

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

**DESARROLLO DE SISTEMA WEB INFORMATIVO DE ASILOS EN LA CIUDAD
DE QUITO**

DESARROLLO DE UN SISTEMA

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

JOSE ANTONIO BORJA FUELTALA

DIRECTOR: BYRON GUSTAVO LOARTE CAJAMARCA

DMQ, septiembre 2022

CERTIFICACIONES

Yo, Jose Antonio Borja Fweltala 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.



Jose Antonio Borja Fweltala

jose.borja@epn.edu.ec

jose.borja_@hotmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por Jose Antonio Borja Fweltala, bajo mi supervisión.



Ing. Byron Loarte, MSc.

DIRECTOR

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.

Jose Antonio Borja Fweltala

DEDICATORIA

En primer lugar, quiero dedicar este proyecto a Dios, por darme la fuerza de no darme por vencido por permitirme culminar una etapa de mi vida, también a mi mamá y mi hermana que estuvieron apoyándome a cada momento por darme ánimos para culminar mi carrera universitaria.

En segundo lugar, quiero agradecer al Ing. Byron Loarte, por la paciencia el apoyo brindado en el proceso de la elaboración del proyecto, como también a todos los docentes que en cada semestre han sabido tener paciencia y sin su ayuda y entrega no podría ser posible culminar este proyecto. También a mis compañeros, que estuvieron en cada momento difíciles y brindarme su tiempo y apoyo.

Jose Antonio Borja Fweltala

AGRADECIMIENTO

En primer lugar, dar gracias a Dios, por siempre darme cada día fuerzas para salir adelante y permitirme llegar hasta este punto de mi vida. A mi madre y mi hermana, les agradezco por que han sido un apoyo indispensable en toda mi carrera universitaria por estar a mi lado apoyarme aconsejarme darme ánimos para no rendirme.

A todos mis compañeros, que han estado en las buenas y malas dándome consejos y animándome y ayudarme en los momentos difíciles

A todos los docentes, que me han impartido conocimiento su apoyo y dedicación las cuales han sido una parte muy importante de mi formación académica y gracias a ellos me forman para una nueva etapa de mi vida.

Agradecer al Ing. Byron Loarte, por su paciencia por su dedicación, por apoyarme a culminar esta etapa de mi formación académica y en especial por ayudarme a culminar este proyecto.

Jose Antonio Borja Fweltala

ÍNDICE DE CONTENIDO

CERTIFICACIONES	II
DECLARACIÓN DE AUTORÍA	III
DEDICATORIA	IV
AGRADECIMIENTO	V
ÍNDICE DE CONTENIDO.....	VI
RESUMEN.....	VIII
ABSTRACT.....	IX
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO	1
1.1 Objetivo general	2
1.2 Objetivos específicos	2
1.3 Alcance	2
1.4 Marco Teórico	4
2 METODOLOGÍA	6
2.1 Metodología de Desarrollo	6
Roles.....	7
Artefactos.....	8
2.2 Diseño de interfaces	11
Herramienta utilizada para el diseño	11
2.3 Diseño de la arquitectura	12
Patrón arquitectónico	12
2.4 Herramientas de desarrollo	13
3 RESULTADOS.....	15
<i>Sprint 0.</i> Configuración del ambiente de desarrollo.	15
<i>Sprint 1.</i> Diseño e implementación de los módulos para el usuario administrador.	18
<i>Sprint 2.</i> Diseño e implementación de los módulos para el usuario administrador del asilo.	21
<i>Sprint 3.</i> Diseño e implementación de los módulos para el usuario ciudadano.	25
<i>Sprint 4.</i> Pruebas del sistema web.....	26
<i>Sprint 5.</i> Despliegue del sistema <i>web</i>	30
4 CONCLUSIONES	32
5 RECOMENDACIONES	33
6 REFERENCIAS BIBLIOGRÁFICAS	34
7 ANEXOS.....	38

ANEXO I	39
ANEXO II	40
ANEXO III	71
ANEXO IV.....	72

RESUMEN

La pandemia del COVID-19 ha afectado en gran medida a todo el mundo y en especial a nuestro país ya que fueron afectados diferentes grupos de personas e incluso llevándolos a la muerte. Además, los más afectados siguen siendo las personas de tercera edad que son muy propensas a contagiarse más rápido y con el retorno de las actividades presenciales la ciudadanía en algunos casos requiere la contratación de los servicios de asilos para que realicen el cuidado de estas personas mayores, sin embargo, esta información no es fácilmente accedida ya que actualmente no existe un medio que permita fácilmente encontrar dicha información.

Con el fin de ayudar a la ciudadanía, en este proyecto de integración curricular se ha desarrollado un sistema *web* que les permita obtener información de todos los asilos que se encuentran disponibles dentro de la ciudad de Quito. Adicional a ello, toda la gestión de la información se lo realiza gracias a una serie de perfiles previamente establecidos, para que de esta manera toda la ciudadanía pueda obtener información siempre actualizada y en tiempo real gracias al uso de la tecnología y herramientas de desarrollo modernas.

El presente documento está diseñado de la siguiente manera, la primera parte está orientada en dar a conocer la problemática, objetivos, alcance y el respectivo marco teórico. En la segunda parte, se especifica como se ha realizado la implementación de la metodología *Scrum*, prototipos y herramientas en el desarrollo del sistema *web*, como también *Firebase* para la creación de la base de datos (NoSQL). En la tercera sección se presenta los resultados que se han obtenido a través de la realización de cada una de las tareas de los *Sprint* y el despliegue a producción respectivamente. Finalmente, en la cuarta sección se especifican las conclusiones y recomendaciones que se han obtenido en todo el desarrollo de este proyecto de Integración Curricular.

PALABRAS CLAVE: *Firebase, Sprint, Scrum, Covid-19, asilos, sistema web*

ABSTRACT

The COVID-19 pandemic has greatly affected the entire world and especially our country since different groups of people were affected and even leading to death. In addition, the most affected continue to be the elderly who are very likely to get infected faster and with the return of face-to-face activities, citizens in some cases require the hiring of nursing home services to carry out the care of these elderly people, however, this information is not easily accessed since there is currently no means to easily find such information.

In order to help citizens, in this Curriculum Integration project, a web system has been developed that allows them to obtain information on all the nursing homes that are available within the city of Quito. In addition to this, all information management will be carried out from a series of previously established profiles, so that in this way all citizens can obtain information that is always up-to-date and in real time thanks to the use of technology and modern development tools.

This document is designed as follows, the first part is oriented to publicize the problem, objectives, scope and the respective theoretical framework. In the second part, it is specified how the implementation of the Scrum methodology, prototypes and tools in the development of the web system has been carried out, as well as Firebase for the creation of the database (NoSQL). The third section presents the results that have been obtained through the completion of each of the Sprint tasks and the deployment to production, respectively. Finally, the fourth section specifies the conclusions and recommendations that have been obtained throughout the development of this Curriculum Integration project.

KEYWORDS: Firebase, Sprint, Scrum, Covid-19, nursing homes, web system

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

El COVID-19 ha tenido y sigue recayendo en las personas mayores de edad [1], en Ecuador se registraron 871,029 casos confirmados por COVID-19. Además, se registran 196,441 casos probables de ser COVID-19 a la espera de su resultado y en alrededor de 24 horas al momento que se realizaron 644 pruebas de ellas fueron positivas [2]. Por otra parte, la forma más clara que una persona mayor se puede ver infectada por el COVID-19 depende de su salud física como mental, las personas mayores de edad tienen una mayor tasa de contagio por lo que se requiere de cuidados especiales [3]. Actualmente en la ciudad de Quito, existen un sin número de asilos lo cuales cuentan con medidas de bioseguridad para evitar que existan más contagios, pero sobre todo cuidar del bienestar de los adultos mayores. La Organización Mundial de la Salud (OMS) señaló que los adultos mayores y las personas con enfermedades crónicas son más propensas a experimentar síntomas severos si contraen la enfermedad e incluso riesgo de mortalidad [4].

No obstante, con el retorno de las actividades presenciales la ciudadanía se ha visto en la necesidad de optar por la contratación de un asilo y de sus servicios, pero no existe un medio donde se encuentre de forma centralizada toda la información de los asilos que se encuentran en la ciudad de Quito, de tal manera que una oportunidad u opción para dar a conocer a un asilo ya sea grande o pequeño es por medio de un sistema *web* informativo el cual brinde toda la información de manera oportuna, actualizada y en tiempo real de todos los asilos que pertenecen a la ciudad de Quito [5].

En la actualidad un sistema *web* se lo puede considerar un medio tecnológico que permita compartir información en tiempo real, actualizada y que pueda ser visualizada por cualquier persona a cualquier hora del día [6]. Sin embargo, al momento de crear un sistema *web* se debe tomar en cuenta la comunicación que debe existir entre el *frontend*, *backend*, Base de datos, librerías, etc. El objetivo de esto es obtener un sistema robusto y que fácilmente pueda escalar con el pasar del tiempo [7]. Es por esta razón, que en la actualidad un sistema *web* tiene cualidades únicas y distintivas ya que interactúan con otras partes de *software* para asegurar a los usuarios una correcta funcionalidad y que los mismos puedan interactuar de una forma correcta con la información almacenada.

Ante la problemática expuesta y con el objetivo de que la ciudadanía pueda obtener un medio digital con información actualizada y en tiempo real, en el presente Trabajo de Integración Curricular se ha desarrollado un sistema *web* para presentar información de los asilos que existen en la ciudad de Quito. No obstante, dentro del sistema que se ha desarrollado se cuenta con un perfil administrador, el cual permite gestionar a cada uno de

los asilos que desean formar parte de este sistema *web*. Mientras que el usuario con perfil fundación, tiene la posibilidad de gestionar toda la información relacionada a su negocio como: perfil de usuario, horario de atención, ubicación, servicios, contactos, entre otros. Otorgando de esta manera a la ciudadanía un medio de conexión más cercana entre asilos y sus determinadas áreas de servicio gracias al uso de la tecnología y herramientas de desarrollo modernas.

1.1 Objetivo general

Desarrollar un sistema *web* informativo de asilos en la ciudad de Quito.

1.2 Objetivos específicos

1. Determinar los requisitos para el sistema *web* tanto funcionales y no funcionales
2. Diseñar los prototipos en base a los requerimientos que se han obtenido.
3. Diseñar el modelo de la Base de datos en base a los requerimientos que se han obtenido.
4. Codificar el sistema *web* en base al lenguaje de programación y cada uno de los *Sprints* que se han obtenido.
5. Verificar el funcionamiento del sistema *web* a través de una serie de pruebas y el respectivo despliegue a producción.

1.3 Alcance

El objetivo del presente proyecto es dotar a la ciudadanía un sistema *web* que les permita obtener información de todos los asilos que se encuentran disponibles dentro de la ciudad de Quito. Adicional a ello, toda la gestión de la información se lo realiza por medio de una serie de diferentes módulos que han sido previamente establecidos, para que de esta manera toda la ciudadanía pueda obtener información siempre actualizada y si fuera el caso contratar sus servicios respectivos.

Por la actual situación que está atravesando el país, el ofrecer servicios mediante plataformas digitales en Internet aumenta considerablemente las ventas ya que se tiene acceso a una mayor cantidad de clientes que puedan visualizar en tiempo real toda la información inherente al negocio cuando lo requieran y desde cualquier dispositivo tecnológico [8].

Por otra parte, en lo que respecta a la parte del *Frontend* del sistema *web* se garantiza la integridad, consistencia y seguridad de los datos con el *backend* para tener un mejor control y una comunicación eficiente. Todo ello, gracias a la utilización de herramientas de desarrollo actuales y modernas, teniendo en cuenta su escalabilidad y robustez. Además, para la implantación del sistema *web* se ha utilizado la metodología *Scrum* para poder entregar iteraciones en un tiempo determinado y cumplir de esta manera con los objetivos que se han propuesto. Adicional a ello, se ha utilizado una herramienta para la elaboración de las interfaces, un patrón arquitectónico, una serie de pruebas para verificar el correcto funcionamiento y una etapa final para realizar el despliegue a producción respectivamente [9].

Por último, el sistema *web* dispone de una serie de perfiles y en cada uno de ellos asociada las tareas que pueden realizar.

Perfiles del sistema *web*

- Administrador principal.
- Administrador del asilo.

El perfil administrador en el sistema *web* permite:

- Iniciar o cerrar sesión.
- Actualizar su perfil.
- Aprobar o rechazar la afiliación de nuevos asilos.
- Eliminar asilos que se han registrado.

El perfil administrador del asilo en el sistema *web* permite:

- Registrarse por medio de un formulario.
- Iniciar o cerrar sesión.
- Actualizar su perfil.
- Gestionar toda la información del asilo.

El usuario con perfil invitado en el sistema *web* permite:

- Visualización de asilos.
- Realizar una proforma de servicios.

1.4 Marco Teórico

En la actualidad, la sociedad tiene el deber de ser solidaria y proteger a las personas mayores, siendo estos uno de los grupos más afectados por la pandemia del COVID-19 debido a que los adultos mayores se les dificulta realizar ciertas tareas y son más propensos a contagiarse o enfermarse [10]. Durante la pandemia algunas de las personas de la tercera edad fueron abandonadas en las calles y otras se quedaron al cuidado de los mismos familiares, pero con el retorno de las actividades presenciales se ha dificultado el cuidado de estas personas mayores por lo que prefieren contratar servicios de terceros para su cuidado y bienestar [11].

El *Framework* Angular, permite desarrollar aplicaciones web escalables y separar la comunicación con el *backend* evitando de esta manera escribir código repetitivo gracias a la integración del patrón de arquitectura MVC (Modelo-Vista-Controlador). Por otra parte, el lenguaje principal de programación que utiliza este maravilloso *Framework* es *TypeScript*, el cual permite que toda la sintaxis sea fácil de codificar e interpretar [12].

Firebase por otra parte, es una plataforma creada por la empresa de Google, teniendo como principal función desarrollar y facilitar la creación de aplicaciones para dispositivos móviles y web. Gracias al uso de varios servicios que proporciona esta plataforma en su capa gratuita. Además, esta plataforma se encuentra alojada en la nube y, por ende, está disponible para diferentes plataformas como Android, iOS, y web. Así mismo, cuenta con diversas funciones para que cualquier desarrollador pueda combinar y adaptar la plataforma a medida de sus necesidades, se puede decir que es un *backend* previamente desarrollado con todas las funcionalidades listas para ser consumidas [13].

Angular Material, es un *Framework* el cual permite agregar estilos y diseños a una aplicación en Angular gracias al uso de componentes previamente desarrollados. Logrando de esta manera tener interfaces de usuario atractivas, sencillas y directas. De igual manera, estos componentes ayudan a que la aplicación sea más consistente, rápida, versátil e incluso adaptable a cualquier dispositivo [14].

Figma es una herramienta de prototipado web y un editor de gráficos vectorial, que, a diferencia de otras herramientas se aloja en la web para trabajar en proyectos de manera mucho más colaborativa y organizada en un solo tablero de trabajo [15].

Visual Studio Code (VSC) es un editor de código que es compatible para todo tipo de Sistemas Operativos como Windows, Linux y macOS. Además, VSC permite trabajar con una serie de *plugins* y un sin número de lenguajes de programación tales

como Java, JavaScript, Python, C++ entre otros. También permite trabajar con *GitHub* para que el versionamiento de código sea mucho más fácil e intuitivo [16].

Node.js permite a los desarrolladores crear todo tipo de aplicaciones de lado servidor gracias al uso del lenguaje de programación JavaScript, también es ampliamente utilizado para desarrollar aplicaciones del lado del cliente. Por otra parte, Node.js al ser multiplataforma es compatible con cualquier Sistema Operativo [17].

Una parte sumamente importante en el desarrollo de cualquier *software* es la etapa de pruebas, en el que se pone a prueba el funcionamiento, validaciones, rendimiento, compatibilidad y la aceptación de los usuarios finales antes de ser puesto en producción. No obstante, existen diferentes tipos de pruebas las cuales ayudan a verificar que el *software* tenga una buena funcionalidad y usabilidad, además, existen otro tipo de pruebas que son usadas para medir o visualizar errores, verificar el rendimiento, pruebas de escalabilidad, pruebas de integración, entre otras [18].

Una parte fundamental dentro de cualquier sistema *software* es la Base de datos debido a que evita que exista redundancia y mejorar la organización de la información que se requiera almacenar sin dejar de lado las reglas de seguridad que se pueda integrar [19].

El despliegue a producción de un sistema *software* a través de una plataforma de manera segura implica que todas las características funcionales y no funcionales funcionen adecuadamente y sigan cumpliendo con todos los requisitos que se han establecido logrando de esta manera certificar la calidad de *software* [20].

Firebase facilita que la creación de aplicaciones web sea mucho más rápida ya que otorga una serie de servicios y métodos disponibles. Por ejemplo, *Firebase Authentication* permite que todo el proceso de autenticación sea fácil a través de usuario y contraseña o plataformas de terceros, por otra parte existe *In-app messaging*, esto permite enviar notificaciones a varias plataformas con *Cloud Messaging* y otros servicios más que otorga esta plataforma en su capa gratuita [21].

La importancia de que un sistema *software* tenga un *backend* y *frontend* es la integración con otras aplicaciones y la separación de actividades. En otras palabras, en el *frontend* es la parte del desarrollo de lado del cliente la cual se dedica a estructurar el contenido y establecer estilos como: colores, fondos, tamaños hasta llegar a las animaciones y efectos, por otra parte, el *backend* procesa toda la información a través de toda la lógica del negocio, comunicación con la base de datos e integración con varias aplicaciones por medio de API's de tipo *RESTful* [22].

2 METODOLOGÍA

Un estudio de casos es una herramienta de investigación y una técnica de aprendizaje, la cual permite conocer el comportamiento y la particularidad de situaciones para poder comparar, evaluar y comprender diferentes aspectos de un problema de investigación. Los estudios de casos permiten obtener un proyecto centrado de manera que permita manejar el tiempo y obtener resultados agradables a través de la investigación [23].

Debido a esto, el presente trabajo de Integración Curricular mantiene un estudio de casos ya que se basa en investigaciones sobre la actual problemática que enfrenta la ciudadanía con respecto a la falta de información oportuna y en tiempo real de todos los asilos que pertenecen a la ciudad de Quito y con ello desarrollar un sistema *web*, el cual les permita brindar toda la información de los asilos que se encuentran disponibles dentro de la ciudad de Quito para que puedan de esta manera contratar sus servicios de una manera segura gracias al uso de las herramientas de desarrollo y al uso de tecnologías modernas y que sean capaces de ajustarse a cualquier escenario.

2.1 Metodología de Desarrollo

Las metodologías de desarrollo de *software* es un conjunto de técnicas y métodos organizados para poder diseñar soluciones de *software* a la diversas empresas, esto se logra a través de equipos organizados de trabajo para así reducir el nivel de dificultad y mejorar los resultados mediante constantes revisiones llamadas iteraciones las mismas que son realizadas con el fin de que el sistema *software* cumpla todas las iteraciones asignadas y la aprobación del dueño del producto [24].

Las metodologías ágiles en la actualidad permiten que el trabajo en equipo sea mucho más ágil y flexible ante cualquier cambio que pueda suceder sin afectar a las demás tareas que ya han sido planificadas. Además, permite que el proyecto pueda ser dividido en diferentes iteraciones con una serie de tareas para que sean asignadas a cada uno de los miembros del equipo de trabajo y de esta manera tener un avance progresivo del sistema *software* a desarrollar sin descuidar el objetivo que se ha planteado al inicio [25]. Es por esta razón, que el desarrollo de este sistema *web* se lo ha realizado bajo la implementación de la metodología ágil *Scrum* para poder tener así un equipo de trabajo organizado, entregar pequeños avances funcionales, implementar buenas practicas, aplicar de manera correcta cada una de las fases de la metodología, entre otras para así obtener un producto de calidad para la ciudadanía.

Roles

Los roles en la metodología *Scrum* pueden ser: *Product Owner*, *Scrum Master* y *Development Team* los cuales son capaces de manejar todas las actividades que han sido designadas, buena organización y una comunicación efectiva para lograr de esta manera un proyecto de calidad en los tiempos establecidos, en sí, son los responsables del éxito de cada *Sprint* y del proyecto en su totalidad [26]. A continuación, se detalla a cada una de las personas que han sido asignadas a estos roles.

Product Owner

Es el responsable de otorgar al equipo *Scrum* todas las necesidades y prioridades que requiere el sistema *software*, además, es el responsable de validar y aprobar cada una de las entregas funcionales comúnmente llamada iteraciones (*Sprints*) [27]. En ese sentido, en la **TABLA I** se puede observar a la persona encargada para este rol.

Scrum Master

Es el líder del equipo, el cual está encargado de establecer las buenas prácticas de *Scrum*, mantener a los miembros del equipo enfocados en el objetivo del proyecto y ayudar a los miembros del equipo a crecer y mejorar en cada avance del proyecto [28]. Por tal razón, en la **TABLA I** se evidencia la persona encargada para este rol.

Development Team

Están estructurados y organizados de manera que puedan gestionar y organizar su propio trabajo, en otras palabras, es el equipo de trabajo en cargo en desarrollar el producto del proyecto con el fin de mejorar su eficiencia y efectividad [29]. Muestra de ello, en la **TABLA I** se evidencia a la persona encargada para este rol.

TABLA I: Asignación de roles.

ROLES	NOMBRES
<i>Product Owner</i>	Ing. Byron Loarte, MSc.
<i>Scrum Master</i>	Ing. Byron Loarte, MSc.
<i>Development Team</i>	José Antonio Borja

Artefactos

Los artefactos en *Scrum* están diseñados para garantizar que toda la información sea lo más transparente posible para todo el equipo de trabajo, evitando la descoordinación y retrasos en el proyecto *software* [30]. Es por esta razón que en el presente proyecto se ha utilizado los siguientes artefactos para obtener una buena fuente de información.

Recopilación de Requerimientos

Gracias a la Recopilación de requerimientos se pueden identificar todas las necesidades de una forma exacta y clara para el proyecto *software* [31]. Por lo tanto, en la **TABLA II** se observa el formato para la elaboración de la recopilación de requerimientos. Sin embargo, toda la tabla y la información completa de la misma se detalla en el **ANEXO II** del presente documento.

TABLA II: Levantamiento de requerimientos.

RECOPIACIÓN DE REQUERIMIENTOS		
TIPO DEL SISTEMA	ID – RR	ENUNCIADO DEL ÍTEM
SISTEMA WEB	RR010	Como usuario ciudadano necesita realizar lo siguiente: <ul style="list-style-type: none">• Visualizar información de asilos.
	RR011	Como usuario ciudadano necesita realizar lo siguiente: <ul style="list-style-type: none">• Realizar una proforma.

Historias de Usuario

Una Historia de usuario es una descripción breve y con un lenguaje sencillo en donde el usuario plasma lo que requiere realizar dentro de un producto *software* y de esta manera obtener un resultado óptimo [32]. Muestra de ello, en la **TABLA III** se observa una de las primeras Historias de usuario que se ha diseñado e implementado. No obstante, el resto de Historias de usuario se detallan en el **ANEXO II** del presente documento.

TABLA III: Recopilación de Historias de Usuario 11.

HISTORIA DE USUARIO	
Identificador (ID): HU011	Usuario: usuario ciudadano
Nombre historia: Visualizar página informativa	
Prioridad en negocio: Alto	Riesgo en desarrollo: Realizar proforma.
Sprint asignado: SB002	
Programador responsable: Jose Antonio Borja	
Descripción: El usuario con perfil ciudadano en el sistema web tiene la posibilidad de visualizar un simulador de los costos por cada uno de los servicios que oferta el asilo asignado.	
Observación: El sistema web una vez que finalice el proceso del simulador esta información es proporcionada por medio de un PDF para que pueda ser descargado.	

Product Backlog

También se lo denomina como un conjunto de tareas donde se diseñan durante el inicio del proyecto, el *Product Backlog* contiene la descripción de las tareas que se deben realizar al momento de ejecutar el proyecto y que se van adaptando según las necesidades del cliente [33]. Seguidamente, la **TABLA IV** muestra el formato que se ha utilizado para enumerar las tareas del *Product Backlog*, mientras que la tabla completa se puede visualizar en el **ANEXO II** del presente documento.

TABLA IV: Diseño del *Product Backlog*.

ELABORACIÓN DEL <i>PRODUCT BACKLOG</i>				
ID-HU	Historia de Usuario	No. Iteración	Prioridad	Estado
HU010	Visualizar información de asilos	2	Planificado	Finalizado

HU011	Realizar una proforma	2	Planificado	Finalizado
--------------	-----------------------	---	-------------	------------

Sprint Backlog

Se representa como un tablero de tareas en donde se hace visible todo el trabajo necesario para alcanzar las actividades que se ha planificado y organizado con el *Product Owner* para cada una de las iteraciones o comúnmente llamado *Sprints* [34]. A continuación, la **TABLA V** se ha diseñado un formato el cual se ha utilizado para el diseño de los 5 *Sprints*, mientras que el resto de la tabla a detalle se puede visualizar en el **ANEXO II** del presente documento.

TABLA V: Formato del *Sprint Backlog*.

ELABORACIÓN DEL <i>SPRINT BACKLOG</i>						
ID-SB	NOMBRE	MÓDULO	ID-HU	HISTORIA DE USUARIO	TAREAS	TIEMPO ESTIMADO
SB003	Diseño e implementación de los módulos para los ciudadanos.	-----	-----	Realizar una proforma	<ul style="list-style-type: none"> • Diseño e implementación de interfaz para realizar una proforma • Diseño e implementación de una proforma la cual permita calcular un aproximado de costo de un asilo • Reglas de validación para cada campo del 	35 H

					formulario. Pruebas de guardar, visualización, que la proforma da un costo de asilo	
--	--	--	--	--	--	--

2.2 Diseño de interfaces

La elaboración de las interfaces permite la interacción entre el usuario y el producto *software*, se puede decir que permite el intercambio de información a través de una serie de interfaces y asegurar que las mismas sean acorde y agradables para los usuarios que van hacer uso de la misma, como también fáciles e intuitivas al momento de usar [35].

Herramienta utilizada para el diseño

La elaboración de cada uno de los prototipos para el sistema *web* se los ha diseñado con la plataforma Figma, la cual permite crear diseños de manera fácil, sin tantas complicaciones y que se puedan trabajar de manera colaborativa. A continuación, la **Fig. 1** se visualiza el diseño para la página de inicio del sistema *web* con distintos componentes como el uso de un *navbar*, *cards*, *caruusel* y un *footer*. Mientras que el resto de los diseños se detallan en el **ANEXO II** del presente documento.

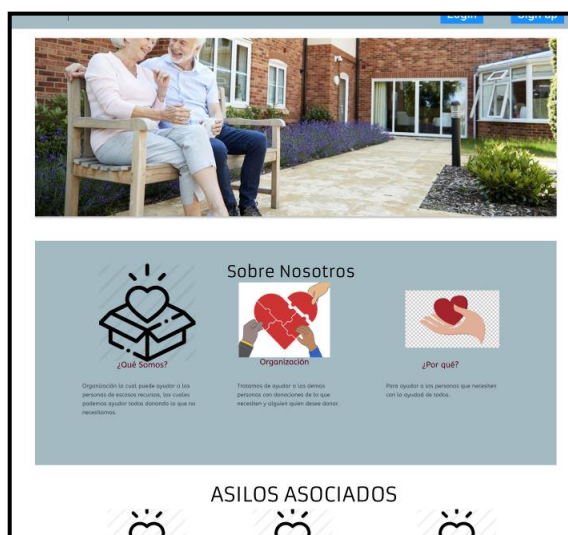


Fig. 1: Prototipo del inicio en el sistema web.

2.3 Diseño de la arquitectura

El diseño de la arquitectura permite dar a conocer sobre la estructura interna de un producto *software*, de tal modo que permita agregar detalles y también permitir a los desarrolladores delimitar la codificación con la integración de nuevos módulos, librerías y herramientas [36]. A continuación, se describe al patrón arquitectónico que se ha utilizado y como se lo implementado en el proyecto de tesis.

Patrón arquitectónico

El patrón Modelo, Vista y Controlador (MVC) es usado para organizar los componentes y separarlos en capas: capa de datos, capa de la lógica y capa de interfaz, permitiendo de esta manera tener un sistema *software* que sea escalable y mantenible para las futuras versiones [37].

- **Modelo:** permite gestionar la información y comunicación con la Base de Datos.
- **Vista:** es la capa responsable de generar las interfaces y la presentación de la información.
- **Controlador:** es la capa que permite recibir y procesar todas las peticiones HTTPS entrantes del usuario.

A continuación, en la **Fig. 2**, se puede visualizar el patrón MVC que se diseñado en el sistema *web* y en donde se puede agregar más funcionalidades a cada una de las capas si se requiere, así como la integración de varias herramientas o librerías de terceros sin que afecte la funcionalidad de las demás capas.

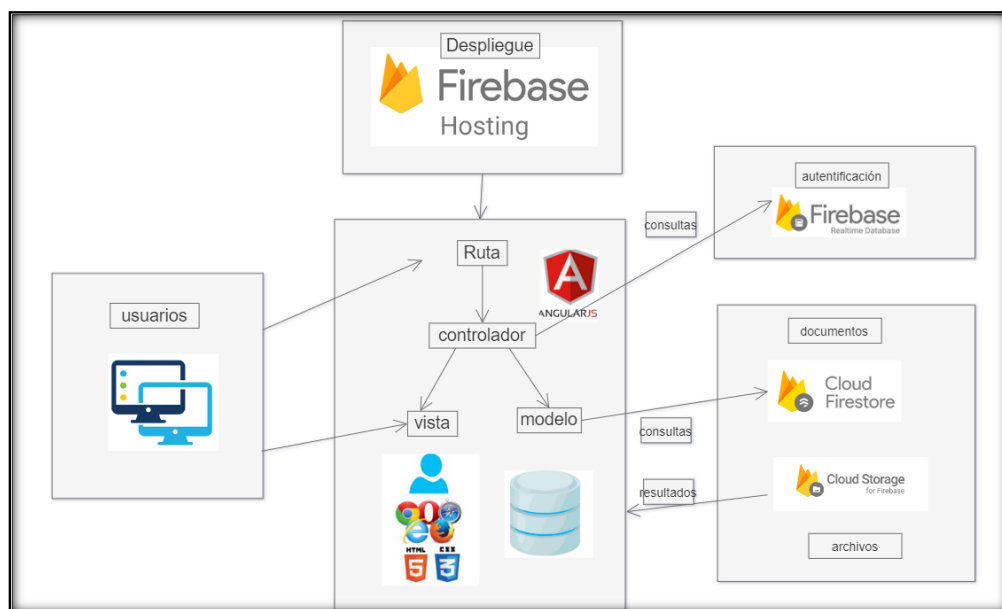


Fig. 2: Patrón arquitectónico.

2.4 Herramientas de desarrollo

Las herramientas de desarrollo son aquellas que facilitan y optimizan la mejora y el desempeño del trabajo. Por otra parte, las herramientas tienen como objetivo disminuir el estrés, tiempos de codificación, pruebas y entregas en cada fase [38]. Es por esta razón, que se ha utilizado múltiples herramientas las cuales han facilitado notablemente la etapa de codificación, pruebas y despliegue a producción para lograr de esta manera obtener un producto de calidad e integración con nuevas herramientas o librerías.

En la **TABLA VI** se puede observar al conjunto de herramientas que se han utilizado para el proceso de codificación de cada uno de los módulos que corresponden al sistema *web*.

TABLA VI: Herramientas para la codificación del sistema *web*.

HERRAMIENTA	JUSTIFICACIÓN
<i>Node.js</i>	Permite a los desarrolladores crear todo tipo de aplicaciones de lado servidor gracias al uso del lenguaje de programación JavaScript [17].
<i>Firebase</i>	Facilita la creación de aplicaciones para dispositivos móviles y web gracias al uso de servicios que son totalmente gratuitos [13].
Angular material	Es un <i>Framework</i> el cual permite agregar estilos y diseños a una aplicación en Angular a través de componentes [14].
Angular	Permite desarrollar aplicaciones web escalables y separar la comunicación con el <i>backend</i> [12].
<i>Visual Studio Code</i>	Permite trabajar con una serie de <i>plugins</i> a través de edición de código [16].

Librerías

En la **TABLA VII** se observa un grupo de librerías que se han implantado para la codificación del sistema *web*.

TABLA VII: Codificación del sistema *web* mediante el uso de las librerías.

LIBRERÍA	DESCRIPCIÓN
"firebase": "^9.8.1",	Proporciona las herramientas y la infraestructura que se necesita para desarrollar web y móviles [39].
"@angular/fire": "7.3.0",	Suaviza las asperezas que un desarrollador de Angular pueda encontrar al implementar el SDK [40].
"ngx-cookie-service": "13.2.0",	Es un servicio para Angular que permite leer, configurar y eliminar las cookies del navegador [41].
"@ng-bootstrap/ng-bootstrap": "12.0.2",	Es una serie de widgets para Angular que han sido creados desde cero utilizando el <i>Framework</i> de Bootstrap [42].
"@angular/material": "13.3.5",	Son una serie de herramientas que ayudan a los desarrolladores a crear sus propios componentes personalizados y posterior a ello utilizarlos en proyectos de Angular [43].

3 RESULTADOS

En esta sección del presente documento, se puede visualizar de mejor manera a cada una de las tareas que se han planificado, como también los resultados que se han obtenido a través de la realización de los 5 *Sprints*. Adicionalmente se presenta las pruebas que se han ejecutado y el despliegue del sistema *web*.

Sprint 0. Configuración del ambiente de desarrollo.

Las actividades para dicho *Sprint* son las siguientes:

- Plataforma *Firebase* para la Base de datos.
- Estructuración del proyecto.
- Clasificación y asignación de roles de usuarios.
- Restricciones del sistema *web*.

Plataforma *Firebase* para la Base de datos

Para gestionar toda la información de cada uno de los usuarios y de todos los asilos se ha utilizado la Base de datos de la plataforma *Firebase* específicamente *Cloud Firestore*, el cual permite administrar en tiempo real la información ya que no utiliza relaciones. Por otra parte, permite ser una Base de datos NoSQL la cual se estructura en base a colecciones y documentos para una fácil integración y ejecución de consultas [21].

En la **Fig. 3** se puede visualizar la estructura de colecciones como parte de la Base de datos la cual ha sido diseñada en base a la Recopilación de requerimientos. Por último, el diseño completo se lo puede apreciar de una mejor forma en el **ANEXO II** del presente documento.

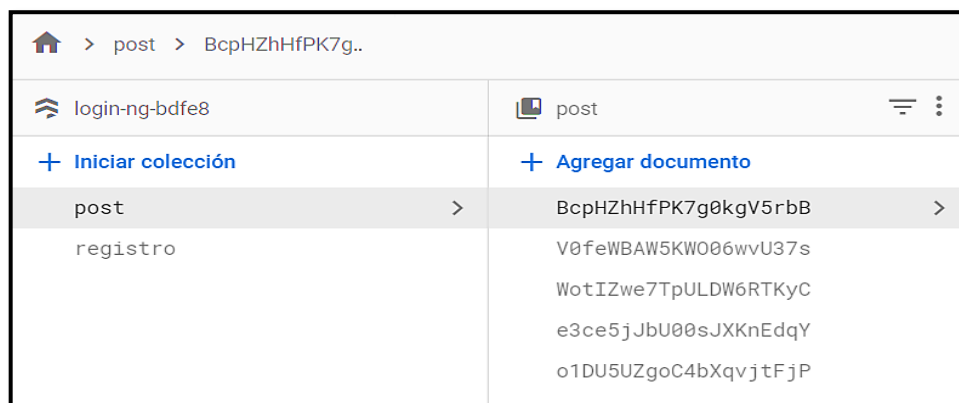


Fig. 3: Esquema NoSQL de la Base de datos.

Estructuración del proyecto

Para la codificación del sistema *web* se ha aplicado un entorno de desarrollo integrado (IDE) que es *Visual Studio Code*, el cual facilita la codificación y edición de código de una forma organizada siendo el más óptimo para la instalación de *plugins* y lograr codificar todas las funcionalidades que requiere el sistema *web* de una forma optimizada y ordenada. Adicional a ello, gracias a este IDE se ha logrado estructurar de una forma adecuada todo el proyecto conjuntamente con la integración del patrón de arquitectura MVC como se puede apreciar en la **Fig. 4**.

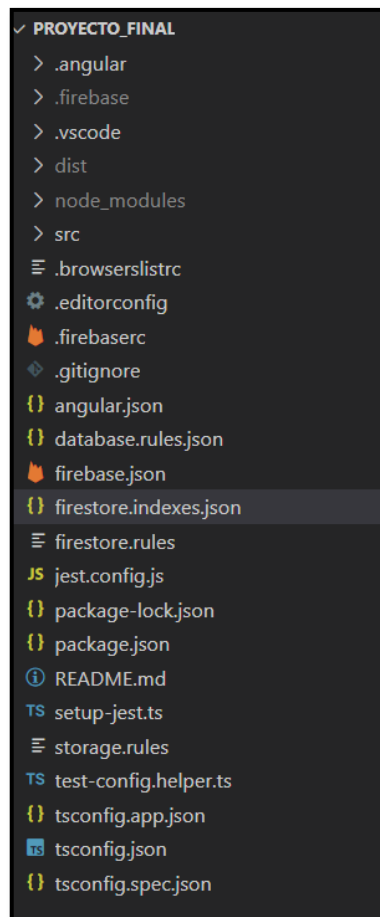


Fig. 4: Organización de archivos y directorios en el sistema *web*.

Clasificación y asignación de roles de usuarios

En la **Fig. 5** se puede visualizar de una forma más detallada a cada uno de los usuarios que hacen uso del sistema *web* que en este caso son principalmente tres: administrador, administrador del asilo y ciudadano, los cuales una vez que inicien sesión en el formulario respectivo se presenta los módulos a los que tienen acceso para poder realizar las tareas asignadas.

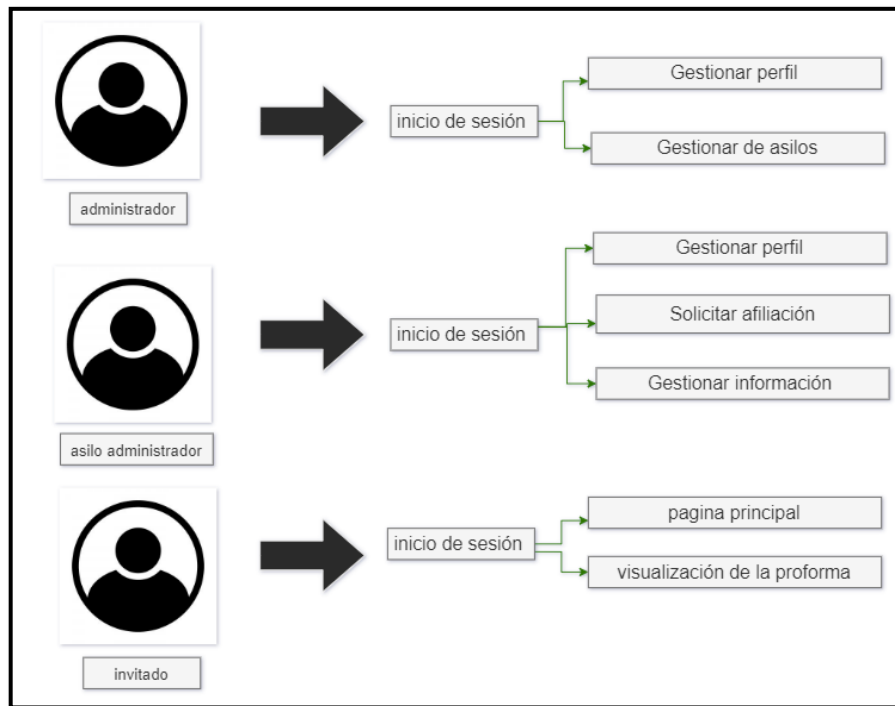


Fig. 5: Asignación de roles y módulos asignados.

Restricciones del sistema web

Autenticación de usuarios

El usuario administrador y administrador del asilo pueden acceder al sistema *web* a través de la autenticación de una contraseña y correo respectivamente.

Gestión de asilos

El administrador tiene la posibilidad de aprobar a cada uno de los asilos que estén pendientes y que los mismos se hayan registrado en el módulo respectivo.

Control de documentos de asilos

El usuario administrador es el encargado de controlar y supervisar toda la información de los asilos, así como del documento que avale que es un asilo verificado para su respectiva aprobación o rechazo.

Recuperación o cambio de contraseña

El usuario administrador o administrador del asilo tienen la opción de cambiar la contraseña o recuperarla a través del módulo perfil el cual dispone la opción de poder cambiar el correo o la contraseña. Al igual que en el módulo de inicio de sesión se puede visualizar el botón

de “restablecer contraseña”. Por último, la **Fig. 6** presenta las acciones que realiza cada uno de los usuarios dentro del sistema *web*.



Fig. 6: módulos para cambiar o restablecer contraseña.

***Sprint 1.* Diseño e implementación de los módulos para el usuario administrador.**

En base a la planificación que se ha realizado, a continuación, se detalla las actividades para dicho *Sprint*:

- Visualizar página informativa.
- Iniciar sesión.
- Gestión de la información del perfil usuario.
- Gestionar peticiones de aprobación, pendiente y rechazo de asilos.

Visualizar página informativa

En la **Fig. 7** se puede visualizar la página informativa la cual permite iniciar sesión y acceder a los módulos respectivos. Además, en esta página informativa se puede visualizar a cada uno de los asilos que han sido aprobados por parte del administrador y que están listos para prestar sus servicios. Las demás secciones se pueden apreciar de mejor manera en el **ANEXO III** del presente documento.

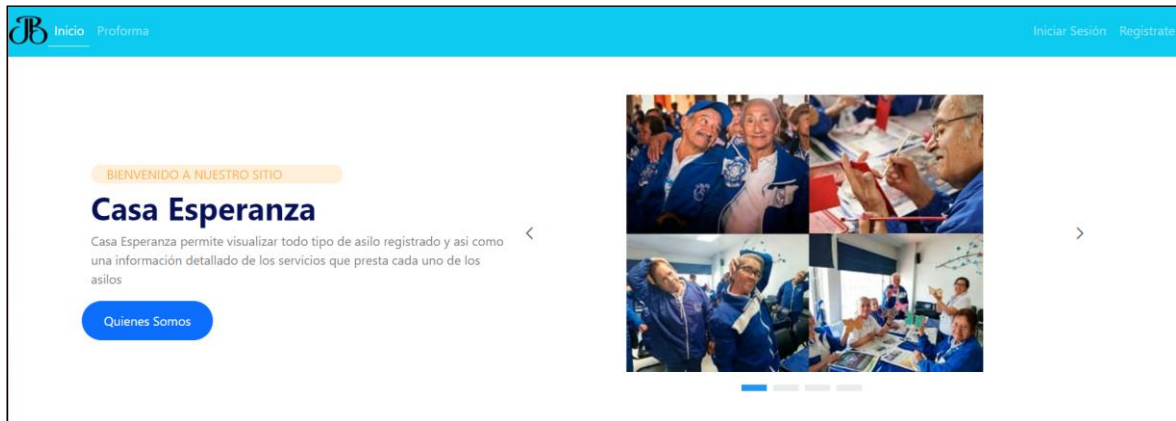


Fig. 7: Visualización de la página informativa.

Iniciar sesión

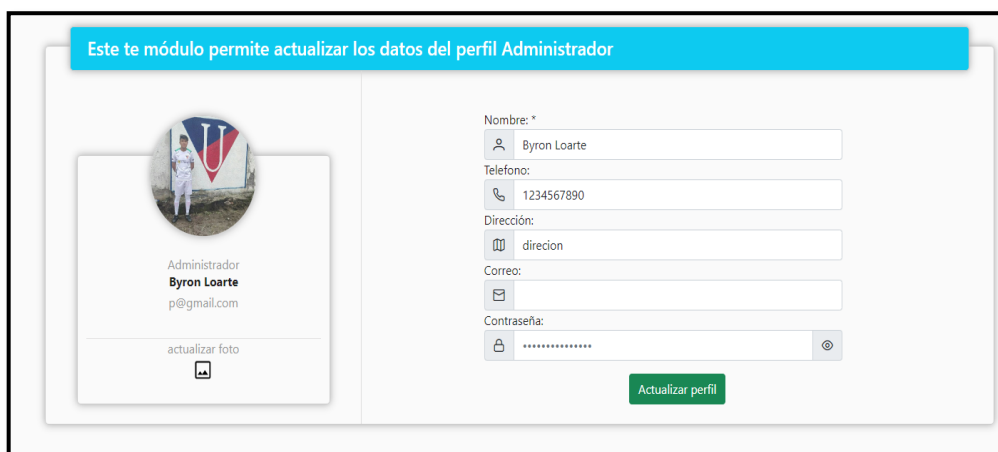
A continuación, en la **Fig. 8** se puede visualizar el módulo de inicio de sesión, el cual a través de las credenciales y validaciones respectivas se puede acceder a los módulos respectivos. Además, cabe recalcar que la funcionalidad de este módulo de autenticación sirve también por el usuario administrador del asilo. El proceso a detalle del inicio de sesión y las validaciones respectivas se las puede apreciar de una mejor forma en el **ANEXO III** del presente documento.

Fig. 8: Inicio de sesión.

Gestión de la información del perfil usuario

En la **Fig. 9** se puede visualizar el módulo perfil, en el cual el administrador tiene la opción de modificar la foto o editar sus datos personales si fuera el caso, además, los cambios se visualizan tras la validación de los campos respectivos. Por otra parte, cabe recalcar que la funcionalidad de este módulo de perfil de usuario sirve también por el usuario administrador del asilo.

La modificación de la información del usuario y las validaciones respectivas se las puede apreciar de una mejor forma en el **ANEXO III** del presente documento.



The screenshot displays a web interface for managing a user profile. At the top, a blue banner reads: "Este te módulo permite actualizar los datos del perfil Administrador". The interface is divided into two main sections. On the left, there is a circular profile picture of a person in a white uniform. Below the photo, the text identifies the user as "Administrador Byron Loarte" with the email "p@gmail.com". A button labeled "actualizar foto" with a camera icon is positioned below the photo. On the right, there is a form with several input fields: "Nombre: *" (containing "Byron Loarte"), "Telefono:" (containing "1234567890"), "Dirección:" (containing "direccion"), "Correo:" (empty), and "Contraseña:" (masked with asterisks). A green button labeled "Actualizar perfil" is located at the bottom right of the form.

Fig. 9: Módulo perfil de usuario.











Gestionar peticiones de aprobación, pendiente y rechazo de asilos

En la **Fig. 10** se visualiza el módulo en donde el administrador tiene la opción de visualizar todos los asilos que se han registrado y los que están pendientes para su aprobación o rechazo con la opción de un buscador. Adicional a ello, el usuario administrador puede gestionar todas las solicitudes de los asilos que deseen forma parte del sistema *web*, el cual puede aceptar la solicitud como se puede visualizar en la **Fig. 11** o si fuera el caso rechazar la solicitud. El proceso a detalle del rechazo de los asilos y las validaciones respectivas se las puede apreciar de una mejor forma en el **ANEXO III** del presente documento.

Este módulo te permite gestionar los asilos que desean formar parte del sistema web

Asilos pendientes Asilos Aprobados Asilos rechazados

Buscar

Cédula	Nombre	Dirección	Correo	Teléfono	Acciones
1750068585005	a	Cerca de Francisco Rueda, Solanda, Quito, Distrito Metropolitano de Quito, Pichincha, 170148, Ecuador	a@gmail.com	0983616234	    
1750068585	asdasdasd	asdasdsasa	e@gmail.com	0983162344	    

Items per page: 4 1 - 4 of 7 |< < > >|

Fig. 10: Tabla de asilos pendientes.

Este módulo te permite gestionar los asilos que desean formar parte del sistema web

Asilos pendientes Asilos Aprobados Asilos rechazados

Buscar

Cédula	Nombre	Dirección	Correo	Teléfono
1751558303	asilo prueba	ajavi y huigra oe5-322	asiloprue@gmail.com	1234567
0930109582001	Casa Aurora	CALLE / ZOILA RENDÓN DE MOSQUERA E 7 - 37 CDLA. KENNEDY	aurora.3ae@gmail.com	0981776117
1750068585001	a	huigra y ajavi prueba	a@gmail.com	0983162344
1801755628	bogati	pollo la kenedi	bogati69@gmail.com	1234567

Items per page: 4 0 of 0 |< < > >|

Fig. 11: Tabla de asilos aprobados o rechazados.

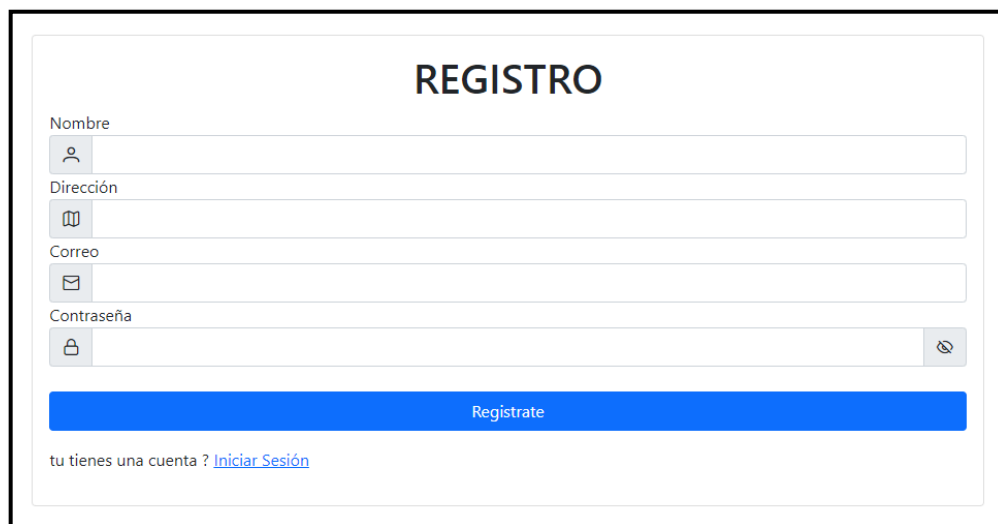
***Sprint 2.* Diseño e implementación de los módulos para el usuario administrador del asilo.**

En base a la planificación que se ha realizado, a continuación, se detalla las actividades para dicho *Sprint*:

- Registro del usuario administrador del asilo.
- Enviar solicitud de afiliación.
- Gestionar información del asilo.
- Visualizar información del asilo

Registro del usuario administrador del asilo

A continuación, en la **Fig. 12** se visualiza el módulo de registro el cual permite que el usuario administrador del asilo se registre en el sistema *web* a través de un formulario con los siguientes datos: nombre del administrador asilo, dirección del administrador asilo, correo y una contraseña. El proceso a detalle y las validaciones respectivas se las puede apreciar de una mejor forma en el **ANEXO III** del presente documento.



The image shows a web registration form titled "REGISTRO". It contains four input fields, each with a small icon to its left: a person icon for "Nombre", a location pin icon for "Dirección", an envelope icon for "Correo", and a lock icon for "Contraseña". The "Contraseña" field has a toggle icon on its right side. Below the fields is a large blue button labeled "Registrate". At the bottom, there is a link that says "tu tienes una cuenta ? [Iniciar Sesión](#)".

Fig. 12: Registro del usuario administrador del asilo.

Enviar solicitud de afiliación

A continuación, en la **Fig. 13** se visualiza el módulo enviar solicitud de afiliación, el cual permite enviar un formulario donde consta los siguientes datos como ruc/cédula, nombre, dirección, teléfono, correo, imagen y documentación por parte del asilo la cual sirve para el proceso de aprobación o rechazo del asilo. El proceso a detalle y las validaciones respectivas se las puede apreciar de una mejor forma en el **ANEXO III** del presente documento.

Fig. 13: Enviar solicitud de afiliación.

Gestionar información del asilo

En la **Fig. 14** se visualiza el módulo para la gestión de la información del asilo el cual permite registrar toda la información del asilo una vez que el mismo ha sido aceptado por parte del administrador en donde puede registrar misión, visión, servicios, horarios, número del personal del asilo, cambiar la imagen y editar la información que fue enviada al momento de solicitar la afiliación. No obstante, en la **Fig. 15** se visualiza cuando el asilo fue rechazado y tiene la opción de poder verificar el motivo del rechazo y poder reenviar el formulario y esperar la aceptación o rechazo por segunda vez. Por último, la modificación de la información y las validaciones respectivas se las puede apreciar de una mejor forma en el **ANEXO III** del presente documento.

Fig. 14: Aceptación del asilo.

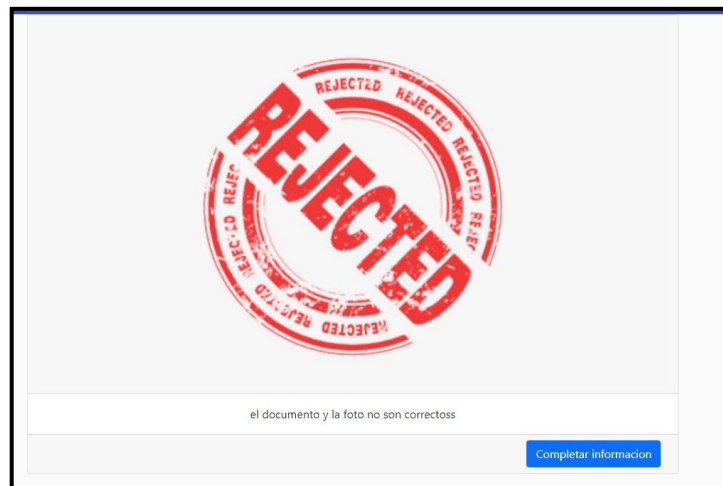


Fig. 15: Rechazo del asilo y el motivo.

Visualizar información del asilo

En la **Fig. 16** se observa el módulo para visualizar la información del asilo el cual tiene como objetivo visualizar toda la información que se ha registrado en el módulo respectivo. Además, cabe recalcar que la funcionalidad de este módulo de visualización de los asilos sirve también para el usuario ciudadano. La modificación de la información y las validaciones respectivas se las puede apreciar de una mejor forma en el **ANEXO III** del presente documento.



Fig. 16: Visualizar información del asilo.

Sprint 3. Diseño e implementación de los módulos para el usuario ciudadano.

En base a la planificación que se ha realizado, a continuación, se detalla la actividad para dicho *Sprint* la cual es realizar y descargar proforma.

Realizar y descargar proforma


En la **Fig. 17** se observa el módulo para realizar y descarga una proforma el cual tiene como objetivo que el usuario ciudadano pueda realizar una proforma y de esta manera conocer el costo aproximado de los servicios ofertados por un asilo. Mientras que en la **Fig. 18** se puede visualizar la proforma descargada en formato pdf con todos los precios y servicios, por último, proceso a detalle y las validaciones respectivas se las puede apreciar de una mejor forma en el **ANEXO III** del presente documento.

Detalles de la proforma	
Descripción	Valor
Tipo de habitación - Compartida	\$ 325
Zona/Ubicación - Valle	\$ 100
Amoblado - No	\$ 0
Cuidado físico - Se le Dificulta	\$ 25
Cuidado Cognitivo - Deterioro Mental Moderado	\$ 25
> Servicios Medicos (2)	\$ 60
> Servicios adicionales (1)	\$ 20
Total	\$ 555
Valor total por hijo	\$ 277.5

Fig. 17: Proforma de servicios.


1 / 1 | - 67% + | [Icono de zoom] [Icono de compartir]

Proforma



Datos			
Nombre	Casa Esperanza	Fecha	13-07-2022
Provincia	Pichincha	Ciudad	Quito
Cédula	1726749056	Correo	casaesperanza@gmail.com

Detalles de la proforma	
Descripcion	Valor
Tipo de Habitación - Compartida	\$ 325
Zona/Ubicación - Valle	\$ 100
Amoblado - No	\$ 0
Cuidado físico - Se le Dificulta	\$ 25
Cuidado cognitivo - Deterioro Mental Moderado	\$ 25
Servicios Médicos - 2	\$ 60
Terapias Musculares	\$ 25
Rehabilitación y fisioterapia	\$ 35
Servicios Adicionales - 1	\$ 20
Acompañamiento a Citas Médicas	\$ 20
Total	\$ 555
Valor total por hijo	\$ 277.5



nombre del sitio

Condiciones y formas de pago
El pago se realiza al instante de realizar el

Fig. 18: Proforma descargada.

Sprint 4. Pruebas del sistema web

En base a la planificación que se ha realizado, a continuación, se detalla las actividades para dicho *Sprint*:

- Ejecución de pruebas unitarias y resultados.
- Ejecución de pruebas de rendimiento y resultados.
- Ejecución de pruebas de compatibilidad y resultados.
- Ejecución de pruebas de aceptación y resultados.

Ejecución de pruebas unitarias y resultados

Las pruebas unitarias permiten detectar errores en la codificación de un sistema *software*, de tal manera, que se evite una mala funcionalidad en su totalidad o presente errores al momento de ser implementada con los demás módulos de un sistema *software*. Es por esta razón, que se debe realizar este tipo de pruebas para poder evitar fallas y garantizar un código de calidad, así como también realizar nuevas mejoras e implementaciones a futuro [44].

Para la ejecución de las pruebas unitarias se ha utilizado la herramienta Karma como se puede ver en la **Fig. 19**, el cual toma una porción del código como por ejemplo del módulo de inicio de sesión y tras la ejecución de la prueba se puede obtener un resultado de la herramienta el cual muestra un resultado favorable y con ello seguir con el resto de las pruebas las mismas que se las puede observar en el **ANEXO II**.

```
//Carga de inicio Login
let component;

describe("<Login />", () =>{
  beforeEach(()=>{
    component = render("<Login />")
  });
  it("Renderiza correctamente", () =>{
    expect(component).toBeDefined();
  })
})
```

Fig. 19: Prueba unitaria al módulo inicio de sesión.

```
<Login />
  ✓ Renderiza correctamente (3346 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        13.57 s, estimated 17 s
Ran all test suites.
```

Fig. 20: resultado módulo inicio de sesión.

Al finalizar el proceso de la prueba unitaria en cada uno de los módulos se ha obtenido un resultado 100% favorable, garantizando de esta manera que el sistema web no presente ningún tipo de fallos a nivel de módulos y que esté listo para nuevas implementaciones si se requiere.

Ejecución de pruebas de rendimiento y resultados

Las pruebas de rendimiento detectan si la funcionalidad de un producto *software* se ve afectada al momento de estar en producción y poder determinar el tiempo de respuesta y la escalabilidad óptima [45]. En ese sentido, para ejecutar dicha prueba se ha utilizado *PageSpeed Insights*, la cual permite evaluar la carga y velocidad de un sitio web a través de los componentes como internos como: archivos, imágenes, módulos, código fuente, etc. [46]. Muestra de ello, en la **Fig. 21** se puede visualizar la obtención del resultado al momento de culminar la prueba respectiva.

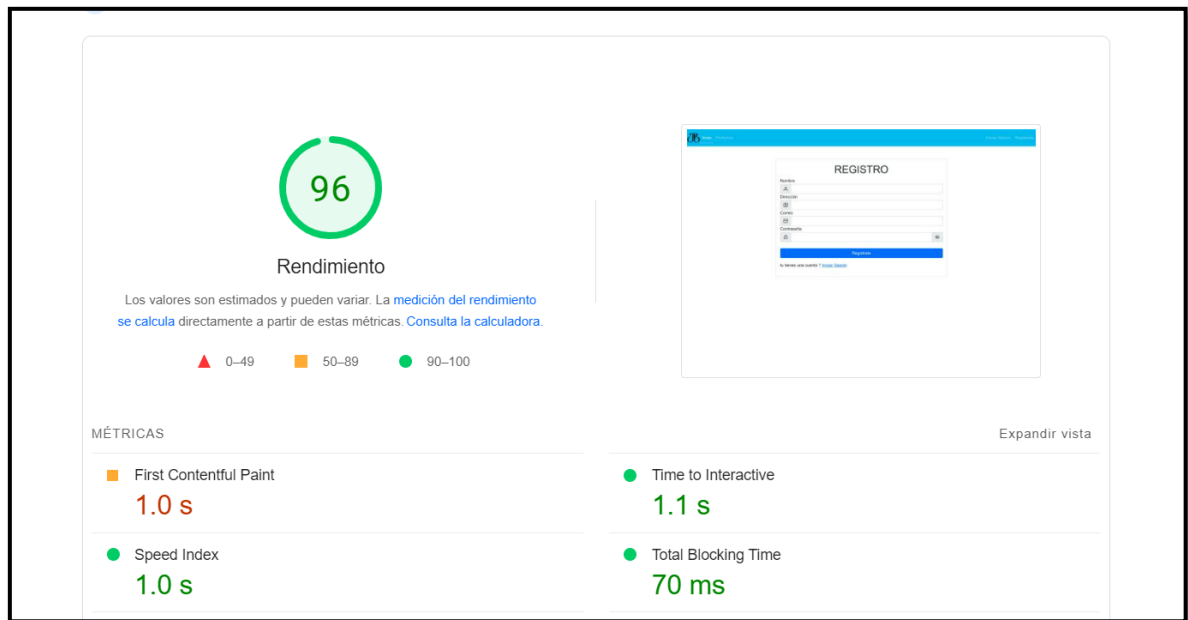


Fig. 21: Prueba de rendimiento al registro.

Al finalizar el proceso de la prueba de rendimiento se ha obtenido un resultado favorable lo que permite que el sistema *web* a futuro no presente ningún tipo de fallo para los usuarios que utilicen dicho sistema y que información sea presentada en tiempo real.

Ejecución de pruebas de compatibilidad y resultados

Este tipo de pruebas deben garantizar que toda la funcionalidad al igual que la información del sistema *software* se presente de manera correcta en los diferentes navegadores ya que la información o contenido multimedia se puede visualizar de forma desproporcionada en otros navegadores o en aquellos que aún se encuentran desactualizados es por esta razón de la implementación de esta prueba [47].

Para la realización de la prueba de compatibilidad se ha utilizado diferentes navegadores con el fin de visualizar todo el contenido del sistema *web* como se observa en la **TABLA VIII**, a continuación, en la **Fig. 22** se puede visualizar la prueba realizada en el navegador *Google Chrome*. Por último, las demás pruebas se las puede observar en el **ANEXO II**.

TABLA VIII: Prueba de compatibilidad en navegadores *web*.

NOMBRE	VERSIÓN	OBSERVACIÓN
<i>Opera</i>	86.0.4363.59	Completamente funcional
<i>Google</i>	101.0.4951.67	Completamente funcional

Chrome		
Microsoft Edge	100.0.1185.44	Completamente funcional
Firefox	100.2	Parcialmente funcional

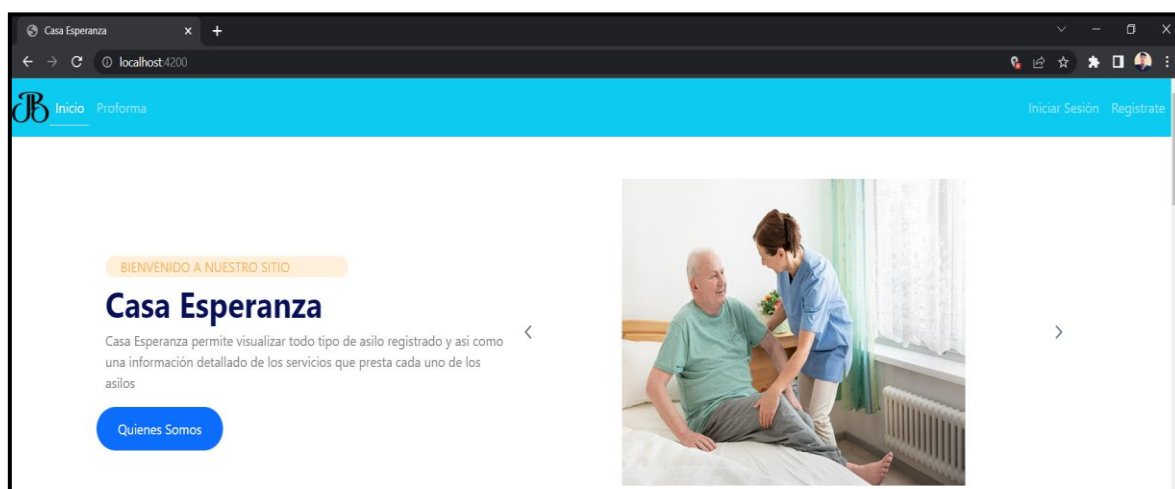


Fig. 22: Resultado de la prueba en el navegador *Google Chrome*.

Al finalizar el proceso de la prueba de compatibilidad se ha obtenido un resultado totalmente favorable ya que el contenido informativo y multimedia se ajustan de manera correcta en los diferentes navegadores.

Ejecución de pruebas de aceptación y resultados

La prueba de aceptación permite identificar si todos requisitos del dueño del producto han sido implementados en el producto *software* logrando de esta manera la aprobación o el rechazo si fuera el caso [48]. En ese sentido en la **TABLA IX** se puede visualizar un ejemplo de la prueba de aceptación, la cual es para verificar la implementación y funcionalidad de la proforma, así como el resultado que se ha obtenido, mientras que las demás pruebas se las puede encontrar en el **ANEXO II** del presente documento.

TABLA IX: Prueba de aceptación 11 – Proforma.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA011	Identificador historia de Usuario: HU011
Nombre: Ingreso hacia la proforma	
Descripción: El usuario con perfil ciudadano en el sistema <i>web</i> tiene la posibilidad de visualizar un	

simulador de los costos por cada uno de los servicios que oferta el asilo asignado.
Pasos de ejecución: <ul style="list-style-type: none"> • Dirigirse al navegador y colocar la URL del sistema • Ingresar al módulo de proforma • Realizar la proforma en base a las necesidades del usuario ciudadano • Imprimir la proforma
Resultado deseado: El sistema <i>web</i> permite que el usuario ciudadano realice una proforma de tal manera que estime un valor o costo sobre los servicios que brinda el asilo, como también imprimir la proforma en formato pdf.
Evaluación de la prueba La prueba tiene la aprobación de un 100% y realiza lo esperado.

Al finalizar el proceso de la prueba de aceptación se obtiene como resultado final que el sistema *web* que se ha desarrollado cumple con todas las necesidades y requerimientos por parte del *Product Owner* y con ello pasar a la siguiente etapa del presente documento.

Sprint 5. Despliegue del sistema web

Al finalizar la planificación que se ha realizado, a continuación, se detalla la actividad para dicho *Sprint* la cual es realizar el despliegue en el *hosting* de *Firebase*.

Despliegue en el *hosting* de *Firebase*

Culminado el proceso de codificación se procede con el proceso de despliegue en donde se utiliza *Firebase* como un servidor de *hosting* y dominio el cual va a permitir desplegar el sistema *web* a producción. No obstante, para el despliegue es necesario realizar varios pasos los cuales se puede observar en el **ANEXO II**, mientras que en la **Fig. 23** se puede visualizar el sistema web en producción totalmente operativo la cual se puede acceder ingresando la siguiente URL.

<https://casaesperanzaweb.web.app/>

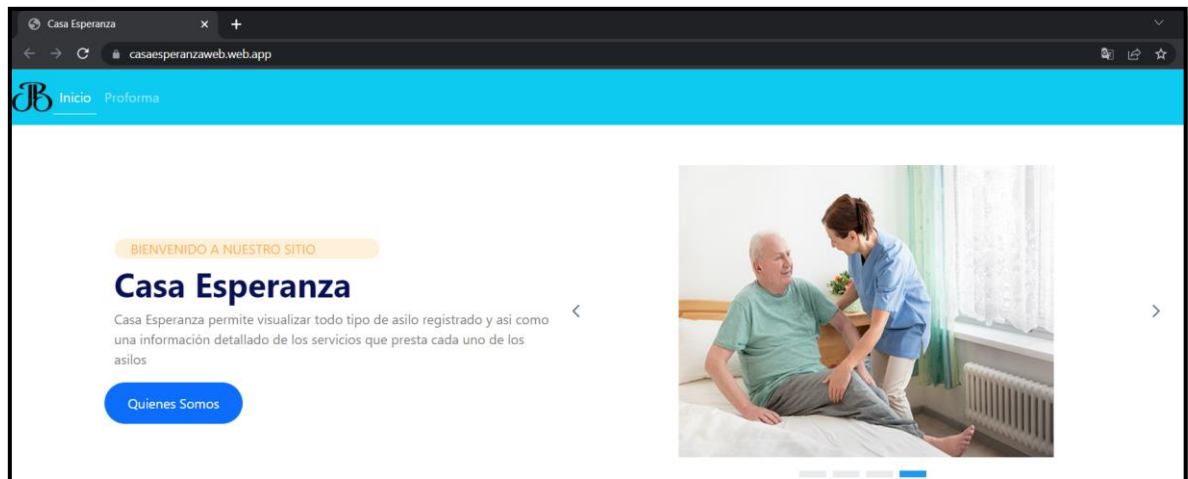


Fig. 23: Despliegue del sistema web bajo un dominio.

4 CONCLUSIONES

En este apartado se presentan las conclusiones que se han obtenido.

- Para la codificación del sistema *web* es muy importante obtener una adecuada recopilación de requerimientos ya que gracias a ello se puede seleccionar de forma acertada cada una de las herramientas de desarrollo y con ello culminar con cada uno de los módulos en los tiempos establecidos.
- El desarrollo del sistema *web* se basa en la implantación de la metodología ágil *Scrum*, la cual ha permitido obtener en un tiempo relativamente corto y bien planificado la culminación de los diferentes *Sprints*, logrando de esta manera cumplir con todas las necesidades del cliente y evitar los posibles errores durante la etapa de codificación, pruebas y despliegue.
- La implantación del patrón arquitectónico o también conocido como MVC, ha permitido que el código sea lo más limpio y estructurado posible. Además, posibilita que la incorporación de algún módulo sea totalmente transparente sin afectar a los demás módulos.
- La implantación de una serie de librerías y herramientas en el sistema *web* permiten la reutilización del código y que sea mucho más ordenado para las futuras integraciones.
- A través de la plataforma *Firebase* como sistema gestor de la base de datos, tiene como función primordial ayudar a resolver, optimizar y mejorar el rendimiento del sistema con el fin de prevenir que exista errores al momento de ser puesto a producción.
- Las distintas pruebas que se han realizado al sistema *web* permiten verificar que cada módulo o componente funcione de manera correcta y evitar problemas a futuro cuando la ciudadanía comience a utilizar el sistema.

5 RECOMENDACIONES

En este apartado se presentan las recomendaciones que se han obtenido.

- Es indispensable que los usuarios tengan un conocimiento básico en el uso de navegadores web con el fin evitar algún problema al momento de usar el sistema *web*.
- Es necesario que los usuarios comprueben que los navegadores estén actualizados para uso del sistema *web* y evitar que el diseño del sistema no se visualice correctamente.
- Se recomienda que el administrador implemente políticas de seguridad para habilitar o deshabilitar a los usuarios, en los casos que no utilicen el sistema *web* correctamente.
- Se recomienda realizar un *backup* de la Base de datos del sistema *web* mensualmente con el fin de evitar la pérdida o como también el robo de la información o por cualquier eventualidad y así evitar tener conflictos con los usuarios.
- Se recomienda comprobar si existe nuevas actualizaciones de las herramientas que se han utilizado en el sistema *web*, en un periodo de 6 meses, con la finalidad de mantener los componentes, librerías de terceros o nativas y ciertas configuraciones actualizadas para asegurar que el sistema no presente ningún tipo de error.
- Se recomienda tener un ambiente de pruebas para la incorporación de alguna funcionalidad o modulo al sistema *web* evitando de esta manera que el sistema a producción se vea completamente afectado.

6 REFERENCIAS BIBLIOGRÁFICAS

- [1] E. Comercio, «El Comercio,» 19 09 2020. [En línea]. Available: <https://www.elcomercio.com/actualidad/quito/fallecidos-covid19-ecuador-adultos-mayores.html>. [Último acceso: 12 05 2022].
- [2] bservatorio, «bservatorio,» 08 05 2022. [En línea]. Available: <https://www.covid19ecuador.org/ecuador>. [Último acceso: 12 05 2020].
- [3] OPS, «OPS,» 30 09 2020. [En línea]. Available: <https://www.paho.org/es/noticias/30-9-2020-personas-mayores-60-anos-han-sido-mas-afectadas-por-covid-19-america#:~:text=Si%20bien%20todos%20est%C3%A1n%20el,veces%20mayor%20que%20la%20media..> [Último acceso: 12 05 2022].
- [4] inredh, «inredh,» 24 07 2020. [En línea]. Available: <https://inredh.org/ecuador-personas-de-tercera-edad-frente-al-covid-19/#:~:text=De%20esta%20cifra%2C%20se%20estima,general%2C%20como%20veremo%20a%20continuaci%C3%B3n..> [Último acceso: 12 05 2020].
- [5] «publications,» publications, 12 2020. [En línea]. Available: <https://publications.iadb.org/publications/spanish/document/Envejecimiento-y-atencion-a-la-dependencia-en-Ecuador.pdf>. [Último acceso: 18 05 2022].
- [6] gcefe, «gcefe,» 2020. [En línea]. Available: <https://grupoconsultorefe.com/servicio/tecnologias-de-la-informacion/sistemas-web#:~:text=Se%20denomina%20sistema%20web%20a,una%20intranet%20mediante%20n%20navegador..> [Último acceso: 12 05 2020].
- [7] d. comunicacion, «descubre comunicacion,» 03 04 2022. [En línea]. Available: <https://descubrecomunicacion.com/que-es-backend-y-frontend/>. [Último acceso: 12 05 2022].
- [8] internetya, «internetya,» 14 11 2016. [En línea]. Available: <https://www.internetya.co/ventajas-y-beneficios-de-las-aplicaciones-web/>. [Último acceso: 16 05 2022].
- [9] wikipedia, «wikipedia,» 03 05 2022. [En línea]. Available: [https://en.wikipedia.org/wiki/Scrum_\(software_development\)](https://en.wikipedia.org/wiki/Scrum_(software_development)). [Último acceso: 16 05 2020].
- [10] N. unidas, «Naciones unidas,» 27 03 2020. [En línea]. Available: <https://news.un.org/es/story/2020/03/1471932>. [Último acceso: 12 05 2022].
- [11] scielo, «scielo,» 01 03 2021. [En línea]. Available: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1028-99332021000200004. [Último acceso: 12 05 2022].
- [12] Q. Devs, «Quality Devs,» 16 09 2019. [En línea]. Available: <https://www.qualitydevs.com/2019/09/16/que-es-angular-y-para-que-sirve/>. [Último acceso: 05 2022].
- [13] OpenWebinars, «OpenWebinars,» 22 06 2021. [En línea]. Available: <https://openwebinars.net/blog/que-es-firebase-de-google/#:~:text=Firebase%20b%C3%A1sicamente%20es%20una%20plataforma,incrementar%20la%20base%20de%20datos>. [Último acceso: 12 05 2022].
- [14] ichlese, «ichlese,» [En línea]. Available: <https://es.ichlese.at/what-is-angular-material>. [Último acceso: 12 05 2020].

- [15] CEI, «CEI,» [En línea]. Available: <https://cei.es/que-es-figma/>. [Último acceso: 12 05 2022].
- [16] wikipedia, «wikipedia,» 03 05 2022. [En línea]. Available: https://en.wikipedia.org/wiki/Visual_Studio_Code. [Último acceso: 16 05 2022].
- [17] crehana, «crehana,» 08 02 2021. [En línea]. Available: https://www.crehana.com/blog/desarrollo-web/que-es-nodejs/?gclid=EAlalQobChMI7df-z4bT9QIVqwaICR2RZQ9REAAAYASAAEgJLzvd_BwE. [Último acceso: 16 05 2022].
- [18] loadview, «loadview,» 16 10 2020. [En línea]. Available: <https://www.loadview-testing.com/es/blog/tipos-de-pruebas-de-software-diferencias-y-ejemplos/>. [Último acceso: 16 05 2022].
- [19] datacentric, «datacentric,» 14 10 2015. [En línea]. Available: <https://www.datacentric.es/blog/bases-datos/importancia-bases-de-datos-2/#:~:text=Las%20principales%20utilidades%20que%20ofrece,la%20organizaci%C3%B3n%20de%20nuestra%20actividad..> [Último acceso: 19 05 2022].
- [20] highscalability, «highscalability,» 27 09 2021. [En línea]. Available: <https://highscalability.wordpress.com/tag/despliegue-de-software/>. [Último acceso: 18 05 2022].
- [21] rockcontent, «rockcontent,» 16 04 2019. [En línea]. Available: <https://rockcontent.com/es/blog/que-es-firebase/>. [Último acceso: 18 05 2022].
- [22] servnet, «servnet,» 30 03 2021. [En línea]. Available: <https://www.servnet.mx/blog/backend-y-frontend-partes-fundamentales-de-la-programaci%C3%B3n-de-una-aplicaci%C3%B3n-web>. [Último acceso: 18 05 2022].
- [23] questionpro, «questionpro,» [En línea]. Available: <https://www.questionpro.com/blog/es/que-es-un-estudio-de-caso/>. [Último acceso: 18 05 2022].
- [24] santander, 21 12 2020. [En línea]. Available: <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html#:~:text=Las%20metodolog%C3%ADas%20de%20desarrollo%20de%20software%20son%20un%20conjunto%20de,dise%C3%B1ar%20soluciones%20de%20software%20inform%C3%A1tico..> [Último acceso: 18 05 2022].
- [25] bbva, «bbva,» 2021. [En línea]. Available: <https://www.bbva.com/en/agile-methodology-la-revolution-ways-working/>. [Último acceso: 18 05 2022].
- [26] proyectum, «proyectum,» 19 10 2016. [En línea]. Available: <https://www.proyectum.com/sistema/blog/los-tres-principales-roles-en-scrum/#:~:text=En%20la%20metodolog%C3%ADa%20Scrum%20podemos,del%20proyecto%20en%20su%20totalidad..> [Último acceso: 18 05 2022].
- [27] proyectum, «proyectum,» 19 10 2016. [En línea]. Available: <https://www.proyectum.com/sistema/blog/los-tres-principales-roles-en-scrum/#:~:text=En%20la%20metodolog%C3%ADa%20Scrum%20podemos,del%20proyecto%20en%20su%20totalidad..> [Último acceso: 18 05 2022].
- [28] asana, «asana,» 3 09 2021. [En línea]. Available: <https://asana.com/es/resources/scrum-master>. [Último acceso: 18 05 2022].

- [29] mamaqueesscrum, «mamaqueesscrum,» 29 04 2020. [En línea]. Available: <https://mamaqueesscrum.com/2020/04/29/que-es-un-development-team-os-proponemos-una-dinamica/>. [Último acceso: 18 05 2022].
- [30] kaizenia, «kaizenia,» [En línea]. Available: <https://kzi.mx/que-son-los-artefactos-de-scrum/>. [Último acceso: 18 05 2022].
- [31] asana, «asana,» 15 11 2021. [En línea]. Available: <https://asana.com/es/resources/requirements-gathering>. [Último acceso: 18 05 2022].
- [32] digite, «digite,» 2020. [En línea]. Available: <https://www.digite.com/es/agile/historias-de-usuarios/>. [Último acceso: 18 05 2022].
- [33] ealde, «ealde,» 27 08 2019. [En línea]. Available: <https://www.ealde.es/product-backlog-sprint-backlog/>. [Último acceso: 18 05 2022].
- [34] integrait, «integrait,» [En línea]. Available: <https://integrait.com.mx/blog/sprint-y-sprint-backlog/>. [Último acceso: 18 05 2022].
- [35] multimedia, «multimedia,» [En línea]. Available: <https://multimedia.uoc.edu/blogs/dii/es/que-es-una-interficie/>. [Último acceso: 18 05 2022].
- [36] lucidchart, «lucidchart,» [En línea]. Available: <https://www.lucidchart.com/blog/es/como-disenar-una-arquitectura-de-software#:~:text=El%20dise%C3%B1o%20de%20una%20arquitectura,comenzar%20a%20elaborar%20un%20prototipo..> [Último acceso: 18 05 2022].
- [37] c. mvp, «campus mvp,» 15 10 2019. [En línea]. Available: <https://www.campusmvp.es/recursos/post/que-es-el-patron-mvc-en-programacion-y-por-que-es-util.aspx#:~:text=MVC%20era%20inicialmente%20un%20patr%C3%B3n,entre%20cada%20uno%20de%20ellos..> [Último acceso: 18 05 2022].
- [38] okhosting, «okhosting,» [En línea]. Available: <https://okhosting.com/blog/herramientas-de-desarrollo-de-software/>. [Último acceso: 18 05 2022].
- [39] npmjs, «npmjs,» 09 05 2022. [En línea]. Available: <https://www.npmjs.com/package/firebase>. [Último acceso: 19 05 2022].
- [40] npmjs, «npmjs,» 19 03 2022. [En línea]. Available: <https://www.npmjs.com/package/@angular/fire>. [Último acceso: 19 05 2022].
- [41] libraries, «libraries,» [En línea]. Available: <https://libraries.io/npm/ngx-cookie-service>. [Último acceso: 19 05 2022].
- [42] npmjs, «npmjs,» 17 05 2022. [En línea]. Available: <https://www.npmjs.com/package/@ng-bootstrap/ng-bootstrap>. [Último acceso: 19 05 2022].
- [43] material, «material,» [En línea]. Available: <https://material.angular.io/>. [Último acceso: 19 05 2022].
- [44] yeeply, «yeeply,» [En línea]. Available: <https://www.yeeply.com/blog/que-son-pruebas-unitarias/>. [Último acceso: 26 05 2022].

- [45] loadview, «loadview,» [En línea]. Available: <https://www.loadview-testing.com/es/pruebas-de-carga/>. [Último acceso: 16 06 2022].
- [46] acens, «acens,» [En línea]. Available: <https://www.acens.com/wp-content/images/2016/02/google-pagespeed-insights-wp-acens.pdf>. [Último acceso: 07 07 2022].
- [47] ebooksonline, «ebooksonline,» 13 03 2021. [En línea]. Available: <https://ebooksonline.es/que-es-una-prueba-de-compatibilidad-prueba-hacia-adelante-y-hacia-atras-ejemplo/>. [Último acceso: 26 05 2022].
- [48] manuel, «manuel,» [En línea]. Available: <https://manuel.cillero.es/doc/metodologia/metrica-3/tecnicas/pruebas/acceptacion/>. [Último acceso: 26 05 2022].
- [49] wikipedia, «wikipedia,» 02 02 2022. [En línea]. Available: [https://en.wikipedia.org/wiki/Figma_\(software\)](https://en.wikipedia.org/wiki/Figma_(software)). [Último acceso: 18 05 2022].

7 ANEXOS

A continuación, se presentan todos los Anexos que se han utilizado en el desarrollo del sistema *web* los cuales se encuentran divididos de la siguiente manera:

- **ANEXO I.** Resultado del programa antiplagio Turnitin.
- **ANEXO II.** Manual Técnico.
- **ANEXO III.** Manual de Usuario.
- **ANEXO IV.** Manual de Instalación.

ANEXO I

A continuación, se presenta el certificado que el Director de Tesis ha emitido y en donde se evidencia el resultado que se ha obtenido en la herramienta antiplagio Turnitin.



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

CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 22 de agosto de 2022

De mi consideración:

Yo, Loarte Cajamarca Byron Gustavo, en calidad de Director del Trabajo de Integración Curricular titulado Desarrollo de un sistema asociado al DESARROLLO DE SISTEMA WEB INFORMATIVO DE ASILOS EN LA CIUDAD DE QUITO elaborado por el estudiante Jose Antonio Borja Fweltala de la carrera en Tecnología Superior en Desarrollo de Software, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito secciones: Descripción del componente desarrollado, Metodología, Resultados, Conclusiones y Recomendaciones, producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 10%.

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,

Loarte Cajamarca Byron Gustavo
Profesor Ocasional a Tiempo Completo
Escuela de Formación de Tecnólogos

ANEXO II

Recopilación de requerimientos

En la **TABLA X** se muestra los requerimientos que se han obtenido al inicio del proyecto de acuerdo con lo solicitado por el dueño del producto.

TABLA X: Recopilación de requerimientos.

RECOPIACIÓN DE REQUERIMIENTOS		
TIPO DEL SISTEMA	ID – RR	ENUNCIADO DEL ÍTEM
SISTEMA WEB	RR001	Como usuario administrador general, administrador del asilo y ciudadano necesitan visualizar una página informativa del sistema web.
	RR002	Como usuario administrador necesita realizar lo siguiente: <ul style="list-style-type: none"> • Iniciar sesión. • Cerrar sesión. • Cambiar contraseña.
	RR003	Como usuario administrador necesita realizar lo siguiente: <ul style="list-style-type: none"> • Gestionar su perfil.
	RR004	Como usuario administrador necesita realizar lo siguiente: <ul style="list-style-type: none"> • Gestionar asilos.
	RR005	Como usuario administrador del asilo necesita realizar lo siguiente: <ul style="list-style-type: none"> • Registrarse mediante un formulario.
	RR006	Como usuario administrador del asilo necesita realizar lo siguiente: <ul style="list-style-type: none"> • Iniciar sesión. • Cerrar sesión. • Cambiar contraseña.
	RR007	Como usuario administrador del asilo necesita realizar lo siguiente:

		<ul style="list-style-type: none"> • Gestionar su perfil.
	RR008	Como usuario administrador del asilo necesita realizar lo siguiente: Enviar solicitud de aprobación
	RR009	Como usuario administrador del asilo necesita realizar lo siguiente: <ul style="list-style-type: none"> • Registrar información del asilo.

Historias de Usuario

Culminada la etapa de Recopilación de requerimientos y el análisis respectivo de las mismas. A continuación, se presentan las 13 Historias de Usuario que se han implementado en este proyecto las cuales van desde la **TABLA XI** hasta la **TABLA XX**.

TABLA XI: Historia de Usuario 11 – Iniciar sesión, cerrar sesión y cambiar de contraseña.

HISTORIA DE USUARIO	
Identificador (ID): HU002	Usuario: Administrador
Nombre historia: Iniciar sesión, cerrar sesión y cambiar de contraseña	
Prioridad en negocio: Medio	Riesgo en desarrollo: Medio
Sprint asignado: SB001	
Programador responsable: Jose Antonio Borja	
Descripción: El usuario con perfil administrador en el sistema <i>web</i> tiene la posibilidad de iniciar sesión mediante un usuario y contraseña. Además, tiene la posibilidad de cerrar sesión cuando lo requiera y cambiar de contraseña.	
Observación: Una vez que se ingresa usuario y contraseña de forma incorrecta, el sistema <i>web</i> presenta un mensaje de alerta cuando los datos ingresados no son correctos.	

TABLA XII: Historia de Usuario 11 – Gestionar perfil.

HISTORIA DE USUARIO	
Identificador (ID): HU003	Usuario: Administrador
Nombre historia: Gestionar perfil	
Prioridad en negocio: Medio	Riesgo en desarrollo: Medio
Sprint asignado: SB001	
Programador responsable: Jose Antonio Borja	
Descripción: El usuario con perfil administrador en el sistema web tiene la posibilidad de modificar su información de perfil una vez que haya iniciado sesión. Los datos que se pueden modificar son: <ul style="list-style-type: none"> • Nombre de usuario • Avatar • Correo • Apellido • Dirección • Ciudad 	
Observación: Todos los campos son obligatorios y validados. En el caso de que la información ingresada no sea correcta se muestra un mensaje de error, caso contrario se presenta un mensaje con la información modificada satisfactoriamente.	

TABLA XIII: Historia de Usuario 13 – Gestionar asilos.

HISTORIA DE USUARIO	
Identificador (ID): HU004	Usuario: Administrador
Nombre historia: Gestionar asilos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio

Sprint asignado: SB001
Programador responsable: Jose Antonio Borja
Descripción: El usuario con perfil administrador en el sistema <i>web</i> tiene la posibilidad de listar la información completa de los nuevos asilos que deseen forma parte del sistema <i>web</i> . Además, puede aprobar a cada uno de los asilos, así como rechazar a cada uno de ellos si fuera el caso.
Observación: El usuario administrador tiene la posibilidad de observar el contenido de las solicitudes y se le presenta la opción de rechazar. Además, el sistema <i>web</i> envía una notificación al usuario administrador del asilo indicando que su petición ha sido rechazada junto con una respuesta.

TABLA XIV: Historia de Usuario 14 – Registrarse mediante un formulario.

HISTORIA DE USUARIO	
Identificador (ID): HU005	Usuario: Administrador del asilo
Nombre historia: Registrarse mediante un formulario	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Sprint asignado: SB001	
Programador responsable: Jose Antonio Borja	
Descripción: El usuario con perfil administrador del asilo en el sistema <i>web</i> tiene la posibilidad de crear una nueva cuenta por medio de un formulario de registro en el que debe llenar la siguiente información: <ul style="list-style-type: none"> • Nombre • Apellido • Correo • Contraseña • Fecha nacimiento • Dirección 	
Observación: El sistema <i>web</i> muestra un mensaje de error cuando se presente problemas en la	

validación de los campos, caso contrario se presenta un mensaje con la información registrada satisfactoriamente y proceder con el inicio de sesión respectivo.

TABLA XV: Historia de Usuario 15 – Iniciar sesión, cerrar sesión y cambiar de contraseña.

HISTORIA DE USUARIO	
Identificador (ID): HU006	Usuario: Administrador del asilo
Nombre historia: Iniciar sesión, cerrar sesión y cambiar de contraseña	
Prioridad en negocio: Medio	Riesgo en desarrollo: Medio
Sprint asignado: SB002	
Programador responsable: Jose Antonio Borja	
Descripción: El usuario con perfil administrador del asilo en el sistema <i>web</i> tiene la posibilidad de iniciar sesión mediante un usuario y contraseña. Además, tiene la posibilidad de cerrar sesión cuando lo requiera y cambiar de contraseña.	
Observación: Una vez que se ingresa usuario y contraseña de forma incorrecta, el sistema <i>web</i> presenta un mensaje de alerta cuando los datos ingresados no son correctos. Caso contrario se procede a ingresar a los módulos asignados a su cargo.	

TABLA XVI: Historia de Usuario 16 – Gestionar perfil.

HISTORIA DE USUARIO	
Identificador (ID): HU007	Usuario: Administrador del asilo
Nombre historia: Gestionar perfil	
Prioridad en negocio: Medio	Riesgo en desarrollo: Medio
Sprint asignado: SB002	
Programador responsable: Jose Antonio Borja	
Descripción: El usuario con perfil administrador del asilo en el sistema <i>web</i> tiene la posibilidad de	

modificar su información de perfil una vez que haya iniciado sesión. Los datos que se pueden modificar son:

- Nombre de usuario
- Avatar
- Correo
- Apellido
- Dirección
- Ciudad o País

Observación:

Todos los campos son obligatorios y validados. En el caso de que la información ingresada no sea correcta se muestra un mensaje de error, caso contrario se presenta un mensaje con la información modificada satisfactoriamente.

TABLA XVII: Historia de Usuario 17 – Enviar solicitud de aprobación.

HISTORIA DE USUARIO	
Identificador (ID): HU008	Usuario: usuario administrador del asilo
Nombre historia: Enviar solicitud de aprobación	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Sprint asignado: SB002	
Programador responsable: Jose Antonio Borja	
<p>Descripción:</p> <p>El usuario con perfil administrador del asilo en el sistema <i>web</i> tiene la posibilidad de enviar una solicitud de afiliación para forma parte del sistema <i>web</i>. Además, para realizar dicha acción el usuario debe llenar la siguiente información:</p> <ul style="list-style-type: none"> • Nombre del asilo • Dirección • Foto • Documento • Correo 	
<p>Observación:</p> <p>El sistema <i>web</i> verifica que no exista campos vacíos al momento de enviar la solicitud, si existe una inconsistencia se muestra un mensaje de “Error”, caso contrario se presenta un mensaje de “Solicitud enviada exitosamente”.</p>	

TABLA XVIII: Historia de Usuario 18 – Registrar información del asilo.

HISTORIA DE USUARIO	
Identificador (ID): HU009	Usuario: usuario administrador del asilo
Nombre historia: Registrar información del asilo.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Sprint asignado: SB002	
Programador responsable: Jose Antonio Borja	
<p>Descripción:</p> <p>El usuario con perfil administrador del asilo en el sistema <i>web</i> una vez que su solicitud fue aprobada tiene la posibilidad de ingresar información del asilo a través de un formulario llenando los siguientes campos:</p> <p>Información del Asilo:</p> <ul style="list-style-type: none"> • Nombre • Ubicación • Dirección • Teléfono • Correo <p>Propietario del Asilo:</p> <ul style="list-style-type: none"> • Nombres y apellidos • Correo electrónico • Número de cédula • Dirección 	
<p>Observación:</p> <p>El sistema <i>web</i> verifica que no exista duplicidad de información revisando que en la Base de Datos el nombre del asilo no haya sido ingresado previamente. Por otra parte, el sistema <i>web</i> muestra un mensaje de error cuando se presente errores en la validación de campos.</p>	

TABLA XIX: Historia de Usuario 19 – Visualizar información de asilos.

HISTORIA DE USUARIO	
Identificador (ID): HU010	Usuario: Ciudadano

Nombre historia: Visualizar información de asilos.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Sprint asignado: SB002	
Programador responsable: Jose Antonio Borja	
Descripción: El usuario con perfil ciudadano en el sistema <i>web</i> tiene la posibilidad de ver la información de los asilos que estén registrados, así como toda la información relacionada a cada uno de ellos.	
Observación: El sistema <i>web</i> presenta todos los asilos registrados y la opción de poder filtrarlos según se requiera.	

TABLA XX: Historia de Usuario 20 – Realizar proforma.

HISTORIA DE USUARIO	
Identificador (ID): HU011	Usuario: usuario ciudadano
Nombre historia: Realizar proforma.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Sprint asignado: SB002	
Programador responsable: Jose Antonio Borja	
Descripción: El usuario con perfil ciudadano en el sistema <i>web</i> tiene la posibilidad de visualizar un simulador de los costos por cada uno de los servicios que oferta el asilo asignado.	
Observación: El sistema <i>web</i> una vez que finalice el proceso del simulador esta información es proporcionada por medio de un PDF para que pueda ser descargado.	

Product Backlog

En la **TABLA XXI** se presenta la prioridad de cada requisito que se ha implementado en el sistema *web*. Estos requisitos se clasifican de acuerdo con las necesidades del dueño del producto y la complejidad del desarrollo.

TABLA XXI: *Product Backlog*.

ELABORACIÓN DEL <i>PRODUCT BACKLOG</i>				
ID – HU	HISTORIA DE USUARIO	ITERACIÓN	ESTADO	PRIORIDAD
HU001	Visualizar página informativa	1	Finalizado	Terminado
HU002	Iniciar sesión, cerrar sesión y cambiar de contraseña.	1	Finalizado	Terminado
HU003	Gestionar perfil	1	Finalizado	Terminado
HU004	Gestionar asilos.	1	Finalizado	Terminado
HU005	Registrarse mediante un formulario.	1	Finalizado	Terminado
HU 006	Iniciar sesión, cerrar sesión y cambiar de contraseña.	2	Finalizado	Terminado
HU007	Gestionar perfil	2	Finalizado	Terminado
HU008	Enviar solicitud de aprobación	2	Finalizado	Terminado
HU009	Registrar información del asilo.	2	Finalizado	Terminado

Sprint Backlog

A continuación, se presenta el detalle de los seis *Sprints* de desarrollo, que han sido implementados en el desarrollo del sistema *web*, como se presenta en la **TABLA XXII**.

TABLA XXII: *Sprint Backlog*.

ELABORACIÓN DEL <i>SPRINT BACKLOG</i>						
ID – SB	NOMBRE	MÓDULO	ID-HU	HISTORIAS DE USUARIO	TAREAS	TIEMPO ESTIMADO
SB000	Estructuración en el ambiente de desarrollo	-----	-----	-----	<ul style="list-style-type: none">• Definir los requerimientos funcionales y no funcionales del sistema <i>web</i>.• Diseño de la base de datos.• Configuración de las herramientas de desarrollo.• Configuración del ambiente de desarrollo para el Sistema	10H
		Módulo – Pagina informativa	HU001	Visualizar página informativa	<ul style="list-style-type: none">• Diseño de la interfaz para la página informativa• Definir métodos para la visualización de restaurantes• Pruebas con cualquier usuario	15H

SB001	Diseño e implementación de los módulos para el usuario administrador.					
		Módulo <i>Login</i>	HU002	Iniciar sesión, cerrar sesión y cambio de contraseña.	<ul style="list-style-type: none"> • Diseño de la interfaz para inicio y cierre de sesión de cada usuario. • Diseño de la interfaz para el cambio de contraseña. • Validación de credenciales de usuarios. • Validación de campos • Pruebas con su respectivo usuario administrador. 	35H
		Módulo del Admin	HU003	Gestionar perfil	<ul style="list-style-type: none"> • Diseño e implementación de interfaz para gestión del perfil • Diseño e implementación de un perfil donde se pueda guardar, visualizar, editar y cambiar información del ADMIN. • Reglas de validación para cada campo del formulario. 	

					<ul style="list-style-type: none"> • Pruebas de guardar, visualización, edición el perfil. 	45h
		Módulo aprobación asilo	HU004	Gestionar asilos.	<ul style="list-style-type: none"> • Diseño e implementación de interfaz para gestionar asilos. • Diseño e implementación de la tabla donde se pueda, visualizar, todos los asilos listos para aprobar o rechazar • Reglas de validación para cada campo del formulario. • Prueba para visualizar, aprobar o rechazar a los asilos 	
		Módulo registro	HU005	Registrarse mediante un formulario.	<ul style="list-style-type: none"> • Diseño e implementación de interfaz para registrar mediante un formulario. • Diseño e implementación formulario para registrar la información del 	

	Diseño e implementación de los módulos para el usuario administrador del asilo.				<p>asilo.</p> <ul style="list-style-type: none"> • Establecer reglas de validación para campo del formulario. • Guardar formulario. • Prueba para registrar, visualizar, la información del asilo. 	
		Módulo <i>login</i> (asilo)	HU006	<p>Iniciar sesión, cerrar sesión y cambiar de contraseña. (asilo)</p>	<ul style="list-style-type: none"> • Diseño de la interfaz para inicio y cierre de sesión de cada usuario. • Diseño de la interfaz para el cambio de contraseña. • Validación de credenciales de usuarios. • Validación de campos • Pruebas con su respectivo usuario administrador. 	
		Módulo del		Gestionar perfil	<ul style="list-style-type: none"> • Diseño e implementación de interfaz para gestión del perfil • Diseño e implementación de un perfil donde se pueda guardar, visualizar, editar y cambiar información del ADMIN asilo. 	45H

SB002		Admin asilo	HU007	Asilo	<ul style="list-style-type: none"> • Reglas de validación para cada campo del formulario. • Pruebas de guardar, visualización, edición el perfil. 	
			HU008	Enviar solicitud de aprobación	<ul style="list-style-type: none"> • Diseño e implementación de interfaz para gestión del perfil • Diseño e implementación de una solicitud de afiliación para forma parte del sistema <i>web</i>. Mediante un formulario • Reglas de validación para cada campo del formulario. • Pruebas de guardar y visualización de la información del asilo registrado y afiliado 	
			HU009	Registrar información del asilo.	<ul style="list-style-type: none"> • Diseño e implementación de interfaz para registrar información del asilo • Diseño e implementación de un registro el cual se pueda llenar más información sobre el asilo mediante un formulario 	

					<ul style="list-style-type: none"> • Reglas de validación para cada campo del formulario. • Pruebas de guardar y visualización de la información del asilo 	
			HU010	Visualizar información de asilos	<ul style="list-style-type: none"> • Diseño e implementación de interfaz para visualizar información de asilos • Diseño e implementación de una página donde se va visualiza la información de los asilos • Pruebas de guardar, visualización de la información de asilos en la página principal 	
SB004	Pruebas y despliegue del sistema de <i>web</i>	<ul style="list-style-type: none"> • Pruebas unitarias. • Prueba de rendimiento. • Prueba de compatibilidad. • Prueba de aceptación. 				10H
SB005	Despliegue del sistema de <i>web</i>	<ul style="list-style-type: none"> • Publicación del sistema <i>web</i> en la consola de <i>Firebase</i> 				10H

Documentación	<ul style="list-style-type: none"> • Trabajo de Integración Curricular. • Anexos. 	35H
TOTAL		240 H

Diseño de interfaces

A continuación se presenta todos los prototipos que se han diseñado las cuales van desde la **Fig. 24** hasta la **Fig. 30**.

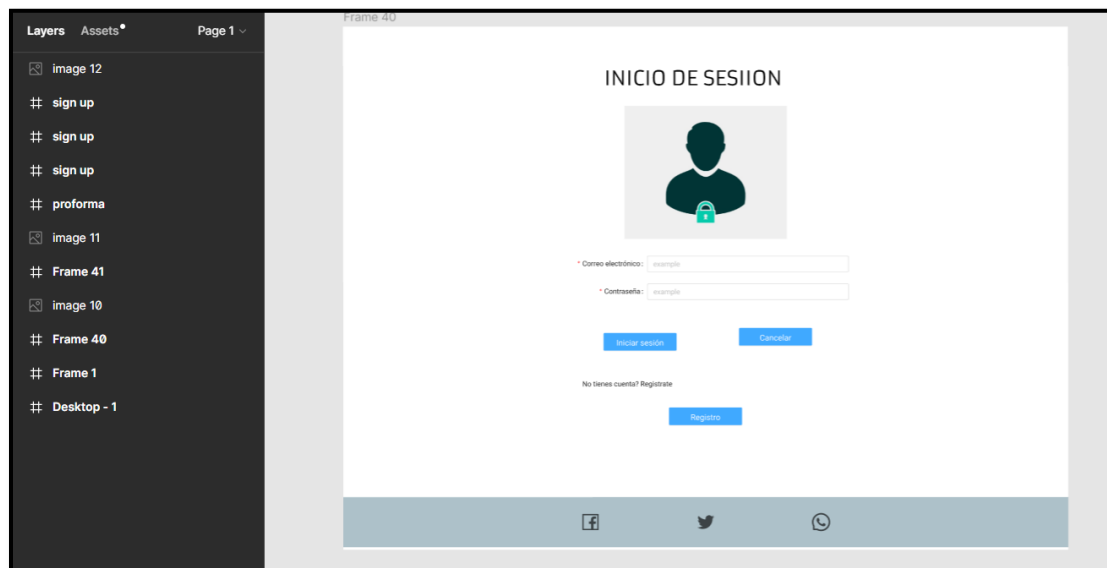


Fig. 24: iniciar sesión.

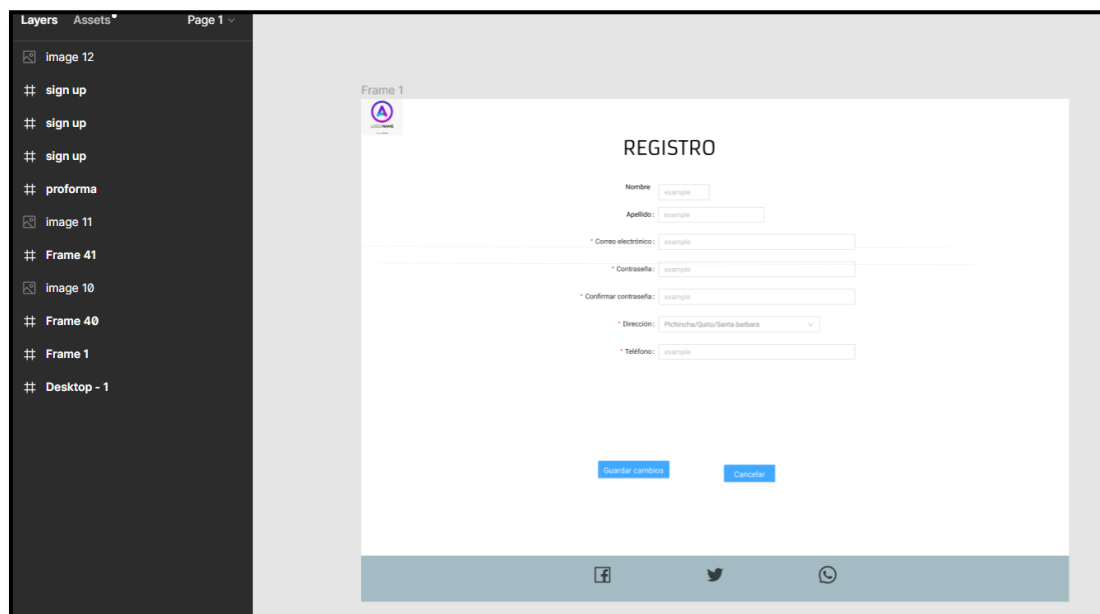


Fig. 25: registro.

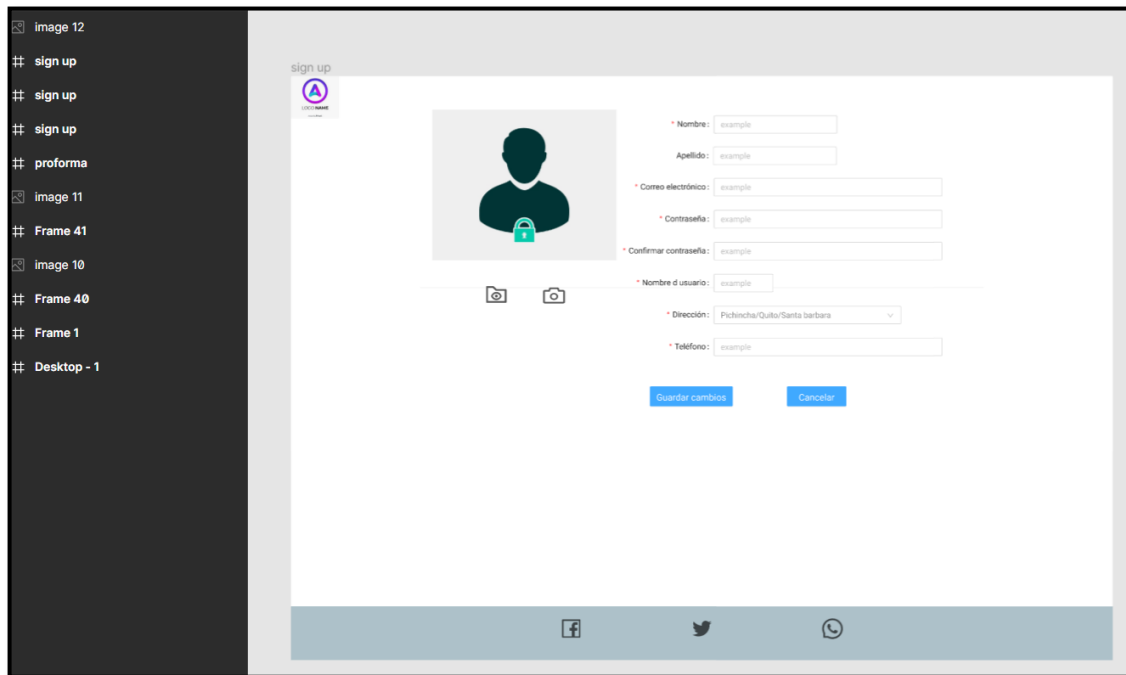


Fig. 26: módulo perfil.

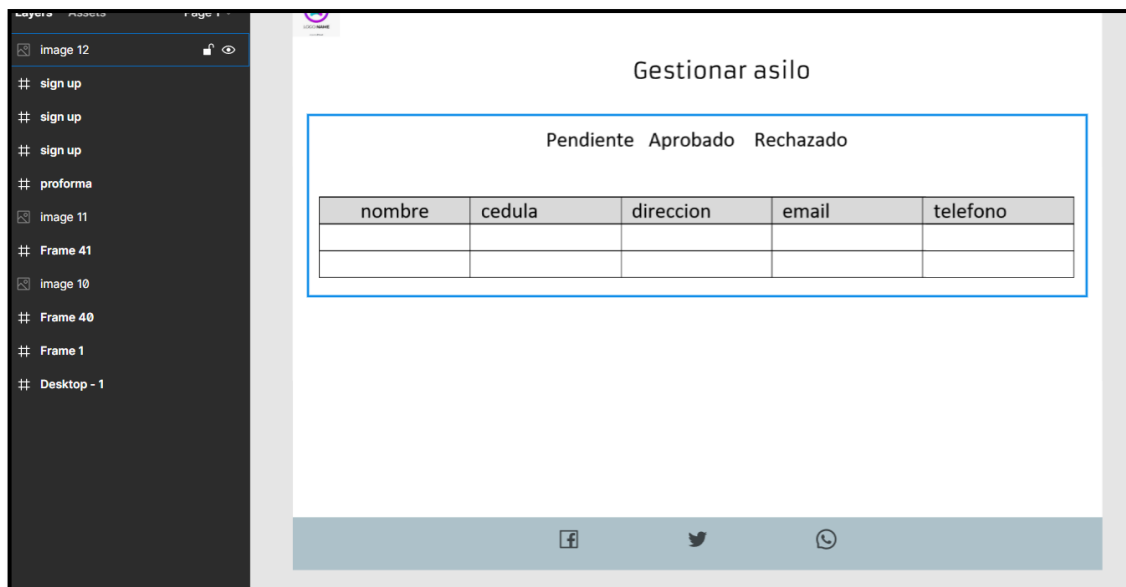


Fig. 27: gestionar asilos pendientes, aceptados o rechazados.

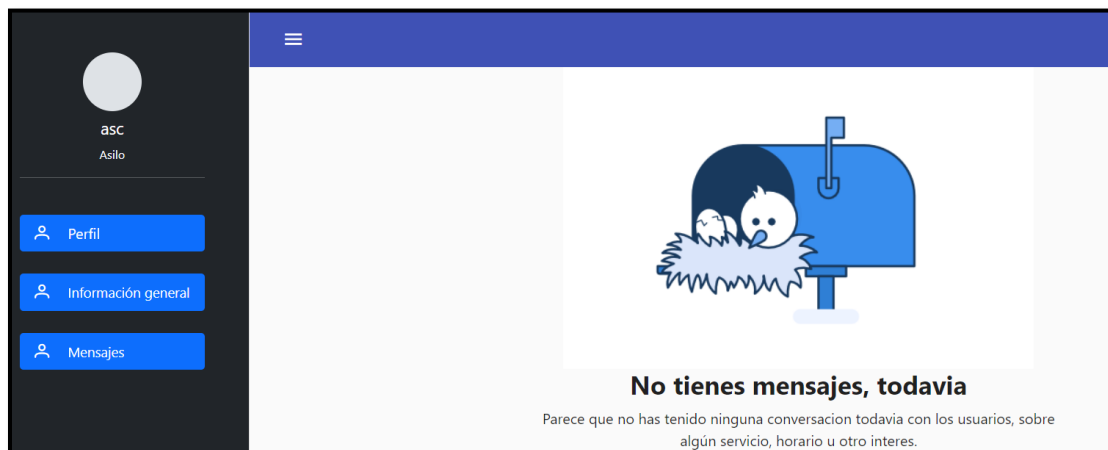


Fig. 28: modulo mensajes.

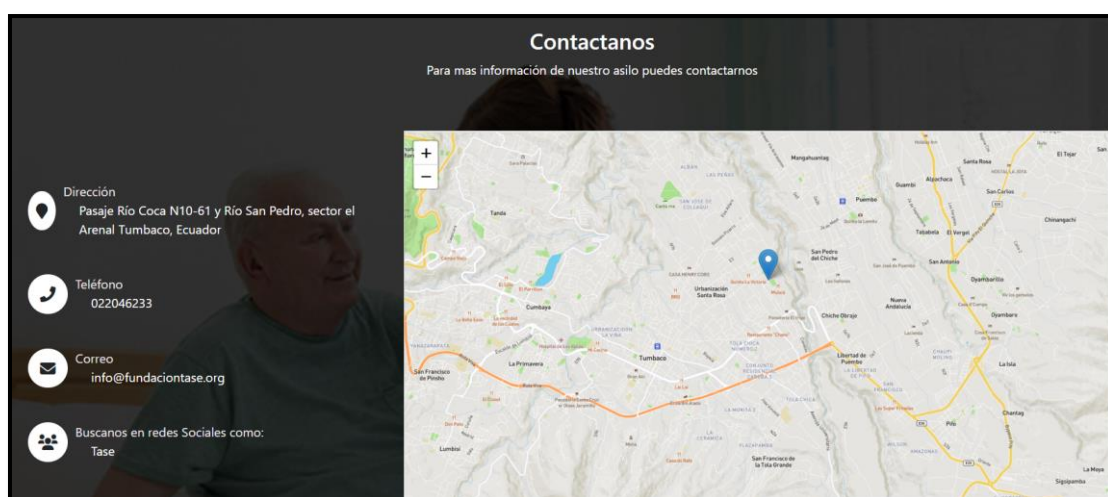


Fig. 29: localización del asilo tiempo real.



Fig. 30: chat para usuario.

Diseño de la Base de datos

En las **Fig. 31** y **Fig. 32** se puede visualizar la base de datos no relacional (NoSQL) utilizada para el sistema web, debido a que la Base de datos es no relacional y para la realización del diagrama como entidad relación no puede ser posible, ya que en este caso se usa colecciones como también documentos.

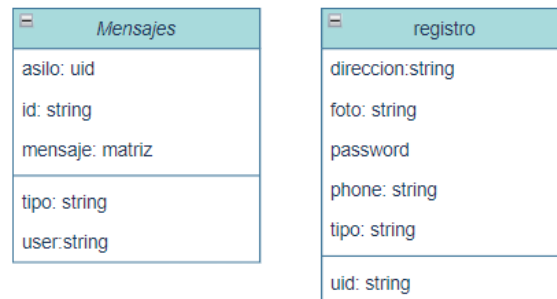


Fig. 31: diseño de la base de datos

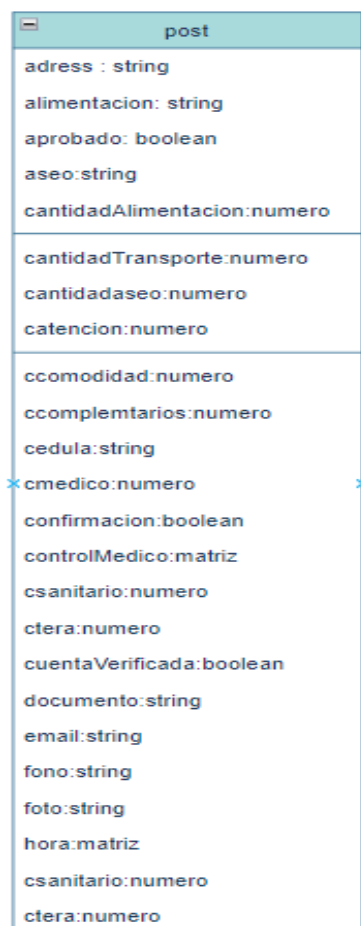


Fig. 32: Diseño de la base de datos

Pruebas

Terminada la etapa de codificación se ha desarrollado la ejecución de pruebas unitarias, rendimiento, compatibilidad y de aceptación para comprobar la calidad del sistema web.

Ejecución de pruebas unitarias y resultados

Desde **Fig. 33** la hasta **Fig. 37** se puede observar la ejecución de la prueba en los módulos principales del sistema web mediante el uso de la herramienta karma.

```
C:\Users\Lenovo\Documents\GitHub\proyecto_final>ng test

"Generating browser application bundles (phase: building)...14 11 2021 00:39:13.843:WARN [karma]: No captured browser, open http://localhost:9876/
```

Fig. 33: Comando ng test.

```
//Registro de usuarios
export const signUp = (email, password) => {
  auth()
    .createUserWithEmailAndPassword(email, password)
    .then((user) => {
      const fbRootReffFS = firebase.firestore();
      const userID = user.user.uid;
      console.log('user ID is:', userID);
      const userRef = fbRootReffFS.collection('users').doc(userID);
      userRef.set({
        email,
        password
      });
    });
}
```

Fig. 34: prueba registro.

```
✓ Registro de usuario - registro@gmail.com: (440 ms)
<Login />
✓ Si existe el boton (850 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        4.205 s, estimated 5 s
Ran all test suites.
```

Fig. 35: resultado

```
//Modulo principal
describe("<HomeUser />", () =>{
  beforeEach(()=>{
    component = render(<HomeUser />)
  });
  it("Renderiza correctamente", () =>{
    expect(component).toBeDefined();
  })
})
```

Fig. 36: función en el inicio.

```
✓ Renderiza correctamente (2882 ms)
Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        11.525 s, estimated 15 s
Ran all test suites.
```

Fig. 37: resultado

Ejecución y prueba de rendimiento y resultados

A continuación, se presenta el resultado de la prueba de rendimiento en donde se utiliza la siguiente página web *Pagespeed Insights* en donde se observa el rendimiento del sistema web como observar desde la Fig. 38 hasta la Fig. 40

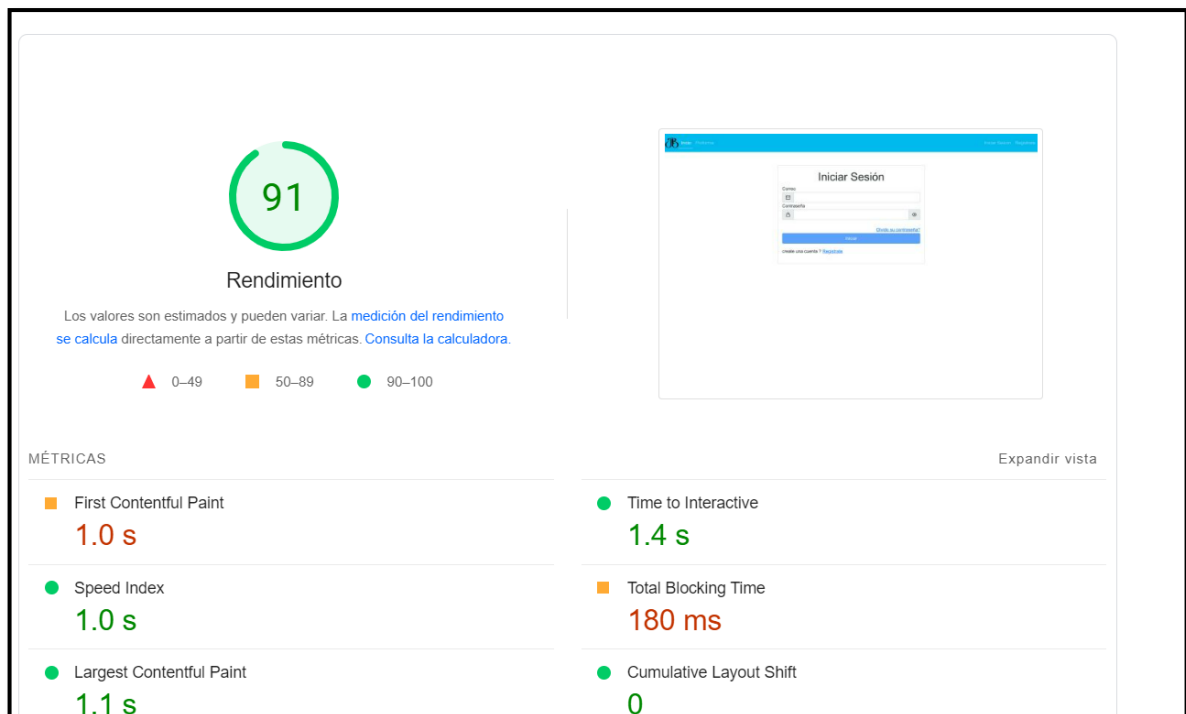


Fig. 38: prueba *login*.

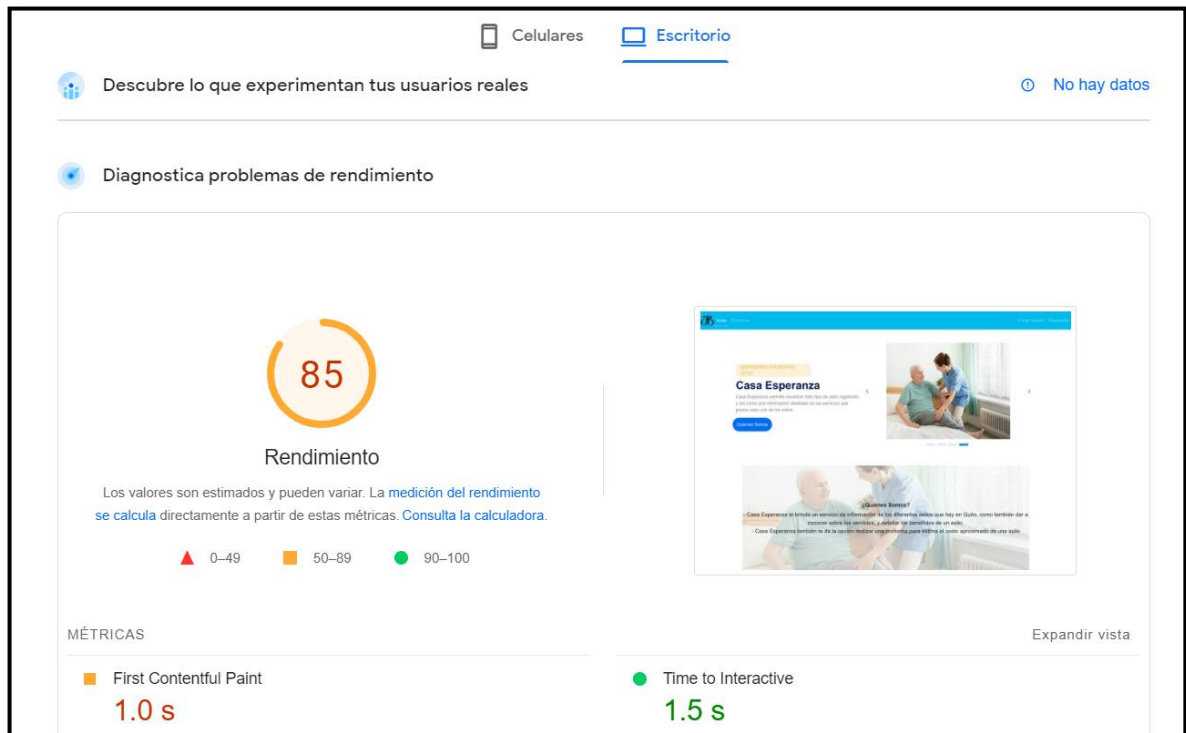


Fig. 39: prueba de inicio del sistema.

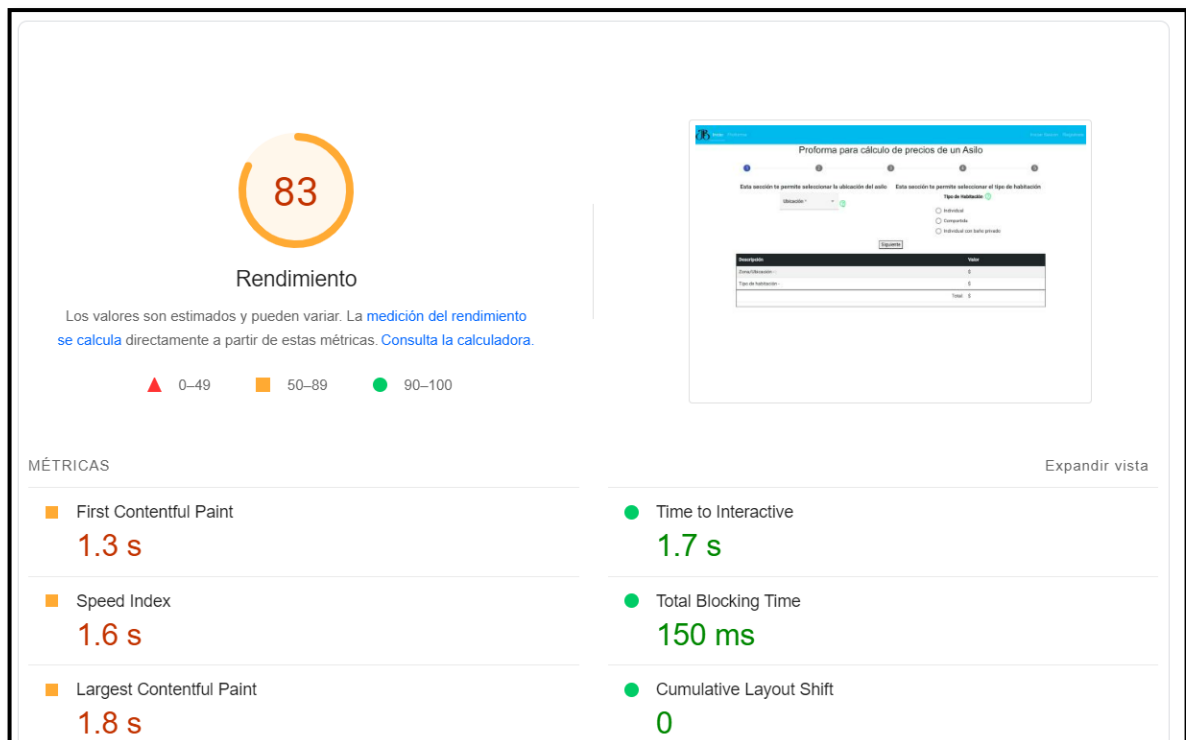


Fig. 40: prueba proforma.

Ejecución de prueba de compatibilidad y resultados

A continuación, se presentan los resultados de las pruebas de compatibilidad, en donde se utilizan los siguientes navegadores como se puede observar en cada una de las figuras que van desde la **Fig. 41** hasta la **Fig. 43**.

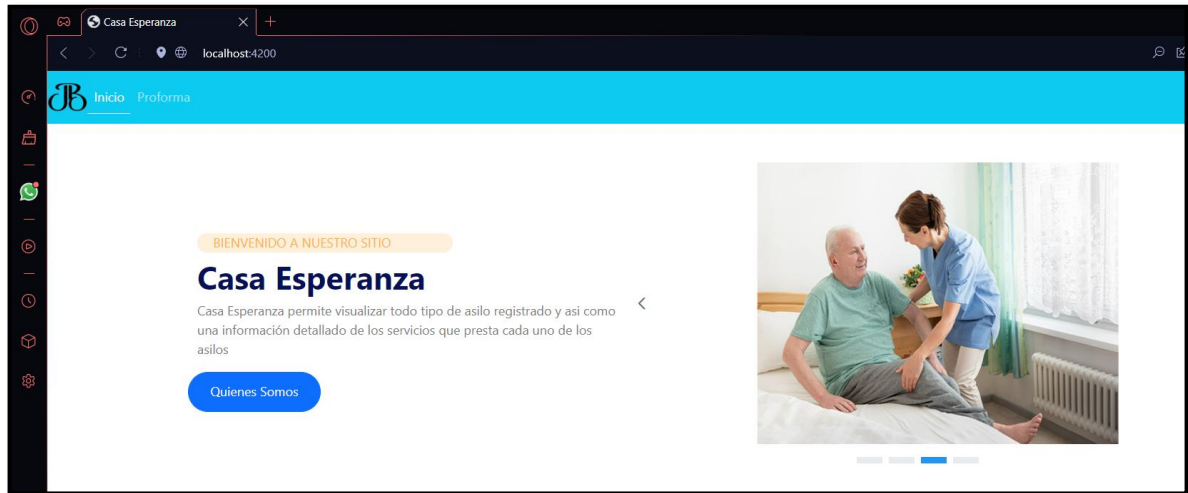


Fig. 41: Resultado de la prueba en el navegador Opera.

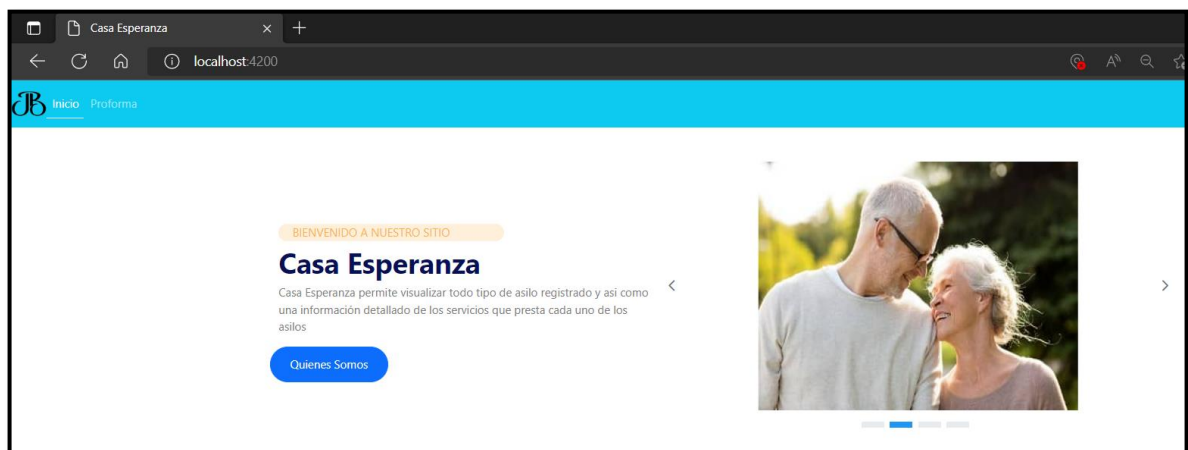


Fig. 42: Resultado de la prueba en el navegador *Microsoft Edge*.

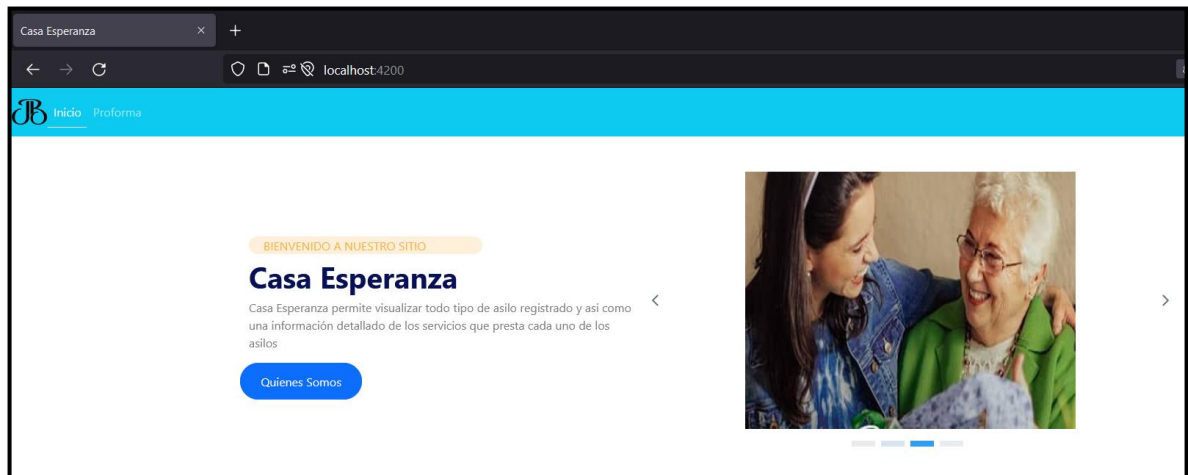


Fig. 43: Resultado de la prueba en el navegador *Firefox*.

Ejecución de pruebas de aceptación y resultados

A continuación, se muestran las 10 pruebas de aceptación que van desde la **TABLA XXII** hasta la **TABLA XXXII**. Por otra parte, cada prueba detalla el procedimiento que el usuario ha realizado para validar su resultado.

TABLA XXII: Prueba de aceptación 1 – visualizar página informativa.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA001	Identificador historia de Usuario: HU001
Nombre: visualización de la página informativa	
Descripción: El usuario ciudadano como también el administrador y asilo administrador puede visualizar la página informativa referente el sistema	
Pasos de ejecución: <ul style="list-style-type: none"> • Dirigirse a la URL del sistema <i>web</i> en el navegador. • Visualizar la información de los asilos • Dar una sugerencia al sistema si se desea 	
Resultado deseado: El sistema <i>web</i> permite que el usuario administrador y asilo administrador, de manera correcta a través de sus credenciales	
Evaluación de la prueba La prueba tiene la aprobación de un 100% y realiza lo esperado	

TABLA XXIII: Prueba de aceptación 2 – Ingresar al sistema web.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA002	Identificador historia de Usuario: HU002
Nombre: Ingreso al sistema <i>web</i>	
Descripción: El administrador puede ingresar al sistema mediante su despectiva credencial	
Pasos de ejecución: <ul style="list-style-type: none"> • Dirigirse a la URL del sistema <i>web</i> en el navegador. • Escribir las credenciales como el correo y la contraseña • Darle clic en Iniciar sesión 	
Resultado deseado: El sistema <i>web</i> permite que el usuario y asilo administradores, de manera correcta a través de sus credenciales	
Evaluación de la prueba La prueba tiene la aprobación de un 100% y realiza lo esperado	

TABLA XXIV: Prueba de aceptación 3 – Gestionar perfil.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA003	Identificador historia de Usuario: HU003
Nombre: Gestiona miento de perfil administrador	
Descripción: El usuario con perfil administrador en el sistema <i>web</i> tiene la posibilidad de editar la información personal.	

Pasos de ejecución: <ul style="list-style-type: none"> • Dirigirse a la URL del sistema web en el navegador. • Ingresar al inicio de sesión • Ingresar con las credenciales de administrador • Ingresar el perfil administrador para modificar sea el caso
Resultado deseado: El sistema web permite que el usuario administrador pueda editar modificar la información registrada y guardarla de manera fácil.
Evaluación de la prueba La prueba tiene la aprobación de un 100% y realiza lo esperado

TABLA XXV: Prueba de aceptación 4 – Gestionar asilos.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA004	Identificador historia de Usuario: HU004
Nombre: Gestionar asilos	
Descripción: El usuario con perfil administrador en el sistema web tiene la posibilidad gestionar los asilos los cuales se registran a través del formulario respectivo en donde el administrador tiene la posibilidad de aprobar o rechazar a los asilos	
Pasos de ejecución: <ul style="list-style-type: none"> • Dirigirse a la URL del sistema web en el navegador. • Ingresar al inicio de sesión. • Ingresar con las credenciales de administrador. • Ingresar el perfil administrador al módulo asilos. • Visualizar los asilos y ver si se los aprueba o rechaza. 	
Resultado deseado: El sistema web permite que el usuario administrador pueda aprobar o rechazar a los asilos que hayan realizado el formulario.	
Evaluación de la prueba La prueba tiene la aprobación de un 100% y realiza lo esperado	

TABLA XXVI: Prueba de aceptación 5 – Registrarse mediante un formulario.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA005	Identificador historia de Usuario: HU005
Nombre: Registrarse mediante un formulario.	
Descripción: El usuario con perfil asilo administrador en el sistema <i>web</i> tiene la posibilidad registrarse mediante un formulario el cual permitirá crearse una cuenta en el sistema	
Pasos de ejecución: <ul style="list-style-type: none"> • Dirigirse a la URL del sistema <i>web</i> en el navegador. • Ingresar a registrarse. • Ingresar todos los datos que se le solicita • Dar le al botón de registrarse. 	
Resultado deseado: El sistema <i>web</i> permite que el usuario asilo administrador pueda registrarse al sistema y forma parte y obtener los beneficios.	
Evaluación de la prueba La prueba tiene la aprobación de un 100% y realiza lo esperado	

TABLA XXVII: Prueba de aceptación 6 – Ingresar al sistema *web*.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA006	Identificador historia de Usuario: HU006
Nombre: Ingreso al sistema <i>web</i> del asilo administrador	
Descripción: El administrador puede ingresar al sistema mediante su despectiva credencial.	
Pasos de ejecución: <ul style="list-style-type: none"> • Dirigirse a la URL del sistema <i>web</i> en el navegador. • Escribir las credenciales como el correo y la contraseña • Darle clic en Iniciar sesión 	

Resultado deseado: El sistema <i>web</i> permite que el asilo administradores, ingrese al sistema de manera correcta a través de sus credenciales y puede observar los módulos que brinda el sistema al asilo.
Evaluación de la prueba La prueba tiene la aprobación de un 100% y realiza lo esperado

TABLA XXVIII: Prueba de aceptación 7 – Gestionar perfil.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA007	Identificador historia de Usuario: HU007
Nombre: Gestiona miento de perfil asilo administrador	
Descripción: El usuario con perfil asilo administrador en el sistema <i>web</i> tiene la posibilidad de editar la información personal.	
Pasos de ejecución: <ul style="list-style-type: none"> • Dirigirse a la URL del sistema <i>web</i> en el navegador. • Ingresar al inicio de sesión • Ingresar con las credenciales de administrador • Ingresar el perfil asilo administrador para modificar sea el caso 	
Resultado deseado: El sistema <i>web</i> permite que el usuario asilo administrador pueda editar modificar la información registrada y guardarla de manera fácil.	
Evaluación de la prueba La prueba tiene la aprobación de un 100% y realiza lo esperado	

TABLA XXIX: Prueba de aceptación 8– Enviar solicitud de aprobación.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA008	Identificador historia de Usuario: HU008
Nombre: Enviar solicitud de aprobación	

Descripción: El usuario con perfil asilo administrador en el sistema <i>web</i> tiene la posibilidad enviar una solicitud de afiliación mediante un formulario.
Pasos de ejecución: <ul style="list-style-type: none"> • Dirigirse a la URL del sistema <i>web</i> en el navegador. • Ingresar al inicio de sesión • Ingresar con las credenciales de administrador • Ingresar al módulo afiliación • Ingresar la información del asilo incluida la foto y documento para validación • Dar guardar y esperar la respuesta
Resultado deseado: El sistema <i>web</i> permite que el usuario asilo administrador enviar una solicitud para la respectiva afiliación con el fin de ser aprobado poder llevar la información o servicios adicionales por parte del asilo
Evaluación de la prueba La prueba tiene la aprobación de un 100% y realiza lo esperado

TABLA XXX: Prueba de aceptación 9 – Registrar información del asilo.

PRUEBA DE ACEPTACIÓN	
Identificador (ID): PA009	Identificador historia de Usuario: HU009
Nombre: Registrar información del asilo.	
Descripción: El usuario con perfil asilo administrador en el sistema <i>web</i> tiene la posibilidad poder registrar la información del asilo en caso de ser aprobado, si es rechazado deberá rectificar la información	
Pasos de ejecución: <ul style="list-style-type: none"> • Dirigirse a la URL del sistema <i>web</i> en el navegador. • Ingresar al inicio de sesión • Ingresar con las credenciales de administrador • Verificar si es aprobado o rechazado el asilo • Si es aprobado debe ingresar la información del asilo adicional • Si es rechazado deberá rectificar la información y enviarla otra vez y esperar 	

ser aprobado
<p>Resultado deseado:</p> <p>El sistema <i>web</i> permite que el usuario asilo administrador pueda registrar la información de todos los servicios.</p>
<p>Evaluación de la prueba</p> <p>La prueba tiene la aprobación de un 100% y realiza lo esperado</p>

Despliegue del sistema de *web*

Terminada la etapa de codificación como también pruebas llega el momento de poner a producción al sistema *web* como se observar en la **Fig. 44** y la **Fig. 45** son los comandos que ayudan al despliegue del sistema.

```
C:\Users\Lenovo\Documents\GitHub\proyecto_final>ng build
✓ Browser application bundle generation complete.
✓ Copying assets complete.
✖ Generating index html...2 rules skipped due to selector errors:
  legend+* -> Cannot read properties of undefined (reading 'type')
  legend+* -> Cannot read properties of undefined (reading 'type')
✓ Index html generation complete.
```

Initial Chunk Files	Names	Raw Size	Estimated Transfer Size
main.e600af26465cf6e5.js	main	4.62 MB	1.15 MB
styles.fbbae0c49176ccf8.css	styles	560.42 kB	44.12 kB
scripts.9f64bae8f755a019.js	scripts	165.46 kB	49.14 kB
polyfills.ae0838f66a606a39.js	polyfills	33.69 kB	10.83 kB
runtime.e3b2d68ea2c7a70a.js	runtime	1.94 kB	967 bytes
Initial Total		5.37 MB	1.25 MB

```
Build at: 2022-07-13T03:57:50.943Z - Hash: 735bba27492325e5 - Time: 30211ms
```

Fig. 44: ejecución del comando *build*.

```
C:\Users\Lenovo\Documents\GitHub\proyecto_final>firebase deploy

=== Deploying to 'login-ng-bdfe8'...

i deploying storage, firestore, hosting
+ storage: released rules storage.rules to firebase.storage
+ firestore: released rules firestore.rules to cloud.firestore
i hosting[login-ng-bdfe8]: finalizing version...
+ hosting[login-ng-bdfe8]: version finalized
i hosting[login-ng-bdfe8]: releasing new version...
+ hosting[login-ng-bdfe8]: release complete

+ Deploy complete!
```

Fig. 45:reestructurar configuración.

ANEXO III

A continuación, para acceder al enlace del Manual de Usuario se debe ingresar a la siguiente URL:

<https://www.youtube.com/watch?v=tRFaw7VKTEM>

en el que se menciona de forma clara y detallada cada una de las funcionalidades del sistema *web*, así como los perfiles que intervienen en la misma.

ANEXO IV

A continuación, se procede a definir las credenciales de acceso para el sistema *web*, así como el enlace al repositorio en GitHub en donde se encuentra el código fuente y en el apartado de README los pasos para realizar la instalación de forma local.

Credenciales de acceso para el sistema *web*

Para acceder al sistema *web* en producción, ingresar a la siguiente URL:

<https://casaesperanzaweb.web.app/>

Credenciales para el perfil administrador:

- **Correo del usuario:** p@gmail.com
- **Contraseña:** 1234567

Credenciales para el miembro administrador asilo:

- **Correo del usuario:** monicaalexandra@sanjose.com
- **Contraseña:** 1234567

Repositorio del código fuente del sistema *web*

El código fuente de todo el proyecto, se encuentra alojado en el repositorio GitHub, el cual se puede acceder a través de la siguiente URL:

https://github.com/JBorja25/AsiloTesis_Jose_Borja.git