

# **ESCUELA POLITÉCNICA NACIONAL**

## **FACULTAD DE INGENIERÍA DE SISTEMAS**

**DESARROLLO DE UNA APLICACIÓN WEB PARA LA GESTIÓN Y ADMINISTRACIÓN DE LAS MATRICULAS DE LOS ESTUDIANTES DEL CENTRO DE EDUCACIÓN INICIAL “SHINY KIDS”**

**COMPONENTE: MÓDULO DE REGISTRO DE MATRÍCULAS DEL CENTRO EDUCATIVO “SHINY KIDS”**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN CIENCIAS DE LA COMPUTACIÓN**

**EDISON ROMEO ALMEIDA CASTRO**

edison.almeida@epn.edu.ec

**DIRECTOR: VICTOR VICENTE VELEPUCHA BONETT**

victor.velepucha@epn.edu.ec

**DMQ, septiembre 2024**

## **CERTIFICACIÓN**

Yo, EDISON ROMEO ALMEIDA CASTRO 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.

---

**(EDISON ROMEO ALMEIDA CASTRO)**

Certifico que el presente trabajo de integración curricular fue desarrollado por EDISON ROMEO ALMEIDA CASTRO, bajo mi supervisión.

---

**(ING. VICTOR VICENTE VELEPUCHA BONETT)**

**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.

ESTUDIANTE: EDISON ROMEO ALMEIDA CASTRO

DIRECTOR: ING. VICTOR VICENTE VELEPUCHA BONETT

## DEDICATORIA

Dedico este trabajo de tesis a mis padres, quienes han sido una fuente constante de apoyo a lo largo de este desafiante viaje en mis estudios universitarios. Gracias a su paciencia y comprensión, siempre han estado a mi lado, respaldándome en cada etapa que he superado. Agradezco profundamente su total apoyo.

A mi hermano, cuyo papel ha sido esencial como fuente de inspiración y alegría. Este logro es un éxito compartido, ya que él ha estado a mi lado en mis momentos de alegría y triunfos. Ahora, me complace compartir esta felicidad con él.

A mi compañera de vida, quien ha sido una fuente incondicional de apoyo, ayudándome a mantenerme firme en la persecución de mis metas. Su presencia en mi vida ha sido fundamental, y este logro también le pertenece. Estoy agradecido por su existencia y por ser mi equipo en todo momento.

A mi papico, quien desde el cielo sigue apoyándome. Su apoyo fue fundamental en mi camino para alcanzar mi sueño de convertirme en profesional.

A toda mi familia, que siempre ha estado a mi lado, brindándome su apoyo incondicional, especialmente a mi mami Lola y a mi tío Wilson. Les agradezco de corazón.

## **AGRADECIMIENTOS**

Quiero expresar mi profundo agradecimiento al ingeniero Víctor Velepucha, mi director de tesis, por su invaluable guía y apoyo constante a lo largo de este proyecto. A pesar de haberme recibido en un momento inoportuno, se convirtió en un pilar esencial gracias a su paciencia y dedicación, lo que permitió el desarrollo de este trabajo en el tiempo limitado disponible.

Asimismo, agradezco a la Escuela Politécnica Nacional por proporcionarme el conocimiento y las herramientas necesarias para convertirme en un profesional competente. Mi gratitud se extiende a la Facultad de Sistemas, a sus profesores y compañeros, quienes jugaron un papel crucial en la culminación de mi formación académica.

Mi reconocimiento también va al Centro de Educación Inicial “Shiny Kids” y a su directora, Tnlga. Rosario Tarupi, por ofrecerme la asistencia necesaria para la realización de este proyecto, facilitando la documentación y el tiempo requerido para las entrevistas, lo que fue fundamental para cumplir con los objetivos de mi tesis.

Finalmente, expreso mi agradecimiento a mi familia y amigos por su constante apoyo y aliento. Su presencia y respaldo incondicional han sido esenciales para la culminación de este proyecto.

# Índice general

<b>1. INTRODUCCIÓN</b>	<b>1</b>
1.1. Objetivo general . . . . .	2
1.2. Objetivos específicos . . . . .	2
1.3. Alcance . . . . .	2
1.4. Marco teórico . . . . .	4
<b>2. METODOLOGÍA</b>	<b>15</b>
2.1. Implementación SCRUM . . . . .	15
2.1.1. Configuración del proyecto . . . . .	16
2.2. Inicialización del proyecto . . . . .	21
2.2.1. Configuración de JIRA . . . . .	21
2.2.2. Diseño de interfaces . . . . .	22
2.3. Implementación de sprints . . . . .	23
2.3.1. Sprint 0 . . . . .	23
2.3.2. Sprint 1 . . . . .	27
2.3.3. Sprint 2 . . . . .	30
2.3.4. Sprint 3 . . . . .	31
2.3.5. Sprint 4 . . . . .	33
2.3.6. Sprint 5 . . . . .	36
2.4. Pruebas de funcionalidad . . . . .	38
2.4.1. Resultados de las pruebas de funcionalidad . . . . .	38
<b>3. CONCLUSIONES Y RECOMENDACIONES</b>	<b>47</b>
3.1. Conclusiones . . . . .	47
3.2. Recomendaciones . . . . .	48
<b>4. REFERENCIAS BIBLIOGRÁFICAS</b>	<b>50</b>



# Índice de figuras

2.1. Configuración de JIRA . . . . .	21
2.2. Planificación de tareas en JIRA . . . . .	22
2.3. Interfaz login . . . . .	23
2.4. Interfaz paralelos . . . . .	24
2.5. Interfaz estudiantes . . . . .	24
2.6. Interfaz matrículas . . . . .	25
2.7. Arquitectura aplicación web . . . . .	26
2.8. Colección en Firestore . . . . .	28
2.9. Archivo firebase.json . . . . .	29
2.10.Sprint 1 Pruebas de funcionalidad de guardar estudiante . . . . .	40
2.11.Sprint 2 Pruebas de funcionalidad de editar estudiante . . . . .	41
2.12.Sprint 3 Pruebas de funcionalidad de listar estudiantes . . . . .	42
2.13.Sprint 4 Pruebas de funcionalidad de carga de archivos . . . . .	43
2.14.Sprint 5 Pruebas de funcionalidad de autenticación . . . . .	45
2.15.Sprint 5 Pruebas de funcionalidad de campos requeridos . . . . .	45
2.16.Sprint 5 Pruebas de funcionalidad de imprimir . . . . .	45

# Índice de Tablas

2.1. Roles del equipo . . . . .	16
2.2. Historias de usuario . . . . .	17
2.3. Valoración de Prioridad . . . . .	18
2.4. Valoración de Complejidad . . . . .	19
2.5. Product Backlog . . . . .	20
2.6. Distribución de tareas por Sprints . . . . .	21
2.7. Herramientas para el desarrollo web . . . . .	27
2.8. Sprint 1 Backlog . . . . .	28
2.9. Sprint 2 Backlog . . . . .	30
2.10.Sprint 3 Backlog . . . . .	32
2.11.Sprint 4 Backlog . . . . .	34
2.12.Sprint 5 Backlog . . . . .	36
2.13.Pruebas de Funcionalidad Sprint 1 . . . . .	39
2.14.Pruebas de Funcionalidad Sprint 2 . . . . .	40
2.15.Pruebas de Funcionalidad Sprint 3 . . . . .	41
2.16.Pruebas de funcionalidad Sprint 4 . . . . .	42
2.17.Pruebas de Funcionalidad Sprint 5 . . . . .	44

## RESUMEN

El presente trabajo de integración curricular se enfoca en el desarrollo de una aplicación web para la gestión de matrículas del centro de educación inicial "Shiny Kids", utilizando la metodología SCRUM. Esta metodología garantiza la efectividad y el cumplimiento de los requisitos del sistema, facilitando una retroalimentación continua durante todo el desarrollo. La herramienta JIRA fue utilizada para la planificación, seguimiento y gestión del proyecto, permitiendo organizar al equipo, rastrear el progreso y asegurar una colaboración eficiente. El desarrollo de la aplicación se llevó a cabo en cinco sprints bien definidos. En el sprint 0, se configuró el entorno de desarrollo y se definió la arquitectura del aplicativo. En el sprint 1, se definieron historias de usuario para el almacenamiento de datos en Firebase. El sprint 2 se centró en el desarrollo del backend con NodeJS, implementando funcionalidades clave como la actualización de datos de los estudiantes matriculados. En el sprint 3, se extrajeron y presentaron de manera estructurada los datos almacenados en Firebase, mostrando a los estudiantes según los cursos inscritos. Finalmente, en el sprint 4, se abordó el manejo de imágenes y archivos PDF relacionados con los estudiantes. Los resultados demuestran que la aplicación web fue planificada y ejecutada de manera efectiva, cumpliendo con todos los elementos del backlog y las historias de usuario establecidas. Esto evidencia la eficacia de SCRUM en el desarrollo de sistemas de gestión educativa y la importancia de una herramienta robusta como JIRA en la organización y ejecución de proyectos complejos.

**Palabras Claves** - Aplicación web, Scrum, Framework, Node.JS, Firebase, Angular

## ABSTRACT

This curricular integration project focuses on the development of a web application for managing enrollments at the "Shiny Kids.early education center, using the SCRUM methodology. This methodology ensures the effectiveness and fulfillment of the system's requirements, facilitating continuous feedback throughout the development process. The JIRA tool was used for project planning, tracking, and management, allowing the team to organize, track progress, and ensure efficient collaboration. The application development was carried out in five well-defined sprints. In sprint 0, the development environment was set up and the application's architecture was defined. In sprint 1, user stories were defined for data storage in Firebase. Sprint 2 focused on backend development with NodeJS, implementing key functionalities such as updating data of enrolled students. In sprint 3, the stored data in Firebase was extracted and presented in a structured manner, displaying students according to their enrolled courses. Finally, in sprint 4, the handling of images and PDF files related to students was addressed. The results demonstrate that the web application was effectively planned and executed, meeting all the elements of the backlog and established user stories. This evidences the effectiveness of SCRUM in the development of educational management systems and the importance of a robust tool like JIRA in the organization and execution of complex projects.

**Keywords** - Web Application, Scrum, Framework, Node.JS, Firebase, Angular

# Capítulo 1

## INTRODUCCIÓN

En la actualidad, la tecnología desempeña un papel crucial en el desarrollo de diversos sistemas web, los cuales son fundamentales para facilitar una amplia gama de funcionalidades destinadas a mejorar procesos en diferentes sectores. Por lo tanto, la usabilidad de estos sistemas es esencial para asegurar su correcto funcionamiento y para garantizar que cumplan efectivamente con el objetivo para el cual fue diseñado e implementado [1].

En ese sentido, el Centro de Educación Inicial "Shiny Kids", ubicado en la ciudad de Tulcán, es un establecimiento privado dedicado a la educación desde el año 2011 para niños y niñas de 1 a 4 años, utilizando técnicas de aprendizaje Montessori. A pesar de su larga trayectoria educativa, el centro carece de un sistema de matriculación que pueda mejorar significativamente el proceso. Debido a que las matrículas se realizan de manera manual, registrando la información en hojas de papel membretadas, lo cual dificulta la búsqueda de datos y la gestión documental, a pesar de que en la actualidad es crucial gestionar los registros de manera ordenada y simplificada mediante sistemas digitales. Por esta razón, el centro educativo ha reconocido la necesidad de implementar un nuevo sistema de matriculación web, este nuevo sistema permitirá un registro más eficiente de cada estudiante matriculado, facilitando a los administradores el acceso rápido a la base de datos para revisar la información cuando sea necesario.

Teniendo en cuenta todos estos aspectos, se planteó la implementación de una aplicación web para el Centro de Educación Inicial "Shiny Kids", con el objetivo de agilizar y simplificar el proceso de matriculación de los estudiantes. Esta aplicación no solo simplificará la interacción del personal administrativo con el sistema, sino que también mejorará la experiencia del usuario final mediante una interfaz intuitiva y amigable, además, se permite almacenar la información en una base de datos en la nube utilizando Firebase, una

plataforma de base de datos no relacional que permite trabajar en tiempo real. Esto facilitará mejorar las interacciones con los clientes, proporcionando a los administradores acceso instantáneo a la información de los estudiantes [2].

Por lo tanto, este proyecto proporcionará a los administradores un mayor control y facilitará el ingreso de datos de cada estudiante del Centro Educativo. A través de una aplicación web con una interfaz amigable, será posible visualizar, actualizar, eliminar e ingresar datos de manera eficiente. Esto evitará la pérdida de información, reducirá los tiempos dedicados al proceso de matriculación y mejorará búsqueda de datos de los estudiantes, además, la implementación del módulo de matrículas garantizará un manejo más efectivo de la información de los estudiantes en el Centro Educativo.

### **1.1. Objetivo general**

Desarrollar la aplicación web para el proceso de registro de la matrícula de los estudiantes de Centro Educativo permitiendo la gestión del proceso de Matriculas.

### **1.2. Objetivos específicos**

1. Definir los requisitos del módulo de matrículas para la correcta implementación de la aplicación web.
2. Definir la arquitectura del sistema a implementar que mejor se adapte para la realización de la aplicación web.
3. Realizar la implementación de los prototipos con base a los requisitos presentados de la aplicación web.
4. Implementar la aplicación web utilizando metodologías ágiles como Scrum para la planificación, diseño, implementación y pruebas de funcionalidad de la aplicación web.
5. Realizar las pruebas de funcionalidad y despliegue de la aplicación web para observar el correcto funcionamiento.

### **1.3. Alcance**

Se desarrolló una aplicación web utilizando la metodología SCRUM para implementar este sistema. SCRUM se caracteriza por su enfoque iterativo e incremental, lo cual fue

fundamental para asegurar la aplicación efectiva de los requisitos del sistema y obtener retroalimentación continua durante todo el desarrollo del proyecto. Esta metodología guió las fases de Diseño, Implementación, Pruebas y Evaluación de manera organizada y progresiva, permitiendo ajustes y mejoras constantes basadas en la retroalimentación continua a lo largo del desarrollo del proyecto.

Para la fase de Diseño de la aplicación web del Centro Educativo, se comenzó con la creación de mockups en Figma. Estos mockups proporcionaron una visualización clara y detallada de cómo funcionaría el módulo de matrículas en la aplicación web del Centro Educativo, dirigida a los administradores. Esta etapa nos permitió identificar y documentar los requisitos funcionales del sistema, los cuales definieron las características específicas del módulo de matrículas, incluyendo el registro de estudiantes y la gestión de matrículas, así como la generación de reportes de estudiantes matriculados. Además, se llevó a cabo la definición de la arquitectura de la aplicación web. Durante este proceso, se seleccionó el marco arquitectónico más adecuado para satisfacer las necesidades del Centro Educativo, asegurando criterios de escalabilidad, mantenibilidad y facilidad de integración con otros módulos del sistema, que serán implementados en futuras etapas del desarrollo del aplicativo web.

En la fase de Implementación se elaboró la aplicación web utilizando frameworks actuales y lenguajes de programación adecuados, que fueran los más utilizados por los desarrolladores en ese momento. Para el desarrollo del Frontend se utilizó Angular en su versión 18.0.5, la más actualizada. Esta elección permitió crear una interfaz de usuario intuitiva y responsive, mejorando la experiencia de los administradores del sistema. En cuanto al desarrollo del Backend se usó tecnologías que soportaran la lógica del negocio, la gestión de datos y la integración con servicios externos, además se optó por Node.JS en su versión 20.15.0, lo cual facilitó la integración a través de Express versión 4.19.2. Esto permitió implementar toda la lógica del negocio de manera eficiente y asegurando la correcta operatividad del sistema.

La base de datos de la aplicación web se alojó en la nube, aprovechando la plataforma de Google Cloud y utilizar la firebase versión 10.12.2 donde se alojará las colecciones en la Firestore y los archivos a través del Storage que nos ofrece Firebase para asegurar la escalabilidad y la disponibilidad del sistema, permitiendo manejar un aumento en la demanda sin afectar el rendimiento.

En la fase de Pruebas de funcionalidad, se llevó a cabo pruebas fundamentales para asegurar la calidad y funcionalidad del módulo de matriculación del Centro Educativo. Estas

pruebas permitieron verificar que la aplicación web funcionara correctamente y cumpliera con las pruebas de usuario especificadas. Durante esta fase se realizó una revisión en base a la retroalimentación de los usuarios para realizar los ajustes y mejoras necesarias, asegurando una entrega satisfactoria y alineada con las expectativas del Centro Educativo.

En ese contexto, el alcance del proyecto incluye un desarrollo estructurado y meticuloso basado en la metodología SCRUM. Esto garantizó que cada fase del proyecto, desde el diseño hasta la implementación y las pruebas, fuera gestionada de manera eficiente, para la entrega de un módulo de matrículas en la aplicación web del Centro Educativo que cumple con las necesidades del proceso de matriculación de los estudiantes.

## **1.4. Marco teórico**

### **Figma**

Figma es una plataforma de diseño de interfaces que destaca por su capacidad para facilitar la colaboración en tiempo real entre equipos de diseño, tanto a través de su versión en línea como de su aplicación de escritorio. Actualmente, se considera la herramienta líder en el ámbito del diseño de interfaces, superando en popularidad y valoración a otras soluciones bien conocidas en el mercado, tales como Adobe XD, InVision, Sketch y Justinmind. Esta preferencia se debe a sus características avanzadas que permiten una integración fluida y eficiente en los procesos de trabajo de los diseñadores, haciendo que la creación, prototipado y colaboración sean más ágiles y efectivas [3].

### **Interfaz de Figma**

La interfaz de FIGMA según, se organiza en un menú superior con varios accesos directos y tres secciones principales.

- En el menú superior, el primer icono con forma de hamburguesa despliega todas las opciones del programa, seguido de herramientas como el cursor, marcos, formas, pluma, texto y comentarios.
- En el centro de este menú se muestra el nombre del documento, y a la derecha se encuentran los accesos directos para compartir y ejecutar el prototipo.
- En el lado izquierdo se ubican las capas (similares a las de Photoshop) y los recursos (como iconos e imágenes).
- La sección central es el área de trabajo, donde se desarrolla el diseño del prototipo.

- En la columna derecha se presentan las opciones específicas del elemento seleccionado, entonces si se selecciona un bloque de texto, se pueden ajustar la fuente, el tamaño y otros atributos [3].

## **Firestore**

Firestore es un conjunto de herramientas proporcionado por Google que facilita la creación de infraestructura para aplicaciones, la ampliación de la base de usuarios, el almacenamiento de datos en formato JSON, la mejora de la calidad de las aplicaciones y el crecimiento del negocio. La aplicación Shiny Kids emplea Firestore para almacenar información en colecciones, las cuales se utilizan para realizar consultas y extraer datos que se muestran en la interfaz de la aplicación. Además, las imágenes subidas por los usuarios se guardan en carpetas utilizando el espacio de Cloud Storage que ofrece Firestore[4].

Del mismo podemos indicar que Firestore es:

Una plataforma de computación en la nube creada por Google, diseñada para desarrollar y facilitar la creación rápida de aplicaciones móviles de alta calidad. Su objetivo principal es permitir el aumento de la base de usuarios y la generación de ingresos. Esta plataforma, disponible para iOS, Android y la web, se adapta a las necesidades de cualquier desarrollador mediante una variedad de funciones que se pueden combinar y ajustar a medida [5].

### **Principales características de Firestore:**

Firestore describe las siguientes características:

- **Desarrollo:** Ofrece numerosas herramientas y funciones para mejorar y crear aplicaciones. Durante la fase de desarrollo, destacan la detección de errores y el testeo. Una de sus principales ventajas es la capacidad de almacenar y gestionar remotamente los datos de los usuarios.
- **Analítica:** Proporciona un control exhaustivo del rendimiento de la aplicación de manera gratuita. Los análisis se basan en datos reales, lo que permite a los desarrolladores tomar decisiones informadas.
- **Capacidad de crecimiento:** Gracias a sus modelos de bases de datos, facilita la gestión de usuarios. Además, permite enviar notificaciones para atraer a más usuarios.
- **Monetización:** En el futuro, se planea integrar AdMob para añadir anuncios y generar ingresos una vez que la aplicación esté lanzada.

- **Rapidez:** Las funcionalidades de Firebase se implementan fácilmente en Android Studio mediante una API intuitiva contenida en un único SDK, lo que ahorra tiempo en el diseño de la infraestructura y permite centrarse en los usuarios.
- **Agilidad:** Permite la creación de aplicaciones multiplataforma con una API integrada en SDKs para iOS, Android y JavaScript, facilitando la gestión de diferentes aplicaciones desde una sola plataforma [6].

### **Servicios de Firebase:**

- **Base de datos en tiempo real:** Firebase Realtime Database es una base de datos en la nube que almacena datos en formato JSON y los sincroniza en tiempo real con todos los clientes conectados. Esta base de datos remota NoSQL permite guardar y sincronizar datos de usuarios, publicaciones, comentarios, rutinas y dietas, manteniendo los datos accesibles incluso sin conexión.
- **Autenticación:** Ofrece un sistema de autenticación simplificado que admite proveedores como Twitter, Facebook, teléfono, correo/contraseña y Google. Esto facilita la gestión de usuarios y mejora la calidad del servicio, garantizando la seguridad de los datos almacenados en la nube y proporcionando una experiencia uniforme en todos los dispositivos Android.
- **Almacenamiento:** Firebase Storage es un servicio de almacenamiento de objetos seguro y escalable, adecuado para guardar fotos, videos y otros archivos de usuarios. Es especialmente útil para almacenar imágenes de publicaciones y documentos generados automáticamente por la aplicación.
- **Funciones en la nube:** Cloud Functions para Firebase permite ejecutar automáticamente código backend en respuesta a eventos de Firebase. Estas funciones, creadas en NodeJS y desplegadas con una URL para llamadas AJAX, se utilizan para enviar notificaciones o correos electrónicos sobre nuevos eventos, como la creación de publicaciones por usuarios registrados [6].

### **Aplicación Web**

Es un tipo de software accesible a través de un navegador web y que se ejecuta en un servidor remoto, eliminando la necesidad de instalarlo en el dispositivo del usuario. Todas las funciones y datos se almacenan en el servidor, proporcionando a los usuarios la flexibilidad de acceder a las aplicaciones web desde diversos dispositivos y sistemas operativos.

Además, se indica que estas aplicaciones pueden ofrecer una amplia variedad de funcionalidades, que van desde sitios web interactivos simples hasta aplicaciones comerciales complejas. Entre los ejemplos comunes se incluyen plataformas de comercio electrónico, aplicaciones de productividad, herramientas de colaboración en línea y sistemas de gestión de contenidos. Utilizan la infraestructura de la nube para el procesamiento y almacenamiento de datos, lo que brinda mayor flexibilidad y escalabilidad [7].

### **Estructura de una Aplicación Web:**

Las aplicaciones web, tienen usualmente tres capas principales: Frontend que contiene la interfaz gráfica, backend con servicios reutilizables y la base de datos.

- **Frontend:** es la capa inicial de cualquier aplicación informática y es la utilizada directamente por los usuarios y clientes. En las aplicaciones web, esta capa se ejecuta mediante un navegador de internet, mientras que en las aplicaciones móviles se ejecuta en un dispositivo móvil.
- **Backend:** esta capa se encarga de recibir todas las solicitudes del usuario provenientes de la interfaz gráfica, procesarlas y ejecutarlas, devolviendo finalmente los datos a la primera capa. El servidor es responsable de establecer una conexión con la base de datos, permitiendo ingresar, actualizar o consultar la información solicitada por el usuario.
- **Base de datos:** constituye la capa fundamental de cualquier aplicación informática, ya que almacena toda la información del sistema [8].

### **Tipo de aplicación web**

- **Aplicaciones web estáticas** Son aplicaciones que muestran una cantidad limitada de información y están diseñadas para no agregar nuevos contenidos con frecuencia. Generalmente desarrolladas con HTML y CSS, pueden incluir elementos multimedia como videos, banners y GIFs. Actualizar el contenido puede resultar complicado, ya que implica descargar el código, editarlo y volver a cargarlo, lo cual no es práctico. Este tipo de aplicación es adecuado para proyectos que no requieren actualizaciones frecuentes o para quienes están comenzando en el ámbito digital [9].

### **Aplicaciones web Dinámicas**

Son considerablemente más complejas que las estáticas, basándose en el uso de bases de datos para cargar información. Los contenidos se actualizan en tiempo real cada

vez que el usuario accede a la aplicación. Suelen contar con un panel de administración o CMS desde el cual se gestionan y publican noticias, publicaciones, imágenes, videos, etc. Se emplean diversos lenguajes de programación como PHP y ASP, que permiten una estructuración eficaz del contenido y facilitan la actualización de los mismos. Además, estas aplicaciones pueden incluir funcionalidades como foros y bases de datos, y permiten ajustar el diseño de manera flexible [10].

### **Frameworks**

Un framework es un entorno que proporciona directrices que facilitan el desarrollo, ofreciendo herramientas poderosas que los desarrolladores pueden utilizar en lugar de crearlas desde cero, lo cual ahorra considerable tiempo y esfuerzo. Por lo tanto, es un entorno que simplifica la programación de software mediante un conjunto de directrices que definen su estructura [11].

Al seleccionar un framework para el desarrollo de aplicaciones híbridas, es crucial considerar características clave como una comunidad activa para soporte continuo, mantenimiento y actualizaciones.

### **Angular**

Angular, en particular, se utiliza para desarrollar sistemas web con la capacidad, eficiencia y seguridad necesarias para satisfacer los requisitos tanto de empresas como de usuarios. Desarrollado por Google como un framework de código abierto, Angular se especializa en la creación de aplicaciones de una sola página (SPA, por sus siglas en inglés, Single Page Application), permitiendo una rápida separación entre el frontend y el backend del sistema mediante la implementación de la arquitectura MVC, Modelo-Vista-Controlador [12].

De la misma manera, Angular es ampliamente reconocido como un robusto marco de JavaScript para el desarrollo de aplicaciones web avanzada y al ser gratuito y de código abierto, Angular es ideal para implementar servicios en línea, formularios interactivos y procesos con respuestas inmediatas desde el servidor. Combinado con recursos como servidores, equipos y conexiones, representa una sólida elección para el desarrollo de sistemas web empresariales [12].

### **Express.js**

Express.js es un framework minimalista y flexible para Node.js, diseñado para construir aplicaciones web y APIs de una manera más rápida y sencilla. Desarrollado por TJ Holowaychuk en 2010, Express.js se ha convertido en uno de los frameworks más populares a nivel mundial en el ecosistema de Node.js debido a que es más simple para su uso. Este

ofrece una serie de características potentes, como enrutamiento robusto, middleware configurable y una fácil integración con bases de datos y otros servicios externos como es las plataformas de Google. Express.js también permite a los programadores crear aplicaciones web escalables y mantenibles con una estructura clara y concisa, lo que acelera significativamente el proceso de desarrollo y mejora la eficiencia en la gestión de proyectos complejos [13].

**Node.js:** Node.js, es un entorno de ejecución de JavaScript de código abierto y multi-plataforma que permite a los desarrolladores crear aplicaciones web escalables y de alto rendimiento. Se basa en el motor V8 de Google Chrome, lo que le otorga una velocidad y eficiencia excepcionales. Node.js utiliza un modelo de ejecución de un solo hilo con E/S asíncrona, lo que lo hace ideal para manejar múltiples conexiones simultáneas y aplicaciones en tiempo real [14].

#### **Lenguajes de programación:**

Los lenguajes de programación son cruciales para enseñar ciertos conocimientos. Existen dos tipos principales de lenguajes; los textuales y los icónicos:

##### ■ **JavaScript:**

JavaScript es el lenguaje de programación que proporciona dinamismo e interactividad a las páginas web. Los navegadores interpretan su código directamente, sin necesidad de intermediarios. Es considerado uno de los tres lenguajes web fundamentales, junto con HTML, que define el contenido y la estructura, y CSS, que se encarga del diseño y la presentación del contenido. Actualmente, JavaScript es el lenguaje de programación más popular. Se desarrolló una versión llamada Node.js, que permite la ejecución del código JavaScript en el lado del servidor. Esto ha ampliado su uso tanto en navegadores como en servidores, creando una comunidad de desarrolladores muy amplia y robusta. En el ámbito del servidor, JavaScript compite directamente con lenguajes como PHP [15].

En ese contexto, JavaScript se establece como un lenguaje de programación principalmente utilizado para crear páginas web dinámicas. Este tipo de páginas se caracteriza por agregar efectos a los textos, hacer que aparezcan y desaparezcan, incluir animaciones, activar acciones al pulsar botones y mostrar ventanas con mensajes al usuario. Además, es un lenguaje interpretado, lo que significa que no requiere compilación para ejecutarse, dado que los programas escritos en JavaScript pueden probarse directamente en cualquier navegador sin necesidad de pasos adicionales [15].

Además, JavaScript se utiliza principalmente en aplicaciones web, pero su uso no se limita a este ámbito. En los últimos años, el lenguaje ha ampliado su alcance a servidores, dispositivos móviles y sistemas embebidos. Esta versatilidad ha llevado a que JavaScript se convierta en uno de los lenguajes más utilizados en el mundo [15].

## HTML

El lenguaje de marcado de hipertexto (HTML), que se basa en etiquetas, que son instrucciones sobre cómo mostrar el texto, y en atributos, que son parámetros que asignan valores a las etiquetas. HTML no define la apariencia de un documento, sino que permite a cada plataforma darle formato según sus capacidades y las del navegador, como el tamaño de la pantalla y las fuentes instaladas. Además, tiene dos ventajas clave que lo hacen esencial para el diseño web: su compatibilidad y su facilidad de aprendizaje debido al pequeño número de etiquetas que emplea. Los documentos HTML consisten en el texto del documento y las etiquetas, que pueden incluir atributos. En la práctica, esto se representa como: '`<etiqueta><atributo>texto </atributo></etiqueta>`'. La primera etiqueta inicia una orden y la última la cierra; no todas las etiquetas tienen un inicio y un final [16].

Además, HTML tiene las siguientes Características:

- **Hipertexto:** varios elementos en la pantalla, como textos o imágenes, están enlazados a otras informaciones, permitiendo acceder a ellas con un clic.
- **Gráficos:** permite la visualización simultánea de texto, imágenes e incluso sonidos.
- **Globalidad:** es accesible desde cualquier plataforma, utilizando cualquier navegador y desde cualquier lugar del mundo.
- **Publicación:** la información está distribuida en numerosos ordenadores que la almacenan y la ponen a disposición del usuario, siendo toda esta información pública.
- **Dinámica:** la información puede ser actualizada por quien la publicó sin necesidad de que el usuario actualice su dispositivo.
- **Independencia:** debido a la gran cantidad de fuentes disponibles, la información es independiente y libre [16].

## Css

“CSS es el lenguaje que proporciona a los navegadores las instrucciones necesarias para presentar una página web. Cada elemento de la página forma parte de un documento escrito en un lenguaje de marcado como HTML o XML” [17].

En similares condiciones, establece que CSS es fundamental para personalizar aplicaciones web, ya que su función principal es aplicar estilos a las etiquetas HTML que han sido previamente estructuradas. Mediante CSS, es posible definir el color, tamaño, posición y otras características de los elementos de una aplicación web [18].

Entre sus características principales se incluyen:

- Definir la etiqueta con una llave de apertura.
- Especificar el nombre del elemento que se desea modificar.
- Asignar el nuevo valor deseado.
- Finalizar con una llave de cierre.

Por lo tanto, CSS es crucial para los desarrolladores, ya que les ayuda a seguir buenas prácticas y a crear código personalizado, lo cual ahorra tiempo y dinero [18].

### **Metodologías Ágiles**

Las metodologías ágiles se destacan por su enfoque adaptativo en lugar de predictivo, priorizando a las personas sobre los procesos. En el ámbito del desarrollo de software, las tendencias innovadoras ofrecen la oportunidad de explorar diversas metodologías ágiles en busca de la eficacia y el éxito en este continuo desafío. En la industria, estas metodologías han sustituido gradualmente los enfoques tradicionales, permitiendo enfrentar de manera efectiva los desafíos en la gestión y entrega de proyectos [19].

### **XP o Extreme Programming**

Se le conoce como programación extrema esta metodología ha dado éxito debido a que se enfatiza en la satisfacción del cliente, muchas veces se ofrece todo el producto en una fecha muy lejana y este ofrece un proceso en el que se trabaja en el software que necesita a medida. Enfatiza en la importancia del trabajo en equipo, todos los miembros deben colaborar gerentes, clientes y desarrolladores, en si XP implementa en un entorno simple y eficaz lo cual permite que los equipos sean altamente productivos [20].

Tenemos diferentes roles en este tipo de metodología ágil como son:

- **Tracker:** Es el encargado de dar seguimiento y va proporcionando la retroalimentación del equipo.
- **Customer:** Es el cliente o usuario y es quien va escribiendo las historias de usuario y va asignado prioridad siempre centrada al valor del negocio.

- **Programmer:** Es el desarrollador más importante de equipo, programa el código del sistema y las pruebas unitarias.
- **Coach:** Este miembro es el responsable del proceso a nivel global, es una guía para los demás miembros del equipo.
- **Tester:** Es el encargado de escribir las pruebas funcionales y las va ejecutando regularmente.
- **Big Boss:** Es el encargado como medio de coordinación entre clientes y programadores.
- **Consultor:** Actúa como miembro externo del equipo y ayuda al equipo a resolver problemas de un tema en específico del cual el conoce.
- **Manager:** Va registrando todas las reuniones y extrae la información importante para mantener al equipo feliz y productivo.
- **Doomsayer:** Ayuda a que todos los miembros del equipo se empapen de los riesgos del proyecto [20].

### **DYNAMIC SYSTEMS DEVELOPMENT METHOD**

El DSDM Agile Project Framework es una metodología ágil que tiene un enfoque líder, probado y ágil que brinda el rigor junto a la flexibilidad y agilidad de las organizaciones. Esta metodología se puede utilizar de manera independiente o combinada es ideal para enfoques ágiles que se aborde todo el ciclo de vida del proyecto, muestra cómo se puede aplicar el ciclo de vida y abordar las necesidades del negocio. Los roles de esta metodología son agrupados de acuerdo con los intereses a continuación listamos los Roles:

- Business Ambassador
- Business Visionary
- Business Sponsor
- Business Advisor
- Technical Advisor
- Solution Tester
- Solution Developer

- Technical Coordinator
- Project Manager
- Team Leader
- Workshop Facilitator
- DSDM Coach
- Business Analyst

Se tiene 13 roles que son mucho más que las otras metodologías debido a que DSDM tiene un alcance mayor que las otras tecnologías, pero así también las personas miembros del equipo pueden tener más de un rol, pero siempre se debe cumplir todas las responsabilidades requeridas [20].

**SCRUM** Se trata de un marco de trabajo que permite a las personas enfrentar problemas complejos y adaptativos, al tiempo que entregan productos de alto valor de manera productiva y creativa. Este framework es conocido por su agilidad y su enfoque centrado en el cliente, estructurándose en roles, artefactos y eventos específicos [21].

Dentro del marco de trabajo de SCRUM, las historias de usuario juegan un papel fundamental durante la fase de desarrollo al facilitar la definición y especificación de los requisitos funcionales y no funcionales del sistema. Escritas en un lenguaje sencillo y accesible, estas historias describen las funcionalidades buscadas por el usuario final y son un componente esencial en la metodología SCRUM [22].

En ese sentido, se establece los componentes principales de SCRUM:

**Roles:**

- **Product Owner:** persona que representa las necesidades del cliente.
- **Scrum Master:** facilitador del proceso y guía al equipo en la adopción de Scrum.

**Artefactos:**

- **Product Backlog:** lista priorizada de requisitos del producto.
- **Sprint Backlog:** elementos seleccionados del Product Backlog para trabajar durante un Sprint.
- **Incremento:** resultado funcional obtenido al finalizar un Sprint.

## Eventos:

- **Sprint:** período fijo para desarrollar un incremento.
- **Reunión de Planificación del Sprint:** selección de elementos del Product Backlog para el Sprint.
- **Daily SCRUM:** reunión diaria para la sincronización del equipo.
- **Revisión del Sprint:** evaluación del incremento al finalizar el Sprint.
- **Retrospectiva del Sprint:** análisis del rendimiento del equipo y mejoras en los procesos [21].

**GitHub** GitHub es una plataforma de alojamiento de código y colaboración para desarrolladores de software que permite a los usuarios crear, compartir y trabajar juntos en proyectos de software de forma eficiente. GitHub funciona como un repositorio centralizado donde los desarrolladores pueden almacenar, gestionar y colaborar en el código fuente de sus proyectos. La plataforma Github utiliza el sistema de control de versiones Git, lo que facilita el seguimiento de los cambios en el código, la reversión a versiones anteriores y la colaboración entre múltiples desarrolladores [23].

**Git** Es un controlador de versión de sistemas, es utilizado como una herramienta que permite obtener la trazabilidad de un desarrollo de Software y diseño de software, nos ayuda a tener un fácil control de las versiones de los sistemas. Este sistema es capaz de ir registrando los cambios realizados en un archivo o en varios archivos que pertenecen al mismo proyecto gracias a esto podemos volver en el tiempo y recuperar la versión específica de un proyecto [24].

Para entender de mejor manera la herramienta GitHub el control de versión de los sistemas es un paquete donde se van registrando las actividades realizadas, esto ayuda cuando se desea volver a las versiones anteriores así se obtiene varias ventajas:

- La realización de cambios localmente sin preocuparse de la conectividad.
- No depender de la copia de archivos en el servidor.
- Se reutiliza el trabajo realizado.
- Se puede realizar un trabajo colaborativo sin depender del historial almacenado en las computadoras de los miembros del equipo de trabajo.

## Capítulo 2

# METODOLOGÍA

La realización de este proyecto del módulo de matrículas del Centro de Educación Inicial “Shiny Kids”, tuvo como enfoque la metodología ágil SCRUM. A lo largo del desarrollo, utilicé los roles, artefactos y eventos que esta metodología proporciona, considerando que SCRUM se distingue por su enfoque en la agilidad, con equipos autosuficientes y autónomos.

Para garantizar la implementación del desarrollo de SCRUM se utilizó la herramienta JIRA, ampliamente reconocida para la gestión de proyectos de software en equipos u organizaciones. Esta herramienta proporcionó funcionalidades robustas para la planificación, el seguimiento y la gestión de proyectos ágiles, facilitando la organización del equipo, el rastreo del progreso y la colaboración eficiente. Además, la adaptabilidad, la transparencia, la comunicación eficiente y las mejoras continuas promovidas por SCRUM a través de JIRA fueron fundamentales para enfrentar los desafíos de este componente y asegurar resultados de alta calidad [25].

### 2.1. Implementación SCRUM

En lo que respecta a la implementación de SCRUM, es crucial definir el equipo SCRUM que aplicará esta metodología ágil para continuar con el desarrollo del módulo de matrículas, conforme se especifica en la Tabla 2.1:

SCRUM TEAM		
Rol	Persona	Función
Scrum Master	Ing. Victor Velepucha	Su rol será como líder del proyecto para que realice las revisiones, los progresos del proyecto para alcanzar los objetivos en el tiempo establecido.
Product Owner	Tnlga. Rosario Tarupi	Su rol será el de transmitir las ideas así garantizando el producto final del proyecto, en este caso, el CEI "Shiny Kids".
Developer	Edison Almeida	Se rol será ser el responsable de todas las tareas asignadas para la realización del proyecto; el análisis, diseño e implementación.

Tabla 2.1: Roles del equipo

### 2.1.1. Configuración del proyecto

#### Historias de usuario

El desarrollo del proyecto web se llevó a cabo a través de diversas reuniones con la directora del Centro Educativo (Product Owner), durante las cuales se han especificado todos los requisitos para la creación del módulo de matrículas. En estas reuniones se detalló cómo debería ser el proceso de registro de cada estudiante, con el fin de satisfacer las necesidades y se elaboró un cuadro de historias épicas, el cual se detalla en la Tabla 2.2

HISTORIAS DE USUARIO		
Código	Nombre	Escenario
HU01	Ingreso a la aplicación mediante autenticación	Como usuario registrado necesito mis credenciales de usuario y contraseña para acceder a la aplicación de matriculación.
HU02	Registro de datos del estudiante	Como usuario necesito registrar los datos personales de cada estudiante para guardar en la aplicación web.
HU03	Validar datos ingresados	Como usuario necesito validar los datos requeridos en el formulario de registro para asegurar que no se ingrese datos erróneos.
HU04	Cargar una imagen de la foto del estudiante	Como usuario necesito adjuntar la foto de un estudiante para tener el registro del perfil del estudiante.
HU05	Registro de datos del representante	Como usuario necesito registrar los datos del representante para mantener una información adicional del estudiante.
HU06	Cargar documento "Carnet de Vacunas"	Como usuario necesito adjuntar el pdf del carnet de vacunas de un estudiante para tener el registro médico del estudiante.
HU07	Seleccionar el nivel en el cual el estudiante va a ser matriculado	Como usuario necesito seleccionar el curso para que el estudiante sea registrado en el nivel respectivo
HU08	Imprimir documento de matrícula del estudiante	Como usuario necesito imprimir la matrícula del estudiante para entregar como respaldo al representante
HU09	Almacenar los datos en una base de datos	Como usuario necesito almacenar todos los registros en una base de datos para tener un respaldo de las matrículas
HU010	Listar los datos de los estudiantes matriculados	Como usuario necesito listar por cursos los estudiantes para visualizar los estudiantes matriculados.

Tabla 2.2: Historias de usuario

## Product Backlog

Después de definir las historias de usuario, se generó el Product Backlog, el cual incluye la complejidad y la prioridad de cada tarea para guiar el desarrollo. Por lo tanto, se emplearon los valores especificados en la Tabla 2.3

VALORACIÓN DE PRIORIDAD	
Valoración	Prioridad
1	Baja
2	Media
3	Alta
4	Muy Alta

Tabla 2.3: Valoración de Prioridad

Siguiendo la planificación de scrum, se definió la complejidad de cada historia de usuario utilizando Planning Poker, la cual es una técnica que permitió medir el esfuerzo requerido para completar cada tarea. Las valoraciones se detallan en la Tabla 2.4

VALORACIÓN DE COMPLEJIDAD	
Valoración	Complejidad
1	Muy Baja
2	Baja
3	Media
4	Alta
5	Muy Alta

Tabla 2.4: Valoración de Complejidad

En la definición del Product Backlog se enfatizó en las tareas de mayor complejidad con las de mayor prioridad para establecer el orden de la lista de producto como se indica en la Tabla 2.5

PRODUCT BACKLOG				
HU	Código	Nombre	Complejidad	Prioridad
HU01	T01	Autenticación exitosa	3	1
HU02	T02	Registro básico de estudiante	3	4
HU02	T03	Edición de datos del estudiante	3	3
HU03	T04	Validación de campos requeridos	3	1
HU04	T05	Subir foto del estudiante	5	2
HU04	T06	Validación de formato de imagen	3	2
HU04	T07	Visualizar imagen subida	3	2
HU05	T08	Registrar representante del estudiante	3	4
HU05	T09	Editar datos del representante	3	3
HU06	T10	Subir Carnet de vacunas del estudiante	2	2
HU06	T11	Validación de formato de documento	1	2
HU06	T12	Visualizar documento subido	1	2
HU07	T13	Selección del nivel académico	3	3
HU07	T14	Cambio de nivel académico	3	3
HU08	T15	Imprimir documento de matrícula	2	2
HU09	T16	Guardar datos del estudiante en la base de datos	4	4
HU010	T17	Ver Lista de estudiantes matriculados	3	3
HU010	T18	Filtrar lista de estudiantes por curso	3	3

Tabla 2.5: Product Backlog

### Planificación de sprints

El desarrollo de la aplicación web, se planifico con la ejecución de 5 sprints para cumplir con el backlog, además del sprint 0, permitió configurar el entorno de desarrollo y la arquitectura del aplicativo. Esta planificación se detalla según la Tabla 2.6

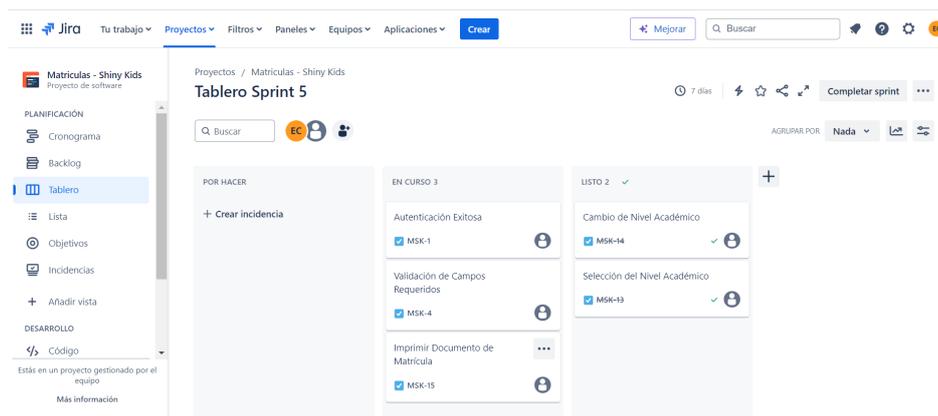


Figura 2.1: Configuración de JIRA

Sprint 0	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5
Preparación del entorno	T16	T03	T17	T05	T01
-	T02	T09	T18	T06	T04
-	T08	-	-	T07	T13
-	-	-	-	T10	T14
-	-	-	-	T11	T15
-	-	-	-	T12	-

Tabla 2.6: Distribución de tareas por Sprints

## 2.2. Inicialización del proyecto

### 2.2.1. Configuración de JIRA

En la implementación respectiva con SCRUM, se procedió con la configuración de JIRA, una herramienta fácil de manejar que facilita el desarrollo y mejora el flujo de trabajo de manera eficiente. A través de JIRA, se cargó las diversas tareas del Product Backlog para asignarlas en el tablero y dar seguimiento conforme al listado de tareas. De esa manera, los sprints detallados se muestran en la Tabla 2.6 y la entrada de tareas se puede visualizar en la Figura 2.1.

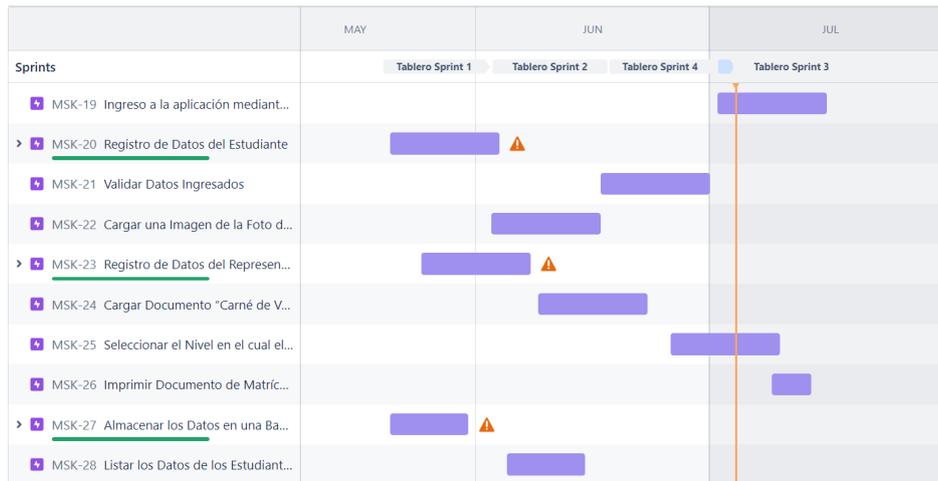


Figura 2.2: Planificación de tareas en JIRA

Una vez creado el proyecto y asignadas las tareas para hacer el seguimiento del desarrollo correspondiente, se configuraron las fechas y los avances del proyecto dentro de los plazos establecidos. Esta configuración se muestra en la Figura 2.2. Además, en el ANEXO I se pueden observar detalles adicionales de la configuración de JIRA, que incluyen el tablero, la lista de incidencias y el cronograma planificado para este proyecto.

## 2.2.2. Diseño de interfaces

Una vez alcanzados los requerimientos del proyecto web del módulo de matrículas, conforme al alcance establecido, se procedió al diseño de cada una de las interfaces y a la creación del prototipo de media fidelidad utilizando la herramienta FIGMA. Este enfoque permitió acercarse más al diseño solicitado por el Centro Educativo y evaluar la navegabilidad del sistema, incluyendo cómo se ingresarán los datos de los estudiantes para su registro de matrícula y la visualización del listado de paralelos y estudiantes matriculados. El enlace al diseño de cada interfaz de la aplicación web está disponible en el ANEXO I.

Para el diseño de la aplicación web, se comienza con la interfaz de inicio de sesión, que facilita el acceso a la aplicación mediante las credenciales correspondientes. La primera pantalla se presenta en la Figura 2.3, donde los usuarios pueden ingresar sus credenciales para autenticarse y acceder al sistema.

Una vez ingresados los datos del usuario, se accede a la pantalla principal que incluye la sección de paralelos, como se muestra en la interfaz de nuestro aplicativo web. Aquí, los usuarios pueden seleccionar cada uno de los paralelos y visualizar la lista de estudiantes matriculados en cada uno, una vez seleccionado el paralelo correspondiente. Esta funcio-

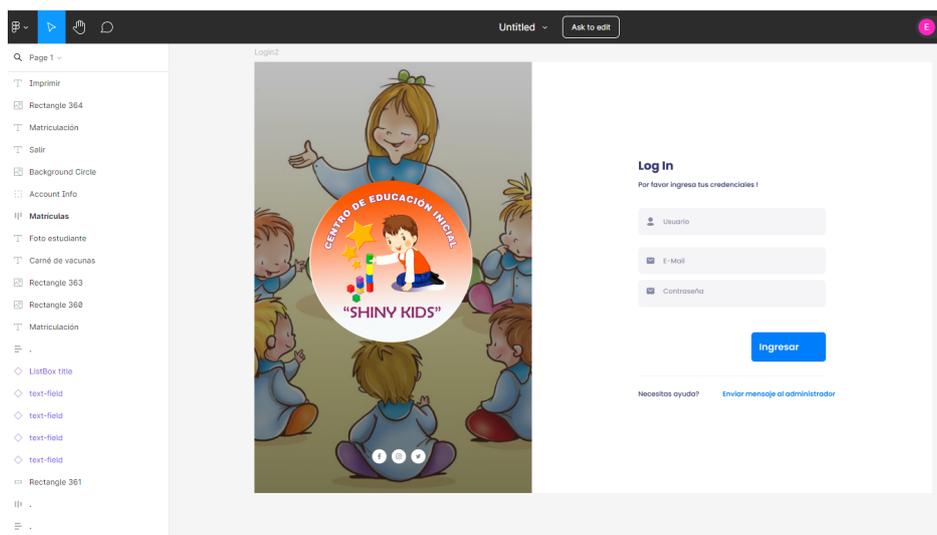


Figura 2.3: Interfaz login

alidad está ilustrada en la Figura 2.4.

De igual manera, es posible navegar en el aplicativo web y acceder a la pantalla de estudiantes. Aquí, los usuarios pueden consultar el listado completo de todos los estudiantes registrados y identificar aquellos que están matriculados. Este listado general de estudiantes del Centro Educativo cumplió con los requisitos establecidos para la visualización de los estudiantes matriculados, como se muestra en la figura 2.5.

Una de las pantallas más importantes para el funcionamiento de la aplicación web es la de registro de estudiantes matriculados. En esta pantalla, se ingresan todos los datos requeridos por el Product Owner, incluyendo información detallada sobre los estudiantes, sus padres y representantes legales, asegurando un registro completo que se almacena en la base de datos. Una vez ingresados los datos, se activa el botón de guardar, y si es necesario, se puede imprimir la información utilizando el botón correspondiente ubicado en la parte inferior. Además, la página web cuenta con un rollo de desplazamiento hacia abajo para facilitar el registro completo de todos los datos, como se muestra en la figura 2.6.

## 2.3. Implementación de sprints

### 2.3.1. Sprint 0

Con base en la información proporcionada por el Centro Infantil Shiny Kids, se ha llevado a cabo un estudio que analiza el desarrollo del proyecto con un diseño factible, que garantice escalabilidad para futuras mejoras del sistema y la integración con diferentes módulos.

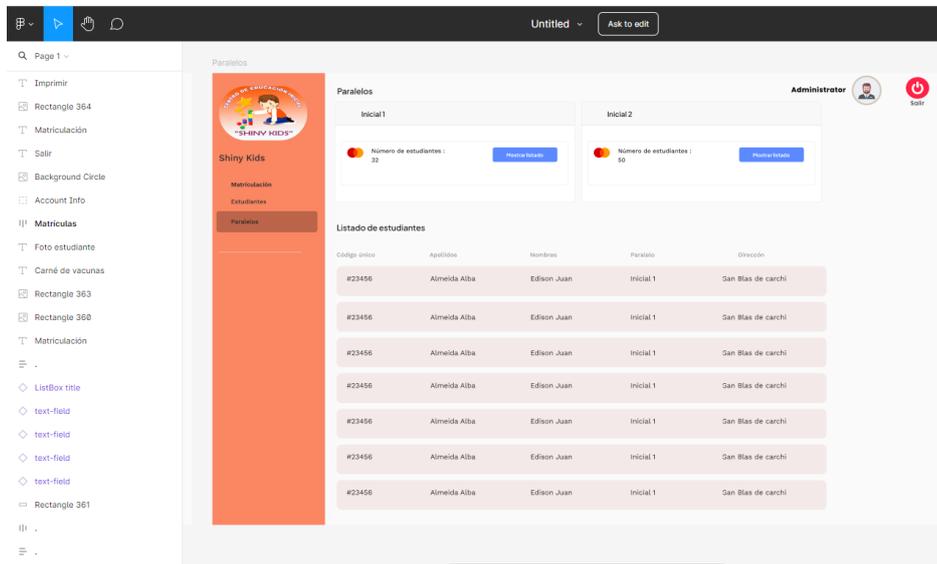


Figura 2.4: Interfaz paralelos

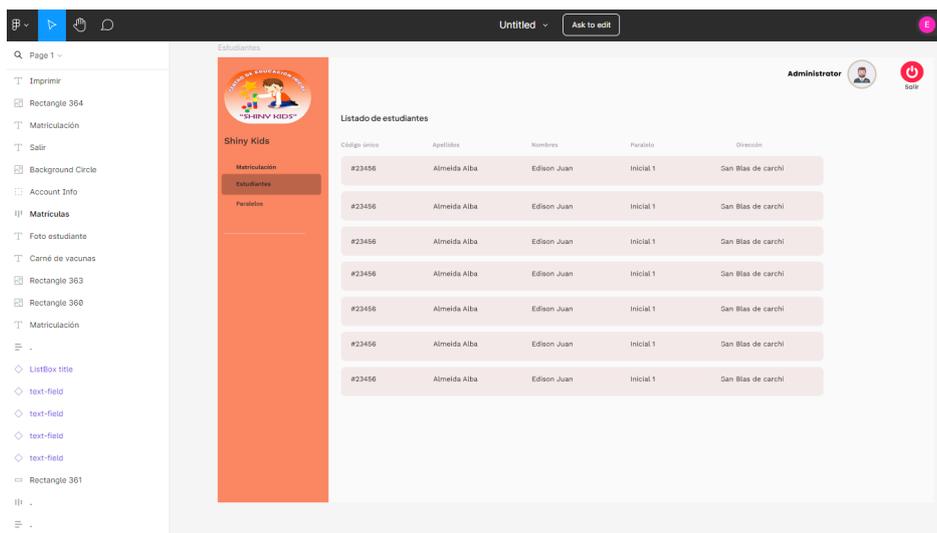


Figura 2.5: Interfaz estudiantes

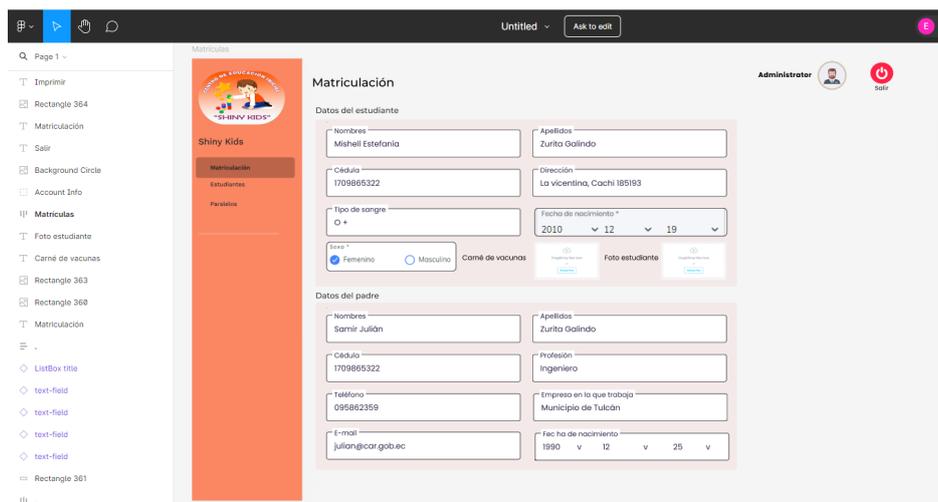


Figura 2.6: Interfaz matrículas

## Ejecución del Sprint 0

### Arquitectura del Proyecto

Se elige una arquitectura para nuestra aplicación web para así establecer una estructura modular y organizada que va a permitir como se ha mencionado el acople de módulos a futuro que necesitará el Centro Infantil y con esta arquitectura tenemos la separación de responsabilidades en la implementación. Como se requiere el fácil acceso a la información y con respaldo de los datos ingresados se opta por la opción de alojar la base de datos usando la plataforma Firebase, que es proporcionado por Google permitiendo un almacenamiento eficiente y escalable donde utilizamos 'Firestore' para como base de datos NoSQL para almacenar información del aplicativo, 'Storage' para almacenar las imágenes y documentos, finalmente 'Authentication' para controlar el ingreso al aplicativo. Estos tres servicios consumimos en nuestra aplicación.

Para el controlador, se implementó el backend utilizando Node.js y Express con el fin de gestionar todas las interacciones entre los usuarios del Centro Educativo y la aplicación. Este enfoque permitirá manejar cada consulta y gestionar las respuestas correspondientes de manera efectiva. La elección de esta tecnología garantiza una flexibilidad eficiente en el desarrollo de rutas y la estructuración de middleware.

Para la capa de presentación, todo el frontend se desarrolló en Angular, encargándose de la maquetación, presentación e interfaces de usuario. Esto proporciona una apariencia mejorada y generará una experiencia visual más atractiva. Esta visión se ilustra claramente

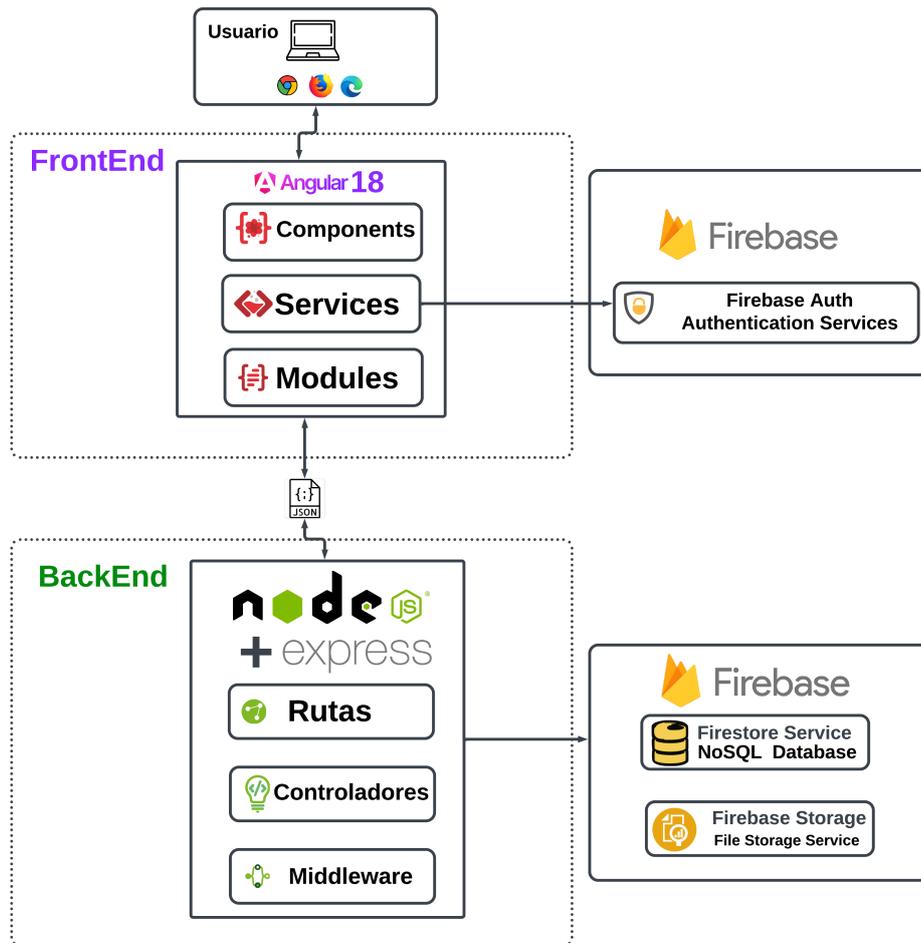


Figura 2.7: Arquitectura aplicación web

en la Figura 2.7.

### Configuración del ambiente de trabajo

Para el desarrollo de la aplicación para la gestión de matrículas del Centro Educativo, se emplearán diversas herramientas y recursos de software libre para así tener un ambiente de fácil mejora para llevar a cabo todo el proceso de implementación. Estas herramientas son esenciales para maximizar la productividad del programador y así se asegurará la eficiencia del proyecto web por lo cual el entorno de trabajo se visualiza de la siguiente manera indicada en la Tabla 2.7

Herramientas de Desarrollo	Software Utilizado
Planificación	Jira Atlassian
Diseño de Interfaces	Figma
Edición de Código	Visual Studio Code
Controlador de Versiones	Github
Base de datos	Firestore
Testing	Postman

Tabla 2.7: Herramientas para el desarrollo web

### **Sprint 0 Review**

Tras culminar el Sprint 0 se logra tener el ambiente de trabajo para desarrollar la aplicación web, además de la arquitectura de como se va a realizar el proyecto y así seguir con la planificación del proyecto.

### **Sprint 0 Retrospective**

- **¿Qué salió bien?** Se logra definir la arquitectura y la perfecta configuración del ambiente de trabajo para continuar con la codificación
- **¿Qué se puede mejorar?** Se debe consultar de mejor manera las versiones compatibles de nodejs con angular para evitar problemas de compatibilidad.

## **2.3.2. Sprint 1**

### **Sprint 1 Planning**

En la ejecución del Sprint, 1 se definió las historias de usuario que facilitan el almacenamiento de datos en la base de Firestore. De esa manera, se planificó la creación de una base de datos en Firestore para el registro de la información correspondiente.

### **Sprint 1 Backlog**

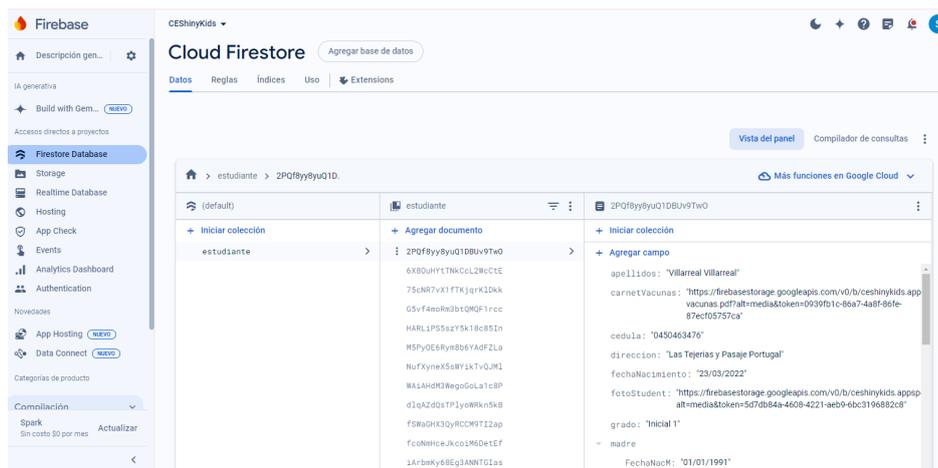


Figura 2.8: Colección en Firestore

Sprint 1 Backlog	
Código	Nombre
T16	Guardar Datos del Estudiante en la Base de Datos
T02	Registro Básico de Estudiante
T08	Registrar Representante del Estudiante

Tabla 2.8: Sprint 1 Backlog

## Ejecución del Sprint 1

En la ejecución del sprint 1, fue necesario almacenar la información en Firestore, donde se realizará el ingreso de documentos para el registro de datos y se creó una colección de datos con las pruebas de los estudiantes insertados directamente en Firestore como se muestra en la Figura 2.8.

Además, a través del backend, se configuró las credenciales para acceder a la base de datos de firestore mediante la creación del archivo firebase.json. Este archivo proporciona las KEY-ID necesarias desde firebase, las cuales permitirán el acceso a la base de datos, como se detalla en la Figura 2.9. En ese sentido, el ingreso de cada uno de los datos de

```
firebase.json ×
backend > firebase.json > client_email
1 {
2   "type": "service_account",
3   "project_id": "ceshinykids",
4   "private_key_id": "c5034a31e45dbfccd7e3013abcb4e09a735c65f5",
5   "private_key": "-----BEGIN PRIVATE KEY-----\nMIIEvQIBADANBgkqhkiG9w0BAQEFAASCBCwgg
6   "client_email": "firebase-adminsdk-zwxci@ceshinykids.iam.gserviceaccount.com",
7   "client_id": "108222336856412418825",
8   "auth_uri": "https://accounts.google.com/o/oauth2/auth",
9   "token_uri": "https://oauth2.googleapis.com/token",
10  "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
11  "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/firebase
12  "universe_domain": "googleapis.com"
13 }
14
```

Figura 2.9: Archivo firebase.json

los estudiantes se realizó a través del backend que se encargó de inicializar la colección e ingresar los datos solicitados del estudiante con el siguiente código:

```
1
2
3 const newStudent = {
4     nombres: params.nombres,
5     apellidos: params.apellidos,
6     cedula: params.cedula,
7     direccion: params.direccion,
8     tipoSangre: params.tipoSangre,
9     fechaNacimiento: params.fechaNacimiento,
10    sexo: params.sexo,
11    fotoCarnetVacunas: null,
12    fotoStudent: null,
13    padre: {
14        nombresP: params.nombresP,
15        apellidosP: params.apellidosP,
16        cedulaP: params.cedulaP,
17        profesionP: params.profesionP,
18        telefonoP: params.telefonoP,
19        lugarTrabajoP: params.lugarTrabajoP,
20        correoP: params.correoP,
21        fechaNacP: params.fechaNacP
22    }
23 }
```

### Sprint 1 Review

Se logró exitosamente la inserción de datos desde el backend a Firestore, permitiendo

ingresar todos los datos solicitados por el usuario.

### **Sprint 1 Retrospective**

#### **¿Qué salió bien?**

Se ha observado un adecuado ingreso de documentos a Firestore de cada estudiante desde el backend, asegurando su almacenamiento en la plataforma.

#### **¿Qué se puede mejorar?**

Sería beneficioso crear más colecciones para mejorar la organización y comprensión de la base de datos en Firestore.

## **2.3.3. Sprint 2**

### **Sprint 2 Planning**

Para la realización de este sprint, se desarrollaron historias de usuario que permiten editar los datos de los estudiantes matriculados y sus representantes. A través del backend, se implementó el CRUD correspondiente, definiendo los procedimientos necesarios para la edición de los datos solicitados.

### **Sprint 2 Backlog**

Sprint 2 Backlog	
Código	Nombre
T03	Edición de datos del estudiante
T09	Editar datos del representante

Tabla 2.9: Sprint 2 Backlog

### **Ejecución del Sprint 2**

Para la realización del sprint 2, se utilizó el backend desarrollado en NodeJS, donde se implementó la función de actualización de los datos de un estudiante matriculado. Primero, se recupera el ID del estudiante y se realiza una llamada a Firestore para proceder con la actualización de los datos. En este proceso, se emplean buenas prácticas de programación, como el uso de try-catch para manejar posibles errores y garantizar su visualización. A

continuación, se presenta un fragmento de código que ilustra cómo se actualizan los datos de un estudiante en una base de datos Firestore:

```
1      updateStudent: async function (req, res) {
      var studentId = req.params.id;
      var update = req.body;

5      try {
          await db.collection('estudiante').doc(studentId).update(update);
          const updatedStudent = await db.collection('estudiante').doc(studentId).
              get();
          return res.status(200).send({ student: updatedStudent.data() });
      } catch (err) {

10         return res.status(500).send({ message: 'Error al actualizar los datos',
            error: err });
      }
  }
```

### **Sprint 2 Review**

Se crea satisfactoriamente el proceso de actualización de datos propuestos para este sprint desde el backend a la base de datos Firestore cuando sean solicitados por el usuario.

### **Sprint 2 Retrospective**

#### **¿Qué salió bien?**

Se puede observar como se van actualizando cada una de los datos a la base de datos Firestore de cada estudiante desde el Backend y se guardan correctamente esta información.

#### **¿Qué se puede mejorar?**

Es posible crear mejores procesos para la validación de datos en el backend, con el fin de tener un control más eficiente sobre los datos a actualizar.

## **2.3.4. Sprint 3**

### **Sprint 3 Planning**

Para la realización de este sprint, se desarrollaron historias de usuario que permiten listar a los estudiantes matriculados y visualizar cada uno de ellos, así como realizar filtros según el grado al que pertenecen en el Centro Infantil, de manera que el usuario pueda

observar la información dependiendo de cada caso.

### Sprint 3 Backlog

Sprint 3 Backlog	
Código	Nombre
T17	Ver lista de estudiantes matriculados
T18	Filtrar lista de estudiantes por curso

Tabla 2.10: Sprint 3 Backlog

### Ejecución del Sprint 3

Para la realización del sprint 3, se tuvo como objetivo extraer los datos desde Firestore y presentar a los estudiantes matriculados según los cursos en los que están inscritos. Cabe destacar que en el Centro Infantil existen cuatro cursos o niveles, por lo que se debe listar a los estudiantes de acuerdo con la elección del usuario. Además, se obtiene una lista completa de todos los estudiantes matriculados para su visualización por parte del usuario.

Para obtener este tipo de datos, se realizó una llamada a la colección "getStudents" que se sincroniza con ".async" para gestionar la solicitud HTTP y su respuesta. Como buena práctica de programación, también implementamos un bloque try-catch para manejar posibles errores. Finalmente, se accedió a la colección de estudiantes en Firestore utilizando "querySnapshot.docs" para recuperar gradualmente los datos. Este proceso se muestra en el siguiente código, diseñado para recuperar los datos de los estudiantes desde el backend:

```
1      getStudents: async function (req, res) {  
    try {  
      const querySnapshot = await db.collection('estudiante').get();  
      const students = querySnapshot.docs.map(doc => ({ id: doc.id, nombres  
        : doc.data().nombres, apellidos: doc.data().apellidos, grado: doc  
        .data().grado, direccion: doc.data().direccion }));  
5      return res.status(200).send({ students });  
    } catch (err) {
```

```
        return res.status(500).send({ message: 'Error al recuperar los datos'
        , error: err });
    }
},
```

### **Sprint 3 Review**

Se cumplió con las tareas propuestas en este sprint como también la tarea de extraer los datos desde Firestore, donde está alojado la colección de estudiantes y se pueda mostrar cada uno de los estudiantes según el usuario solicite de cada uno de los cursos o listar la lista de estudiantes matriculados para su correcta visualización.

### **Sprint 3 Retrospective**

#### **¿Qué salió bien?**

Se puede observar como se muestran todos los datos que se obtiene desde la Firestore conforme el usuario solicite, ya sea por cursos o si prefiere puede listar todos los estudiantes matriculados.

#### **¿Qué se puede mejorar?**

Se puede crear mejores procesos para la extracción de datos desde Firestore ajustándolos de mejor manera, como también crear mas filtros solicitados por el usuario para tener un mejor uso de los datos ingresados de los estudiantes matriculados.

## **2.3.5. Sprint 4**

### **Sprint 4 Planning**

Durante la ejecución del sprint 4, se abordó una de las partes más críticas de la aplicación: el manejo de imágenes de carnet y PDFs de carnet de vacunación de los estudiantes. Esto incluye la gestión de la foto del estudiante y el archivo PDF de las vacunas de los niños que van a ser matriculados. Para ello, se utilizó el servicio de almacenamiento de Firebase - Storage, donde se establece el acceso a los documentos y se almacenan tanto las imágenes como los PDFs.

### **Sprint 4 Backlog**

Sprint 4 Backlog	
Código	Nombre
T05	Subir foto del estudiante
T06	Validación de formato de imagen
T07	Visualizar imagen ingresada
T10	Subir Carnet de vacunas del estudiante
T11	Validación de formato de documento
T12	Visualizar documento ingresado

Tabla 2.11: Sprint 4 Backlog

## Ejecución del Sprint 4

Para la ejecución de este sprint, se consideró el manejo de archivos en diferentes formatos, como imágenes y archivos PDF de diversos tamaños, que se desean cargar en la base de datos de cada estudiante y como se interactúa con el servicio de almacenamiento de Firestore. El objetivo principal es permitir la carga de archivos con un token único que identifique tanto la imagen como el PDF pertenecientes al mismo estudiante.

Se implementó la lógica necesaria para generar un nombre de archivo único para las imágenes utilizando “uniqueFileName”, facilitando así la carga de la imagen con un nombre único en el almacenamiento, como se detalla en el siguiente código:

```

1      if (profileImage) {
          const uniqueFileName = `${docID}.${profileImage.mimetype.split('/')
            [1]} `;
          const tempLocalFile = profileImage.buffer;

```

También fue fundamental implementar try-catch para identificar posibles errores durante

la carga de la imagen. Para guardar la imagen en Google Cloud Storage, se creó una referencia al archivo en el bucket y lo almacenamos en la carpeta "images". Posteriormente, se sube el archivo al bucket junto con los datos necesarios, incluyendo el token único para asegurar descargas seguras y gestionar correctamente los archivos. A continuación, se muestra el fragmento de código donde se implementa la lógica para la subida de archivos:

```
1      try {
          const file = bucket.file('images/${uniqueFileName}');
          await file.save(tempLocalFile, {
              metadata: {
                contentType: profileImage.mimetype,
                metadata: {
                  firebaseStorageDownloadTokens: uuidv4(),
                }
              }
            });

          imageUrl = `https://firebasestorage.googleapis.com/v0/b/${
            bucket.name}/o/images%2F${encodeURIComponent(
              uniqueFileName)}?alt=media&token=${uuidv4()}`;
          newStudent.fotoStudent = imageUrl; // Guardar la URL en el
            campo correspondiente
        } catch (uploadErr) {
          return res.status(500).send({
            message: 'Error al subir la imagen',
            error: uploadErr,
          });
        }
      }
```

### **Sprint 4 Review**

Se cumple con las tareas propuestas en este sprint que son la parte central de la aplicación e importante en la gestión de las matrículas de subida de la foto del estudiantes y el carnet de vacunas que se debe tener en el almacenamiento del Firebase Storage donde se ha propuesto para guardar los archivos.

### **Sprint 4 Retrospective**

#### **¿Qué salió bien?**

Se puede observar como se suben los archivos tanto como la foto del estudiante como el pdf de carnet de vacunas del estudiante y se alojan en el Storage del Firebase, tanto las imágenes en la carpeta 'images' y los archivos pdf en la carpeta 'documentos'.

### ¿Qué se puede mejorar?

Se puede mejorar el manejo de archivos mediante la optimización de la interfaz, haciéndola más intuitiva para el usuario. De esta manera, el usuario podrá seleccionar los archivos con la extensión adecuada necesaria para subirlos correctamente a la base de datos.

## 2.3.6. Sprint 5

### Sprint 5 Planning

En este sprint se realiza las tareas de autenticación del usuario en la aplicación web utilizando Firebase Authentication, como se planteó en la arquitectura. Además, desde el frontend, se realizará la validación de datos en la interfaz de matriculación para garantizar que los datos ingresados por el usuario sean válidos.

Igualmente, las tareas de elección del curso para la matrícula se llevarán a cabo a través del frontend, permitiendo al usuario seleccionar el curso en el que desea matricularse. Para la fase final, se realizará la impresión del formulario en el cual se incluirán los datos más relevantes especificados por el Product Owner para esta tarea.

### Sprint 5 Backlog

Sprint 5 Backlog	
Código	Nombre
T01	Autenticación exitosa
T04	Validación de campos requeridos
T13	Selección del nivel académico
T14	Cambio de nivel académico
T15	Imprimir documento de matrícula

Tabla 2.12: Sprint 5 Backlog

## Ejecución del Sprint 5

Para realizar la ejecución de este sprint se va a realizar la validación de datos a través de Angular, en el frontend con Firebase Authentication que nos facilita el logueo del usuario y nos proporciona la seguridad. Para la realización de esta parte utilizamos el método `onSubmit()`, que nos ayudará a manejar el inicio de sesión. Si la validación es exitosa nos redirigirá a la página de paralelos, caso contrario nos enviará una alerta al usuario. Este detalle lo podemos observar en el siguiente código:

```
1      onSubmit() {
      signInWithEmailAndPassword(this.auth, this.email, this.password)
        .then(userCredential => {
          // Logged in successfully
5         console.log('Logged in:', userCredential);
          this.router.navigate(['/paralelos']); // Redirige al componente Paralelos
        })
        .catch(error => {
          // Handle errors here
10         console.error('Login error:', error);
          window.alert('Correo o contraseña incorrectos. Inténtalo de nuevo.');
```

Al igual debemos validar los campos de la aplicación, para esto necesitamos a través del frontend realizar las validaciones respectivas. A continuación mostramos la validación de campos poniendo como ejemplo el ingreso de la fecha de nacimiento y como se realiza la validación, como indicamos en el siguiente código:

```
1      <div class="form-floating">
          <input type="date" class="form-control" name="fechaNacimiento" #
            fechaNacimiento="ngModel" placeholder="Fecha de nacimiento"
            [(ngModel)]="newStudent.fechaNacimiento" required>
          <span *ngIf="fechaNacimiento.touched && !fechaNacimiento.valid"
            style="color: red;">Fecha de nacimiento de sangre requerida
            !</span>
          <label for="fechaNacimiento">Fecha de nacimiento</label>
5      </div>
```

## Sprint 5 Review

Se cumple con las tareas propuestas en este sprint que son la autenticación del usuario a través de Firebase Autentication que se cumple y es de gran importancia para la seguridad de la aplicación web.

### **Sprint 5 Retrospective**

#### **¿Qué salió bien?**

Se puede observar como se autentica el usuario a través de los usuarios creados en Firebase Autentication así se puede observar la seguridad que ofrece la plataforma de Google, además el usuario cuenta con la validación de campos requeridos en el formulario de registro.

#### **¿Qué se puede mejorar?**

Se puede mejorar el manejo de las interfaces de login para mayor facilidad del usuario al momento de su autenticación, como por ejemplo, que en el formulario el usuario pueda tener mejores mensajes de campos no válidos en el ingreso de un estudiante al momento que procesa la validación en el aplicativo.

## **2.4. Pruebas de funcionalidad**

Las pruebas de funcionalidad se proponen con el objetivo principal de verificar el funcionamiento de la aplicación web. Estas pruebas son esenciales, ya que permitirán comprobar que todos los requisitos especificados por el Product Owner han sido desarrollados correctamente [26].

Las pruebas de funcionalidad se realizarán utilizando principalmente un enfoque funcional, que se implementa mediante el método de Caja Negra. Este enfoque se centra en verificar el comportamiento esperado del módulo de matrículas, buscando identificar casos en los que el módulo no cumple con sus especificaciones. En este método, se proporcionan datos de entrada y se evalúan los resultados obtenidos sin considerar la implementación interna del módulo. Las pruebas de Caja Negra se basan especialmente en los requisitos del módulo, asegurando así la cobertura de especificación. Durante las pruebas, se documentará el cumplimiento de los requisitos que han sido validados [27].

### **2.4.1. Resultados de las pruebas de funcionalidad**

Para llevar a cabo las pruebas de funcionalidad del módulo de matrículas de la aplicación web del Centro Educativo, se realizará una revisión de sprint con la directora del Centro

Educativo (Product Owner), donde se analiza el Sprint Review de cada uno de los sprints propuestos junto con las tareas asignadas. Durante este proceso, se examinan detalladamente las tareas para gestionar adecuadamente los resultados obtenidos.

A continuación presentamos la tabla 2.13 con el resumen de los resultados de las pruebas de funcionalidad del Sprint 1

<b>Historia de Usuario</b>	<b>Criterio de Aceptación</b>	<b>Cumplido</b>
T16-Guardar datos del estudiante en la base de datos	Como administrador cuando ingreso los datos del estudiante y hago clic en el botón Guardar, entonces los datos deberían ser almacenados correctamente en la base de datos	Si
T02-Registro básico de estudiante	Como administrador quiero registrar los datos personales básicos de un estudiante para mantener un registro detallado de cada estudiante matriculado	Si
T08-Registrar representante del estudiante	Como administrador quiero registrar los datos personales del representante del estudiante para mantener un registro de contacto y responsabilidad	Si

Tabla 2.13: Pruebas de Funcionalidad Sprint 1

Para completar las pruebas de funcionalidad del Sprint 1 se realiza los criterios de aceptación de cada una de las tareas asignadas en el Product Backlog del Sprint 1.

A continuación mostramos como se realizó la pruebas de funcionalidad del Sprint 1 en el cual ingresamos los datos básicos de un estudiante y representante para que sean guardados en la Firestore como se visualiza en la Figura 2.10:

Siguiendo con las pruebas de funcionalidad se presenta la tabla 2.14 con los criterios de aceptación del Sprint 2:

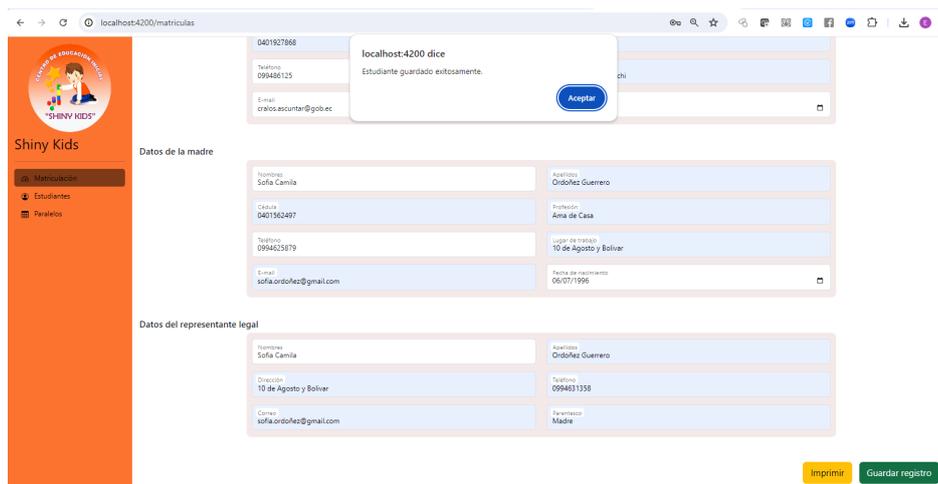


Figura 2.10: Sprint 1 Pruebas de funcionalidad de guardar estudiante

Historia de Usuario	Criterio de Aceptación	Cumplido
T03-Edición de datos del estudiante	Como administrador quiero poder editar los datos de un estudiante existente para mantener la información del estudiante actualizada	Si
T09-Editar datos del representante	Como administrador quiero poder editar los datos de un representante existente para mantener la información actualizada	Si

Tabla 2.14: Pruebas de Funcionalidad Sprint 2

Las pruebas de funcionalidad del Sprint 2 se basaron en la funcionalidad de edición del estudiante y del representante al momento de realizar el ingreso de datos a la aplicación.

Para esto observamos como podemos ingresar los nuevos datos del representante legal y nuevamente dar clic en el botón guardar para que los cambios sean realizados de forma correcta como podemos observar en la figura 2.11:

A continuación vamos a observar los criterios de aceptación que se proponen en el Sprint 3 en la tabla 2.15:



Figura 2.11: Sprint 2 Pruebas de funcionalidad de editar estudiante

Historia de Usuario	Criterio de Aceptación	Cumplido
T17-Ver lista de estudiantes matriculados	Como administrador quiero ver una lista de todos los estudiantes matriculados para tener un registro accesible y detallado de las matrículas	Si
T18-Filtrar lista de estudiantes por curso	Como administrador quiero filtrar la lista de estudiantes por nivel académico para encontrar rápidamente estudiantes en un nivel específico	Si

Tabla 2.15: Pruebas de Funcionalidad Sprint 3

Las pruebas de funcionalidad del Sprint 3 se basaron en la visualización de los estudiantes matriculados donde el usuario requiere el filtrado de cada uno de los cursos como visualizar los datos de los estudiantes ingresados.

Para esto debemos estar en la sección de paralelos y elegir cada uno de los cursos para filtrar los estudiantes que se han matriculado en cada uno de los cursos como se muestra en la figura 2.12

Continuando con los criterios de aceptación de las pruebas de funcionalidad se realiza del sprint 4 detallado en la tabla 2.16:

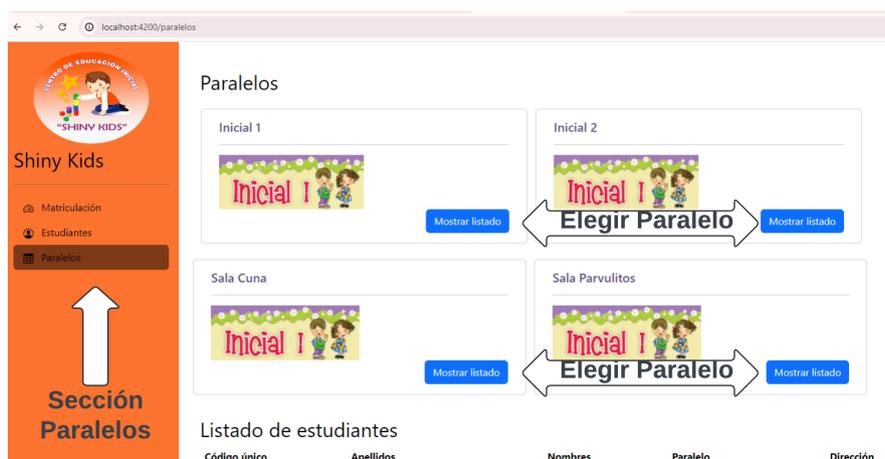


Figura 2.12: Sprint 3 Pruebas de funcionalidad de listar estudiantes

Historia de Usuario	Criterio de Aceptación	Cumplido
T05-Subir foto del estudiante	Como administrador quiero poder subir una foto del estudiante para tener una imagen de identificación en el perfil del estudiante	Si
T06-Validación de formato de imagen	Como administrador quiero que el sistema valide el formato de la imagen subida para asegurarme de que solo se acepten formatos de imagen válidos	Si
T07-Visualizar imagen subida	Como administrador quiero visualizar la imagen del estudiante después de cargarla para asegurarme de que se ha subido correctamente	Si
T10-Subir Carnet de vacunas del estudiante	Como administrador quiero poder subir el carnet de vacunas del estudiante para mantener un registro de su estado de vacunación	Si
T11-Validación de formato de documento	Como administrador quiero que el sistema valide el formato del documento cargado para asegurarme de que solo se acepten formatos válidos	Si
T12-Visualizar documento subido	Como administrador quiero visualizar el carnet de vacunas del estudiante después de cargarlo para asegurarme de que se ha subido correctamente	Si

Tabla 2.16: Pruebas de funcionalidad Sprint 4

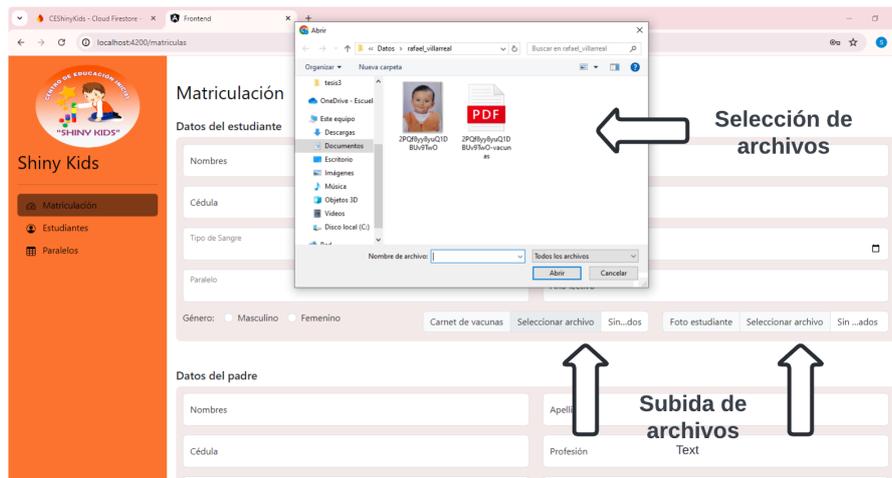


Figura 2.13: Sprint 4 Pruebas de funcionalidad de carga de archivos

En la realización de las pruebas de funcionalidad del Sprint 4 tenemos como el usuario puede subir la foto del estudiante y así mismo el botón de ingreso del PDF del carnet de vacunas como lo podemos observar en la Figura 2.13

Para finalizar las pruebas de funcionalidad en el sprint 5, se enfatizó en los criterios de aceptación del login del sistema, que incluyen la autenticación, la validación de los campos requeridos, la selección del curso para la matrícula del estudiante y la impresión del documento de matrícula, como se detalla en la Tabla 2.17:

Historia de Usuario	Criterio de Aceptación	Cumplido
T01-Autenticación exitosa	Como usuario registrado quiero ingresar mis credenciales de usuario y contraseña para acceder a la aplicación de matriculación	Si
T04-Validación de campos requeridos	Como administrador quiero que el sistema valide los campos requeridos en el formulario de registro para asegurarme de que no falte información crucial	Si
T13-Selección del nivel académico	Como administrador quiero ingresar el nivel académico del estudiante para asegurarme de que el estudiante sea matriculado en el nivel correcto	Si
T14-Cambio de nivel académico	Como administrador quiero poder cambiar el nivel académico del estudiante para ajustar la matrícula según sea necesario	Si
T15-Imprimir documento de matrícula	Como administrador quiero imprimir el documento de matrícula generado para entregar una copia física al estudiante o su representante	Si

Tabla 2.17: Pruebas de Funcionalidad Sprint 5

Para realizar la validación del login asignamos el correo y contraseña al usuario para que pueda realizar el ingreso respectivo como lo podemos visualizar en la figura 2.14

Así también, el usuario tendrá las advertencias de campos erróneos cuando vaya ingresando los datos solicitando siempre que los campos deben ser ingresados del estudiante que esta realizando su matrícula como lo observamos en la Figura 2.15

Finalmente para completar las pruebas de funcionalidad del sprint 5 tenemos la impresión del estudiante que ha realizado la matrícula donde el usuario da clic en el botón de imprimir y podrá visualizar la impresión del estudiante matriculado como se observa en la figura 2.16

Después de revisar las pruebas de funcionalidad, se verificará que los objetivos planteados en la aplicación web han sido cumplidos satisfactoriamente según los criterios de

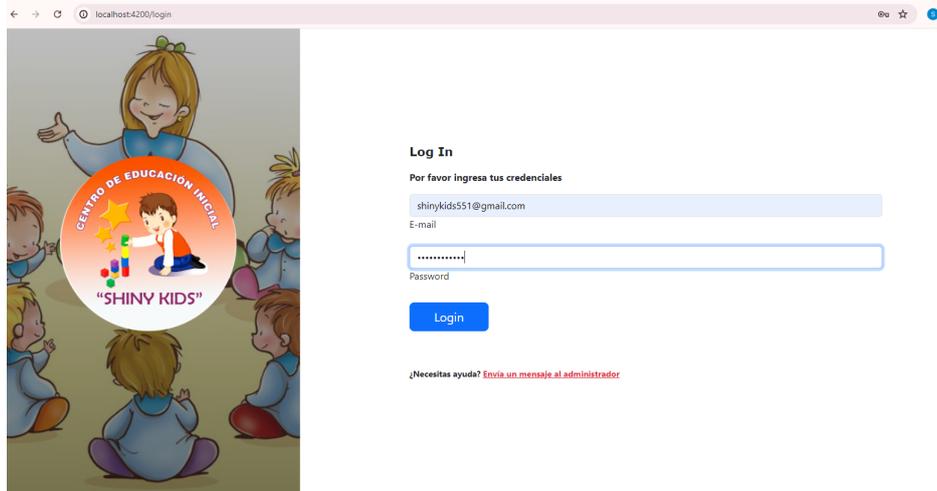


Figura 2.14: Sprint 5 Pruebas de funcionalidad de autenticación

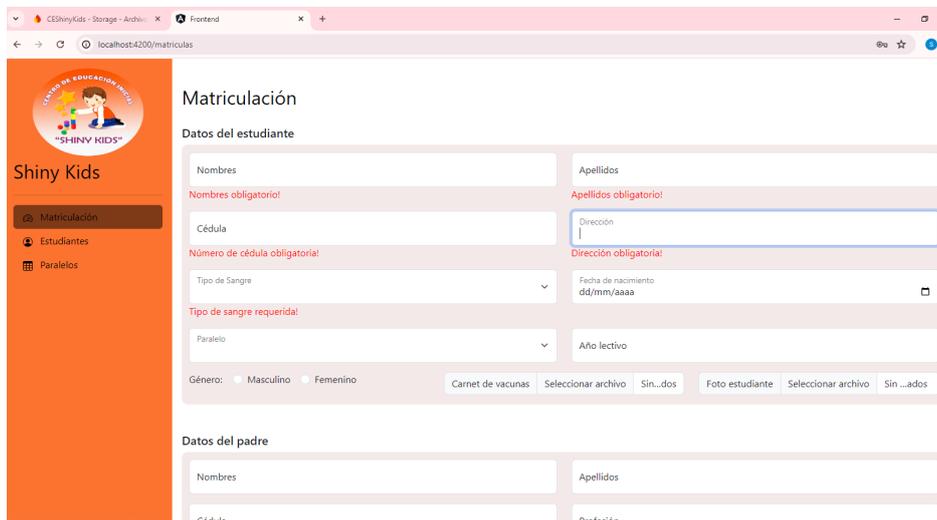


Figura 2.15: Sprint 5 Pruebas de funcionalidad de campos requeridos

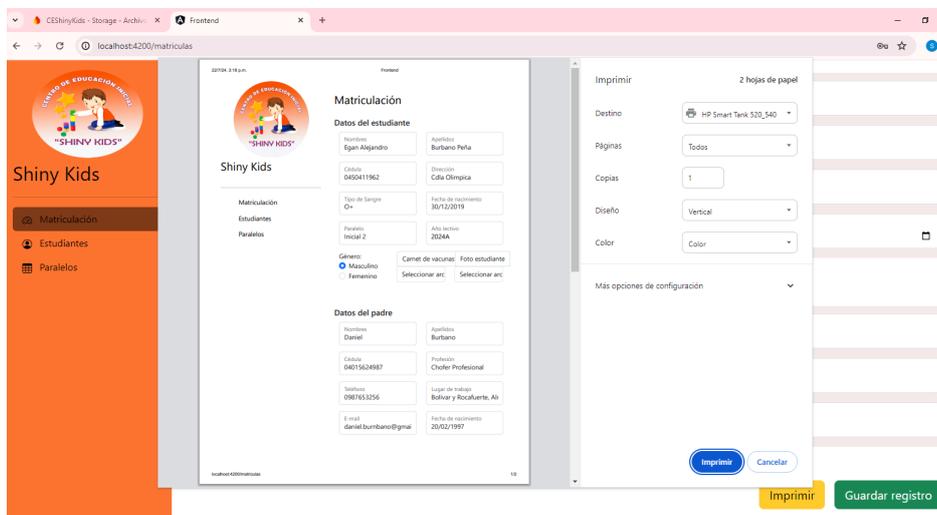


Figura 2.16: Sprint 5 Pruebas de funcionalidad de imprimir

aceptación establecidos. Además, se presentan las tablas de verificación de funcionalidad con los resultados obtenidos, evidenciando que el módulo de matrículas de la aplicación web funciona correctamente, sin errores en la programación y sin alterar los procesos descritos dentro del navegador web.

## Capítulo 3

# CONCLUSIONES Y RECOMENDACIONES

### 3.1. Conclusiones

- La implementación del sistema web para la gestión de matrículas cumple con todos los requerimientos establecidos por el Product Owner. Este sistema no solo facilita la gestión de los estudiantes que deben ser matriculados en el Centro Educativo, sino que también proporciona una plataforma centralizada que optimiza el flujo de trabajo administrativo. La plataforma permite registrar, rastrear y gestionar de manera eficiente toda la información relacionada con las matrículas, lo que resulta en una administración más organizada y menos propensa a errores.
- Con este proyecto se han alcanzado todos los objetivos propuestos en el plan, evidenciando un cumplimiento exhaustivo de las metas establecidas. La aplicación fue desarrollada utilizando tecnologías avanzadas en sus últimas versiones, como Node.js y Angular, lo que permitió un desarrollo ágil y una interfaz de usuario moderna y responsiva. Además, la integración con el servicio en la nube de Firebase de Google para el almacenamiento y gestión de datos, con lo cual aseguramos un rendimiento eficiente y una alta disponibilidad. La adopción de estas tecnologías modernas ha contribuido significativamente a la automatización del proceso de matrículas, reduciendo el tiempo requerido para completar cada proceso, simplificando la administración. Esto se traduce en una experiencia más fluida tanto para los administradores como para los usuarios finales, mejorando así la eficiencia operativa del Centro Educativo.

- El uso de la metodología Scrum ha sido fundamental para la realización del proyecto de manera ágil y efectiva. Esta metodología permitió una adaptación continua a los cambios y ajustes necesarios durante el desarrollo, asegurando que el producto final cumpliera con las expectativas del Product Owner. La comunicación constante y efectiva entre el Product Owner y el desarrollador fue crucial para mantener una alineación precisa de las expectativas y prioridades del proyecto. Esta colaboración continua facilitó la identificación y resolución temprana de problemas, garantizando que el proyecto avanzara de acuerdo con los plazos y objetivos establecidos.
- La utilización de varios servicios de Firebase, para la gestión de bases de datos, autenticación y almacenamiento de archivos proporciona un alto nivel de seguridad para la información. Firebase ofrece robustas características de seguridad, incluyendo autenticación avanzada y controles de acceso granular, que protegen los datos sensibles contra accesos no autorizados. Esta capa adicional de seguridad es esencial para salvaguardar la privacidad y la integridad de la información de los estudiantes, garantizando que los datos estén protegidos frente a posibles amenazas y cumpliendo con los estándares de protección de datos vigentes.
- El modelo de arquitectura propuesto para la aplicación web está diseñado para permitir una gestión eficiente de datos y recursos, lo cual facilita las funcionalidades críticas como la autenticación de usuarios y el almacenamiento de archivos. Esta arquitectura no solo optimiza la eficiencia operativa de la aplicación, sino que también facilita su escalabilidad y mantenimiento. De esta manera, la aplicación podrá crecer y adaptarse a nuevas necesidades y requisitos sin comprometer el rendimiento o la estabilidad del sistema. La capacidad de escalar y mantener el sistema de forma efectiva asegura que pueda evolucionar con el tiempo, incorporando mejoras y nuevas funcionalidades sin afectar su funcionamiento general.

## **3.2. Recomendaciones**

- Es recomendable utilizar frameworks y tecnologías actualizadas debido a que los cambios frecuentes en los servicios de Google pueden generar inconvenientes al momento de integrar estos servicios, dado que ofrecen una infraestructura robusta. Por esta razón, es importante mantener la aplicación actualizada, incorporando nuevas funcionalidades y mejoras basadas en los comentarios de los usuarios.

- Dado que el Centro Educativo está en crecimiento, se debe planificar la escalabilidad de la aplicación, continuando con el desarrollo modular. Esto facilitará tanto el mantenimiento como las actualizaciones futuras, asegurando que la aplicación pueda adaptarse a las necesidades cambiantes del Centro.
- Se recomienda proporcionar capacitación a los usuarios del Centro Infantil sobre el uso de la aplicación web. Esto permitirá una completa familiarización con el sistema, de modo que los usuarios puedan aprovechar al máximo las funcionalidades disponibles. Además, una adecuada capacitación permitirá a los usuarios proporcionar información valiosa que contribuye a la mejora del sistema y a resolver cualquier duda o problema que pueda surgir.
- Es esencial implementar un sistema de respaldo regular de todos los datos almacenados en la aplicación. Esto ayudará a mitigar los efectos de cualquier fallo o pérdida de datos que pueda ocurrir. Con un respaldo actualizado, se podrá restaurar la información de manera más rápida y efectiva, garantizando la integridad y disponibilidad de los datos.

## Capítulo 4

# REFERENCIAS BIBLIOGRÁFICAS

- [1] D. J. Barrientos Mogollon, «Técnicas, métodos y herramientas para la medición de Usabilidad en sistemas web y aplicaciones móviles: Una revisión sistemática de la literatura,» 2020.
- [2] D. Villalón Pardo, «Crear y desarrollar una aplicación de alto rendimiento con bajo coste utilizando flutter y firebase,» Tesis doct., Universitat Politècnica de València, 2021.
- [3] R. Farinella y A. E. Fuentes, «Desarrollo de herramienta para la prevención de Smells de UX en prototipos tempranos,» Tesis doct., Universidad Nacional de La Plata, 2023.
- [4] K. P. Naranjo Páez y O. E. Pullay Vinueza, «Desarrollo de una aplicación móvil que facilite la repartición de alimentos para el proyecto Pancomido,» B.S. thesis, Quito: EPN., 2019.
- [5] G. Cardiel Altemir, «Desarrollo de una aplicación Fitness en el sistema operativo de Android con la plataforma Firebase,» 2019.
- [6] J. Tomás, V. Carbonell, J. Bataller y J. Lloret, *Firestore: trabajar en la nube*. Alpha Editorial, 2019.
- [7] B. R. Jiménez Bustamante, «Aplicación web usando contenedores para la gestión de cursos de actualización de conocimientos de estudiantes del Centro de Posgrados de la Universidad Técnica de Ambato,» B.S. thesis, Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas . . ., 2024.
- [8] A. Piedrahíta-Carvajal, P. A. Rodríguez-Marín, D. F. Terraza-Arciniegas, M. Amaya-Gómez, L. Duque-Muñoz y J. D. Martínez-Vargas, «Aplicación web para el análisis

- de emociones y atención de estudiantes,» *TecnoLógicas*, vol. 24, n.º 51, págs. 62-76, 2021.
- [9] M. A. M. Rodríguez, J. A. Zambrano, I. P. Acosta y C. Z. Cedeño, «Aplicación web para gestionar los procesos administrativos de las Asociaciones Agropecuarias del cantón Chone de la provincia de Manabí.,» *Revista Científica Sinapsis*, vol. 2, n.º 15, 2019.
- [10] J. R. M. Ríos, M. P. Z. Ordóñez, M. J. C. Segarra y F. G. G. Zerda, «Comparación de metodologías en aplicaciones web,» *3C Tecnología: glosas de innovación aplicadas a la pyme*, vol. 7, n.º 1, págs. 1-19, 2018.
- [11] J. Forcada Sanz, «Frameworks para desarrollo de aplicaciones móviles híbridas: análisis comparativo y aplicación a servicios de emergencia,» Tesis doct., ETSI\_Informatica, 2020.
- [12] L. Puciarelli, *Angular: TypeScript–Arquitectura–Instalación–Directivas y Bindings–Forms–Ruteo y más*. RedUsers, 2020.
- [13] E. Haro, T. Guarda, A. O. Z. Peñaherrera y G. N. Quiña, «Desarrollo backend para aplicaciones web, servicios web restful: Node. js vs spring boot,» *Revista Ibérica de Sistemas e Tecnologías de Informação*, n.º E17, págs. 309-321, 2019.
- [14] D. Flanagan, *JavaScript: The definitive guide: Activate your web pages*. .ºReilly Media, Inc.", 2011.
- [15] J. C. O. Cabello, *Diseño de páginas Web con XHTML, JavaScript y CSS*. Ra-Ma Editorial, 2010.
- [16] S. Vargas, «Diseño e implementación de un sitio web institucional para la Unidad Educativa Municipal “Julio Enrique Moreno” en el período académico 2021-2022,» Tesis de pregrado, Universidad Central del Ecuador, 2022. dirección: <https://www.dspace.uce.edu.ec/server/api/core/bitstreams/924387ad-b781-4939-b33d-053ce094ff4c/content>.
- [17] R. Barahona, «Diseño, desarrollo e implementación de sitio web para alertar violencia de género,» Tesis de pregrado, Universidad Católica de Guayaquil, 2022. dirección: <http://repositorio.ucsg.edu.ec/bitstream/3317/18433/1/T-UCSG-PRE-ING-CIC-17.pdf>.

- [18] Universidad de Murcia, *UM.es*, Retrieved from UM.es: <https://www.um.es/docencia/barzana/DAWEB/2017-18/daweb-tema-1-introduccion-html-css.html>, feb. de 2019.
- [19] N. J. Merchán-Narváez, E. E. Palma-Peralta, D. X. Poma-Japón et al., «Comparación de metodologías ágiles para el desarrollo de software,» *MQR Investigar*, vol. 8, n.º 1, págs. 5052-5074, 2024.
- [20] A. López Gil et al., «Estudio comparativo de metodologías tradicionales y ágiles para proyectos de Desarrollo de Software,» 2018.
- [21] L. Ballesteros, «Implementación y práctica de scrum en la asignatura de formulación y evaluación de proyectos en la Facultad de Ciencias Económicas y Administrativas de la Universidad El Bosque,» *Panorama*, vol. 15, n.º 29, pág. 7, 2021.
- [22] W. D. Sierra Bolaños, «Desarrollo de una aplicación web para la automatización del proceso de seguimiento de trabajos de grado e informes docentes de la carrera de Ingeniería en Sistemas Computacionales,» B.S. thesis, 2020.
- [23] R. E. Silverman, *Git Pocket Guide: A Working Introduction*. .ºReilly Media, Inc.", 2013.
- [24] W. I. Rochina Rea, «Desarrollo de una guía metodológica para la gestión de proyectos de software con la herramienta GitHub y el marco de trabajo SCRUM.,» B.S. thesis, 2023.
- [25] V. J. Alarcon Eslava y R. S. Cabrales Oviden, «Aplicación de la metodología ágil SCRUM con la herramienta JIRA para mejorar la gestión de proyectos de telecomunicaciones,» 2023.
- [26] C. E. Vizúete Allauca, «Desarrollo e implementación de un sistema web para la automatización de gestión de procesos BC3-System, utilizando criterios de funcionalidad y usabilidad.,» B.S. thesis, Escuela Superior Politécnica de Chimborazo, 2019.
- [27] T. N. Vásquez, *ANÁLISIS COMPARATIVO DE LOS MÉTODOS DE CAJA NEGRA, BLANCA Y GRIS DE TESTING DE SOFTWARE, PARA LA PROPUESTA DE UN MÉTODO ESTÁNDAR DE TESTING*, 2012.
- [28] L. Puciarelli, *Node JS-Vol. 1: Instalación-Arquitectura-node y npm*. RedUsers, 2020, vol. 1.

# ANEXO I

## ANEXOS

### DISEÑO DE INTERFACES

#### Interfaces en Figma

A través de la herramienta FIGMA se realizó el diseño de media fidelidad, en el siguiente enlace se puede encontrar las diferentes interfaces:

[Link Interfaces Figma](#)

### IMPLEMENTACIÓN DE SCRUM

#### Configuración Jira

A través de la herramienta JIRA se realizó el cronograma y la implementación de los sprints, donde podemos visualizar todo el cronograma y la tareas realizadas.

Puedes encontrar el cronograma Jira en el siguiente enlace:

[Link Cronograma Jira](#)

### PROYECTO COMPLETO EN GITHUB

#### GITHUB

Se puede visualizar el proyecto completo a través de la plataforma GITHUB el proyecto completo

Puedes visualizar el código del proyecto en el siguiente enlace:

[Link Proyecto Github](#)