mensaje: Completar campos.

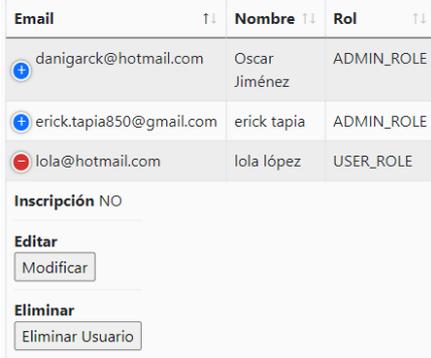
			<p>Envío del formulario para la edición de usuarios con la contraseña sin “@”.</p> <p>Mensaje: incluir el signo “@”.</p>
--	--	---	--

Nota. Resultados según la prueba de caja negra para el cuarto requerimiento funcional.

Elaborado por: Jiménez Oscar y Tapia Erick

Tabla 18

Resultados de la prueba de caja negra 5

Requerimiento Funcional	Acción a analizar	Resultado	Descripción
REQ-F5	Eliminación de clientes registrados		<p>Botón asignado a cada usuario para su procedente eliminación.</p> <p>Acción: si no acciona el</p>

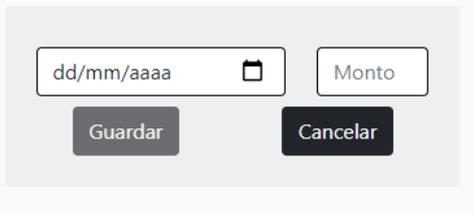
			botón no sucederá nada.
--	--	--	----------------------------

Nota. Resultados según la prueba de caja negra para el quinto requerimiento funcional.

Elaborado por: Jiménez Oscar y Tapia Erick

Tabla 19

Resultados de la prueba de caja negra 6

Requerimiento Funcional	Acción a analizar	Resultado	Descripción
REQ-F6	Registro de cobros de cada usuario por entrenador		Envío del formulario para el registro de cobros con los campos vacíos. Acción: El botón “Guardar” permanece bloqueado.
			Envío del formulario para el registro de

			cobros solo con el campo “monto” vacío. Acción: El botón “Guardar” permanece bloqueado.
--	--	--	---

Nota. Resultados según la prueba de caja negra para el sexto requerimiento funcional.

Elaborado por: Jiménez Oscar y Tapia Erick

Tabla 20

Resultados de la prueba de caja negra 7

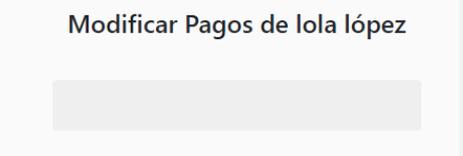
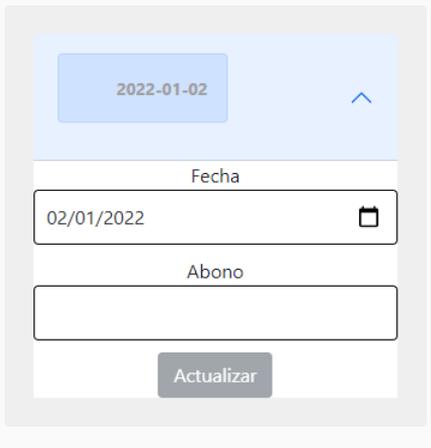
Requerimiento Funcional	Acción a analizar	Resultado	Descripción
REQ-F7	Visualización de pagos de los clientes por entrenador		Si no existe ningún registro de pagos del usuario, la pantalla permanece vacía.

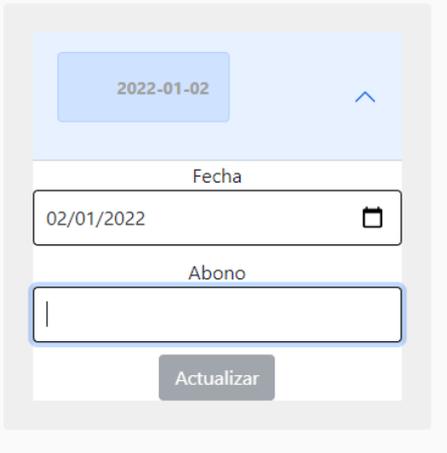
Nota. Resultados según la prueba de caja negra para el séptimo requerimiento funcional.

Elaborado por: Jiménez Oscar y Tapia Erick

Tabla 21

Resultados de la prueba de caja negra 8

Requerimiento Funcional	Acción a analizar	Resultado	Descripción
REQ-F8	Edición de los cobros de los clientes por entrenador		Si no existe ningún registro de pagos del usuario, la pantalla permanece vacía.
			Envío del formulario para la edición de cobros con los campos vacíos. Acción: El botón “Actualizar”

			permanece bloqueado.
			Envío del formulario para la edición de cobros solo con el campo “monto” vacío. Acción: El botón “Actualizar” permanece bloqueado

Nota. Resultados según la prueba de caja negra para el octavo requerimiento funcional.

Elaborado por: Jiménez Oscar y Tapia Erick

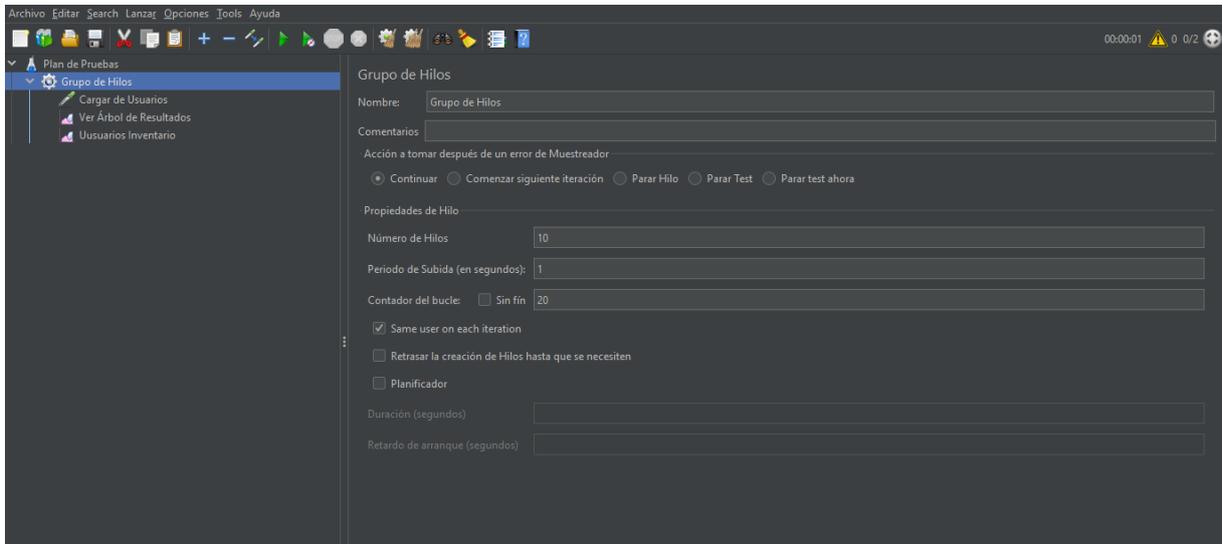
2.5.2 Pruebas de carga. Para las pruebas tanto de carga como de stress, se utilizará la herramienta JMeter, la cual es gratuita e intuitiva y entrega informes en tablas o gráficos fáciles de interpretar.

2.5.2.1 Carga Tabla de usuarios en el inventario con 10 entrenadores y 20 alumnos para cada entrenador. Se realizó una prueba de carga en la pantalla de inventario, específicamente al momento de mostrar los usuarios de la academia. Para ello se simuló un escenario donde, al mismo tiempo, 10 entrenadores ingresaban 20 veces a la pantalla de inventario para visualizar la tabla de usuarios, todo esto en el lapso de 1 segundo. Finalizada

la simulación se observó que todas las cargar fueron exitosas sin ningún error ocasionado, dando un tiempo total de ejecución de 3,7 segundos.

Figura 26

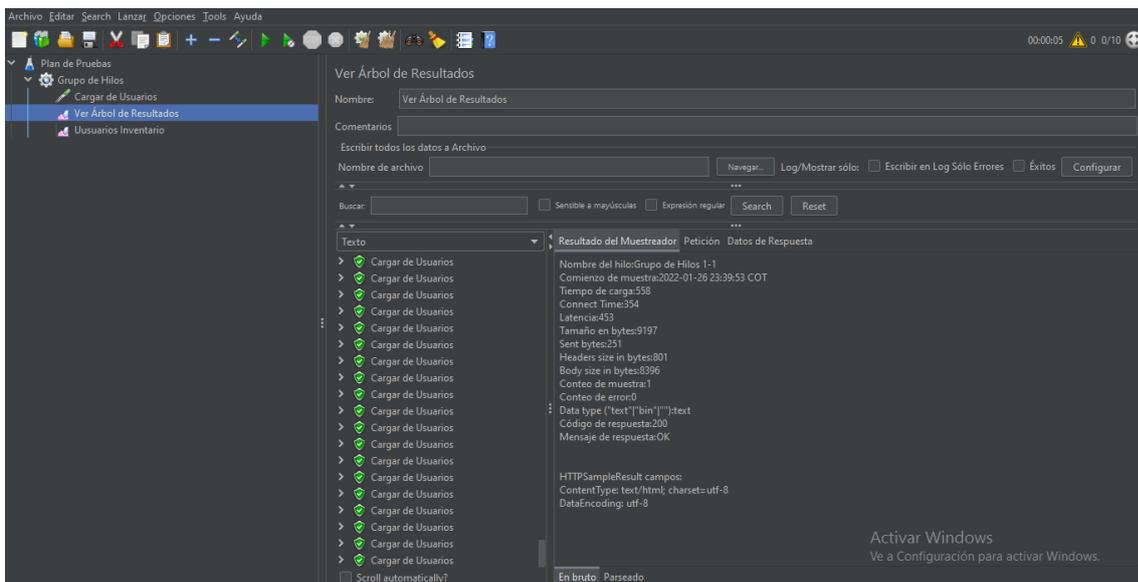
Primera prueba de carga



Nota. Carga de hilos para la simulación para la tabla de usuarios en el inventario con 10 entrenadores y 20 alumnos para cada entrenador. Elaborado por: Jiménez Oscar y Tapia Erick

Figura 27

Resultados primera prueba de carga



Nota. Resultados de cargar para la tabla de usuarios en el inventario con 10 entrenadores y 20 alumnos para cada entrenador. Elaborado por: Jiménez Oscar y Tapia Erick

Figura 28

Tabla de resultados primera prueba de carga

Etiqueta	# Muestras	Media	Min	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Byt...
Cargar de Us...	204	240	193	818	84,30	0,00%	3,7/sec	33,16	0,91	9194,0
Total	204	240	193	818	84,30	0,00%	3,7/sec	33,16	0,91	9194,0

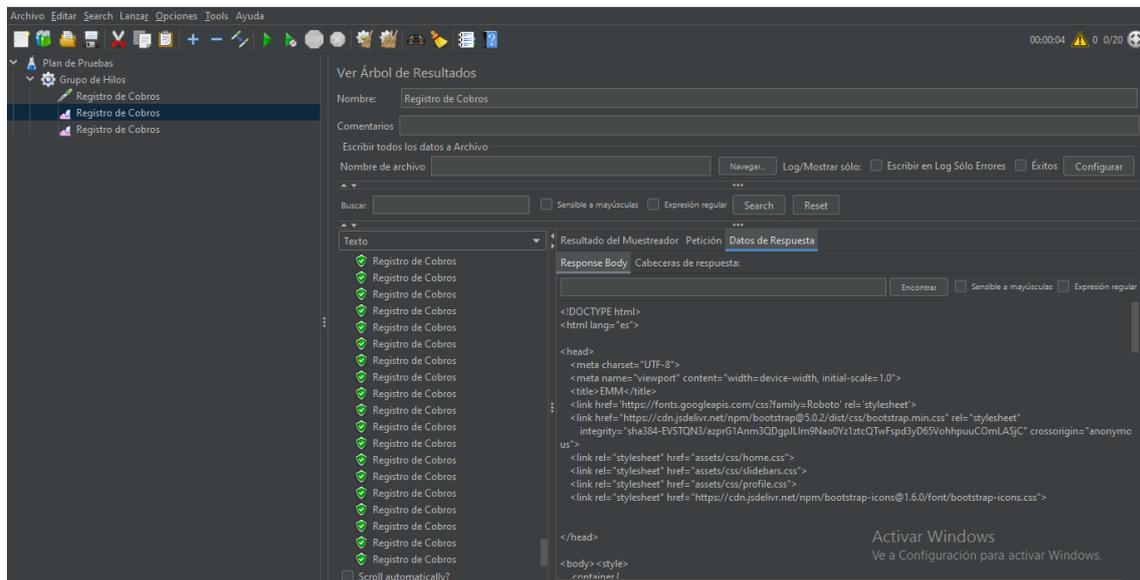
Nota. Tabla de resultados de carga para la tabla de usuarios en el inventario con 10 entrenadores y 20 alumnos para cada entrenador. Elaborado por: Jiménez Oscar y Tapia Erick

2.5.2.2 Registrar cobros de los clientes de la academia en el inventario con 10

entrenadores y 20 alumnos para cada entrenador. Se realizó una prueba de carga para el registro de cobros de los alumnos de la academia. Para ello se simuló un escenario donde, al mismo tiempo, 10 entrenadores registraban 20 cobros en un lapso de 1 segundo. Finalizada la simulación se observó que no hubo ningún fallo al momento de realizar los registros y de igual forma el tiempo que se demoró la aplicación en realizar estos procesos fue de 4 segundos.

Figura 29

Resultados segunda prueba de carga



Nota. Resultado del registro de cobros a clientes de la academia en el inventario con 10 entrenadores y 20 alumnos para cada entrenador. Elaborado por: Jiménez Oscar y Tapia Erick

2.5.2.3 Editar cobros de los clientes de la academia en el inventario con 10 entrenadores y 20 alumnos para cada entrenador. En el caso de la actualización de cobros se realizó una prueba de carga que simuló el escenario donde, al mismo tiempo, 10 entrenadores editaban 20 cobros en un lapso de 1 segundo. Finalizada la simulación se observó que no hubo ningún fallo al momento de realizar los registros y de igual forma el tiempo que se demoró la aplicación en realizar estos procesos fue de 4,1 segundos.

Figura 30

Tabla de resultados tercera prueba de carga

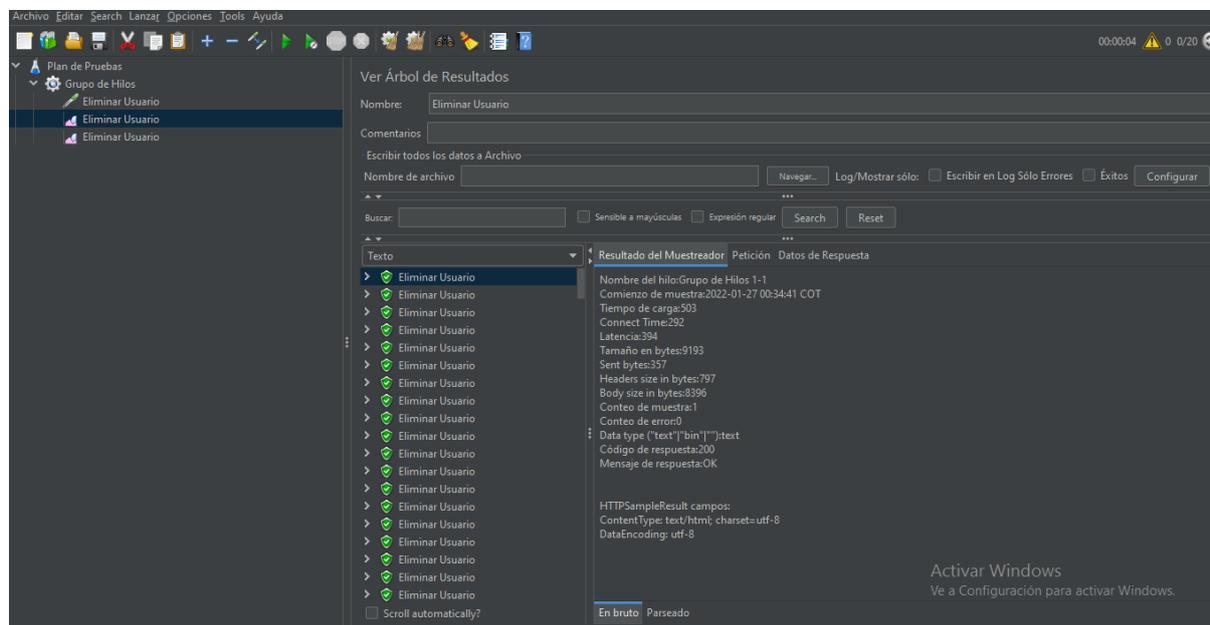
Etiqueta	# Muestras	Media	Min	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Byt...
Editar Cobros	201	356	185	835	204,26	0,00%	4,1/sec	36,74	1,39	9194,5
Total	201	356	185	835	204,26	0,00%	4,1/sec	36,74	1,39	9194,5

Nota. Tabla de resultados de carga para la edición de cobros para los alumnos en el inventario con 10 entrenadores y 20 alumnos para cada entrenador. Elaborado por: Jiménez Oscar y Tapia Erick

2.5.2.4 Eliminar usuarios de la academia con 10 entrenadores y 20 alumnos para cada entrenador. Por último, para simular la carga al momento de eliminar usuarios las condiciones fueron practicante las mismas que se realizaron anteriormente, las cuales fueron que, al mismo tiempo, 10 entrenadores editaban 20 cobros en un lapso de 1 segundo. Finalizada la simulación se observó que no hubo ningún fallo al momento de realizar los registros así mismo el tiempo que se demoró la aplicación en realizar estos procesos fue de 4,1 segundos.

Figura 31

Resultados cuarta prueba de carga



Nota. Tabla de resultados de carga para la edición de cobros para los alumnos en el inventario con 10 entrenadores y 20 alumnos para cada entrenador. Elaborado por: Jiménez Oscar y Tapia Erick

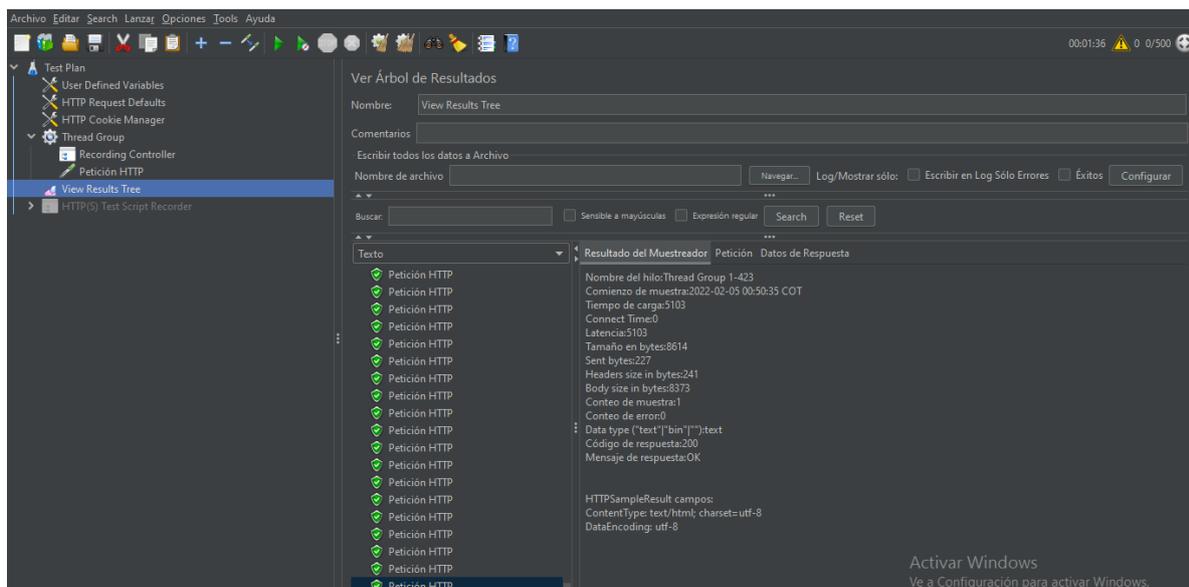
2.5.3 Pruebas de Stress

2.5.3.1 500 usuarios ingresando a la página de inicio usuarios haciendo 10

peticiones en 1 segundo. Para esta prueba se simuló a 500 usuarios intentado entrar 10 veces a la página de inicio, dando como exitosa cada una de las ejecuciones.

Figura 32

Resultados primera prueba de stress

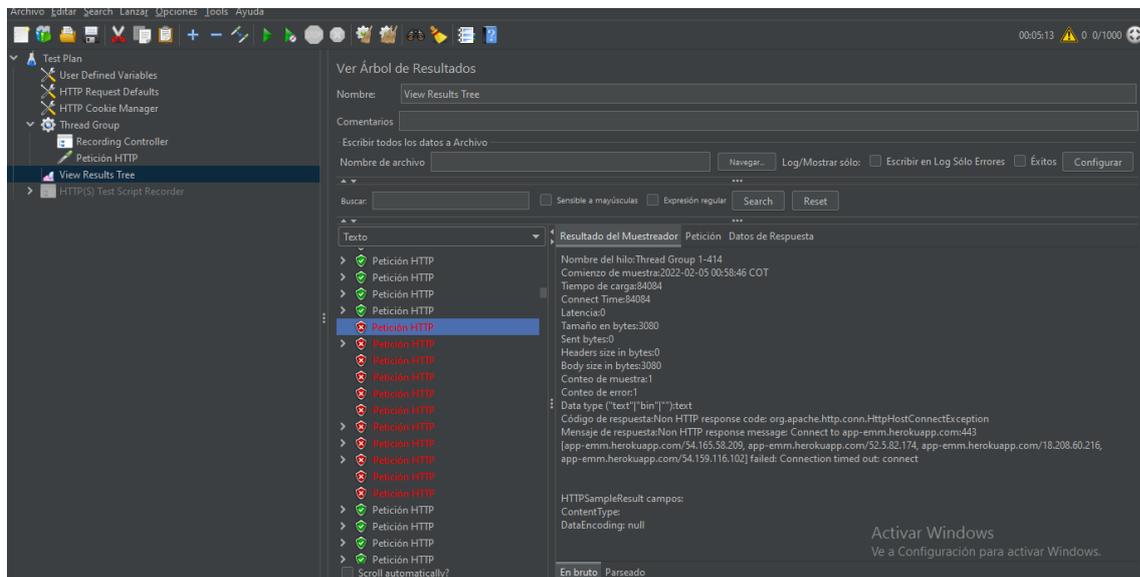


Nota. Resultados de stress para la página de inicio con 500 usuarios. Elaborado por: Jiménez Oscar y Tapia Erick

2.5.3.2 1000 entrenadores realizando 10 peticiones en 1 segundo en la pantalla de inventario. Como la anterior prueba resulto exitosa se decidió generar una simulación, pero esta vez en la pantalla del inventario, esta vez con 1000 entrenadores intentando entrar 10 veces a la página cada uno, todo esto en un lapso de 1 segundo. En este caso hubo fallos al momento de querer ingresar a la aplicación, dejando fuera de servicio a varios entrenadores simulados.

Figura 33

Resultados segunda prueba de stress



Nota. Resultados de stress para la página de inventario con 1000 entrenadores. Elaborado por: Jiménez Oscar y Tapia Erick

2.5.4 Pruebas de diseño e interacción

Para la realización de esta prueba se utilizó una encuesta, la cual tiene preguntas referentes al tema. Este formulario se lo entrego a 20 personas para que las puedan responder, según su criterio y opinión.

La primera pregunta a los encuestados fue que navegador utilizaron para ejecutar la aplicación web, en este caso un 75% utilizó Chrome y el 25% Mozilla. Esta pregunta se la realizó para analizar las diferentes ideas o posibles problemas que podría presentar cada usuario.

Figura 34

Resultados pregunta 1



Nota. Resultados primera pregunta en la prueba de diseño e interacción. Elaborado por: Jiménez Oscar y Tapia Erick

La segunda pregunta quiso proporcionar información por si algún usuario tuvo problemas al registrarse en la aplicación, tanto los usuarios de Mozilla como de Chrome respondieron que no.

Figura 35

Resultados pregunta 2



Nota. Resultados segunda pregunta en la prueba de diseño e interacción. Elaborado por: Jiménez Oscar y Tapia Erick

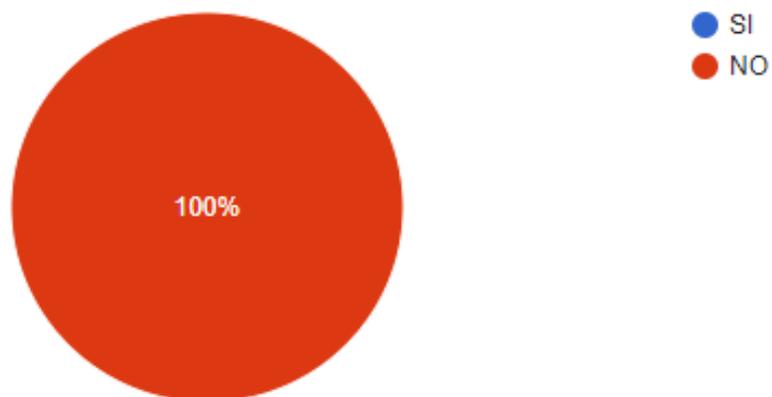
Seguido del registro, es necesario saber si el inicio de sesión fue exitoso, y para todos los casos si lo fue, dando un 100% de satisfacción en el usuario.

Figura 36

Resultados pregunta 3

¿Tuvo problemas cuando inicio sesión?

20 respuestas



Nota. Resultados tercera pregunta en la prueba de diseño e interacción. Elaborado por: Jiménez Oscar y Tapia Erick

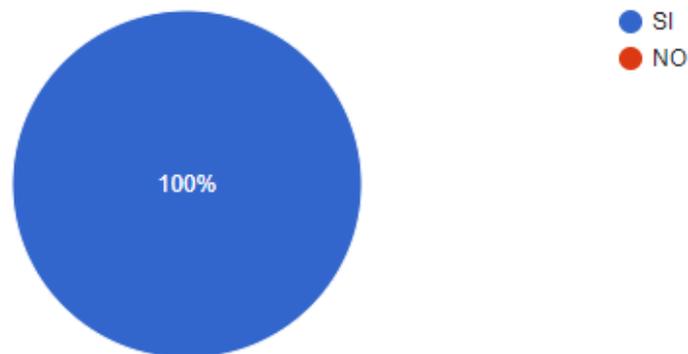
Se pidió de antemano editar algún dato de su perfil, por ende, esta pregunta va relacionada a este aspecto. Como se puede observar no hubo ningún caso en el cual se pierdan datos y no se actualice la información proporcionada por parte del usuario final.

Figura 37

Resultados pregunta 4

Al momento de guardar un dato y refrescar la pantalla, ¿el dato se guardó con éxito?

20 respuestas



Nota. Resultados cuarta pregunta en la prueba de diseño e interacción. Elaborado por: Jiménez Oscar y Tapia Erick

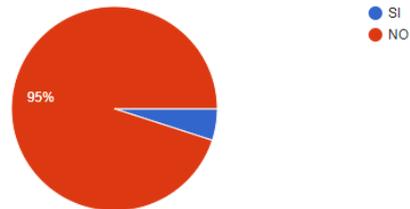
A diferencia de las demás respuestas, en esta hubo pequeños inconvenientes para algunos usuarios ya que el 5% tuvo problemas en algún despliegue, ya sea de algún gráfico, botón o palabra que existe en la aplicación. Por otro lado, es alentador ver que el 95% de los usuarios no tuvieron ninguna dificultad en la visualización.

Figura 38

Resultados pregunta 5

¿Ha tenido problemas al momento de seleccionar o visualizar, palabras, botones o gráficos dentro de la aplicación?

20 respuestas



Nota. Resultados quinta pregunta en la prueba de diseño e interacción. Elaborado por: Jiménez Oscar y Tapia Erick

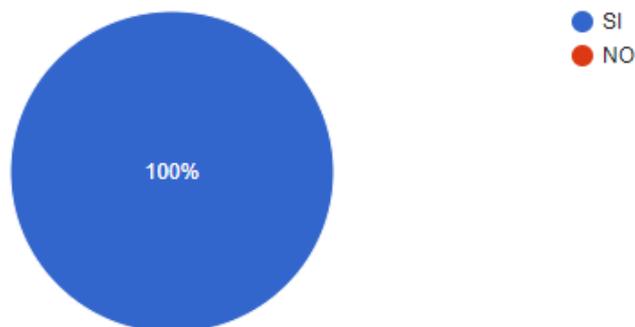
Otra de las preguntas fue enfocada a la parte de los mensajes que la aplicación muestra al usuario. Se puede observar que son de gran ayuda, ya que todos que probaron la aplicación dieron su visto bueno en este apartado.

Figura 39

Resultados pregunta 6

¿Los mensajes proporcionados por la aplicación fueron de ayuda para la interacción de usted con el programa?

20 respuestas



Nota. Resultados sexta pregunta en la prueba de diseño e interacción. Elaborado por: Jiménez Oscar y Tapia Erick

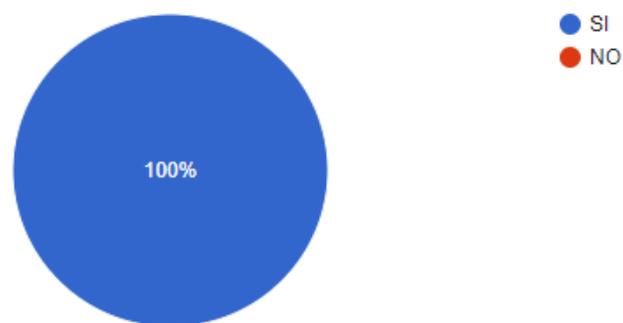
De igual forma se preguntó si hubo algún problema cuando querían regresar a la pantalla anterior, y como en el caso anterior ninguno tuvo inconvenientes en regresar de página.

Figura 40

Resultados pregunta 7

¿Pudo volver siempre atrás, cuando ingresaba a una página que no quería entrar?

20 respuestas



Nota. Resultados séptima pregunta en la prueba de diseño e interacción. Elaborado por:

Jiménez Oscar y Tapia Erick

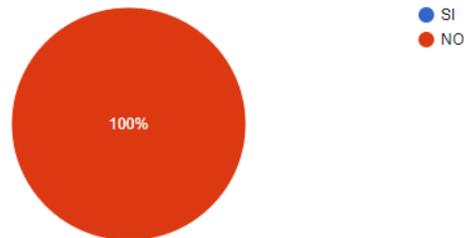
Anteriormente se realizaron pruebas de carga y de stress para la aplicación dando buenos resultados. Aun así, se quiso preguntar nuevamente este aspecto a los usuarios, simplemente para corroborar la carga de la aplicación web, solo que esta vez la prueba sería realizada en los distintos dispositivos que los usuarios usaron para abrir el programa.

Figura 41

Resultados pregunta 8

¿Esperó demasiado tiempo en que se cargue alguna pantalla?

20 respuestas



Nota. Resultados octava pregunta en la prueba de diseño e interacción. Elaborado por: Jiménez Oscar y Tapia Erick

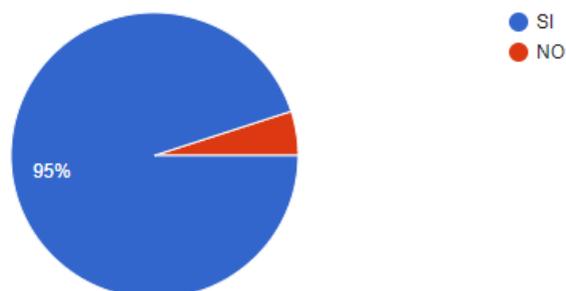
Esta pregunta es un tanto subjetiva, pero viene bien para saber que tal les pareció los colores que se utilizaron para la aplicación. Como se observa el 95% aprobó la mezcla de colores y solo un 5% dijo que no.

Figura 42

Resultados pregunta 9

¿La mezcla de colores de la aplicación fue de su agrado?

20 respuestas



Nota. Resultados novena pregunta en la prueba de diseño e interacción. Elaborado por: Jiménez Oscar y Tapia Erick

Al momento de que los usuarios utilicen las funcionalidades que tiene el inventario, el 45% dijo que la interacción con las tablas fue buena, y el 55% dijo que muy buena. Se cree que esto se debe ya que el usuario aún no está acostumbrado a realizar este tipo de transacciones de datos, y por eso quizás algunos tuvieron ciertos enredos, aun así, la mayoría de personas encuestadas tuvieron una experiencia con la aplicación sumamente satisfactoria.

Figura 43

Resultados pregunta 10



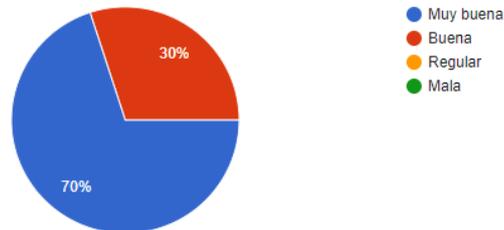
Nota. Resultados decima pregunta en la prueba de diseño e interacción. Elaborado por: Jiménez Oscar y Tapia Erick

En cuanto a la organización de la aplicación web, el 70% de los usuarios votaron como muy buena y un 30% como buena. Esto da a entender que la información que se muestra en el programa es entendible para el usuario que la utiliza, pero aún tiene índices de mejora, los cuales viene bien saber para próximas versiones.

Figura 44

Resultados pregunta 11

¿Qué tal le pareció la organización que presentaba la aplicación, en cuanto a botones, títulos, cajas de texto, etc.?
20 respuestas



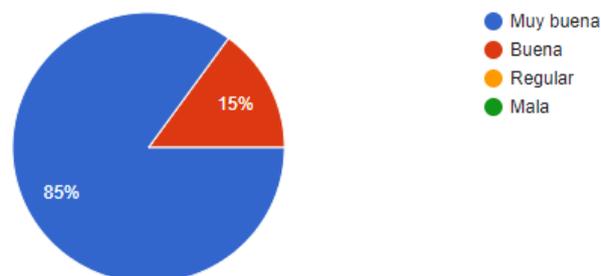
Nota. Resultados onceava pregunta en la prueba de diseño e interacción. Elaborado por: Jiménez Oscar y Tapia Erick

Esta pregunta resultó clave para conocer qué tan indispensable es seguir escribiendo una palabra en concreto en algún botón para que el usuario lo entienda. Como se puede ver, el 85% de las personas encuestadas para este trabajo, dijeron que gustan y prefieren los íconos predeterminados en los botones, y un 15% les agrada la idea, pero aún creen necesario recibir información mediante texto.

Figura 45

Resultados pregunta 12

¿El usar íconos en vez de letras para las acciones de los botones le parece una idea?
20 respuestas



Nota. Resultados doceava pregunta en la prueba de diseño e interacción. Elaborado por: Jiménez Oscar y Tapia Erick

Uno de los aspectos necesarios para saber si el diseño gusta a la vista de las personas es conocer si, según sus ideologías, la aplicación está entre los estándares que rigen a las páginas web actuales. En este caso, a todas las personas les pareció una aplicación actualizada, dejando a entender que, entre las demás aplicaciones que existen en el mercado, esta cumple con el criterio de los usuarios.

Figura 46

Resultados pregunta 13



Nota. Resultados treceava pregunta en la prueba de diseño e interacción. Elaborado por: Jiménez Oscar y Tapia Erick

Una vez terminado la ejecución, se les pregunto si hubo algún tipo de problema en cerrar la sesión de la aplicación, dando como resultado que ninguna persona tuvo inconvenientes en este proceso.

Figura 47

Resultados pregunta 14



Nota. Resultados catorceava pregunta en la prueba de diseño e interacción. Elaborado por: Jiménez Oscar y Tapia Erick

Para finalizar se realizó una simple pregunta, la cual fue si después de haber usado la aplicación, recomendarían la misma. A pesar de desacuerdos en ciertas partes, todos los que utilizaron el programa seleccionaron la opción “SI”, por lo que existen aún mejoras, pero en si la aplicación para el registro de cobros estaría lista para su funcionamiento en el mundo real.

Figura 48

Resultados pregunta 15



Nota. Resultados quinceava pregunta en la prueba de diseño e interacción. Elaborado por: Jiménez Oscar y Tapia Erick

CAPÍTULO 3

3. RESULTADOS

3.1 Análisis de pruebas caja negra

Todos los requerimientos que se propusieron a un principio fueron cumplidos de la mejor manera, con sus respectivas pantallas y mensajes. Esto facilita el uso de la aplicación ya que cumple con lo que la academia desea, que en este caso es gestionar los datos de todos los clientes que están inscritos para los entrenamientos, y no solo eso, sino que esta aplicación sirve para seguir registrando nuevos usuarios, editarlos, o a su vez eliminarlos si es necesario.

En el caso principal que es el registro de cobros, la aplicación de igual manera cumple con lo propuesto ya que se registran los pagos de todos los estudiantes en sus respectivas cuentas, y lo pueden visualizar desplegando la aplicación en el navegador web Chrome o Mozilla, donde constata la actualización de sus pagos. También existe la corroboración del mismo vía correo electrónico ya que cada vez que se realiza un nuevo pago se envía automáticamente un mensaje al estudiante comunicándole que la cancelación fue realizada con éxito.

3.2 Análisis pruebas de carga

Las pruebas fueron realizadas en las pantallas donde los entrenadores estarna más tiempo y realizaran peticiones al servidor de forma simultánea prácticamente cada día. Estas pruebas simularon peticiones al servidor de 10 entrenadores los cuales registran, editan, buscan o eliminar 20 alumnos, todo esto de forma simultánea, lo que quiere decir que se realizaron 200 peticiones al servidor en el lapso de un segundo, finalizando la ejecución con éxito y sin pérdida de datos. Esto permite ver que el tener una aplicación para gestiona datos es sumamente factible y no provoca tiempos de espera largos o molestos, y que puede ser utilizada sin interrupciones de otros usuarios al estar dentro del sistema al mismo tiempo. Así

mismo la aplicación puede ser utilizada tanto en el navegador Chrome como Mozilla sin problema alguno, ajustándose a las necesidades de la empresa.

Es de suma importancia aclarar que estas pruebas fueron realizadas con el despliegue de la aplicación en Heroku, una plataforma gratuita para el alojamiento de páginas web, dando a entender que, si la academia decide contratar un servidor de pago mensual para alojar la aplicación web, la ejecución del programa tendrá mejores resultados de respuesta ya que un servicio pagado es sumamente mejor que uno gratuito, por el simple hecho de tener mejores características, optimas, precisas y adecuadas para saciar con las necesidades que el cliente tenga.

3.3 Análisis pruebas de stress

Las pruebas fueron dos, las cuales sirvieron como eje para demostrar que tan potente es el servidor, en la primera ocasión fueron simulados 500 usuarios intentado entrar al inicio de la aplicación, siendo todos satisfechos con su petición. Por otro lado, en la pantalla del inventario, se simuló a 1000 entrenadores intentado ingresar 10 veces a la página en un lapso de 1 segundo, siendo este el punto de quiebre de la aplicación ya que hubo errores para la gran mayoría de ejecuciones, lo que significa que la página tiene un límite de 1000 usuarios. Esto se entiende ya que la base de datos es remota y a su vez gratuita, así como el servidor donde se encuentra alojada la aplicación, por ende, se tiene resultados utilizando este tipo de plataformas, aun así, los resultados no son desfavorables, ya que todo esto fue simulado en la situación de que se utilice la aplicación diez mil veces en solo un segundo. Son diez mil ya que como se explicó anteriormente, en cada una de ellas, un solo usuario intentaba entrar diez veces, como fueron mil usuarios los simulados entonces se multiplica 1000 usuarios por los 10 intentos de cada uno, todo en el mismo segundo. Con esto se finalizó las pruebas de stress y se pudo concluir lo ya dicho.

3.4 Análisis pruebas de diseño e implementación

Esta prueba se la realizó por medio de un formulario, el cual constaba con 15 preguntas de selección múltiple. El resultado fue, de forma general, bueno para el agrado de los usuarios, señalando que tanto la mezcla de colores, como la organización que presentaba la aplicación al momento de entregar información, era aceptada firmemente por las personas que probaron el software. Así mismo, en la parte de interacción, no hubo mayores dificultades, ya que la muestra de tablas, botones y mensajes, sirvieron de gran ayuda para que los usuarios finales puedan interactuar de manera precisa con el programa. Cabe reconocer que aún existen ciertos desacuerdos e imprecisiones en la parte del diseño, más que nada en las tablas, imprecisiones que no son problemas mayores pero que si se mejoran, la experiencia del usuario con la aplicación será aún mejor, por lo que se tomará en cuenta para próximas versiones.

CRONOGRAMA

Aquí se presentan todas las actividades a realizar en el tiempo establecido, para culminar con el proceso de elaboración del proyecto.

Tabla 22

Cronograma de actividades a realizar

Nombre Tarea	Duración	Fecha Inicio	Fecha Fin
FASE 1			
Recopilación de requerimientos para el análisis de funcionalidades del sistema.	2 horas	11/12/2021	11/12/2021

Modelamiento del sistema con el uso de tres diagramas principales	6 horas	13/12/2021	14/12/2021
FASE 2			
Levantamiento del servidor NodeJs	2 horas	15/12/2021	16/12/2021
Creación pantalla de Inicio de la aplicación web	20 horas	17/12/2021	19/12/2021
Creación pantalla de registro para usuarios de la aplicación web	20 horas	20/12/2021	22/12/2021
Conexión a la base de datos Mongo DB	3 horas	23/12/2021	26/12/2021
Creación Página principal del usuario	20 horas	27/12/2021	29/12/2021
Implementación de funcionalidades CRUD	64 horas	03/01/2022	10/01/2022
Creación del inventario de	20 horas	11/01/2022	14/01/2022

usuarios por cada entrenador			
Implementación del registro de cobros de los clientes por cada entrenador	35 horas	15/01/2022	20/01/2022
Creación del buscador de clientes por cada entrenador	4 horas	21/01/2022	23/01/2022
FASE 3			
Implementación de mensajes más relevantes para informar al usuario sobre las acciones que realiza	3 horas	24/01/2022	25/01/2022
Diseño y ajuste de iconos y colores de la aplicación web	3 horas	25/01/2022	26/01/2022
FASE 4			
Pruebas en los dos navegadores Mozilla y Chrome	10 horas	26/01/2022	31/01/2022

Mejoras	3 horas	27/01/2022	01/02/2022
---------	---------	------------	------------

Nota. Fechas dispuestas para la realización del proyecto. Elaborado por: Jiménez Oscar y

Tapia Erick

CONCLUSIONES

El proyecto técnico realizado culminó con la creación de la aplicación web, la cual ayuda a gestionar la información de todo el personal de manera digital, ya que antes, este proceso se lo hacía manualmente. Esto ha de ocasionar que se deje a un lado, en su totalidad, el uso de la ficha, la cual se utilizaba para el registro de datos personales, pagos e inscripciones en la academia.

Con respecto al registro de cobros, se pudo observar que la accesibilidad a los datos para 500 usuarios es exitosa en el 100%, lo que antes, para poder visualizar la ficha se demoraba un día, hasta que el estudiante la entregue; siempre y cuando esta no se la olvidara, ahora, ingresando a la aplicación la respuesta es de 0.00813 segundos aproximado por cada entrenador que utilice la aplicación, ayudando de esta manera a la disponibilidad de datos.

En lo que se refiere a un software para la gestión de datos mediante el uso de un inventario, se observó que la aplicación web cumple con las acciones de crear, leer, editar y eliminar información que se desee, permitiendo realizar esto a 500 usuarios a la vez, sin ninguna interrupción, lo cual da a entender que, tener una aplicación que permita este tipo trabajo es factible para cualquier organización que lo necesite implementar, contribuyendo así con el uso de la tecnología en las pequeñas y grandes empresas del Ecuador, digitalizando los datos y acercándose cada vez más a la innovación.

Para finalizar, el intercambio de una ficha entre personas ya no es necesario, al ser un programa personal, cada uno revisa su perfil en su propia computadora o celular, bloqueando así una posible vía de transmisión y contagio por el nuevo virus COVID-19.

RECOMENDACIONES

Para un incremento del rendimiento en la aplicación, es apropiado contratar un servidor aparte, tanto para el alojamiento del sitio web como de la base de datos, y verificar su mejora realizando pruebas de stress y carga.

Sería de gran utilidad enfocarse profundamente en la base de datos, creando derivaciones con respecto a las colecciones, separando ciertos datos a un conjunto similar entre ellos, optimizando de esta forma las búsquedas que se realiza por medio del servidor.

Crear un proceso aparte para resguardar la información que se genera en la base de datos, teniendo así un módulo de restauración, por si en algún momento se pierden datos y se deseen recuperar, o a su vez, tener antecedentes de pagos de los usuarios que hayan estado inscritos en la academia.

Realizar mejoras a nivel de diseño creando, logos, imágenes, íconos, o a su vez dedicarse a la parte gráfica de acciones como son, botones, tarjetas de información, formularios, entre otros aspectos. Aportando aún más a la armonía que se tiene entre las funcionalidades internas del programa con la vista que se le presenta al usuario final, mejorando así la experiencia del software con las personas.

REFERENCIAS BIBLIOGRÁFICAS

- Camille Cobb, S. S. (2017). Computer Security for Data Collection Technologies. *Development Engineering*, 12. <https://doi.org/10.1016/j.deveng.2017.12.002>
- Cardoso, R. (2021). Evaluación constructiva de la tecnología. *Dialnet*, <https://doi.org/10.19053/01211129.v30.n55.2021.12459>.
- Domazet, I. a. (2016). New technologies aimed at improving the competitiveness of companies in the services sector. *CEMAFI International*, 1. <http://ebooks.iien.bg.ac.rs/id/eprint/1036>. COBISS.ID=512399458
- Edward Haro, T. G. (2018). Desarrollo backend para aplicaciones web, Serciocios web Restful: Node.js vs Spring Boot. *Revista Iibérica de Sistemas e Tecnologías de Informacao*, 14. <https://creativecommons.org/licenses/by-nc-nd/4.0>, Lousada N.º E17.
- García, F. (2020). Efficient distance join query processing in distributed. *sciencedirect*, 2. <https://doi.org/10.14778/3415478.3415570>
- Hanin M. Abdullah, A. M. (2014). *Frontend and Backend Web Technologies in Social Networking Sites: Facebook as*. Dammam: 3rd International Conference on Advanced Computer Science Applications and Technologies.
- Jimmy Jhonon Diaz Ortiz, M. A. (2017). *DESARROLLO E IMPLEMENTACIÓN DE UN APLICATIVO WEB, UTILIZANDO LA METODOLOGÍA SCRUM, PARA MEJORAR EL PROCESO DE ATENCIÓN AL CLIENTE EN LA EMPRESA Z ADITIVOS S.A.* Lima: Universidad Autónoma de Perú, doi: 10.1109/ACSAT.2014.22.
- jQuery. (Noviembre de 2021). *jquery*. Obtenido de <https://jquery.com>
- Julia Stoyanovich, B. H. (2020). Responsible data management. *VLDB Endowment*, 2.

- Ken Schwaber, J. S. (Noviembre de 2020). *scrumguides*. Obtenido de <https://scrumguides.org>: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-European.pdf>
- Kohl, W. (2013/2014). *Backend As A Service*. Forschungsstarke: Institute of Distributed Systems.
- Mongo DB. (22 de Septiembre de 2015). *Mongo DB*. Obtenido de www.docs.mongodb.org: <http://docs.mongodb.org/manual/core/crud-introduction>
- Moreno Arboleda Francisco Javier, Q. R. (2016). Una comparación de rendimiento entre Oracle y Mongo DB. *Ciencia y Tecnología Neogranadina*, 21.
- Muhammad Yahya, H. W. (2020). Developing School Information Program: Integrated. *IJAE*, 7. <https://doi.org/10.46966/ijae.v1i3.70>
- Navarrete, T. (2006). *El lenguaje JavaScript*.
- Nodemailer. (Diciembre de 2021). *nodemailer*. Obtenido de <https://nodemailer.com>
- Onofa. (2017). *Factores asociados a las exportaciones de las pequeñas y medianas empresas-pymes*. H. Jácome & K. King.
- SpryMedia. (Diciembre de 2021). *Cloud Tables*. Obtenido de <https://datatables.net/>
- Villegas, L. M. (2018). *Análisis y mejora de los procesos en la gestión de cobranzas*. Guayaquil: Universidad de Guayaquil, <http://repositorio.ug.edu.ec/handle/redug/36902>
- Yance Carvajal, S. G. (2017). La importancia de las pymes en el Ecuador. *Observatorio de la Economía*.