

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

DESARROLLO DE UN SISTEMA WEB DE AGENDAMIENTO DE CITAS PARA LA PELUQUERIA UNISEX “SEBITAS”

DESARROLLO DE UN SISTEMA WEB

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR
EN DESARROLLO DE SOFTWARE**

NATALY MARIUXI GUALLICHICO PAUCAR

DIRECTORA: ALVAREZ JIMENEZ MAYRA ISABEL

DMQ, febrero 2023

CERTIFICACIONES

Yo, Nataly Mariuxi Guallichico Paucar declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



NATALY MARIUXI GUALLICHICO PAUCAR

nataly.guallichico@epn.edu.ec

natilu.naty@gmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por Nataly Mariuxi Guallichico Paucar, bajo mi supervisión.



Ing. Mayra Alvarez, MSc.

DIRECTORA

mayra.alvarez@epn.edu.ec

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

Nataly Mariuxi Guallichico Paucar

DEDICATORIA

Dedico el presente trabajo de integración curricular a Dios, por brindarme sabiduría y fuerza con las que he contado para finalizar esta etapa, a mi madre Patricia Paucar por el apoyo incondicional ofrecido durante toda su vida, por enseñarme a ser fuerte y perseverante, por animarme con cada palabra a culminar mi carrera, a mi esposo Kevin Gramal y a mi hija Christina Gramal por su incondicional amor, por la confianza, por las tantas noches que pensaba que no lo lograría, por estar presentes para ayudarme a levantar, por sus palabras de aliento y su confianza, por enseñarme que cada día se puede ser mejor.

Nataly Mariuxi Guallichico Paucar

AGRADECIMIENTO

A Dios por que sin él no lo hubiese logrado, nuevamente a mi madre, a mi esposo, a mi hija por su apoyo incondicional, sus palabras de aliento, sus consejos, por no dejarme caer en este camino. Agradezco a mis padrinos Rene Paucar y Maribel Moposita, quienes siempre han estado presentes demostrando su amor y preocupación por mí, incentivándome a luchar por mis sueños y a cumplir mis metas, a mis hermanos Verónica, Mishell, Jhonny, Mateo, Milena, especialmente a mi hermano Giovanni por estar pendiente de mí, principalmente cuando empecé mis estudios en la EPN, a mi abuelita Fabiola por el cariño, por sus palabras, por apoyarme en el transcurso de mis estudios, a los Ings Juan Pablo Zaldumbide, Byron Loarte, Ivonne Maldonado por sus enseñanzas, sus palabras de aliento y la confianza brindada, a la Ing. Mayrita por su dedicación para que el presente proyecto sea finalizado con éxito, a mis suegros Sr. Cesar Gramal y Sra. Marisol Casa por sus palabras de aliento y por el cariño incondicional hacia mi pequeña, a mis compañeros que han hecho que el transcurso de la carrera sea más amena.

Nataly Mariuxi Guallichico Paucar

ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN	VII
ABSTRACT	VIII
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO	1
1.1 Objetivo general.....	2
1.2 Objetivos específicos	2
1.3 Alcance	2
1.4 Marco Teórico	3
2 METODOLOGÍA	9
2.1 Metodología de Desarrollo	9
<i>Roles</i>	9
<i>Product Owner</i>	9
<i>Scrum Master</i>	10
<i>Development Team</i>	10
<i>Artefactos</i>	10
<i>Recopilación de Requerimientos</i>	11
<i>Historias de Usuario</i>	11
2.2 Diseño de interfaces (mockups).....	13
<i>Figma</i>	13
<i>Sistema Web</i>	13
2.3 Diseño de la arquitectura	14
<i>Patrón arquitectónico</i>	14
2.4 Herramientas de desarrollo	16
2.5 Librerías	17
3 RESULTADOS.....	18
3.1 <i>Sprint 0</i> . Configuración del ambiente de desarrollo	18
3.2 <i>Sprint 1</i> . Autenticación	22
3.3 <i>Sprint 2</i> . Módulo de Información.	31

3.4 <i>Sprint</i> 3. Módulo Complementario.....	37
3.5 <i>Sprint</i> 4. Pruebas y despliegue del sistema.	42
4 CONCLUSIONES	45
5 RECOMENDACIONES.....	46
6 REFERENCIAS BIBLIOGRÁFICAS.....	47
7 ANEXOS.....	50
ANEXO I.....	51
ANEXO II.....	52
ANEXO III.....	81
ANEXO IV	82

RESUMEN

En la actualidad, existen varios salones de belleza que ofrecen sus servicios acorde a necesidades de los clientes. Estos servicios en la gran mayoría no tienen mayor publicidad que la que colocan en sus locales para atraer clientes del sector. Sin embargo, dado los avances tecnológicos y el uso del internet, las personas buscan información y servicios de manera rápida y eficaz haciendo uso del internet. La peluquería Unisex “Sebitas”, ha enfrentado ciertos inconvenientes donde los clientes han expresado su malestar con respecto al manejo de la atención al cliente debido a la espera por los servicios que ofrece el establecimiento. En el siguiente proyecto de titulación se plantea desarrollar un sistema web para la gestión de citas, horarios de atención y publicitar los servicios que ofrece la peluquería Unisex “Sebitas”. Para ello, se manejan dos perfiles: perfil administrador encargado de gestionar el sistema web y perfil usuario.

El presente trabajo de integración curricular tiene la siguiente estructura: en la primera sección se detallan los antecedentes, objetivos, alcance y marco teórico del proyecto. En la segunda sección se menciona la metodología ágil SCRUM, que fue la implementada en este proyecto, además, de prototipos y las herramientas que han sido utilizadas para desarrollar el sistema web. En la tercera sección se presentan las historias de usuario, actividades realizadas en cada iteración y resultados obtenidos de cada Sprint con sus respectivas pruebas unitarias. Finalmente, en la cuarta sección se mencionan las conclusiones y recomendaciones obtenidas en base al desarrollo del sistema web propuesto.

PALABRAS CLAVE: *Sistema web, peluquería, SCRUM, sprint, citas.*

ABSTRACT

Currently, there are several beauty salons that offer their services according to the needs of customers. These services in the vast majority do not have more publicity than the one they place in their premises to attract customers from the sector. However, given technological advances and the use of the Internet, people seek information and services quickly and efficiently using the Internet. The Unisex hairdresser "Sebitas" has faced certain inconveniences where customers have expressed their discomfort regarding the handling of customer service due to waiting for the services offered by the establishment. In the following degree project, it is proposed to develop a web system for managing appointments, opening hours and advertising the services offered by the Unisex hairdresser "Sebitas". For this, two profiles are managed: administrator profile in charge of managing the web system and user profile.

This curricular integration work has the following structure: the first section details the background, objectives, scope and theoretical framework of the project. The second section mentions the agile SCRUM methodology, which was implemented in this project, as well as prototypes and the tools that have been used to develop the web system. The third section presents the user stories, activities carried out in each iteration and results obtained from each Sprint with their respective unit tests. Finally, in the fourth section the conclusions and recommendations obtained based on the development of the proposed web system are mentioned.

KEYWORDS: *Web system, hairdressing, Scrum, Sprint, quotes.*

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

Actualmente, es preciso que la gestión de citas de los negocios que requieren contacto físico tenga presencia en internet sin importar el tamaño del negocio, si es un negocio familiar, una franquicia o una gran firma y así captar la atención de un mayor número de usuarios. Un sitio *web* bien diseñado y accesible da una mayor probabilidad de prosperidad al negocio independientemente del servicio o producto que brinde [1]. Por tanto, la implementación de sitios *web* permite al usuario visualizar los servicios y horarios de atención las 24 horas del día, además de agendar citas sin necesidad de ir al local de forma física, sino únicamente al momento de recibir el servicio previa cita, disminuyendo así el tiempo de espera en la atención y la aglomeración innecesaria de usuarios.

Existen varios salones de belleza que se han acoplado a la prestación de servicios utilizando medios digitales y técnicas como *marketing*, haciendo de esta una herramienta eficaz y los costos son más bajos que los métodos tradicionales [1], dentro este contexto implementar un sistema *web* que oferte servicios de salones de belleza es una manera de llegar clientes potenciales. Sin embargo, una cita mal gestionada puede ocasionar malestar al cliente, si la atención no es oportuna podría involucrar la pérdida del cliente, por tanto, es importante considerar varios parámetros como accesibilidad, navegabilidad, validaciones, entre otros para el desarrollo del sitio *web*.

En la parroquia de Conocoto se pueden encontrar varios salones de belleza que ofrecen sus servicios con pancartas y folletos, pero no cuentan con una herramienta digital donde el cliente pueda visualizar los servicios ofertados, precios y horarios de atención al público. Cabe mencionar que una página *web* que oferte servicios de peluquería sirve como punto de referencia para que clientes potenciales puedan encontrar el negocio en Internet independientemente de la ubicación geográfica. Además, es posible ofrecer la función de una tarjeta de visita, catálogo de los servicios e información en general [2].

La peluquería Unisex “Sebitas”, fundada en el año 2019 ha ido posicionándose en el mercado de manera continua, logrando así tener un alcance significativo dentro del área de la belleza en la parroquia de Conocoto. Sin embargo, no cuenta con un sistema *web* para poder gestionar las citas de manera oportuna y eficaz que ofrezca un mejor tiempo de servicio a los clientes. Por tanto, se ha visto la necesidad de implementar un sistema de agendamiento de citas, logrando así que el cliente pueda visualizar los servicios ofertados por la peluquería y agendar una cita desde cualquier lugar, a cualquier hora del día, evitando así filas de espera y pérdida de tiempo para ser atendido.

Con la finalidad de eludir puntos anteriormente señalados, el presente Trabajo de Integración Curricular tiene como propósito desarrollar un sistema *web* de agendamiento de citas para la peluquería Unisex “Sebitas”, ofreciendo beneficios tales como: mejor manejo de las citas agendadas, mejor planificación del tiempo de atención, evitar aglomeraciones innecesarias dentro del salón de belleza, mejor visualización de los servicios ofertados lo que dará como resultado un mejor servicio y la fidelización del cliente. Para este propósito, el sistema contara con dos roles el administrador será la persona que gestione el sistema *web* y el cliente la persona a quien estará dirigida los servicios que oferta el establecimiento.

1.1 Objetivo general

Desarrollar un sistema *web* de agendamiento de citas para la peluquería Unisex “Sebitas”.

1.2 Objetivos específicos

1. Identificar los requerimientos del sitio *web* a desarrollar para la peluquería Unisex “Sebitas”.
2. Elaborar el prototipado del sistema *web* tomando como base los requerimientos para la peluquería Unisex “Sebitas”.
3. Desarrollar los módulos para el sistema *web* en base a requerimientos.
4. Verificar el correcto funcionamiento del sistema *web* para su despliegue final.

1.3 Alcance

El sistema *web* será desarrollado con el objetivo que los clientes de la Peluquería Unisex “Sebitas” puedan realizar el agendamiento de citas de los servicios que ofrece el establecimiento. El sistema constará de una página con login y manejará dos perfiles: cliente y administrador. Al perfil de cliente se le permitirá realizar agendamiento de citas acorde a disponibilidad de servicios del establecimiento, mientras que el perfil de administrador será el encargado de gestionar el sistema *web*. Mismo que estará compuesto por los siguientes módulos: registro, inicio de sesión, reestablecer contraseña, servicios que presta la peluquería, información de las citas realizadas. Cabe mencionar, que los módulos mencionados están diseñados en función a los requerimientos que se han obtenido de las personas que trabajan en la peluquería Unisex “Sebitas”.

Por otro lado, se proporcionará veracidad, confianza y consistencia de los datos con la ayuda del *backend* con la finalidad de controlar mejor los datos proporcionados por el *frontend*. Para ello se utilizan herramientas de desarrollo de última generación, considerando su robustez y escalabilidad. Igualmente, se integra una metodología de desarrollo ágil llamada Scrum, un mecanismo de diseño de interfaces, un modelo arquitectónico, una sucesión de pruebas para corroborar la funcionalidad correcta del *backend* y *frontend*, y una fase final para elaborar el despliegue mandándolo así a producción.

Perfiles dentro del sistema web

- Perfil Cliente y Perfil administrador

Usuario Administrador

- Consumo de *endpoints* y uso de interfaces para inicio y cierre de sesión.
- Consumo de *endpoints* y uso de interfaces para gestionar el agendamiento de citas de los servicios ofertados.
- Consumo de *endpoints* y uso de interfaces para la gestión de clientes y citas agendadas.
- y uso de interfaces *endpoints* y uso de interfaces para gestionar el sitio *web* en general

Usuario Cliente

- Consumo de *endpoints* y uso de interfaces para registrarse en el sistema.
- Consumo de *endpoints* y uso de interfaces para iniciar y cerrar sesión, además de restablecimiento de contraseñas.
- Consumo de *endpoints* y uso de interfaces para agendar una cita.
- Consumo de *endpoints* y uso de interfaces para visualizar la información ingresada.
- Consumo de *endpoints* y uso de interfaces para visualizar los servicios.

1.4 Marco Teórico

Desde luego no cabe duda de que estamos en la era digital, y avances tecnológicos tales como *blockchain*, el Internet o Negocios inteligentes se han convertido en factores decisivos de evolución, no solo en el ámbito de los negocios y pequeñas y medianas

empresas (pymes), sino también, en un entorno social y personal que poco a poco, va dejando atrás lo analógico, con todas las consecuencias que ello implica [3].

Los cambios que han sufrido las pymes desde los últimos 30 años hasta la actualidad ha sido considerable, no solo las pymes, el entorno, el mercado, la sociedad en general, el contexto es completamente diferente. En ese entonces solo unos pocos visionarios, lo que ahora llamamos *early adopters*, trabajaban en la implementación de soluciones de las tecnologías de la información (TI), especialmente enfocadas en las pymes, cuando la tecnología era aún una disciplina emergente [3].

En la actualidad la probabilidad que un negocio tenga éxito depende mucho de su presencia en sitios digitales, ya que es ahí donde el cliente potencial podrá dirigir su atención a los distintos servicios ofertados. En base a este criterio el *frontend* es de vital importancia debido a que la interfaz proporcionada al sitio *web* es el primer contacto que se tiene con el cliente y la experiencia de este dependerá de esta.

Un *frontend* dentro del desarrollo *web* es una parte fundamental ya que se dedica a la parte frontal de las páginas *web*, es decir, es un diseño que incluye, partiendo del esqueleto del sitio *web* hasta estilos, colores, efectos, etc. [4]. Una de las principales tareas de un *frontend* es asegurar la compatibilidad entre diferentes navegadores, diferentes sistemas operativos y en diferentes dispositivos, además del acceso y uso del sistema *web* [5].

El diseño del prototipado para generar una experiencia de usuario (UX) es importante ya que permite que la interacción de los componentes del sitio *web* con el usuario sea similar a la que ofrecerá el sistema *web* final. El prototipado permite probar, evaluar y validar si efectivamente la idea general que se tiene del sistema cumple los objetivos y requerimientos del negocio y de los clientes, ideas que inicialmente resulta complejo reflejarlas y llevarlas a un terreno palpable [6].

El desarrollo de prototipos requiere el uso de herramientas como Figma que contribuyan al diseño. Figma es una herramienta *online* para diseñar interfaces que permite el trabajo colaborativo, donde los miembros del equipo o el propio diseñador podrán iniciar sesión y realizar los ajustes necesarios [7].

Una aplicación *web* es un programa que funciona gracias al internet, en otras palabras, los registros o las notas en los que se labora son desarrollados y luego guardados en la *web*. Dichas aplicaciones, comúnmente, no necesitan instalación en el computador [8]. Las páginas *web* tienen piezas que ayudan a comunicarse asertivamente entre los usuarios y la información, permitiendo acceder a datos de forma interactiva, ya que las páginas

responderán a sus interacciones, como llenar y mandar formularios, jugar en distintos juegos y acceder a los distintos administradores de las bases de datos [9].

La implementación de un sistema *web* que lleve a cabo los procesos del negocio es una opción viable si el propósito es tener un orden y gestionarlo de manera fácil y rápida [10]. Una buena opción de poner en marcha un sistema *web* que ayude a agendar citas en una peluquería es a través de internet, ya que los clientes podrán consultar las horas disponibles y reservar la hora que mejor se adapte a sus horarios, esto ayudara a ahorrar horas de trabajo en el teléfono gestionando las citas [11].

La definición del contenido de la página *web* se realiza empleando el lenguaje de marcado de hipertexto (HTML), a través de las etiquetas nos ayuda a definir el texto y demás elementos que componen la página *web*, como por ejemplo imágenes, secciones, listas, enlaces, entre otros [12]. HTML es utilizado con la finalidad de agregar partes de texto y generar una estructura de contenido. Aunque, para poder construir un sitio *web* profesional y receptivo no es suficiente, por tal razón, HTML que esta basado en hojas de estilo en forma de cascada (CSS) y *JavaScript* con el fin de crear contenido del sitio *web* más interactivo.

CSS es el quien crea elementos como los fondos, el espaciado y las animaciones. Por otro lado, *JavaScript* añade funciones dinámicas tales como controles que se deslizan, ventanas desplegadas emergentes, etc. HTML, CSS y *JavaScript* son un pilar fundamental para el desarrollo de un *frontend* en un sistema *web* [13].

JavaScript es un lenguaje de programación que utilizan los sistemas *web*, para incluir interactividad y funciones con el fin de enriquecer la experiencia de usuario. Además, según estudios de W3techs, un índice superior al 90% de los sitios *web* utilizan *JavaScript* para agregar características interactivas al sistema *web* [14].

El lenguaje de procesador de hipertexto (PHP), es un lenguaje de programación que ayuda en la creación de sitios *web*, logrando una conexión favorable entre los usuarios y los servidores *web* [15]. Cabe mencionar que PHP se puede utilizar en cualquier proyecto *web*.

Ahora bien, el mecanismo que permitirá comunicarse entre el cliente y el servidor es la interfaz de programación de aplicaciones (APIs), este conjunto de protocolos y definiciones se utilizara en el desarrollo del *software* a través de algunas reglas tales como la especificación formal de un modelo de software que está en comunicación con otro, con la finalidad de concluir con diversas funciones. Dichas APIs no son visibles a los usuarios,

sino más bien son circuitos internos que podrán ser conectados por los desarrolladores con la finalidad de lograr el funcionamiento de una herramienta [16].

Las pruebas realizadas previo al despliegue del sistema *web* son de vital importancia ya que estas permitirán determinar qué tan eficiente es el sistema desarrollado, considerando también compatibilidad, rendimiento y experiencia de usuario. Así pues, se realicen los *tests* de forma manual o mediante automatización, cuanto más rápido y con mayor frecuencia se puedan ejecutar las pruebas, existe mayor probabilidad de encontrar posibles errores y darles una solución inmediata, garantizando así que la aplicación de *software* se ha examinado y ha sido auditada minuciosamente previo a ser presentado a los usuarios. En caso de que los problemas llegasen a infiltrarse en el ambiente de producción, mayor será el costo y el tiempo para solucionarlos [17].

El despliegue del sistema permitirá que cualquier persona con los recursos y herramientas necesarias pueda enlazarse a la aplicación gracias a una URL, es decir se pasara la aplicación de una maquina local a un servidor externo [18].

Metodología Ágil

En el campo de la programación se utiliza metodologías para el desarrollo de software con el fin de entregar el producto a tiempo. Estas metodologías han evolucionado con el tiempo desde un simple procedimiento organizativo hasta ser un pilar fundamental en el desarrollo de software, siendo así competente y favorable. Trabajar con una metodología dentro del desarrollo de software puede disminuir la dificultad, agilizar las tareas y procesos, además de mejorar los resultados finales de las aplicaciones y tiempos de entrega [19]. La metodología con la que se realiza el presente Trabajo de Integración Curricular es la metodología ágil.

Las metodologías ágiles ayudan a acoplar la forma en la que se trabaja a las solicitudes del proyecto, logrando la flexibilidad y capacidad de resolución con el fin de que el proyecto y la elaboración del mismo se acoplen a estas condiciones delimitadas dentro del ambiente de trabajo [20]. Bajo este concepto *Agile* adopta una manera diferente de trabajo y organización, logrando así que cada uno de los proyectos sean cortadas en reducidas piezas que se completaran y entregaran en unas semanas [21]. Algunas de las ventajas que ofrecen las metodologías ágiles son: La mejora del producto en calidad, el cliente tendrá mayor satisfacción, trabajo colaborativo, utilizaran métricas relevantes, reducir costos [22].

Entre las metodologías ágiles más populares se pueden mencionar *Extreme Programming* (XP), Kanban, *Scrum*, *Agile Inception*, etc. En la TABLA I. se muestra una tabla comparativa entre las distintas metodologías ágiles [23].

TABLA I. Metodologías ágiles – tabla comparativa.

Metodología	XP	SCRUM	Kanban	Agile Inception
Característica	Define el alcance del proyecto mediante historias de usuario. El Cliente ayuda en el desarrollo del proyecto. Retroalimentación continua.	Los proyectos se ejecutan en bloques temporales. Flexibilidad en adoptar cambios. Interacción con cliente. Trabaja por Sprints.	Asignar personas a tarjetas. Incluir listas de control. Etiquetado de tarjetas. Visualización de tareas.	Establece objetivos y fundamentos al iniciar un proyecto. Diferentes perspectivas sobre los asuntos importantes del proyecto.
Procesos	Se asegura que al menos un programador conozca y controle la labor del otro. Técnicas que ofrecen agilidad.	Se realiza en periodos momentáneos y de permanencia estable. El proceso parte de la lista de objetivos. El proceso corresponde a las tareas y a la fluidez de dichas tareas en el sistema.	Tiene algunos preámbulos elementales con la finalidad de lograr una mayor productividad en el trabajo, tales como: visualizar que se hace, limitar la carga de codificación en proceso.	Establece el objetivo del producto que se va a crear.
Herramientas	Historias de usuario, plan de entrega, plan de iteraciones.	Historias de usuario, plan de entrega, plan de iteraciones.	Monday.com Kanbanize, meisterTask, clarizen.	<i>Realtime board.</i>
Ventajas	Desarrollo de proyectos de una manera eficiente y menor tiempo. Segmentación del proyecto en pequeños	Eficiencia en proceso de pruebas y planificación. Programación organizada y	Medición del rendimiento. Organización del flujo de trabajo.	Permite al equipo de trabajo separar el proyecto en subprocesos logrando así centrarse en cada uno de

	bloques mucho más gestionable.	satisfacción del programador. Permite dimensionar mejor los proyectos. <i>Feedbacks</i> rápidos y precisos.	Cumplir con los tiempos de entrega.	manera individual.
--	--------------------------------	---	-------------------------------------	--------------------

Por lo mencionado anteriormente para el desarrollo del presente sistema web se ha tomado la decisión de ocupar la metodología ágil *Scrum*, por sus ventajas, tales como: gestionar la regularización de las aspiraciones del cliente, obtención de resultados previos, amoldamiento y adaptabilidad, atenuación de riesgos[24].

2 METODOLOGÍA

Un estudio de caso es el estudio puntual de un contenido en específico utilizado comúnmente en la investigación educativa, social, tecnológica, entre otras. Un diseño de investigación de un caso de estudio generalmente incluye técnicas tanto cualitativas y cuantitativas. Un estudio de casos se utiliza para explicar, equiparar, diagnosticar y entender diferentes puntos de vista de las interrogantes dentro de investigación, es decir, nos ayuda a mantener un proyecto en la línea de enfoque haciéndolo manejable especialmente cuando se realiza una investigación a una escala mayor [25].

El presente Trabajo de Integración Curricular hace referencia a un caso de estudio, dado que se realizaron entrevistas a los administradores de la peluquería Unisex “Sebitas” y encuestas a sus clientes, logrando así determinar las solicitudes de los usuarios y del administrador conjuntamente con el levantamiento de requerimientos para la creación de un sistema *web* que permita gestionar citas de modo que los clientes puedan programar una cita evitando perder tiempo y en algunos de los casos no ser atendidos debido a la gran afluencia de clientes sin cita en el establecimiento.

2.1 Metodología de Desarrollo

Scrum permite el manejo flexible de proyectos difíciles codificados en un ambiente activo y aleatorio. Se sustenta en respuestas fragmentadas y constantes del producto terminado en función del precio que ofertan a los usuarios. Esta es la elección de gestión ideal de los proyectos desarrollados en un entorno complejo ya que nos permite obtener un resultado rápidamente. *Scrum* nos brinda agilidad en el resultado [26].

Roles

Un *Team Scrum* está compuesto por grupos designados a una tarea en específico de al menos 3-9 integrantes dentro del ambiente de desarrollo. Cada rol tiene distintas responsabilidades y debe dar informes de manera distinta, tanto para los miembros del *Team* como para los demás integrantes de la organización [24]. Los siguientes roles a utilizarse dentro del desarrollo del presente sistema *web* son:

Product Owner

El *product owner* usualmente es el rol ocupado por alguien con un conocimiento amplio de los clientes, competencia y preferencias futuras acorde con el modelo de proyecto que se esté desarrollando [27]. A este rol también se lo conoce como la voz del cliente debido a que no solo comprende las necesidades de la empresa, sino que además conoce también

las necesidades de los usuarios [28]. En la TABLA II, se puede visualizar a la persona que será la encargada de cumplir con ese rol.

Scrum Master

Un *Scrum Máster* es quien evita que el grupo designado para el trabajo se distraiga mientras trabajan para lograr completar el *sprint* y resolver los impedimentos que quieran evitar que el *sprint* alcance su objetivo. Es el encargado de asignar tareas a los equipos de trabajo y además hará un seguimiento de las tareas que ya fueron asignadas [29]. La persona que se hará cargo de desempeñar el rol mencionado se puede visualizar en la TABLA II.

Development Team

Son personas quienes se encargan de desarrollar el proyecto, todos comparten responsabilidades dentro del proyecto y son responsables de la calidad de este. El equipo frecuentemente tiene interacciones con los *stakeholders* con la finalidad de compartir el conocimiento de negocio, alinean prioridades y miden los resultados de lo que se entregara al cliente [30]. En la TABLA II, se puede evidenciar a la persona quien será la encargada de asumir las responsabilidades de este rol.

TABLA II: Designación de Roles.

ROLES	NOMBRES
<i>Product Owner</i>	Lic. Janneth Gualotuña
<i>Scrum Master</i>	Ing. Mayra Alvarez, MSc.
<i>Development Team</i>	Nataly Guallichico

Artefactos

Un artefacto es el resultado obtenido a través de un procedimiento dentro del desarrollo de *software*, está asociado a algunos métodos o procesos dentro del desarrollo que son específicos. En algunas ocasiones se puede decir que un artefacto dentro del desarrollo de *software* hace referencia a un producto finalizado, por ejemplo, el ejecutable del proyecto, pero más hace referencia a la documentación obtenida en el proceso del desarrollo del producto [31]. Además, dentro del trabajo *Scrum* los principales utilizados son: el *Product Backlog*, el *Sprint Backlog* y el Incremento [32].

Recopilación de Requerimientos

Dentro del proceso de recopilación de requerimientos se debe tomar en cuenta la identificación y la documentación de los requisitos que necesitan las partes interesadas, siendo estas los usuarios o los clientes, etc., mismos que estarán ligados con el proyecto. Los resultados obtenidos serán utilizados con la finalidad de desarrollar soluciones dentro del proyecto de *software*. Algunos métodos que se utilizan para la recopilación de requerimientos son la entrevista, lluvia de ideas, debates, cuestionarios, etc. [33]. Por lo mencionado, la TABLA III, nos indica los resultados obtenidos de la recopilación de requerimientos que se realizó a la dueña de la peluquería a través de la encuesta. No obstante, dicha tabla completa se puede visualizar de manera más detallada en el **ANEXO II**, en la sección de recopilación de requerimientos del presente documento.

TABLA III: Recopilación de requerimientos.

RECOPIACIÓN DE REQUERIMIENTOS		
TIPO DEL SISTEMA	ID - RR	ENUNCIADO DEL ÍTEM
SISTEMA WEB	RR001	Como usuario administrador necesita. <ul style="list-style-type: none">• Identificarse dentro del sistema <i>web</i>. Finalidad. Iniciar y cerrar sesión para poder ingresar dentro del sistema <i>web</i> .

Historias de Usuario

Una historia de usuario es utilizada como un artefacto para la comunicación entre la parte verbal y la parte escrita dentro de la ingeniería de los requisitos ágiles, con la finalidad de poder construir una solución a una necesidad o problema. También agilizan la administración de requerimientos, logrando así reducir un sin número de documentos formales y tiempos necesarios [34]. Por otro lado, se realizan las historias de usuario para obtener las funcionalidades dentro del sistema *web*, la

TABLA IV, muestra una historia de usuario obtenida para el desarrollo del sistema *web*. Es bueno recalcar que las demás historias de usuario se pueden visualizar de manera más detallada en el **ANEXO II**, en la sección Historias de Usuario del presente documento.

TABLA IV: Historia de usuario 01 –Inicio y cierre de sesión.

Historia de Usuario	
Identificador: HU001	Usuario: Administrador
Nombre historia: Inicio y cierre de sesión.	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Iteración asignada: 1	
Responsable: Nataly Guallichico	
Descripción: El administrador necesita identificarse dentro del aplicativo con la finalidad de iniciar y cerrar sesión. Los datos del usuario administrador estarán guardados previamente en la base de datos y se otorgara su respectiva cuenta y contraseña.	
Observación: El usuario administrado necesita estar registrado previamente en la base de datos del sistema con la finalidad de iniciar y cerrar sesión.	

Product Backlog

Es una serie de requerimientos dados por el dueño del producto. La lista en mención no está terminada debido a que irá evolucionando de acuerdo al tiempo que se tome para desarrollar el proyecto [35]. En ese sentido, la TABLA V, indica el modelo seleccionado para mostrar el listado del *Product Backlog*, la tabla final se puede ver en el **ANEXO II**, sección Product Backlog del documento actual.

TABLA V: *Product Backlog*.

ELABORACIÓN DEL <i>PRODUCT BACKLOG</i>				
ID – HU	Historia de usuario	Iteración	Estado	Prioridad
HU001	Inicio y cierre de sesión	1	Finalizada	Alta
HU002	Visualizar los servicios.	2	Finalizada	Media

Sprint Backlog

Se denomina *Sprint Backlog* a una nómina por realizar dentro del *sprint*, es decir es un número de cosas que faltan por hacer dentro del *Product Backlog* en el tiempo en el que se realiza el *Sprint Planning* para el actual *sprint*, se realiza el trabajo que se necesita para

un aumento del proyecto [36]. La Tabla VI, se puede visualizar un modelo de las tareas realizadas en el *Sprint Backlog*, la tabla final se puede ver en el **ANEXO II**, sección *Sprint Backlog* del documento actual.

Tabla VI. Modelo primer ítem del *Sprint Backlog*

ID-SB	Nombre	Historia de Usuario	Tareas	Tiempo Estimado (horas)
S0	Configuración del ambiente de desarrollo	N/A	<ul style="list-style-type: none"> • Instalación de paquetes y extensiones necesarios para Visual Studio Code. • Creación de prototipado. • Estructura del proyecto. • Creación de proyecto en GitHub. 	40 h

2.2 Diseño de interfaces (mockups)

Ya identificados los requerimientos, se realiza la elaboración de los prototipados o conocidos también como *mockups* que representará cada interfaz de usuario, mismas que permitirá observar previamente como sería la interfaz gráfica del sistema *web*.

Figma

Como ya se mencionó en el Marco Teórico, Figma es una herramienta que nos ayuda en la creación de interfaces amigables con el cliente ofreciendo un prototipado semejante a lo que será el sistema *web*.

Sistema Web

En la Fig. 1, se puede visualizar un modelo de los *mockups* realizados para el sistema *web*. En el **ANEXO II**, en la sección Diseño de Prototipos, se puede visualizar el enlace y los demás *mockups* del sitio *web*.

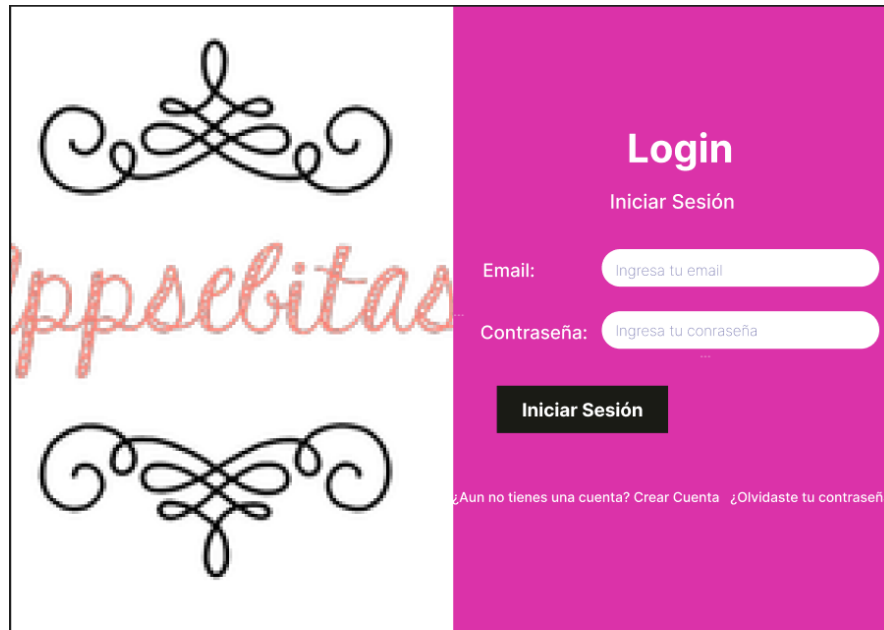


Fig. 1: Prototipo de Inicio de sesión del sistema web

2.3 Diseño de la arquitectura

El diseño arquitectónico de *software* ocupa el conocer de la programación con la finalidad de planear la vista general del *software* logrando así que los detalles puedan añadirse a futuro, logrando así que los grupos de desarrollo puedan minimizar la vista general y empezar a elaborar prototipos [37]. El patrón arquitectónico utilizado para el presente sistema web es el siguiente:

Patrón arquitectónico

Un patrón arquitectónico es una solución genérica y reutilizables para problemas comunes de arquitectura de software. El Modelo Vista Controlador (MVC) es un modelo arquitectónico, una guía que muestra como elaborar y delimitar los elementos de un proyecto de software, responsabilidades y relaciones entre el modelo, la vista y el controlador [38].

Modelo: tiene principalmente partes que representan la lógica, el dominio y los mecanismos de insistencia del sistema web.

Vista: se encuentran los elementos que son encargados de realizar la interconexión con el exterior, es decir la Interfaz de Usuario (UI) del sistema web.

Controlador: realiza tareas de transformación de datos con la finalidad de que los elementos del Modelo y la Vista logren entenderse. Es decir, traduce la información enviada

desde la interfaz a objetos que pueden ser interpretados por el Modelo, y de esa manera el Controlador recibirá información proveniente del Modelo y ajustara a la misma a modelos o estructura de datos para que la Vista lo pueda emplear [38].

La Fig. 2, indica el patrón arquitectónico elegido para el desarrollo del sistema web, el patrón utilizado es el MVC, con el fin de que el sistema tenga un formato de datos, una organizada estructura y compatibilidad con los artefactos utilizados para la codificación, los tests y el *deploy*.

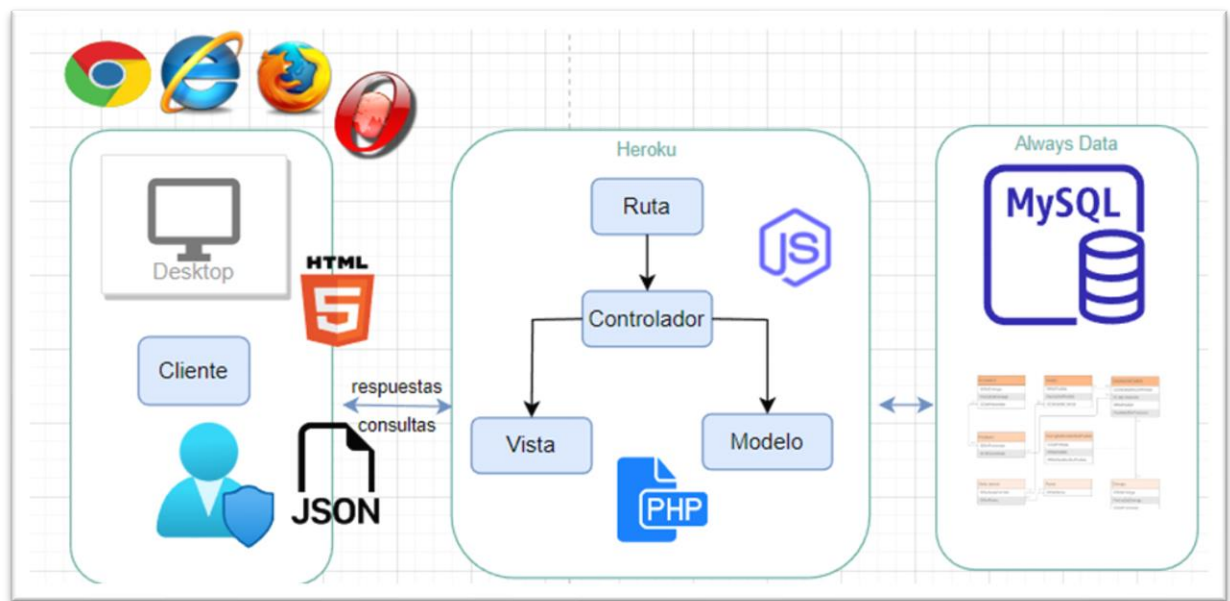


Fig. 2: Diseño del patrón arquitectónico del sistema *web*.

Modelo Entidad Relación

Un modelo entidad relación o ERD, es un diagrama de flujo que presenta cómo los objetos o conceptos se relacionan unos con otros en un sistema. La Fig. 3 representa el modelo de la base de datos que se ha usado en la codificación del presente sistema *web*, donde se detalla la relación que existe entre cada una de las tablas.

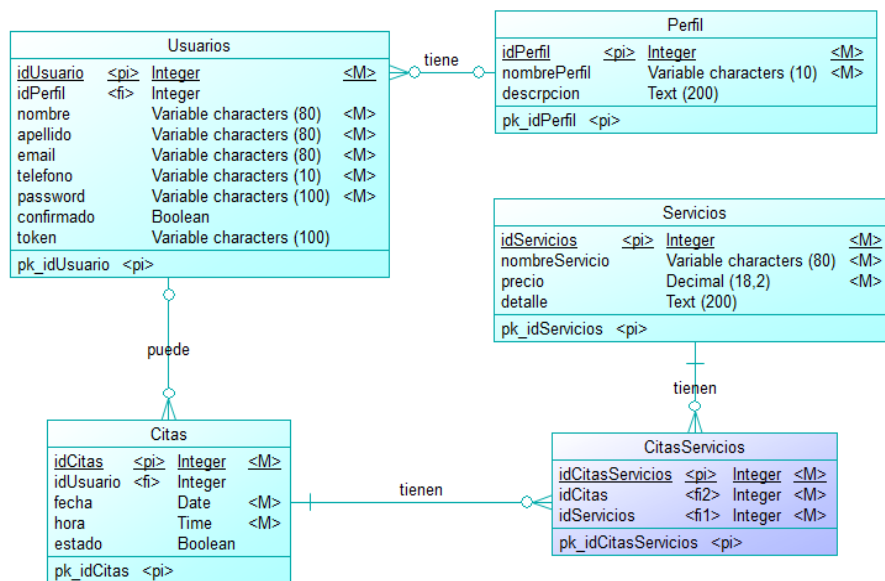


Fig. 3: Modelo Lógico

2.4 Herramientas de desarrollo

Las herramientas utilizadas para el desarrollo del sistema *web*, fueron escogidas a través de los requerimientos que se obtuvieron. La TABLA VII, muestra herramientas seleccionadas para el desarrollo del presente sistema *web*, además se detalla la justificación por la cual se han utilizados las mismas.

TABLA VII. Herramientas para el desarrollo del sistema *web*.

Herramienta	Justificación
PHP	Php permite desarrollar aplicaciones y crear sitios <i>web</i> logrando obtener más seguidores. Además, es fácil de usar y está en constante perfeccionamiento obteniendo así una alternativa fiable para los desarrolladores.
<i>JavaScript</i>	<i>JavaScript</i> es utilizado para agregar características interactivas al sistema <i>web</i> , es compacto y muy flexible, ofreciendo así a los desarrolladores un sin número de funcionalidades con un mínimo esfuerzo.
HTML	HTML es fácil de utilizar y entender, además es de código abierto lo que quiere decir que es gratis y se puede ejecutar en cualquier navegador.

Visual Studio Code	Es un editor robusto ya que ofrece varias extensiones para el desarrollo de los programas, estas ayudan a personalizar y agregar funcionalidades de manera modular.
Gulp	Gulp apoya en la mecanización de trabajos dentro de la elaboración de un sistema, moviendo archivos entre carpetas, eliminándolos, modificando la codificación, ajustar el navegador cuando el código haya sido modificado.
Figma	Figma es intuitiva y fácil de utilizar, ofrece la sistematización de los diseños y a su vez los aloja en la nube, además es gratuita.
Heroku	Heroku ofrece una tranquilidad ya que se ocupa de toda la infraestructura que hay que importar para que el servicio que se encuentra en la nube esté disponible siempre, permitiendo así enfocarse en el desarrollo del sistema <i>web</i> .

2.5 Librerías

En la TABLA VIII, se puede visualizar las librerías que se han utilizado para el desarrollo del sistema *web*, mismo que son importantes ya que permiten la incorporación con las herramientas antes nombradas.

TABLA VIII. Librerías para el desarrollo del sistema web

Librería	Justificación
<i>Composer</i>	<i>Composer</i> es un administrador de paquetes que facilita es modelo para poder administrarlo, descarga e instalación de dependencias.
<i>jQuery</i>	La biblioteca de <i>jQuery</i> facilita el uso de <i>JavaScript</i> en el sistema <i>web</i> ya que conecta recursos <i>JavaScript</i> a una página <i>web</i> .
<i>SweetAlert2</i>	Es una librería de <i>JavaScript</i> que ayuda en la creación de ventanas emergentes con una interfaz profesional, es compatible con la mayoría de los navegadores.

3 RESULTADOS

En el presente apartado se resume las respuestas logradas durante la implementación del sistema *web*, además se podrá visualizar las respuestas obtenidas de las pruebas realizadas, así como el *deploy* del presente sistema *web*. Los resultados de cada *Sprint* se podrán visualizar en el **ANEXO II** del documento actual.

3.1 *Sprint* 0. Configuración del ambiente de desarrollo

El *Sprint* 0, detalla la configuración del ambiente de desarrollo, en donde se realizará las respectivas configuraciones para que el ambiente en el cual se desarrollara el sistema *web* sea adecuado.

Los trabajos realizados en el presente *Sprint* se las detalla a continuación:

- Instalación de paquetes y extensiones necesarios para Visual Studio Code.
- Creación de prototipado.
- Estructura del proyecto.
- Creación del proyecto en GitHub.
- Pruebas unitarias *Sprint* 0.

Instalación de paquetes y extensiones necesarios para Visual Studio Code

El sistema *web* será elaborado en *php* y para hacerlos más interactivo con el usuario se utilizará *JavaScript*, por tal razón es necesario descargar e instalar algunos paquetes como *Node.js*, *PHP*, además de extensiones tales como *ESLint*, *JavaScript*, *PHP Server*, *PHP Intelephense*, que ayudaran a la codificación del presente sistema. Las instalaciones correspondientes se pueden visualizar en la Fig. 4, del presente documento.



Fig. 4: Extensiones para Visual Studio Code

Creación de prototipado

Luego de la recopilación de requerimientos, se realizó la creación de lo que sería el sistema web, con la finalidad que el administrador pueda visualizar un boceto del sistema a implementar. En la Fig. 5, se visualiza el *mockup* del inicio de sesión.

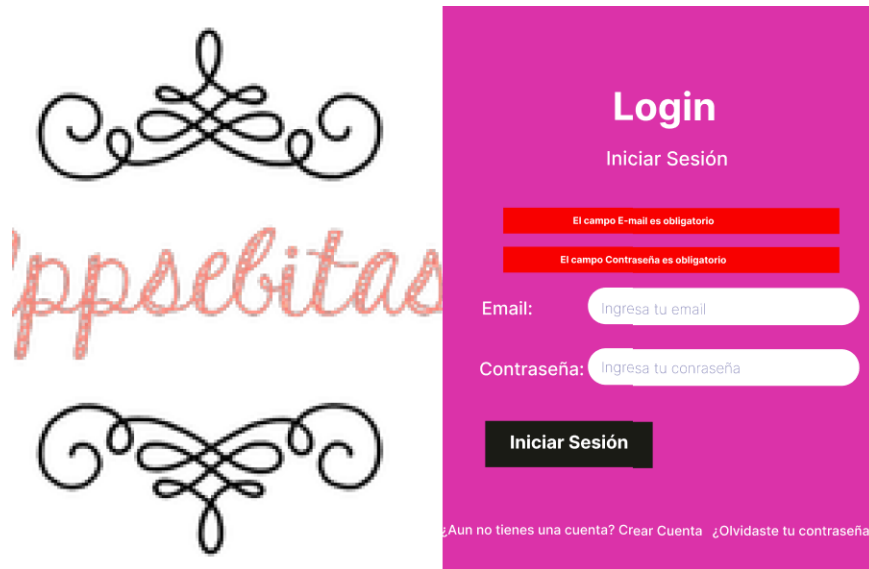


Fig. 5: Validación Login

Estructura del proyecto

El editor de textos utilizado para la implementación del sistema *web* ha sido *Visual Studio Code*, en la Fig. 6, se visualiza la organización del presente sistema basado en el patrón arquitectónico MVC mencionado anteriormente.

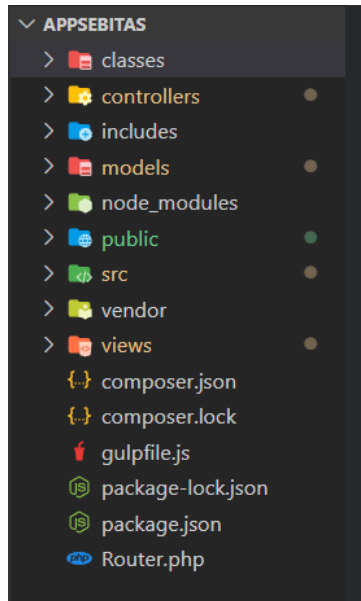


Fig. 6: Estructura del proyecto.

Creación del proyecto en GitHub

En GitHub se crear un nuevo repositorio para almacenar el código fuente del proyecto del sistema *web*, en la Fig. 7, se visualiza el repositorio para el proyecto creado de manera correcta.

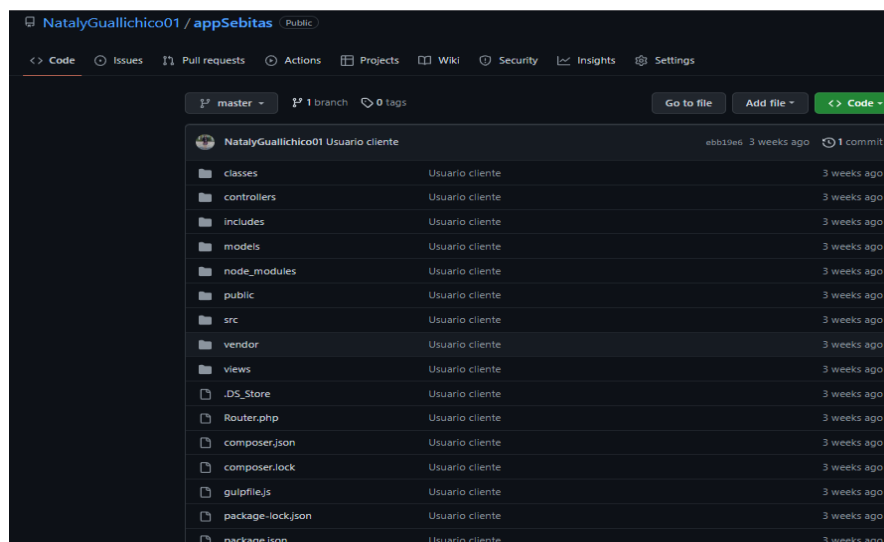
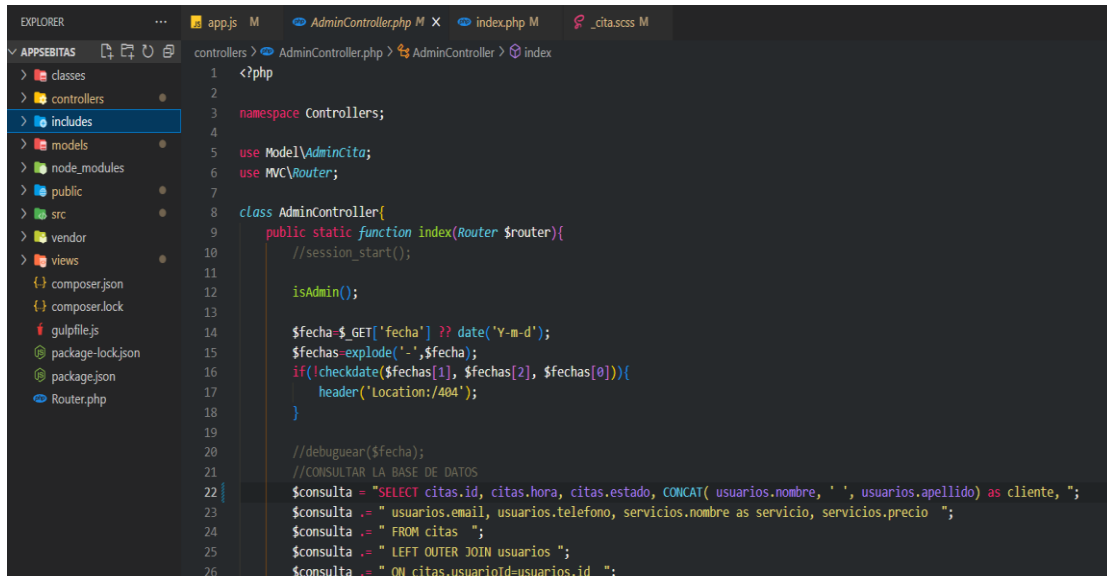


Fig. 7: Proyecto en *GitHub*

Pruebas Unitarias *Sprint 0*.

En este apartado se puede evidenciar que la configuración del ambiente de desarrollo se ha realizado de manera correcta, para evidencia de lo mencionado se muestra en la Fig.

8, el proyecto creado en Visual Studio Code, mismo que se encuentra almacenado en un repositorio de *GitHub*, como se visualiza en la Fig. 9.



```
1 <?php
2
3 namespace Controllers;
4
5 use Model\AdminCita;
6 use MVCRouter;
7
8 class AdminController{
9     public static function index(Router $router){
10         //session_start();
11
12         isAdmin();
13
14         $fecha=$_GET['fecha'] ?? date('Y-m-d');
15         $fechas=explode('-', $fecha);
16         if(!checkdate($fechas[1], $fechas[2], $fechas[0])){
17             header('Location:/404');
18         }
19
20         //debuguear($fecha);
21         //CONSULTAR LA BASE DE DATOS
22         $consulta = "SELECT citas.id, citas.hora, citas.estado, CONCAT( usuarios.nombre, ' ', usuarios.apellido) as cliente, ";
23         $consulta .= " usuarios.email, usuarios.telefono, servicios.nombre as servicio, servicios.precio ";
24         $consulta .= " FROM citas ";
25         $consulta .= " LEFT OUTER JOIN usuarios ";
26         $consulta .= " ON citas.usuarioId=usuarios.id ";
```

Fig. 8: Prueba Unitaria del ambiente de desarrollo

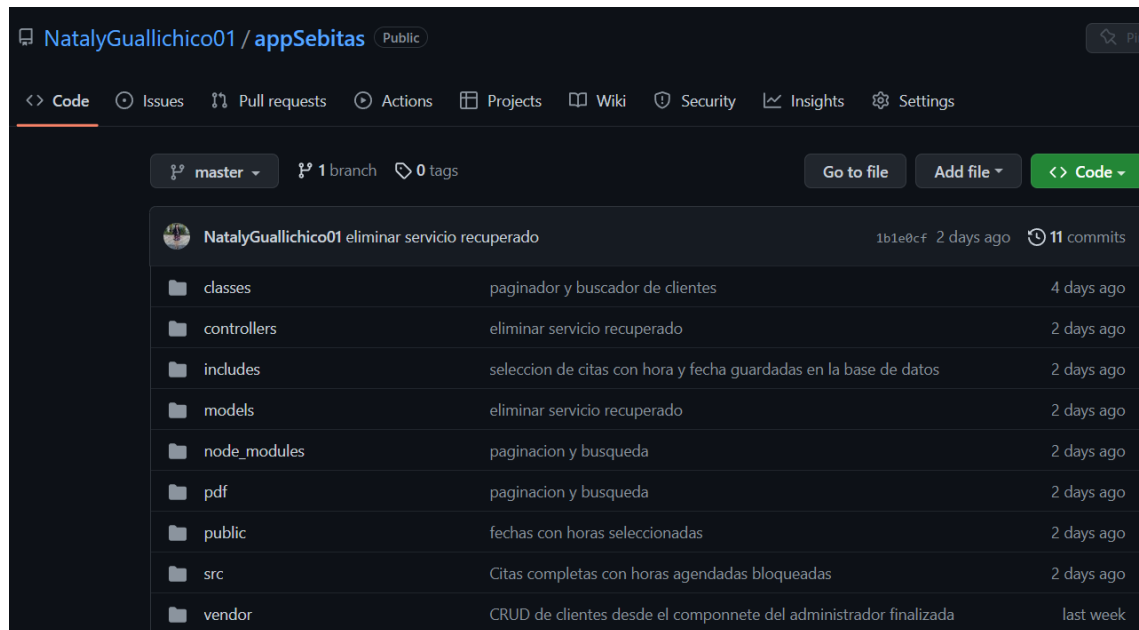


Fig. 9: Actualización de repositorio en GitHub

Una vez finalizada la configuración del ambiente de desarrollo se procede a elaborar el siguiente Sprint que en este caso corresponde a la autenticación.

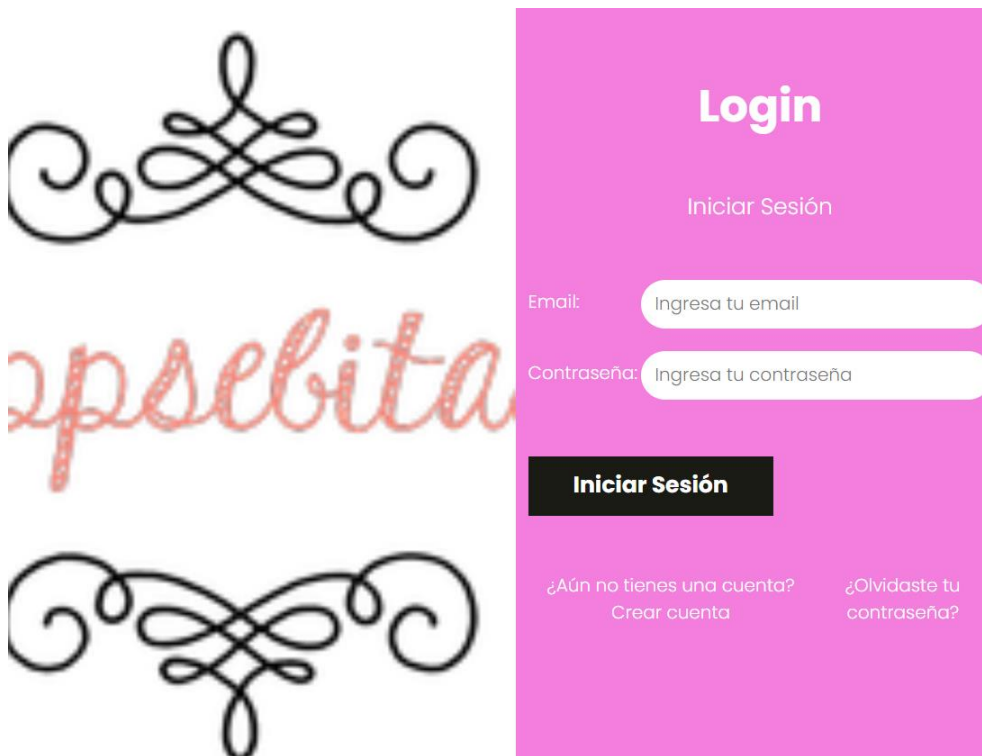
3.2 *Sprint 1*. Autenticación

El presente *Sprint 1*, retribuye a la autenticación, donde se puede visualizar las tareas realizadas con el fin de que el usuario logre autenticarse dentro del sistema *web*. Las tareas realizadas en la presente iteración son las siguientes:

- Delimitar campos en el registro e inicio de sesión.
- Delimitar las reglas de validación.
- Implementar un registro, un inicio de sesión y reestablecer contraseña.
- Verificación de que correos registrados no se repitan.
- Verificación de campos obligatorios.
- Pruebas unitarias *Sprint 1*.

Delimitar campos en el registro e inicio de sesión

La Fig. 10, muestra cada uno de los campos utilizados para el inicio de sesión, los cuales son email y contraseña; además cuenta con una ventana en donde el usuario podrá registrarse en el caso de no tener una cuenta previa, los campos para el registro son nombre, apellido, teléfono, *e-mail* y contraseña tal y como se evidencia en la Fig. 11.



The image shows a login form on a pink background. At the top, the word "Login" is written in white. Below it, the text "Iniciar Sesión" is displayed. The form contains two input fields: "Email:" with the placeholder "Ingresa tu email" and "Contraseña:" with the placeholder "Ingresa tu contraseña". A black button labeled "Iniciar Sesión" is positioned below the input fields. At the bottom, there are two links: "¿Aún no tienes una cuenta? Crear cuenta" and "¿Olvidaste tu contraseña?". The form is framed by decorative black flourishes on the left and bottom, and the word "psebita" is written in a red, cursive font across the middle.

Fig. 10: Formulario de login de usuario



Crear Cuenta

Llena el siguiente formulario para obtener tu cuenta

Nombre:

Apellido:

Teléfono:

E-mail:

Contraseña:

Crear Cuenta

Fig. 11: Formulario de registro de usuario

Delimitar reglas de validación

Las reglas de validación contempladas para el presente sistema *web* son:

- Reglas de validación de correo electrónico, donde el correo deberá ser ingresado con un formato similar a [correo@dominio.de.puebra](#), tal y como se evidencia en la Fig. 12.



Login

Iniciar Sesión

Email:

! Incluye un signo "@" en la dirección de correo electrónico. La dirección "natilu" no incluye el signo "@".

Iniciar Sesión

¿Aún no tienes una cuenta?
Crear cuenta

¿Olvidaste tu contraseña?

Fig. 12: Validación email

- Regla para validar contraseña, en donde la misma no podrá ser menor a 6 caracteres, y deberá confirmar la cuenta aceptando un mensaje enviado a su correo ingresado en el registro de datos, tal y como se evidencia en la Fig. 13.

The image shows a login interface on a pink background. At the top, it says "Login" and "Iniciar Sesión". A red error banner displays the message: "CONTRASEÑA INCORRECTA O NO HAS CONFIRMADO TU CUENTA AÚN". Below this, there are input fields for "Email" (containing "Ingresa tu email") and "Contraseña" (containing "Ingresa tu contraseña"). A black "Iniciar Sesión" button is positioned below the fields. At the bottom, there are two links: "¿Aún no tienes una cuenta? Crear cuenta" and "¿Olvidaste tu contraseña?". To the left of the form, there is a decorative graphic with the word "psebita" in a red, cursive font, flanked by black scrollwork.

Fig. 13: Validación de contraseña

Implementar registro, inicio de sesión y reestablecer contraseña

En el módulo de registro de usuario se procede a llenar los campos solicitados, como se evidencia en la Fig. 14.

The image shows a user registration form on a pink background. The title is "Crear Cuenta". Below the title, it says "Llena el siguiente formulario para obtener tu cuenta". The form contains several input fields: "Nombre" (filled with "Mariuxi"), "Apellido" (filled with "Paucar"), "Teléfono" (filled with "0999059146"), "E-mail" (filled with "natilu@gmail.com"), and "Contraseña" (filled with "....."). A black "Crear Cuenta" button is located at the bottom of the form. To the left of the form, there is a decorative graphic with the word "psebita" in a red, cursive font, flanked by black scrollwork.

Fig. 14: Ingreso de datos registro de usuario

Una vez ingresados todos los campos en el registro al presionar en crear cuenta nos mostrara un mensaje en donde solicitara la confirmación de la cuenta a través de un e-mail, como se evidencia en la Fig. 15.



Fig. 15: Confirmar Cuenta

Revisar en el correo un mensaje de confirmación para que pueda finalizar con la creación de la cuenta, para esto se dará clic en el enlace de Confirmar cuenta y la cuenta será registrada en el caso de no requerirla se deberá ignorar el mensaje de confirmación, como se evidencia en la Fig. 16.



Fig. 16: Email de confirmación de cuenta

Al confirmar la cuenta a través del enlace enviado al correo electrónico ingresado en el registro nos mostrara un mensaje de que la cuenta ha sido creada exitosamente, como se evidencia en la Fig. 17.

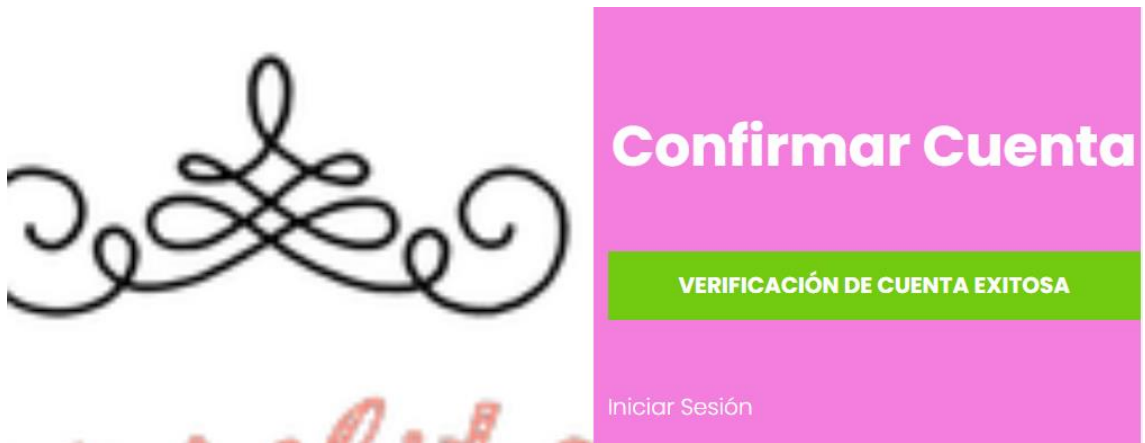


Fig. 17: Mensaje de Cuenta Confirmada

Una vez finalizado el registro de usuario, ingresa el email y contraseña registrados anteriormente para poder visualizar los servicios que ofertados por el sistema *web*, como se evidencia en la Fig. 18.

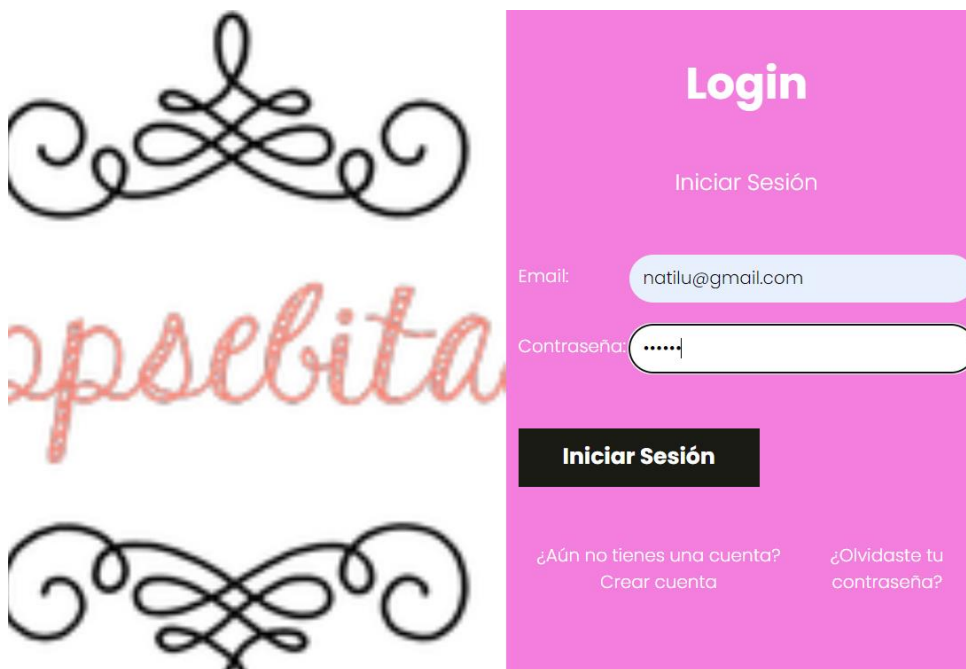


Fig. 18: Ingreso datos a Login

En el módulo de reestablecer la contraseña se pedirá al usuario ingresar el correo con el que fue registrado en el sistema, mismo al que le llegará un email para poder reestablecer la contraseña, se puede visualizar en la Fig. 19, mostrará un mensaje en donde nos pedirá que revisar el email, como se evidencia en la Fig. 20.



Fig. 19: Olvidar Contraseña



Fig. 20: Revisar email

Revisar en el correo un mensaje de reestablecer contraseña para reestablecer la misma, para esto se dará clic en el enlace de Reestablecer Contraseña, tal como se puede visualizar en la Fig. 21, le redirigirá a una venta para recuperar su contraseña en donde deberá ingresar su nueva contraseña, como se evidencia en la Fig. 22.

Reestablece tu contraseña

From: <cuentas@appsebitas.com>
To: AppSebitas.com <cuentas@appsebitas.com>

Show Headers

HTML HTML Source Text Raw Spam Analysis HTML Cr



Hola Nataly Has solicitado reestablecer tu contraseña, presiona el siguiente enlace

Presiona aquí: [Reestablecer Contraseña](#)

Si no solicitó esta cuenta ignore el mensaje

Fig. 21: Email reestablecer contraseña



Fig. 22: Reestablecer contraseña

Verificación de que correos registrados no se repitan

La Fig. 23, muestra la validación de que el email ingresado ya existe para otro usuario, además de que nos indica si el correo ingresado está registrado en la base de datos, como se evidencia en la Fig. 24.



Fig. 23: Usuario ya existente

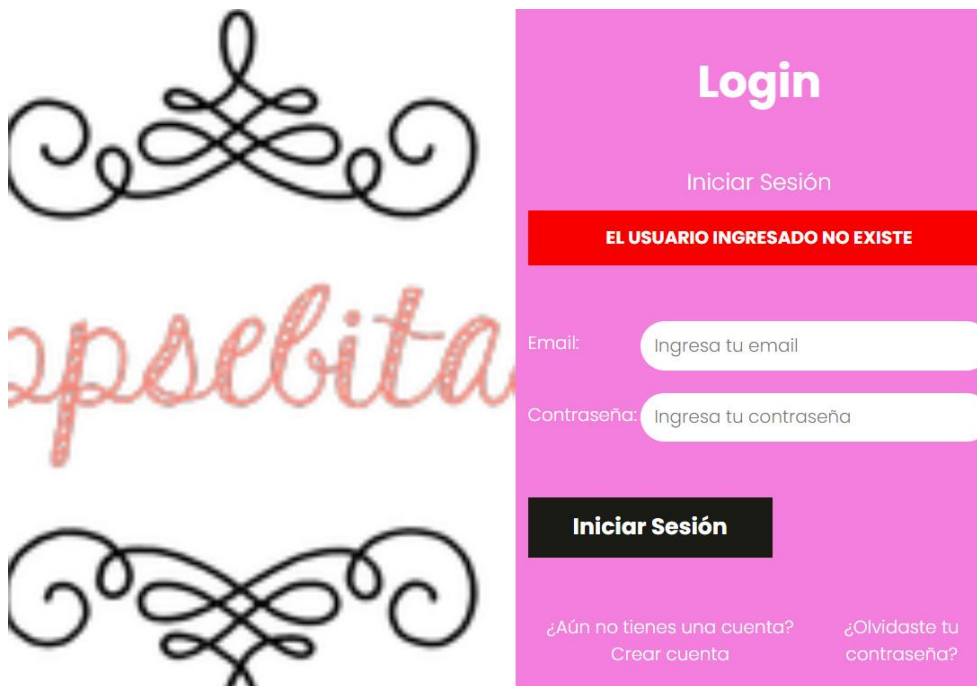


Fig. 24: Validación de existencia de Usuario

Verificación de campos obligatorios

Una de las funcionalidades importantes para el que el usuario pueda acceder al sistema web es estar registrado, por tal razón es importante que todos los campos estén llenos en el formulario de registro, caso contrario se mostrara una alerta al usuario indicando los campos que estén vacíos, tal y como se evidencia en la Fig. 25.



Fig. 25: Validación de datos llenos

Pruebas Unitarias de autenticación.

En este apartado se puede evidenciar que la autenticación se ha realizado de manera correcta, para evidencia de lo mencionado se muestra en la Fig. 26, donde se puede visualizar un inicio de sesión con un usuario previamente registrado y el acceso dentro del sistema, como se evidencia en la Fig. 27.

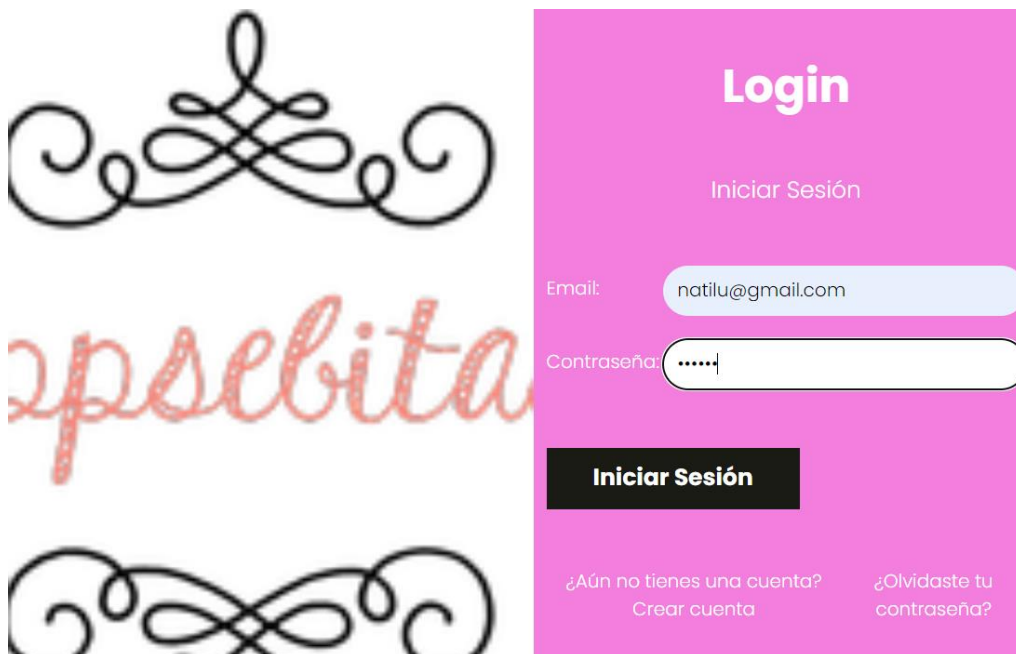


Fig. 26: Prueba Unitaria Autenticación Login



Fig. 27: Prueba Unitaria Autenticación ingresando con Login

3.3 *Sprint 2*. Módulo de Información.

El presente *Sprint 2*, retribuye al módulo de espacio virtual e información, donde se puede visualizar los servicios que ofrece el sistema *web*. Las tareas realizadas en la presente iteración son las siguientes:

- Presentar los servicios ofertados por la peluquería.
- Acceder al módulo de citas agendadas.
- Validar el horario de citas disponibles.
- Validar que todos los campos hayan sido ingresados.
- Pruebas unitaria *Sprint 2*.

Presentar los servicios ofertados por la peluquería.

La Fig. 28, muestra los servicios que ofrece la peluquería, donde se puede visualizar la información acerca de los servicios, en donde se puede apreciar el servicio, precio y lo que incluye, una vez que se haya seleccionado los servicios solicitados el color de servicio ofertado cambiara, como se evidencia en la Fig. 29.

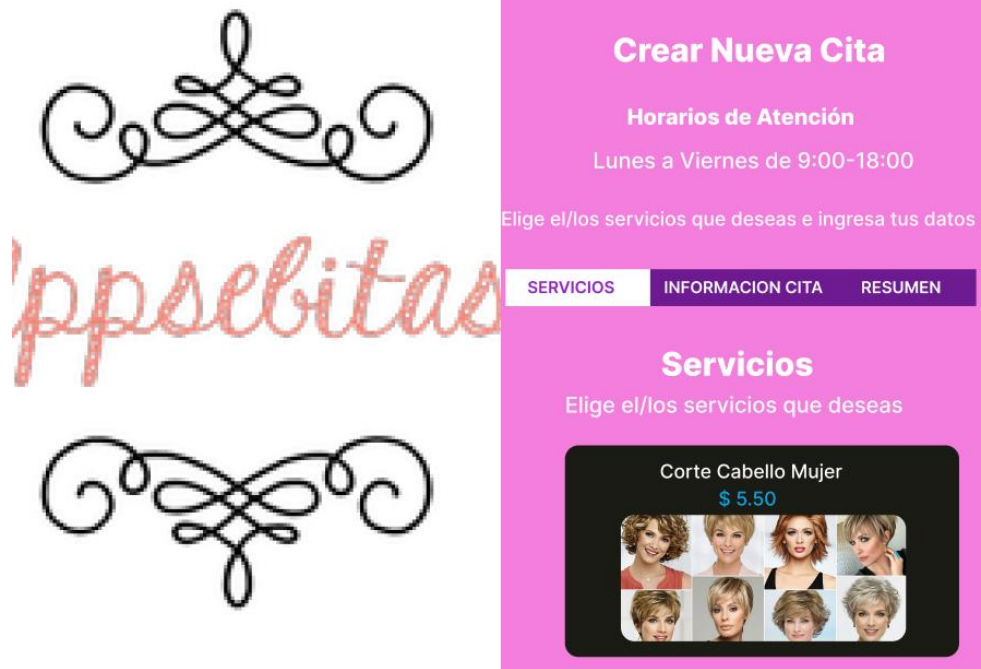


Fig. 28: Servicios

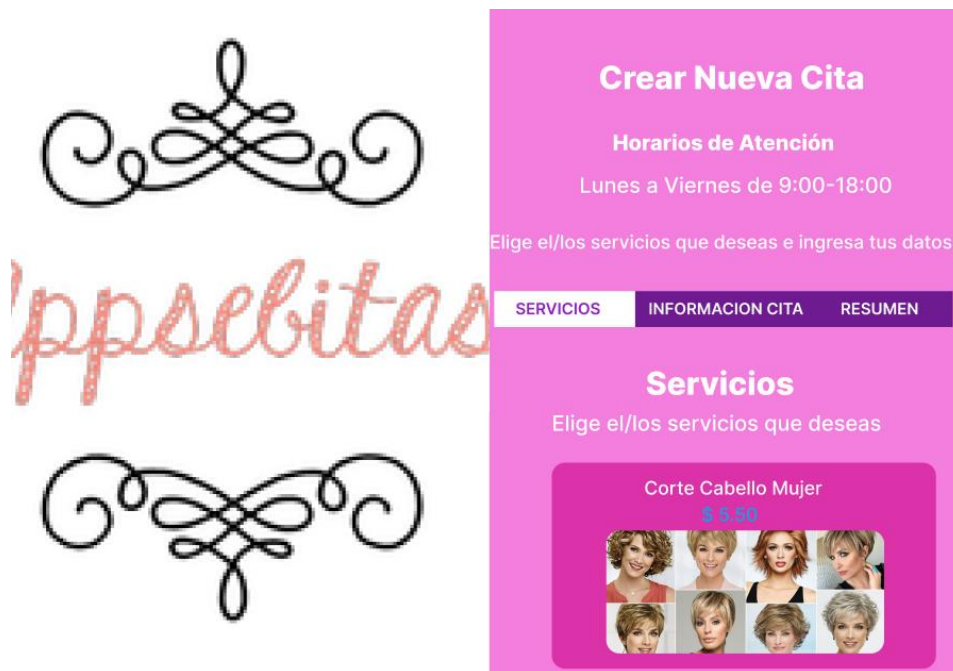


Fig. 29: Selección de servicios

Acceder al módulo de citas agendadas.

La Fig. 30, muestra las citas agendadas dentro del sistema *web* de la peluquería, donde el usuario administrador podrá visualizar las citas agendadas, en donde se seleccionará una fecha para realizar la búsqueda de las citas por atender en la fecha señalada.



Fig. 30: Administrador - citas agendadas

Validar el horario de citas disponibles.

La Fig. 31, muestra la información de la cita en donde podremos seleccionar una cita por fecha y hora, una vez que se haya seleccionado la fecha el sistema nos mostrara los horarios disponibles y los horarios que no están disponibles los bloqueara, tal como se puede visualizar en la Fig. 32, si el cliente selecciona un día en que la peluquería no atienda se mostrara una alerta, como se evidencia en la Fig. 33.



Elige el/los servicios que deseas e ingresa tus datos

Hola Nataly Mariuxi Guallichico **Cerrar Sesión**

SERVICIOS **INFORMACIÓN CITA** RESUMEN

Tus Datos y Cita

Ingresar tus datos y fecha de tu cita

Nombre:

Fecha:

Hora:

« Anterior » **Siguiente »**

Fig. 31: Información Cita



Hola Nataly Mariuxi Guallichico **Cerrar Sesión**

SERVICIOS **INFORMACIÓN CITA** RESUMEN

Tus Datos y Cita

Ingresar tus datos y fecha de tu cita

Nombre:

Fecha:

Hora:

« Anterior » **Siguiente »**

Fig. 32: Citas disponibles

A screenshot of a user profile page. At the top, it says 'Hola Nataly Mariuxi Guallichico' and has a 'Cerrar Sesión' button. Below is a navigation bar with 'SERVICIOS', 'INFORMACIÓN CITA', and 'RESUMEN'. The main section is titled 'Tus Datos y Cita' and asks the user to 'Ingresa tus datos y fecha de tu cita'. There are input fields for 'Nombre' (filled with 'Nataly Mariuxi Guallichico'), 'Fecha' (with a date picker showing 'dd/mm/aaaa'), and 'Hora'. A red banner below the form reads 'LOS FINES DE SEMANA NO ATENDEMOS'. At the bottom are '« Anterior' and 'Siguiete »' buttons.

Fig. 33: Alerta de días no laborales

Validar que todos los campos hayan sido ingresados.

La Fig. 34, muestra el apartado de resumen, donde se puede visualizar la información del servicio y de la cita, en el caso de que no se hayan ingresado toda la información para la cita se mostrara una alerta, como se evidencia en la Fig. 35.

A screenshot of a summary page. The navigation bar shows 'SERVICIOS', 'INFORMACIÓN CITA', and 'RESUMEN'. The main section is titled 'Resumen de Servicios' and lists 'Corte Cabello Mujer' with 'Incluye Corte + Cepillado' and 'Precio: \$5.50'. Below is a section titled 'Resumen de Cita' with 'Nombre: Nataly Mariuxi Guallichico', 'Fecha: jueves, 26 de enero de 2023', and 'Hora: 10:30 Horas'. A 'Reservar Cita' button is at the bottom.

Fig. 34: Resumen

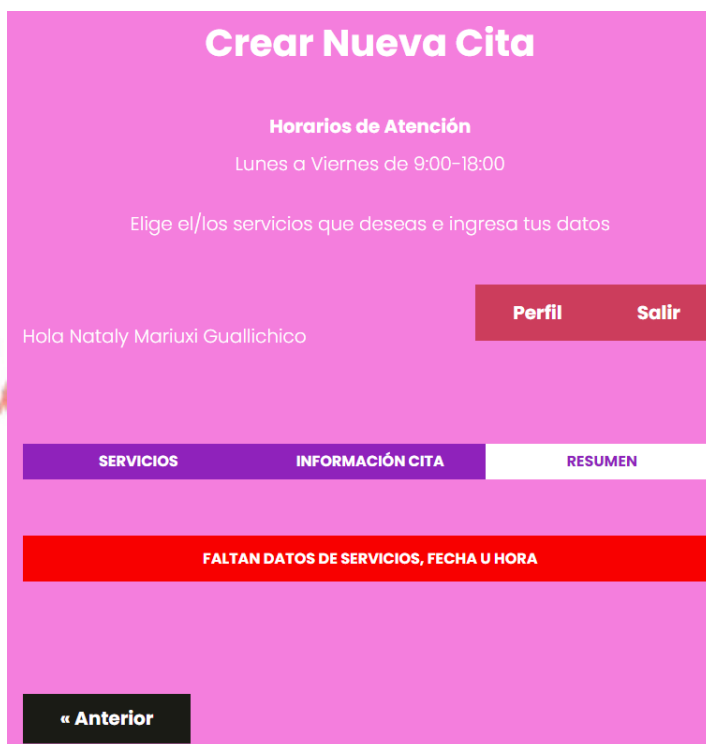


Fig. 35: Alerta de datos faltantes

Pruebas Unitarias *Sprint 2*.

En este apartado se puede evidenciar que una vez que el cliente haya ingresado los campos solicitados para el agendamiento de citas de manera correcta, como se evidencia en la Fig. 36, al presionar en reservar cita nos mostrara una alerta de cita creada, tal y como se evidencia en la Fig. 37.



Fig. 36: Validación campos cita

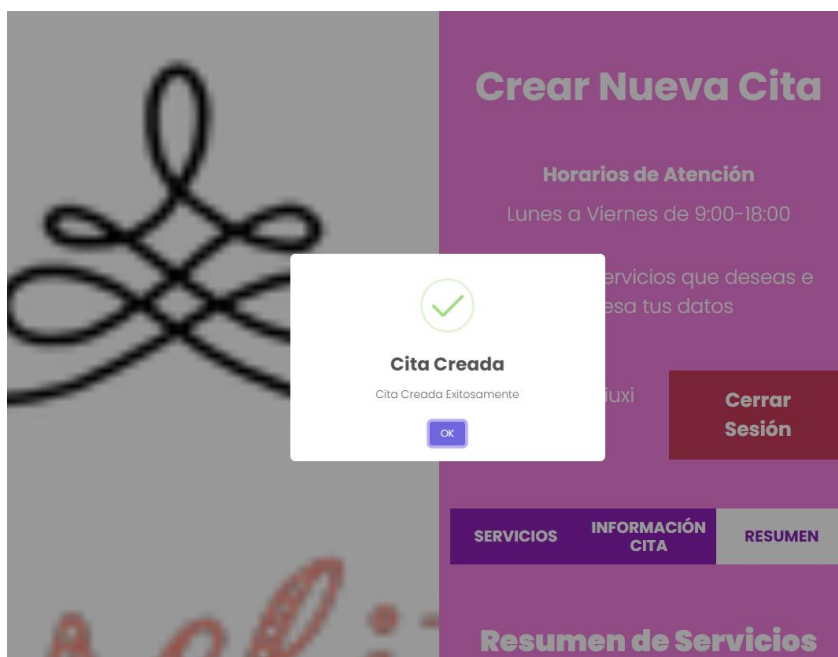


Fig. 37: Creación de cita exitosa

3.4 *Sprint 3*. Módulo Complementario.

El presente *Sprint 3*, retribuye al módulo complementario, logrando obtener los requerimientos solicitados por el administrador. Las tareas realizadas en la presente iteración son las siguientes:

- CRUD de servicios ofertados mediante el sistema *web*.
- Ingresar y actualizar datos de nuevos clientes.
- CRUD de roles dentro del sistema *web*.
- Pruebas unitarias *Sprint 3*.

CRUD de servicios ofertados mediante el sistema *web*.

El usuario administrador podrá ingresar un nuevo servicio, tal como se puede visualizar en la Fig. 38, además también podría actualizar o eliminar algún servicio de ser necesario, tal como se visualiza en la Fig. 39.



Nuevo Servicio

Ingresa los campos para agregar un nuevo Servicio

Hola Admin Paucar Salir

Ver Citas	Ver Servicios	Nuevo Servicio	Ver Clientes
-----------	---------------	----------------	--------------

Nombre:

Precio:

Descripción:

Guardar Servicio

Fig. 38: Nuevo Servicio



Servicios

Administración de Servicios

Hola Admin Paucar Salir

Ver Citas	Ver Servicios	Nuevo Servicio	Ver Clientes
-----------	---------------	----------------	--------------

Nombre: Uñas Encapsuladas
Precio: \$15.00
Descripción: Mantenimiento gratis

Actualizar Eliminar

Fig. 39: Actualizar y eliminar servicios

Ingresar y actualizar datos de nuevos clientes.

Únicamente el usuario administrador podrá ingresar un nuevo cliente, tal como se puede visualizar en la Fig. 40, además también podría actualizar algún cliente de ser necesario, como se evidencia en la Fig. 41.



Nuevo Cliente

Ingrese todos los campos requeridos para ingresar un nuevo Cliente

Hola Admin Paucar **Cerrar Sesión**

Ver Citas **Ver Clientes** **Nuevo Cliente**

Nombre:

Apellido:

Teléfono:

E-mail:

Guardar Cliente

Fig. 40: Ingresar Cliente



Administración de Clientes

Hola Admin Paucar **Cerrar Sesión**

Ver Citas **Nuevo Cliente**

Buscar Cliente

Nombre	Apellido	Correo	Telefono	Acciones
Nombre: Admin	Apellido: Paucar	Teléfono: 1234567890	E-mail: admin@admin.com	Actualizar
Nombre: Nataly Mariuxi	Apellido: Guallichico	Teléfono: 0999059146	E-mail: natilu@gmail.com	Actualizar

Fig. 41: Actualizar Cliente

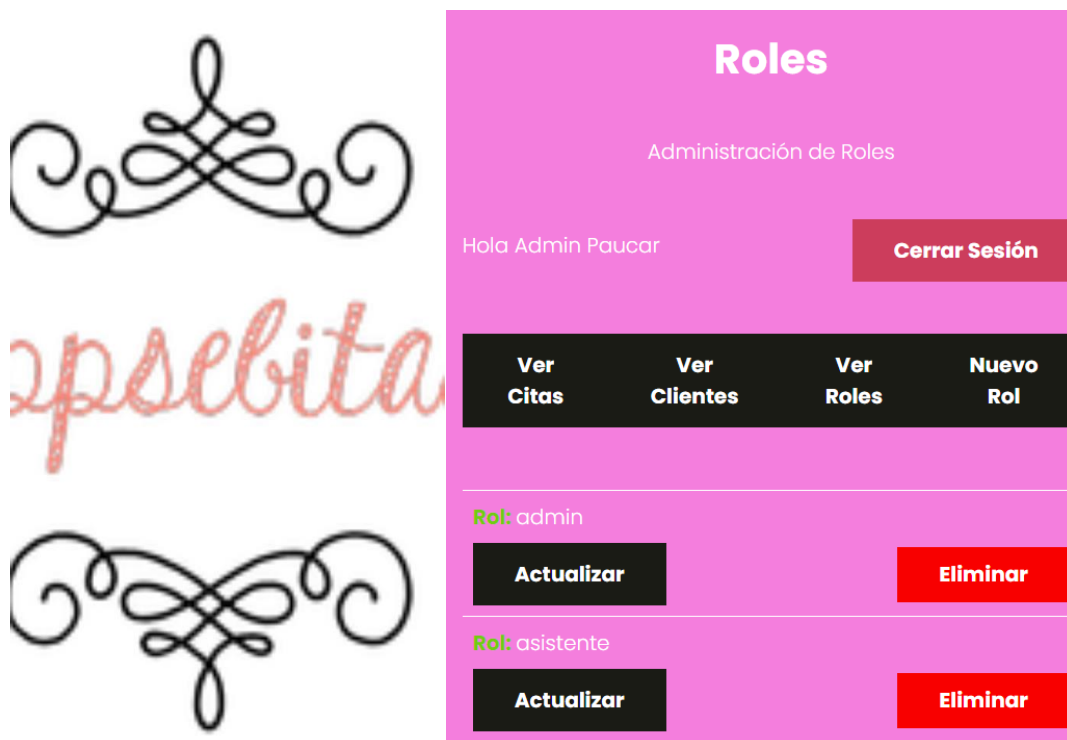
CRUD de roles dentro del sistema web.

Únicamente el usuario administrador podrá ingresar un nuevo rol, tal como se puede visualizar en la Fig. 42, además también podría actualizar o eliminar algún rol de ser necesario, como se evidencia en la Fig. 43.



The screenshot shows a web interface for creating a new role. On the left, there is a decorative header with a black flourish, the text 'xpsebita' in a red cursive font, and another black flourish. The main content area has a pink background. At the top, it says 'Nuevo Rol' in white. Below that, a message reads 'Ingrese todos los campos requeridos para ingresar un nuevo Rol'. The user is identified as 'Hola Admin Paucar' with a 'Cerrar Sesión' button. A navigation bar contains 'Ver Citas', 'Ver Clientes', 'Ver Rol', and 'Nuevo Rol'. The 'Nuevo Rol' form includes a 'Perfil:' label and an input field with the placeholder 'Ingresa el Rol'. A 'Guardar Rol' button is at the bottom.

Fig. 42: Ingresar Rol



The screenshot shows a web interface for managing roles. On the left, there is a decorative header with a black flourish, the text 'xpsebita' in a red cursive font, and another black flourish. The main content area has a pink background. At the top, it says 'Roles' in white. Below that, it says 'Administración de Roles'. The user is identified as 'Hola Admin Paucar' with a 'Cerrar Sesión' button. A navigation bar contains 'Ver Citas', 'Ver Clientes', 'Ver Roles', and 'Nuevo Rol'. The 'Ver Roles' section lists two roles: 'admin' and 'asistente'. Each role has an 'Actualizar' button and an 'Eliminar' button.

Fig. 43: Actualizar o eliminar Rol

Pruebas Unitarias *Sprint 3.*

En este apartado se puede evidenciar como el usuario administrador puede hacer CRUD de servicios, clientes y usuarios, para verificar lo mencionado se puede visualizar la Fig. 44, Fig. 45 y Fig. 46.



Fig. 44: CRUD Servicios



Fig. 45: Crear y actualizar Cliente



Fig. 46: CRUD Rol

3.5 *Sprint 4*. Pruebas y despliegue del sistema.

Una vez terminada la fase de codificación, el presente *Sprint 4* retribuye a las pruebas y despliegue del sistema *web*. Las tareas realizadas en la presente iteración son las siguientes:

- Pruebas de usabilidad.
- Pruebas de compatibilidad.
- Despliegue en *Heroku*.

Pruebas de usabilidad.

Las pruebas de usabilidad son realizadas por usuarios reales, ya que pueden identificar problemas que el desarrollador ya no puede identificar pues estaría familiarizado con el sistema que el desarrollo. Las pruebas de usabilidad permiten al *team* de desarrollo y diseño identificar problemas y así solucionarlos en la codificación [39].

La medición de usabilidad dentro sistema *web* se elaboró basado en la escala de usabilidad SUS (*System Usability Scale*), donde luego de realizar la encuesta a un total de 22 personas se obtuvo un porcentaje promedio de 82.38, que al realizarse una comparación en la escala de evaluación de SUS se puede decir que se encuentra en un rango aceptable tal como se puede visualizar en la Fig. 47. El procedimiento realizado para esta prueba se encuentra detallado en el **ANEXO II**, en la sección pruebas de usabilidad del presente documento.

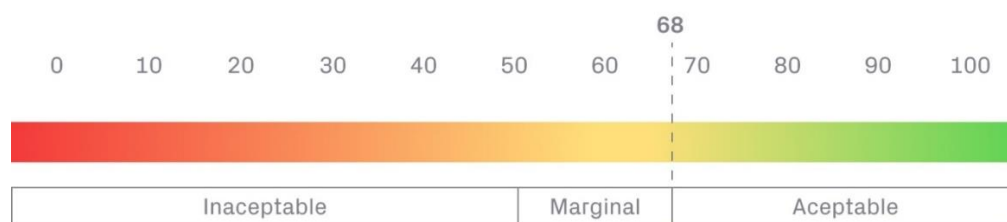


Fig. 47: Escala SUS.

Pruebas de compatibilidad.

Son utilizadas con la finalidad de garantizar el agrado del cliente. Son utilizadas para determinar si un proyecto de software es apto para ser ejecutado en distintos navegadores, sistemas operativos, dispositivos móviles, etc. Además, el proyecto de software puede verse afectado por distintas versiones, resoluciones, velocidades, etc. Además, ayudan a que el sistema se ejecute con éxito y correctamente en distintos navegadores, versiones,

etc. [40]. En la Tabla IX, se puede visualizar los distintos navegadores que fueron utilizados para verificar la compatibilidad. Los resultados obtenidos de la prueba de usabilidad que pueden visualizar en el **ANEXO II**, en la sección pruebas de compatibilidad del presente documento.

Tabla IX: Pruebas de compatibilidad-navegadores.

NOMBRE	VERSIÓN	OBSERVACIÓN
Google Chrome	109.5.54.14.120	Completamente funcional
Microsoft Edge	109.0.1518.61	Completamente funcional
Brave Browser	1.47.186	Completamente funcional

Una vez finalizada las pruebas de compatibilidad y con las respuestas adquiridas se puede decir que el sistema *web* es compatible con los distintos navegadores utilizados para la ejecución de estas pruebas.

Despliegue en Heroku.

El despliegue del presente sistema *web* se llevó a cabo en la plataforma de *heroku*, en donde se creó un nuevo proyecto, tal como se puede visualizar en la Fig. 48, posterior se puede observar ya como se logró el despliegue del sistema *web* en la plataforma y el enlace proporcionado, tal como se puede visualizar en la Fig. 49. Las credenciales de acceso se muestran en el **ANEXO IV**, del documento actual.

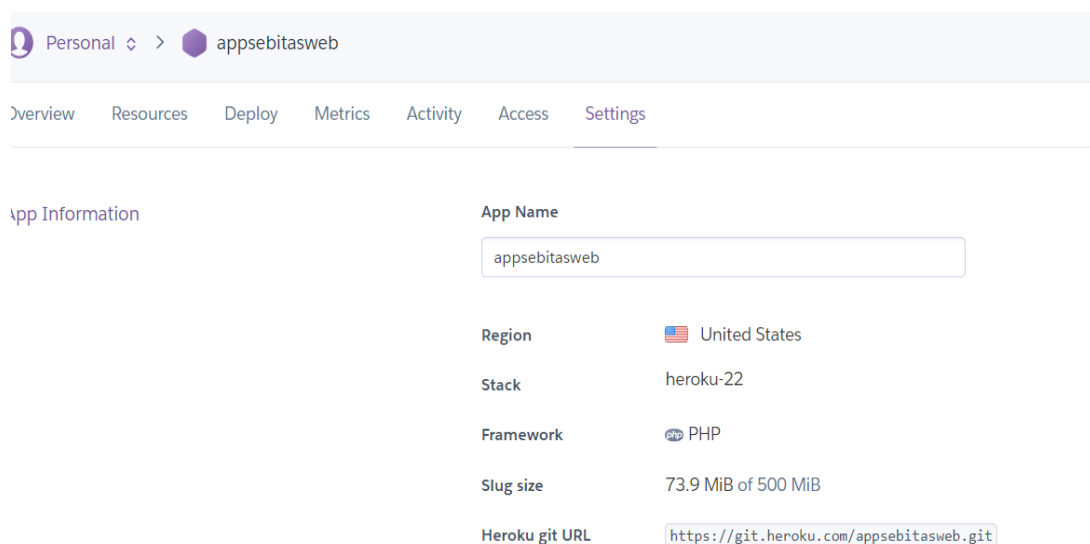


Fig. 48: Nuevo proyecto - Heroku.

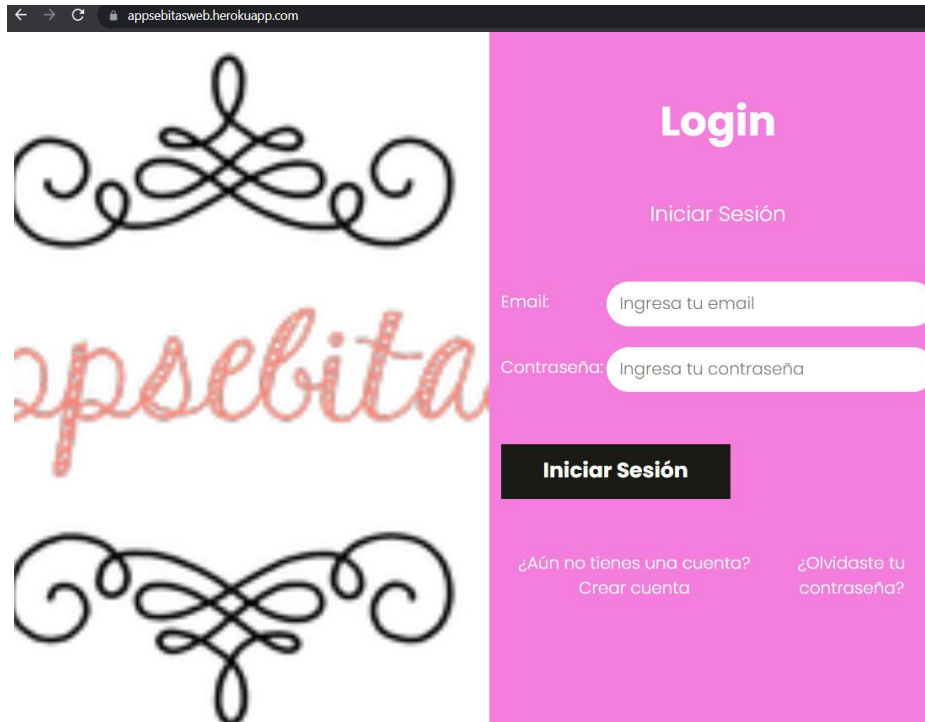


Fig. 49: Deploy en Heroku.

4 CONCLUSIONES

Se presenta a continuación los resultados acordes a los objetivos planteados en el actual proyecto de titulación con las siguientes conclusiones.


- La implementación del sistema *web* de agendamiento de citas para la peluquería Unisex “Sebitas” se realizó en base a los requerimientos obtenidos previamente con las entrevistas realizadas, logrando así cumplir con los objetivos propuestos y el alcance determinado.
- La metodología *Scrum* seleccionada para el presente proyecto permitió que el mismo pueda avanzar según los tiempos propuestos en cada *sprint*, logrando así culminar con el proyecto.
- El patrón arquitectónico modelo, vista, controlador que fue seleccionado para el presente sistema *web* ayudo a que cada endpoint sea consumido de manera correcta dado que nos ayuda a separar, los datos, la lógica y las interfaces de usuario.
- El diseño y elaboración del prototipado fue de gran ayuda ya que se pudo plasmar la idea requerida, así como también se pudo realizar los cambios solicitados por la persona que utilizar el sistema previo a la codificación, logrando así optimizar tiempo.
- Las distintas pruebas realizadas en el presente sistema *web*, permitieron corroborar que el proyecto funcione correctamente, además de que cumpla con los requerimientos solicitados previamente.

5 RECOMENDACIONES

Se presenta a continuación las recomendaciones que se ha obtenido durante el proyecto de titulación actual.

- El desarrollar un sistema *web* eficiente dependerá mucho de los requerimientos realizados, por tal razón es importante tener varias reuniones con el cliente que maneja el sistema, dado que pueden darse pequeños cambios durante el proceso de desarrollo.
- Es importante tener claro el modelo de la base de datos que se va a utilizar para el sistema *web*, ya que del mismo dependerá que el sistema no arroje errores, una herramienta para elaborar el modelo es *power designer* que nos facilita generar los distintos modelos de la base de datos, también el script para generar los campos de la misma.
- Es importante generar una copia de seguridad cada cierto tiempo ya que esto ayudara a resguardar la información del sistema *web* que se encuentra almacenada dentro de la base de datos.
- La usabilidad del sistema *web* depende mucho del cliente, por tal razón se recomienda tener nuevos acercamientos con los usuarios involucrados con sistema *web* para mejorar usabilidad del mismo.
- Si se solicita modificar algún componente dentro del sistema *web*, se sugiere contactarse con el *team* de desarrollo, para que trabajando juntos se logre obtener el sistema *web* actualizado.

6 REFERENCIAS BIBLIOGRÁFICAS

- [1] J. J3W, «Página web para Salón de Belleza y Peluquería», 14 de octubre de 2016. <https://imagen3web.com/pagina-web-salon-belleza-peluquer/> (accedido 14 de noviembre de 2022).
- [2] IONOS, «Página web para peluquería: presencia online rápida, sencilla y asequible», *IONOS Digital Guide*. <https://www.ionos.es/digitalguide/paginas-web/creacion-de-paginas-web/crear-paginas-web-de-peluquerias/> (accedido 14 de noviembre de 2022).
- [3] J. Turina, «La evolución de la tecnología y la gestión en el entorno de las pymes», *Datasa ERP*, 13 de diciembre de 2018. <https://datisa.es/evolucion-las-pymes-la-tecnologia/> (accedido 14 de noviembre de 2022).
- [4] servnet, «Backend y Frontend, ¿Qué es y cómo funcionan en la programación?» <https://www.servnet.mx/blog/backend-y-frontend-partes-fundamentales-de-la-programacion-de-una-aplicacion-web> (accedido 15 de noviembre de 2022).
- [5] Edix, «Front End Developer: ¿cuáles son sus funciones y salidas?», *Edix España*, 22 de octubre de 2021. <https://www.edix.com/es/instituto/front-end-developer/> (accedido 15 de noviembre de 2022).
- [6] IeBS, «¿Qué es el prototipado y cómo prototipar un producto?» <https://www.iebschool.com/blog/que-es-el-prototipado-digital-business/> (accedido 15 de noviembre de 2022).
- [7] crehana, « ¿Qué es Figma? Conoce la app para proyectos colaborativos», <https://www.crehana.com>. <https://www.crehana.com/blog/estilo-vida/que-es-figma/> (accedido 15 de noviembre de 2022).
- [8] bewe, «Informática Básica: ¿Qué son las aplicaciones web?», *GCFGlobal.org*. <https://edu.gcfglobal.org/es/informatica-basica/que-son-las-aplicaciones-web/1/> (accedido 15 de noviembre de 2022).
- [9] J. León, «App Web SCRUM», jul. 2014, Accedido: 21 de diciembre de 2022. [En línea]. Disponible en: <https://upcommons.upc.edu/handle/2099.1/23350>
- [10] M. Ramírez, «BEWE.io programa para peluquería: conoce sus funcionalidades». <https://blog.bewe.io/bewe-io-programa-para-peluqueria-conoce-sus-funcionalidades> (accedido 15 de noviembre de 2022).
- [11] Versum, «Por qué debes tener un sistema de cita previa en tu peluquería - Versum», *Versum ES*, 18 de mayo de 2020. <https://www.versum.com/m/es/blog/por-que-sistema-de-cita-previa-en-tu-peluqueria/> (accedido 15 de noviembre de 2022).
- [12] desarrolloweb, «Qué es HTML». <https://desarrolloweb.com/articulos/que-es-html.html> (accedido 22 de noviembre de 2022).
- [13] G. Bravo, «¿Qué es HTML? Explicación de los fundamentos del Lenguaje de marcado de hipertexto», *Tutoriales Hostinger*, 16 de noviembre de 2018. <https://www.hostinger.es/tutoriales/que-es-html> (accedido 22 de noviembre de 2022).

- [14] M. Coppola, «Qué es JavaScript, para qué sirve y cómo funciona». <https://blog.hubspot.es/website/que-es-javascript> (accedido 22 de noviembre de 2022).
- [15] rockcontent, «PHP: ¿qué es, para qué sirve y cuáles son sus características?», *Rock Content - ES*, 9 de marzo de 2020. <https://rockcontent.com/es/blog/php/> (accedido 22 de noviembre de 2022).
- [16] xataka, «API: qué es y para qué sirve». <https://www.xataka.com/basics/api-que-sirve> (accedido 22 de noviembre de 2022).
- [17] G. Lee, «Tipos de pruebas de software: diferencias y ejemplos», *LoadView*, 16 de octubre de 2020. <https://www.loadview-testing.com/es/blog/tipos-de-pruebas-de-software-diferencias-y-ejemplos/> (accedido 27 de noviembre de 2022).
- [18] UNIR, «El despliegue de aplicaciones web, ¿en qué consiste? | UNIR». <https://www.unir.net/ingenieria/revista/despliegue-aplicaciones-web/> (accedido 23 de noviembre de 2022).
- [19] Santander, «Metodologías de desarrollo software | Blog Becas Santander». <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html> (accedido 23 de noviembre de 2022).
- [20] S. Sotomayor, «Las metodologías ágiles más utilizadas y sus ventajas dentro de la empresa», *Think. Innov.*, dic. 2021, Accedido: 23 de noviembre de 2022. [En línea]. Disponible en: <https://www.iebschool.com/blog/que-son-metodologias-agiles-agile-scrum/>
- [21] BBVA, «¿Qué es la metodología “agile”? ¿Revolución de las formas de trabajo?». <https://www.bbva.com/es/metodologia-agile-la-revolucion-las-formas-trabajo/> (accedido 23 de noviembre de 2022).
- [22] Deloitte, «Scrum: roles y responsabilidades | Deloitte España», *Deloitte Spain*. <https://www2.deloitte.com/es/es/pages/technology/articles/roles-y-responsabilidades-scrum.html> (accedido 23 de noviembre de 2022).
- [23] J. Hernandez, «Cuadro comparativo metodologías ágiles - Metodología XP SCRUM Kanban Agile Inception Característica - StuDocu». <https://www.studocu.com/co/document/universidad-cooperativa-de-colombia/ingenieria-de-requisitos/cuadro-comparativo-metodologias-agiles/10231316> (accedido 5 de diciembre de 2022).
- [24] proyectosagiles, «Beneficios de Scrum», *Proyectos Ágiles*, 4 de agosto de 2008. <https://proyectosagiles.org/beneficios-de-scrum/> (accedido 5 de diciembre de 2022).
- [25] QuestionPro, «¿Qué es un estudio de caso y cómo realizarlo?», *QuestionPro*, 10 de junio de 2021. <https://www.questionpro.com/blog/es/que-es-un-estudio-de-caso/> (accedido 23 de noviembre de 2022).
- [26] J. Hurtado, «Cómo funciona la Metodología Scrum: Qué es y cómo utilizarla», *Think. Innov.*, dic. 2021, Accedido: 6 de diciembre de 2022. [En línea]. Disponible en: <https://www.iebschool.com/blog/metodologia-scrum-agile-scrum/>
- [27] J. Meles, «El desafío de ser un Product Owner», presentado en XX Congreso Argentino de Ciencias de la Computación (Buenos Aires, 2014), 2014. Accedido: 27 de noviembre de 2022. [En línea]. Disponible en: <http://sedici.unlp.edu.ar/handle/10915/42283>

- [28] A. Arboleda, «Product Owner: ¿qué es y cuáles son sus funciones?», *Rock Content - ES*, 13 de marzo de 2020. <https://rockcontent.com/es/blog/product-owner/> (accedido 27 de noviembre de 2022).
- [29] X. Campaña, «Qué es la gestión de proyectos Scrum?», 20 de agosto de 2019. <https://www.itconsultors.com/metodologia-scrum> (accedido 28 de enero de 2023).
- [30] proyectosagiles, «Equipo de desarrollo (Development Team)», *Proyectos Ágiles*, 27 de septiembre de 2008. <https://proyectosagiles.org/equipo-team/> (accedido 27 de noviembre de 2022).
- [31] academic, «Artefacto (diseño de software)», *Los diccionarios y las enciclopedias sobre el Académico*. <https://es-academic.com/dic.nsf/eswiki/109351> (accedido 27 de noviembre de 2022).
- [32] Deloitte, «Artefactos Scrum: las 3 herramientas clave de gestión», *Deloitte Spain*. <https://www2.deloitte.com/es/es/pages/technology/articles/artefactos-scrum.html> (accedido 27 de noviembre de 2022).
- [33] lalindri, «9 herramientas para recopilar requisitos para el desarrollo de software», *Blog de Creately*, 29 de diciembre de 2021. <https://creately.com/blog/es/diagramas/herramientas-recopilar-requisitos-software/> (accedido 27 de noviembre de 2022).
- [34] A. Menzinsky, «Historias de Usuario», p. 62.
- [35] A. Cadavid, «Revisión de metodologías ágiles para el desarrollo de software», *Prospectiva*, vol. 11, n.º 2, p. 30, sep. 2013, doi: 10.15665/rp.v11i2.36.
- [36] BeAgile, «En qué consiste el Sprint Backlog», *Be Agile*, 5 de abril de 2020. <https://beagilemyfriend.com/en-que-consiste-el-sprint-backlog/> (accedido 7 de diciembre de 2022).
- [37] Lucidchart, «Cómo diseñar una arquitectura de software: consejos y prácticas recomendadas», *Lucidchart*, 29 de marzo de 2021. <https://www.lucidchart.com/blog/como-disenar-una-arquitectura-de-software> (accedido 7 de diciembre de 2022).
- [38] Platzi, «¿Qué es el patrón MVC?», *Platzi*. <https://platzi.com/tutoriales/1248-pro-arquitectura/5466-que-es-el-patron-mvc/> (accedido 7 de diciembre de 2022).
- [39] M. Narvaez, «Pruebas de usabilidad: Qué son y cuáles son sus beneficios», *QuestionPro*, 1 de febrero de 2022. <https://www.questionpro.com/blog/es/pruebas-de-usabilidad/> (accedido 27 de enero de 2023).
- [40] myservername, «¿Qué son las pruebas de compatibilidad de software? - Otro». <https://spa.myservername.com/what-is-software-compatibility-testing> (accedido 27 de enero de 2023).

7 ANEXOS

En este apartado se muestran los anexos elaborados en conjunto con el desarrollo del presente trabajo de titulación.

- ANEXO I. Certificación de originalidad del proyecto realizado.
- ANEXO II. Manual técnico.
- ANEXO III. Manual de usuario (video funcionalidad del sistema *web*).
- ANEXO IV. Manual de instalación.

ANEXO I

En la presente sección se muestra el certificado otorgado por el director de tesis, donde se puede visualizar la originalidad del proyecto realizado basado en la herramienta del Turnitin.



**ESCUELA POLITÉCNICA NACIONAL
ESCUELA DE FORMACIÓN DE TECNÓLOGOS
CAMPUS POLITÉCNICO "ING. JOSÉ RUBÉN ORELLANA"**

CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 27 de Febrero de 2023

De mi consideración:

Yo, MAYRA ISABEL ALVAREZ JIMÉNEZ, en calidad de directora del Trabajo de Integración Curricular componente SISTEMA WEB, titulado "DESARROLLO DE UN SISTEMA WEB DE AGENDAMIENTO DE CITAS PARA LA PELUQUERIA UNISEX SEBITAS", elaborado por la estudiante GUALLICHICO PAUCAR NATALY MARIUXI de la carrera en DESARROLLO DE SOFTWARE, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito completo, producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 11%.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente documento para los trámites de titulación.

NOTA: Se adjunta el informe generado por la herramienta Turnitin.

Atentamente,



Ing. Mayra Alvarez MSc.
Técnico docente EPN - ESFOT

ANEXO II

En la presente sección de muestra a detalle el trabajo realizado en las secciones anteriores incluye: Recopilación de Requerimientos, Historias de Usuario, *Product Backlog*, *Sprint Backlog*, diseño de prototipos y las pruebas realizadas.

Recopilación de requerimientos

El proceso de recopilación de requerimientos se realizó en base a una encuesta aplicada a 21 clientes de la peluquería Unisex “Sebitas” con la finalidad de obtener un mejor panorama de las necesidades que tienen los usuarios de la menciona peluquería, se adjunta el enlace de la encuesta realizada:

<https://forms.gle/rQraYwTUfKuaMp4d6>

A continuación, se visualiza los resultados obtenidos en base a las preguntas realizadas a los clientes de la peluquería.

La Fig. 50, indica el resultado de la pregunta 1, donde se puede visualizar que el 100% de los encuestados considera que un sistema web para agendar una cita optimizaría su tiempo.

Pregunta 1

¿Crees que un sistema web de agendamiento de citas ayudaría a optimizar tiempo?

19 respuestas

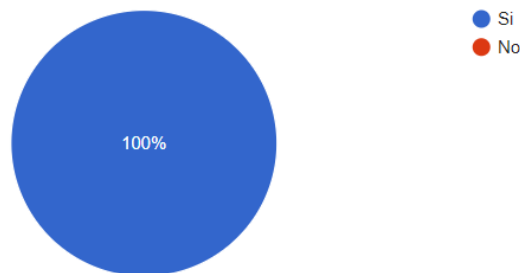


Fig. 50: Encuesta – Pregunta 1

La Fig. 51, indica el resultado de la pregunta 2, donde se puede visualizar que al 84.2% de los encuestados le molesta llegar a la peluquería y tener que esperar para ser atendido.

Pregunta 2

¿Te molesta llegar a la peluquería y esperar a que te atiendan?

19 respuestas

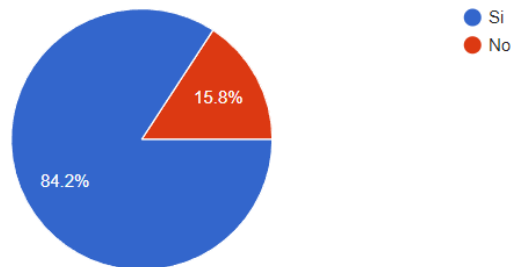


Fig. 51: Encuesta – Pregunta 2

La Fig. 52, indica el resultado de la pregunta 3, donde se puede visualizar que el 68.4% de los encuestados considera que tan importante es poder agendar una cita a través de *Internet*.

Pregunta 3

¿Consideras que es importante poder agendar una cita a través del Internet?

19 respuestas

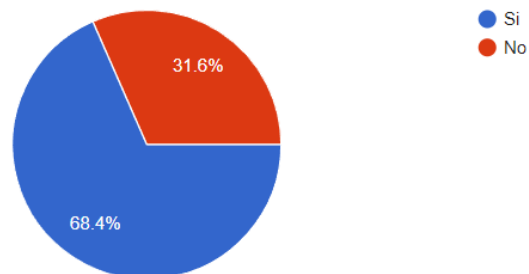


Fig. 52: Encuesta – Pregunta 3

La Fig. 53, indica el resultado de la pregunta 4, donde se puede visualizar que el 78.9% de los encuestados considera que poder agendar una cita en línea sería innovador ya que las personas aún tienen que esperar para ser atendidas.

Pregunta 4

¿Crees que poder agendar citas en línea es innovador?

19 respuestas

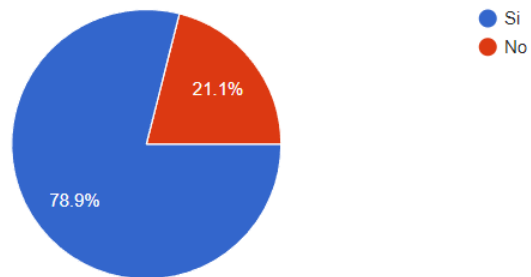


Fig. 53: Encuesta – Pregunta 4

La Fig. 54, indica el resultado de la pregunta 5, donde se puede visualizar que el 78.9% conoce los servicios que ofrece la peluquería, notando así que se debería hacer énfasis a los servicios que se ofertan.

Pregunta 5

¿Conoces los servicios que ofrece la peluquería?

19 respuestas

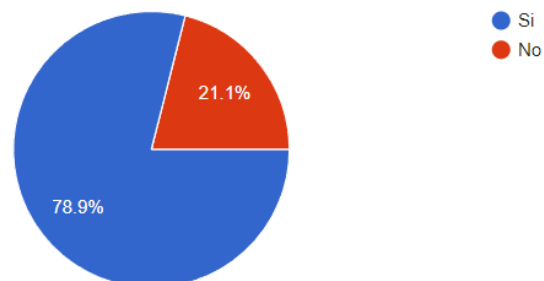


Fig. 54: Encuesta – Pregunta 5

La Fig. 55, indica el resultado de la pregunta 6, donde se puede visualizar que el 100% de los encuestados recomendaría esta peluquería a sus amistades, por ende, si es viable poder ofrecer un sistema *web*.

Pregunta 6

¿Recomendarías esta peluquería a tus amistades?

19 respuestas

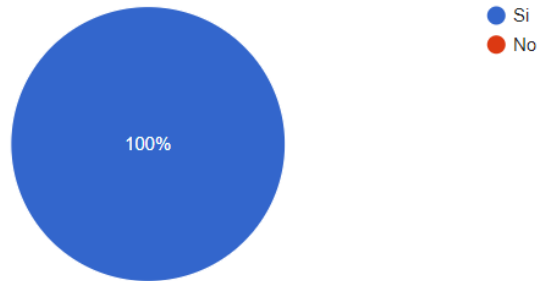


Fig. 55: Encuesta – Pregunta 6

La Fig. 56, indica el resultado de la pregunta 7, donde se puede visualizar que el 73.7% de los encuestados utiliza dicha peluquería con frecuencia.

Pregunta 7

¿Utilizas esta peluquería muy a menudo?

19 respuestas

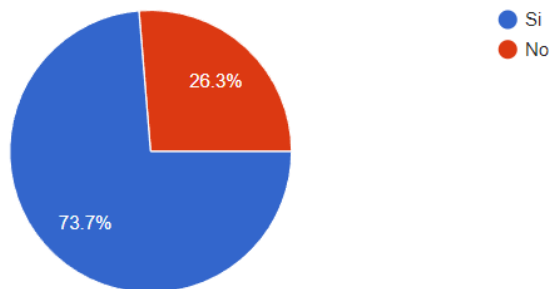


Fig. 56: Encuesta – Pregunta 7

La Fig. 57, indica el resultado de la pregunta 8, donde se puede visualizar que el 68.4% de los encuestados pagaría un valor extra por utilizar un sistema de agendamiento de citas.

Pregunta 8

¿Pagarías un valor extra por utilizar el sistema de agendamiento de citas?

19 respuestas

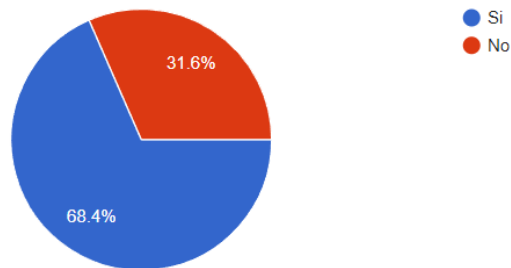


Fig. 57: Encuesta – Pregunta 8

En la **TABLA X**, se puede visualizar los requerimientos obtenidos al comienzo del proyecto donde, concuerda con lo requerido por el dueño del producto.

Tabla X: Recopilación de requerimientos.

Recopilación de requerimientos		
Tipo del sistema	ID - RR	Enunciado del ítem
	RR002	Como usuario administrador y usuario cliente, necesitan. • Visualizar los servicios que oferta el sistema de agendamiento de citas. Finalidad: Conocer los servicios ofertados por la peluquería Unisex “Sebitas”.
	RR003	Como usuario administrador necesita. • Consumir varios métodos del sistema de agendamiento de citas. Finalidad: Visualizar las citas agendadas y los horarios que están disponibles.
	RR004	Como usuario administrador necesita. • Buscar de acuerdo con las fechas las citas agendadas. Finalidad: Conocer el número de citas que se atenderán durante el día o los días siguientes.
	RR005	Como usuario cliente necesita.

		<ul style="list-style-type: none"> • Registrarse en el sistema web a través de un formulario. <p>Finalidad: Identificarme dentro del sistema web y navegar dentro de los servicios ofertados.</p>
	RR006	<p>Como usuario cliente necesita.</p> <ul style="list-style-type: none"> • Identificarme dentro del sistema <i>web</i> a través de las credenciales ingresadas en el formulario de registro. <p>Finalidad: Iniciar sesión y cerrar sesión, además de poder reestablecer la contraseña.</p>
	RR007	<p>Como usuario cliente necesita.</p> <ul style="list-style-type: none"> • Agendar una cita dentro del sistema web. <p>Finalidad: Ser atendido en el establecimiento a la hora y fecha agendada.</p>
	RR008	<p>Como usuario cliente necesita.</p> <ul style="list-style-type: none"> • Visualizar los datos de la cita agendada dentro del sistema web. <p>Finalidad: Verificar que los datos ingresados sean correctos.</p>
	RR009	<p>Es necesario verificar el correcto funcionamiento del sistema <i>web</i> integrando todos los módulos desarrollados, además de realizar el despliegue del sistema.</p>

Historias de Usuario

Ahora que ha finalizado la recopilación de requerimientos, con la información obtenida se puede elaborar las distintas historias de usuario que se implementara en el sistema *web*. Por tal razón, se presenta las 09 Historias de Usuario basadas en los requerimientos obtenidos que empieza desde la **TABLA XI** hasta la **TABLA XVIII**.

TABLA XI: Historia de usuario 02 – Visualizar los servicios.

HISTORIA DE USUARIO	
Identificador: HU002	Usuario: Administrador y Cliente
Nombre historia: Visualizar los servicios.	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Media

Iteración asignada: 2
Responsable: Nataly Guallichico
Descripción: El usuario administrador y el usuario cliente necesitan visualizar los servicios ofertados por la peluquería Unisex “Sebitas”. En la sección citas dentro del sistema <i>web</i> , en el módulo “SERVICIOS”, se puede visualizar todos los servicios ofertados.
Observación: Ninguna.

TABLA XII: Historia de usuario 03 – Visualizar citas agendadas.

HISTORIA DE USUARIO	
Identificador: HU003	Usuario: Administrador
Nombre historia: Visualizar citas agendadas.	
Prioridad en Negocio: Media	Riesgo en Negocio: Alta
Iteración asignada: 1	
Responsable: Nataly Guallichico	
Descripción: El usuario administrador necesita visualizar las citas agendadas. En la sección citas dentro del sistema <i>web</i> , en el módulo “Ver Citas”, se puede visualizar las citas agendadas.	
Observación: El usuario administrador será el único que podrá visualizar a este módulo.	

TABLA XIII: Historia de usuario 04 – Buscar citas.

HISTORIA DE USUARIO	
Identificador: HU004	Usuario: Administrador
Nombre historia: Buscar citas.	
Prioridad en Negocio: Baja	Riesgo en Negocio: Media
Iteración asignada: 1	
Responsable: Nataly Guallichico	
Descripción: El usuario administrador necesita buscar las citas agendadas de acuerdo con la hora y fecha agendada. En la sección citas dentro del sistema <i>web</i> , en el módulo	

“Panel de Administrador”, se puede buscar las citas agendadas de acuerdo con una fecha asignada.
Observación: El usuario administrador será el único que podrá visualizar a este módulo.

TABLA XIV: Historia de usuario 05 – Registro de cliente.

HISTORIA DE USUARIO	
Identificador: HU005	Usuario: Cliente
Nombre historia: Registro de cliente.	
Prioridad en Negocio: Media	Riesgo en Negocio: Alta
Iteración asignada: 3	
Responsable: Nataly Guallichico	
Descripción: El usuario cliente necesita registrarse dentro del sistema <i>web</i> a través de un formulario. En el apartado de “Crear Cuenta” se abrirá un formulario donde el usuario Cliente deberá ingresar sus datos para ser registrado dentro del sistema.	
Observación: El usuario cliente debe ingresar un correo valido debido a que se enviará a ese correo la verificación de la cuenta creada, en el caso de que el usuario cliente no otorgue ese permiso no será autenticado dentro del sistema.	

TABLA XV: Historia de usuario 06 – Inicio y cierre de sesión, además de reestablecer la contraseña.

HISTORIA DE USUARIO	
Identificador: HU006	Usuario: Cliente
Nombre historia: Inicio y cierre de sesión, además reestablecer contraseña.	
Prioridad en Negocio: Alta	Riesgo en Negocio: Alta
Iteración asignada: 2	
Responsable: Nataly Guallichico	
Descripción: El usuario cliente necesita un inicio y un cierre de sesión, adicional para reestablecer su contraseña con la finalidad de identificarse dentro del sistema <i>web</i> . El	

usuario cliente deberá llenar un formulario para poder identificarse dentro del sistema logrando así obtener una cuenta dentro del sistema <i>web</i> .
Observación: El usuario cliente necesita registrarse para que pueda iniciar y cerrar sesión, además de solicitar reestablecer la contraseña.

TABLA XVI: Historia de usuario 07 – Agendar una cita dentro del sistema *web*.

HISTORIA DE USUARIO	
Identificador: HU007	Usuario: Cliente
Nombre historia: Agendar una cita dentro del sistema <i>web</i> .	
Prioridad en Negocio: Media	Riesgo en Negocio: Media
Iteración asignada: 1	
Responsable: Nataly Guallichico	
Descripción: El usuario cliente necesita agendar una cita dentro del sistema <i>web</i> . Una vez que inicie sesión el usuario Cliente podrá agendar una cita y posterior a eso podrá ver la información de la cita agendada.	
Observación: Ninguna	

TABLA XVII: Historia de usuario 08 – Visualizar los datos de la cita agendada.

HISTORIA DE USUARIO	
Identificador: HU008	Usuario: Cliente
Nombre historia: Visualizar los datos de la cita agendada.	
Prioridad en Negocio: Media	Riesgo en Negocio: Media
Iteración asignada: 1	
Responsable: Nataly Guallichico	
Descripción: El usuario cliente necesita visualizar los datos de la cita agendada. Una vez que el usuario Cliente seleccione los servicios que necesita y elija la hora y fecha, en la sección “RESUMEN” podrá visualizar los datos de la cita agendada.	
Observación: El usuario cliente puede verificar los datos de la cita agendada.	

TABLA XVIII: Historia de usuario 09 – Pruebas y despliegue del sistema web.

HISTORIA DE USUARIO	
Identificador: HU009	Usuario: Todos
Nombre historia: Pruebas y despliegue del sistema <i>web</i> .	
Prioridad: Alta	Riesgo: Alta
Iteración asignada: 2	
Responsable: Nataly Guallichico	
Descripción: El sistema <i>web</i> debe funcionar acorde a requerimientos recopilados, para verificación de que dichos requerimientos se cumplan se realizan pruebas al sistema, siendo la prueba final la de integración para lo cual se realiza el despliegue.	
Observación: Verificación del funcionamiento del sistema <i>web</i> acorde a requerimientos recopilados en la fase inicial del proyecto.	

Product Backlog

La TABLA XIX, indica la preferencia de acuerdo con cada requerimiento obtenido dentro del sistema *web*, los mismo se clasifican de acorde con su prioridad y nivel de complejidad.

TABLA XIX: Product Backlog.

ELABORACIÓN DEL PRODUCT BACKLOG				
ID – HU	HISTORIA DE USUARIO	ITERACIÓN	ESTADO	PRIORIDAD
HU003	Visualizar las citas agendadas.	2	Finalizada	Alta
HU004	Buscar citas.	1	Finalizada	Baja
HU005	Registro de cliente.	3	Finalizada	Media
HU006	Inicio y cierre de sesión, reestablecer contraseña.	2	Finalizada	Alta
HU007	Agendar una cita dentro del sistema <i>web</i> .	1	Finalizada	Media
HU008	Visualizar los datos de la cita	1	Finalizada	Media

	agendada.			
HU009	Pruebas y despliegue del sistema <i>web</i>	1,2,3	Finalizada	Alta

Sprint Backlog

La **TABLA XX**, muestra las actividades que se ejecutaran en cada uno de los sprint para el desarrollo del sistema *web*, cada tarea realizada ha sido determinada a través de *Product Backlog*.

TABLA XX. Sprint Backlog

ID-SB	Nombre	Historia de Usuario	Tareas	Tiempo Estimado (horas)
S0	Configuración del ambiente de desarrollo	N/A	<ul style="list-style-type: none"> • Instalación tanto de extensiones como de paquetes para el buen funcionamiento del editor de texto utilizado. • Creación de prototipado. • Estructura del proyecto. • Creación de proyecto en GitHub. 	40 h
S1	Autenticación	HU-001. Inicio y cierre de sesión. HU-005. Registro de cliente. HU-006. Inicio y cierre de sesión, además de reestablecer contraseña.	<ul style="list-style-type: none"> • Delimitar los parámetros para el registro y para el inicio de sesión. • Delimitar reglas de comprobación. • Realizar un registro, un inicio de sesión y reestablecer contraseña. • Verificación de que el mismo correo no se repita y que este registrado. • Verificación de campos obligatorios. 	50h
S2	Módulo de espacio virtual e información.	HU-002. Visualizar los servicios.	<ul style="list-style-type: none"> • Presentar los servicios ofertados por la peluquería. 	50 h

		HU-003. Visualizar las citas agendadas.	<ul style="list-style-type: none"> • Acceder al módulo de citas agendadas. • Validar el horario de citas disponibles. • Validar que el cliente tenga una cuenta para poder agendar la cita. 	
		Hu-007. Agendar una cita dentro del sistema <i>web</i> .		
S3	Módulo complementario	HU-004. Buscar citas.	<ul style="list-style-type: none"> • Buscar las citas agendadas a través de fecha en el calendario. • Verificar que los datos ingresados para el agendamiento de cita sean correctos. • Validar que todos los campos estén llenos correctamente y que ningún campo quede en blanco. 	40 h
		HU-008. Visualizar los datos de la cita agendada.		
S4	Pruebas y Despliegue	HU-009. Pruebas y despliegue del sistema <i>web</i>	<ul style="list-style-type: none"> • Pruebas unitarias • Pruebas de usabilidad. • Pruebas de compatibilidad. • Despliegue 	20 h
Documentación				40 h
Total de horas				240

Diseño de prototipos

El prototipado realizado para el desarrollo del sistema web se realizó con la herramienta de Figma y se la puede encontrar a través del enlace presentado a continuación:

<https://www.figma.com/file/fJqcyGxADWa4nfjez86mhK/AppSebitas?node-id=0%3A1&t=sCHSo3Pt7oSVyS5C-0>

La Fig. 58, muestra la vista de validación de campos del inicio de sesión ya que si algún campo está vacío o los datos ingresados son incorrectas el usuario cliente/administrador no podrá iniciar sesión.

Fig. 58: Prototipado validación de campos de Login

La Fig. 59, muestra la vista del registro de usuarios que se indica al usuario cliente para que pueda registrarse dentro del sistema *web* a través de un formulario.

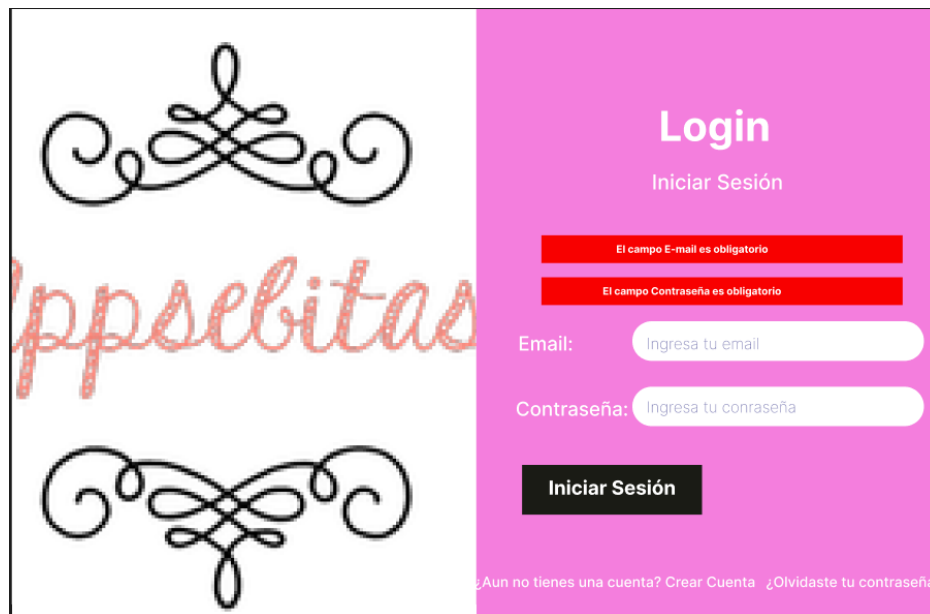


Fig. 59: Prototipado del registro del Sistema *web*.

La Fig. 60, evidencia la vista de validación de campos del registro de sesión ya que si algún campo está vacío el usuario cliente no se podrá registrar dentro del sistema *web*.



Fig. 60: Prototipado de validación de campos de registro del Sistema *web*

La Fig. 61, evidencia la vista de reestablecer contraseña, donde el usuario cliente podrá reestablecer su contraseña a través de un mensaje de reestablecer contraseña enviado a su correo electrónico.



Fig. 61: Prototipado de reestablecer contraseña del sistema *web*

La Fig. 62, evidencia la vista de los servicios para crear una cita, donde el usuario podrá visualizar los servicios ofertados por el sistema *web*.



Fig. 62: Prototipado de los Servicios ofertados por el Sistema *web*

La Fig. 63, evidencia la vista de la información para crear una cita, donde el usuario podrá seleccionar la fecha y hora que necesita la cita dentro el sistema *web*.



Fig. 63: Prototipado de información de la cita del Sistema *web*

La Fig. 64, evidencia la vista del resumen de la cita agenda por el usuario cliente, donde se detalle los datos del cliente y la información acerca de la cita agendada dentro del sistema web.



Fig. 64: Prototipado del resumen de la cita del Sistema web

La Fig. 65, evidencia la vista del panel del administrador, donde podrá visualizar las citas agendadas de acuerdo con su fecha y eliminar las citas que ya han sido atendidas dentro del sistema web.

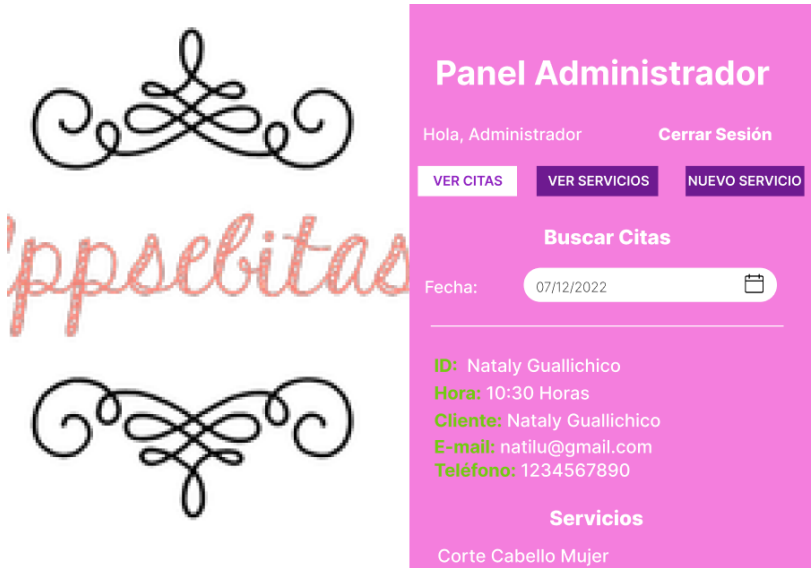


Fig. 65: Prototipado del Panel del Administrador del Sistema web

Pruebas del sistema web

Ahora que se ha finalizado la codificación del sistema web se continuo con la realización de las pruebas de usabilidad, aceptación y compatibilidad.

Pruebas de usabilidad

En este apartado se muestra los resultados de la encuesta que se realizó a los usuarios, en donde según el método de SUS y en base a sus 10 preguntas se debe sumar el valor de las respuestas de las preguntas de los números impares y se les debe restar 5, por otro lado, en las preguntas pares se debe sumar el valor de cada respuesta y se deberá restar 25 – el valor obtenido y a la suma de las 2 operaciones anteriores se le debe multiplicar por 2.5. La Tabla XXII, muestra los resultados obtenidos de los 21 usuarios a quienes se les realizo la encuesta con el método SUS.

El enlace de la encuesta realizada se encuentra a continuación:

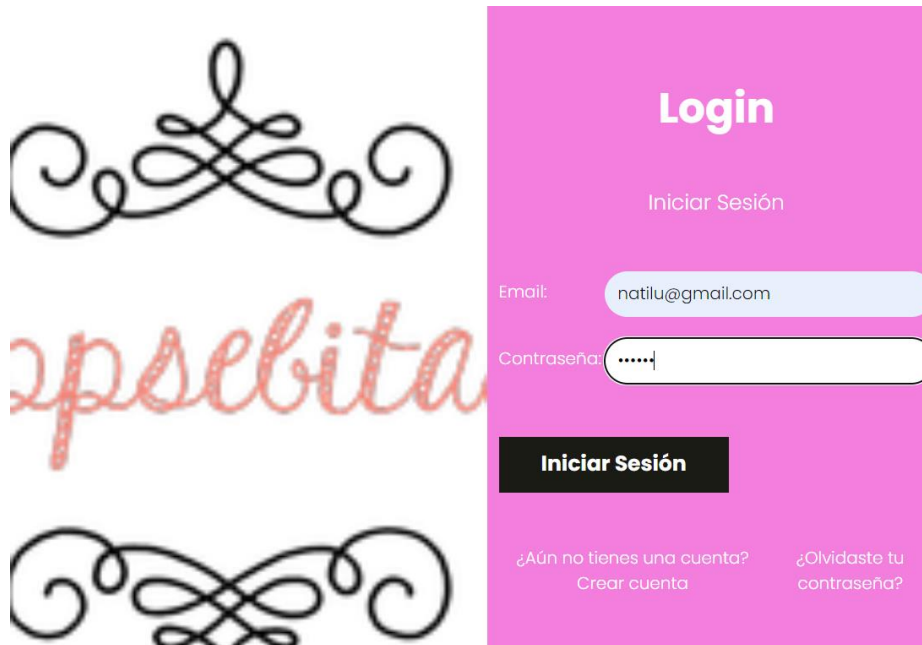
<https://forms.gle/YBAyvnCYbudoaFoc8>

Tabla XXII: Resultados método SUS.

No.-	impares	pares	resulimpares	resul pares	total sus
1	21	5	16	20	90
2	24	12	19	13	80
3	16	7	11	18	72,5
4	19	10	14	15	72,5
5	20	11	15	14	72,5
6	24	11	19	14	82,5
7	20	8	15	17	80
8	20	9	15	16	77,5
9	25	9	20	16	90
10	17	7	12	18	75
11	24	14	19	11	75
12	17	8	12	17	72,5
13	22	5	17	20	92,5
14	22	13	17	12	72,5
15	22	9	17	16	82,5
16	18	6	13	19	80
17	22	11	17	14	77,5
18	23	6	18	19	92,5
19	25	6	20	19	97,5
20	25	7	20	18	95
21	25	6	20	19	97,5
22	22	8	17	17	85
				promedio:	82,3863636

Pruebas de compatibilidad

En este apartado se muestra los resultados obtenidos de las pruebas de compatibilidad, desarrollado en los navegadores Google Chrome, Microsoft Edge y Brave Browser, mismos que se pueden visualizar desde la Fig. 66, hasta la Fig. 83 del presente documento.



Login

Iniciar Sesión

Email: natilu@gmail.com

Contraseña:

Iniciar Sesión

¿Aún no tienes una cuenta? [Crear cuenta](#)

¿Olvidaste tu contraseña?

Fig. 66: Login de Usuario desde Google Chrome.



Crear Cuenta

Llena el siguiente formulario para obtener tu cuenta

Nombre: Ingresar tu nombre

Apellido: Ingresar tu apellido

Teléfono: Ingresar tu teléfono

E-mail: Ingresar tu e-mail

Contraseña: Ingresar tu contraseña

Crear Cuenta

Fig. 67: Registro de usuarios desde Google Chrome.

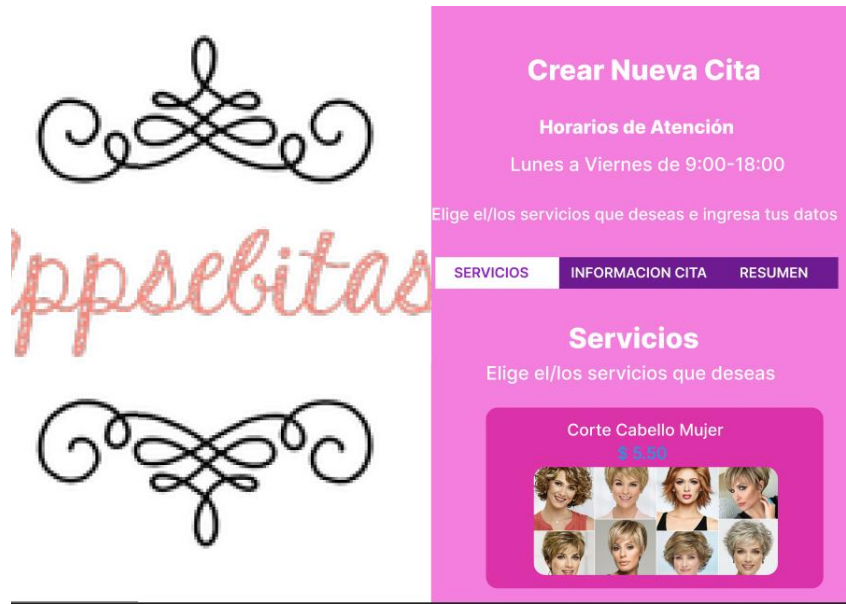


Fig. 68: Módulo de servicios desde Google Chrome.



Fig. 69: Módulo de agendamiento de citas desde Google Chrome.



Fig. 70: Módulo de resumen desde Google Chrome.



Fig. 71: Panel de administrador desde Google Chrome.



Fig. 72: Login de usuario desde Microsoft Edge.



Fig. 73: Registro de usuario desde Microsoft Edge.



Fig. 74: Módulo de servicios desde Microsoft Edge.



Fig. 75: Módulo de agendamiento de citas de Microsoft Edge.



Fig. 76: Módulo de resumen desde Microsoft Edge.



Fig. 77: Panel de administrador desde Microsoft Edge.



apsebita



Login

Iniciar Sesión

Email:

Contraseña:

Iniciar Sesión

[¿Aún no tienes una cuenta?
Crear cuenta](#) [¿Olvidaste tu
contraseña?](#)

Fig. 78: Login de usuario desde Brave Browser.



apsebita



Crear Cuenta

Llena el siguiente formulario para obtener tu cuenta

Nombre:

Apellido:

Teléfono:

E-mail:

Contraseña:

Crear Cuenta

Fig. 79: Registro de usuario desde Brave Browser.

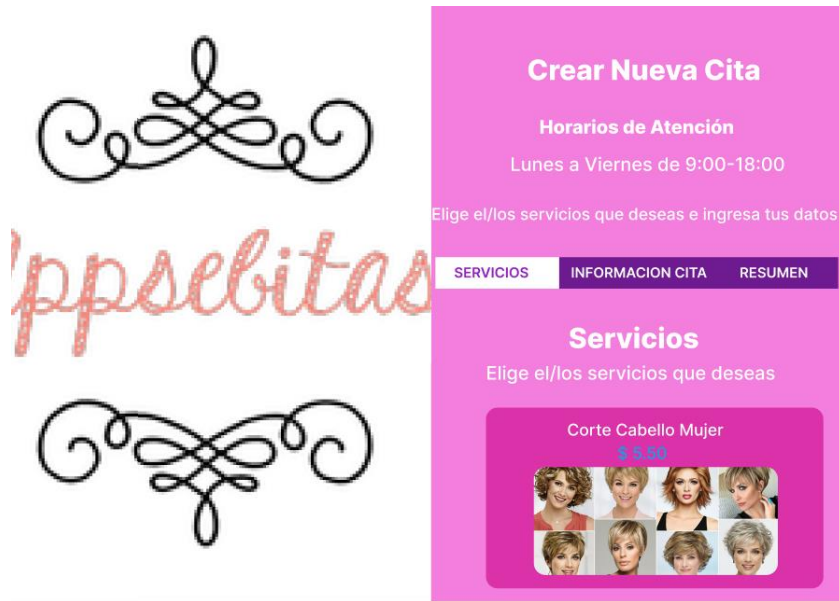


Fig. 80: Módulo de servicios desde Brave Browser.



Fig. 81: Módulo de agendamiento de citas desde Brave Browser.



Fig. 82: Módulo de resumen desde Brave Browser.



Fig. 83: Panel de administrador desde Brave Browser.

Despliegue del sistema web

El despliegue del presente sistema web, se lo realizo en la plataforma de heroku en donde luego de iniciar sesión se procedió con la configuración de las variables de entorno dentro

del sistema con se evidencia en la Fig. 84, y la modificación del archivo .env del proyecto como se evidencia en la Fig. 85.

CLEARDB_DATABASE_URL	[REDACTED]
DB_BD	[REDACTED]
DB_HOST	[REDACTED]
DB_PASS	[REDACTED]
DB_USER	[REDACTED]

Fig. 84: Heroku - variables

```
7 DB_HOST=[REDACTED]
8 DB_USER=[REDACTED]
9 DB_PASS=[REDACTED]
10 DB_BD=[REDACTED]
```

Fig. 85: Credenciales .env

ANEXO III

En este apartado se puede evidenciar el manual de usuario que corresponde al enlace del video elaborado para la explicación del funcionamiento del presente sistema *web*. Donde, se puede visualizar los módulos funcionales y actividades que realizan los usuarios acorde al perfil.

- Url del video en *YouTube*: <https://youtu.be/o6SeuqI9I2Y>

ANEXO IV

En esta sección se encuentra el enlace del repositorio alojado en GitHub donde está almacenado el código del sistema y un archivo de nombre *readme* que detalla los pasos de instalación del sistema de manera local. Además, se proporcionan las credenciales de usuario administrador y cliente para el acceso al sistema *web* desplegado.

Repositorio en GitHub

La url donde se evidencia el código almacenado del proyecto es el siguiente:

<https://github.com/NatalyGuallichico01/appSebitasWebFinal.git>

Sistema *web* desplegado

El presente enlace lleva al sistema web que ha sido desplegado a producción:

<https://appsebitasweb.herokuapp.com>

Credenciales administrador:

- Usuario: admin@admin.com
- Contraseña: 123456

Credenciales del usuario cliente:

- Usuario: natilu.naty@gmail.com
- Contraseña: 123456