**

**Capstone**

**Proyecto APT “SafeKIDS”**



**Docente:** MARILUZ ALEJANDRA RODRIGUEZ DONOSO

**Integrantes:** Brayan Galaz Esperguel, Nicolás Aliaga. Sebastian Pinto.

**Sección**:001D.

**Fecha**: 21/08/2024.

**Datos del documento**

**Histórico de Revisiones**

| Versión | Fecha | Descripción/cambio | Autor |
| --- | --- | --- | --- |
| 1.0 | 19/08 | Inicio informe | Product Owner |
| 1.1 | 23/08 | Ajustes en el informe | Product Owner |
| 1.2 | 13/10 | Entrega 2 | Product Owner |

**Información del Proyecto**

| Organización | Duoc UC |
| --- | --- |
| Sección | 001D |
| Proyecto (Nombre) | SafeKids |
| Fecha de Inicio | 19/08/2024 |
| Fecha de Término | XXX |
| Patrocinador principal | Munistgo |
| Docente | Mariluz Alejandra Rodríguez |

**Integrantes**

| Sección | Nombre | Correo |
| --- | --- | --- |
| **001D** | **Sebastián Pinto** | **seba.pinto@duocuc.cl** |
| **001D** | **Brayan Galaz** | **br.galaz@duocuc.cl** |
| **001D** | **Nicolás Aliaga** | **ni.aliaga@duocuc.cl** |

**Índice del Proyecto APT**

[**1. Introducción/Abstract**](#_heading=h.7jg1tyhbetzb) **4**

[1.1 Descripción del proyecto APT](#_heading=h.3l00yccmxul4) 4

[1.1 Description of the APT project](#_heading=h.m1o0p2kk7qom) 4

[1.2 Justificación de la Relevancia del Proyecto SafeKIDS](#_heading=h.pwq0ehlxln87) 4

[1.2 Justification of the Relevance of the SafeKIDS Project](#_heading=h.aksjspiiwgy9) 5

[**2. Relación del Proyecto APT con el Perfil de Egreso**](#_heading=h.ilo2wjochhit) **5**

[2.1 Competencias del perfil de egreso relacionadas](#_heading=h.q8ur9847ofao) 5

[2.2 Relación del proyecto con tus intereses profesionales](#_heading=h.8taf2atmvrqr) 5

[**3. Factibilidad del Proyecto**](#_heading=h.u6j8u01ko731) **6**

[**4. Análisis de las Competencias**](#_heading=h.w044tce5311m) **6**

[**5. Objetivos del Proyecto**](#_heading=h.gwrpy68i2ujy) **7**

[5.1 Objetivos Generales](#_heading=h.a3qzp1od522u) 7

[5.2 Objetivos Específicos](#_heading=h.afrv3c3q6ch6) 7

[5.3 Usuarios Finales](#_heading=h.m4qktmwerhqo) 8

[**6. Metodología: “SafeKIDS” METODOLOGÍA SCRUM – PLANNING POKER**](#_heading=h.3ri372vzlrs7) **8**

[6.1 Propuesta de Metodología de Trabajo](#_heading=h.n12wcks7k2ve) 8

[6.2 Definición de Roles](#_heading=h.bdmbi23l2pzu) 9

[6.3 Épicas para el Proyecto “SafeKIDS”](#_heading=h.tuy6whuxi0z8) 9

[6.4 Definición de Historias de Usuario](#_heading=h.4g0f2nspdwer) 9

[6.5 Product Backlog del Proyecto SafeKIDS](#_heading=h.57xibuhzivn) 10

[6.5 Planificación de Sprints](#_heading=h.s52s7sp67q58) 10

[6.6 Etapas del Proyecto](#_heading=h.rpba5o8s8il7) 15

[**7. Plan de trabajo**](#_heading=h.whsxmwhyhduv) **16**

[7.1 Descripción del plan de trabajo](#_heading=h.djkfxxw3gr9u) 16

[7.2 Tabla de plan de trabajo](#_heading=h.6qcnb4dykua8) 16

[**8. Evidencias**](#_heading=h.6kz0qc6h6ze4) **17**

[8.1 Determinación de las Evidencias](#_heading=h.rd8ipv5lmv16) 17

[8.2 Justificación de las Evidencias](#_heading=h.6l8wduv5w7dg) 18

[**9. Tecnologías Utilizadas**](#_heading=h.vjew3q3zar85) **18**

[**10. Carta Gantt Actualizada (11/10/2024)**](#_heading=h.vjew3q3zar85) **23**

[**11. Mockups**](#_heading=h.vjew3q3zar85) **24**

[11.1 Vista Apoderado](#_heading=h.qyxc40sdzil) 24

[11.2 Vista Conductor](#_heading=h.7mid5fivichr) 27

[11.3 Vista Administrador Principal](#_heading=h.mevf435csx5a) 29

[11.4 Vista Administrador Corriente](#_heading=h.1gypeshxod5p) 31

[**12. Diagrama Base de Datos**](#_heading=h.2u1j6yic815j) **32**

[**13. Diagrama de arquitectura de alto nivel**](#_heading=h.7rn3yybxl87q) **34**

[13.1 Herramientas en el diagrama](#_heading=h.k5ch5td9uqw4) 34

[13.2 Descripción de las Relaciones del Stack Tecnológico](#_heading=h.pt5da83t21h7) 36

[**14. Evidencias del Desarrollo**](#_heading=h.7rn3yybxl87q) **37**

[14.1 Código Inicio Web](#_heading=h.doi4crulyaxu) 37

[14.2 Código Login Móvil](#_heading=h.1fkz33ci8xge) 38

[**15. Conclusiones**](#_heading=h.7rn3yybxl87q) **39**

# 1. Introducción/Abstract

## 1.1 Descripción del proyecto APT

El proyecto APT denominado ”SafeKIDS” surge de la necesidad de garantizar la seguridad de los estudiantes durante su trayecto diario desde el hogar hasta la escuela, buscando desarrollar un sistema integral de seguimiento en tiempo real que permita registrar y monitorear de manera eficiente y segura el trayecto de los estudiantes desde sus hogares hasta la escuela. Esto se logrará mediante el diseño e implementación de una aplicación móvil para conductores de furgones escolares, que les permita escanear códigos QR únicos asociados a cada estudiante al momento de su embarque. Estos datos se almacenarán en una base de datos centralizada, accesible a través de una página web para padres y personal escolar.

## 1.1 Description of the APT project

The APT project called "SafeKIDS" comes from the need to guarantee the safety of students during their daily journey from home to school, seeking to develop a comprehensive real-time monitoring system that allows the efficient and safe recording and monitoring of students' journey from their homes to school. This will be achieved through the design and implementation of a mobile application for school van drivers, which allows them to scan unique QR codes associated with each student at the time of boarding. This data will be stored in a centralized database, accessible through a website for parents and school staff.

## **1.2 Justificación de la Relevancia del Proyecto SafeKIDS**

El proyecto SafeKIDS es relevante debido a la creciente inseguridad en el país y la poca información relacionada al traslado de los estudiantes durante su trayecto diario hacia la escuela. En un contexto donde la seguridad infantil es una prioridad para padres y autoridades escolares, SafeKIDS ofrece una solución tecnológica que responde a estas preocupaciones mediante un sistema de seguimiento y monitoreo en tiempo real, abordando la falta de un mecanismo eficaz y seguro para supervisar el transporte escolar.

SafeKIDS no solo garantiza que los estudiantes lleguen de manera segura a sus destinos al mantener registro de quien sube al transporte, los antecedentes del conductor y la ubicación del vehículo, sino que también proporciona tranquilidad a los padres.

## **1.2 Justification of the Relevance of the SafeKIDS Project**

The SafeKIDS project is relevant due to the growing insecurity in the country and the lack of information related to the transfer of students during their daily commute to school. In a context where child safety is a priority for parents and school authorities, SafeKIDS offers a technological solution that responds to these concerns through a real-time tracking and monitoring system, addressing the lack of an effective and safe mechanism to monitor school transport.

SafeKIDS not only ensures that students arrive safely at their destinations by keeping track of who gets on the transport, the driver's background, and the vehicle's location, but it also provides peace of mind for parents.

# 2. Relación del Proyecto APT con el Perfil de Egreso

## **2.1 Competencias del perfil de egreso relacionadas**

Este proyecto es fundamental para el campo de la Ingeniería en Informática, implicando el diseño y desarrollo de sistemas de software complejos, la gestión de proyectos informáticos, la integración de tecnologías, programación de plataformas móviles y web, y la aplicación de medidas de seguridad informática.

## **2.2 Relación del proyecto con tus intereses profesionales**

El proyecto APT se alinea perfectamente con nuestros intereses para el perfil de egreso de Ingeniería en Informática al requerir habilidades en el diseño y desarrollo de sistemas complejos, integración de tecnologías móviles y web, seguridad informática y gestión de proyectos informáticos. Además, proporcionándonos la oportunidad de mejorar en los campos anteriormente mencionados mientras que innovamos con una propuesta de valor indispensable para el sistema actual de transporte estudiantil.

# 3. Factibilidad del Proyecto

El Proyecto APT es realizable dentro del semestre en el tiempo asignado debido a la su duración y la distribución de horas asignadas a la asignatura (CAPSTONE), lo que permite dedicar el tiempo necesario para cada etapa del desarrollo programación. Además, los materiales requeridos, como software de programación y acceso a tecnologías móviles y web, son ampliamente accesibles. Factores externos que facilitan su desarrollo incluyen el acceso a recursos en línea, bibliotecas de código abierto y la disponibilidad de documentación técnica. Sin embargo, posibles factores externos que podrían dificultar su desarrollo podrían ser problemas de conectividad a internet o cambios en los requisitos del proyecto. Estos podrían solucionarse manteniendo una comunicación abierta con el equipo y anticipando posibles contratiempos mediante la planificación de contingencias para los posibles riesgos.

# 4. Análisis de las Competencias

Entre las empresas, la competencia es un factor clave para determinar quién será más exitoso, ya que por parte de los clientes se evalúa continuamente quién le brindará más beneficios. Se analizarán las competencias directas del proyecto, identificando tanto sus fortalezas como debilidades. Con este enfoque, se logrará tomar las mejores decisiones para fortalecer el proyecto y ser el más competitivo del mercado.

| Mi Furgón (<https://www.kuantum.cl/mifurgon.html>) | Skole  (<https://skole.cl>) |
| --- | --- |
| **Ventajas:**  **Información del trayecto:** Seguimiento durante todo el trayecto del transporte.  **Notificaciones automáticas:** Notificaciones sobre cambios de hora, inasistencias, atrasos, etc.  **Interfaz Amigable:** Interfaz sencilla para el usuario, con la menor cantidad de interacción al volante automatizando la mayoría de sus funciones.  **Desventajas:**  **Proyecto de pago:** Accesible solo para los centros de educación que lo adquieren mediante un pago. | **Ventajas:**  **Gestión de Pasajeros y Pagos**: Permite conocer la cantidad de pasajeros y el estado de pago de los apoderados de manera sencilla.  **Comunicación Efectiva**: Facilita la notificación y comunicación entre transportistas y apoderados para coordinar la recogida y cualquier cambio en la rutina.  **Navegación fácil**: Utiliza GPS para encontrar y seguir rutas, mejorando la eficiencia y puntualidad.  **Información en Tiempo Real**: Mantiene a los apoderados informados sobre el origen y destino de sus hijos, proporcionando mayor seguridad.  **Desventajas:**  **Aplicación de paga:** Accesible solo realizando un pago mensual.  **Enfoque al conductor:** Está enfocada para contratar conductores de bus particular y no asociados al centro de educación. |

Para que podamos cumplir con el objetivo de ser el proyecto más competitivo, cumpliremos con beneficios que ya otorgan la competencia y agregando beneficios únicos en el producto, tales como:

* Interfaz amigable para todos los usuarios
* Información del trayecto
* Notificaciones en caso de algún inconveniente
* Información sobre el conductor y vehículo
* Ver en tiempo real la ubicación en todo momento

Con todas estas características, el proyecto no sólo proporcionará las ventajas ya ofrecidas por la competencia, sino que también proporcionará un valor añadido único. Esto permitirá a los usuarios disfrutar de una experiencia más segura, eficiente y confiable.

# 5. Objetivos del Proyecto

## **5.1 Objetivos Generales**

El objetivo de este proyecto es desarrollar un sistema de seguimiento en tiempo real denominado "SafeKIDS" que garantice la seguridad de los estudiantes durante su trayecto diario desde el hogar hasta la escuela, mediante la implementación de una aplicación móvil para conductores de furgones escolares y una plataforma web accesible para padres y personal escolar. Esta herramienta cuenta con un sistema de registro para los estudiantes que abordan al vehículo mediante un código QR el cual deberá ser escaneado tanto al subir como al bajar del vehículo.

## **5.2 Objetivos Específicos**

* **Diseñar y desarrollar una aplicación móvil** para conductores de furgones escolares que permita escanear códigos QR únicos asociados a cada estudiante, registrando su embarque en el transporte escolar y proporcione información del conductor y vehículo a los apoderados de los alumnos además de mantener un seguimiento en tiempo real del vehículo.
* **Implementar una base de datos centralizada** que almacene de manera segura los registros de los estudiantes escaneados, incluyendo detalles del conductor, del vehículo y la ubicación del vehículo en tiempo real.
* **Desarrollar una plataforma web** accesible para los usuarios administradores del sistema, donde los usuarios permitidos luego de loguearse puedan crear, modificar y eliminar (CRUD) conductores, furgones y alumnos.
* **Integrar medidas de seguridad informática** que garanticen la protección de los datos personales de los estudiantes y la confidencialidad de la información almacenada en el sistema.
* **Realizar pruebas y validaciones** del sistema para asegurar su correcto funcionamiento y usabilidad, involucrando a los usuarios finales (conductores, padres y usuarios administradores registrados) en el proceso de mejora continua.
* **Capacitar a los usuarios finales** (conductores de furgones escolares) en el uso de la aplicación móvil y a los padres en la utilización de la plataforma web, para garantizar una adopción exitosa del sistema SafeKIDS.

## **5.3 Usuarios Finales**

1. **Conductores de furgones escolares**: Utilizarán la aplicación móvil para escanear los códigos QR de los estudiantes al subir y bajar del vehículo, registrar la información de los viajes.
2. **Padres o apoderados**: Utilizarán la aplicación móvil para recibir notificaciones, ver la información del conductor y vehículo, y seguir en tiempo real el trayecto de sus hijos desde el hogar hasta la escuela y viceversa.
3. **Usuarios administradores registrados (personal autorizado de la escuela)**: Accederán a la plataforma web para realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) en relación con conductores, vehículos y estudiantes. Este acceso estará restringido únicamente a personal autorizado para garantizar la seguridad y confidencialidad de la información.

# 

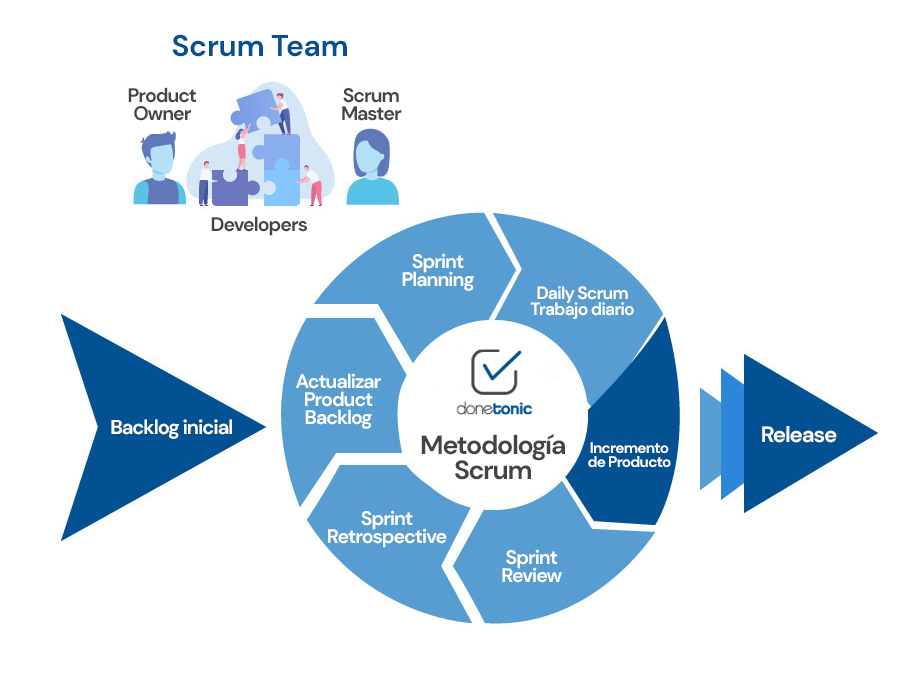
# 

# 

# 6. Metodología: “SafeKIDS” METODOLOGÍA SCRUM – PLANNING POKER

## **6.1 Propuesta de Metodología de Trabajo**

Para la implementación del proyecto SafeKids hemos seleccionado la metodología ágil “**SCRUM**” debido a su flexibilidad durante el proyecto y su enfoque comunicativo lo cual es útil para realizar en plazos cortos y permite solucionar errores con facilidad, otro de los motivos fue por su familiaridad con el equipo desarrollador al ya haber desarrollado soluciones con dicha metodología.

****

**“SafeKIDS”**

**METODOLOGÍA SCRUM – PLANNING POKER**

## **6.2 Definición de Roles**

| **Rol** | **Nombre de los Integrantes** | **Responsabilidades** |
| --- | --- | --- |
| Product Owner | Nicolás Aliaga | Encargado de tomar todos los requisitos del cliente y priorizar el backlog del producto. |
| Scrum Máster | Brayan Galaz | Encargado de facilitar el proceso Scrum, eliminar obstáculos y asegurar la eficiencia del equipo. |
| Desarrollador | Sebastián Pinto | Encargado de programar las funcionalidades del sistema SafeKIDS. |

## **6.3 Épicas para el Proyecto “SafeKIDS”**

| **ID** | **ÉpicaS** |
| --- | --- |
| E01 | Registro y Gestión de Usuarios |
| E02 | Monitoreo en Tiempo Real |
| E03 | Gestión de Rutas y Escaneo QR |
| E04 | Notificaciones y Alertas |
| E05 | Gestión de Antecedentes del Conductor |
| E06 | Plataforma Web para Gestión |

**Priorización de Épicas**

La priorización de las épicas, se utilizó la técnica de **Planning Poker**, Los resultados fueron los siguientes:

| **ID** | **Épica** | **Priorización** |
| --- | --- | --- |
| E01 | Registro y Gestión de Usuarios | 85 |
| E03 | Gestión de Rutas y Escaneo QR | 80 |
| E02 | Monitoreo en Tiempo Real | 75 |
| E04 | Notificaciones y Alertas | 70 |
| E05 | Gestión de Antecedentes del Conductor | 65 |
| E06 | Plataforma Web para Gestión | 60 |

## **6.4 Definición de Historias de Usuario**

| **ID** | **Historia de Usuario** |
| --- | --- |
| HU1 | Registro de Usuarios |
| HU2 | Inicio de sesión de Usuarios |
| HU3 | Visualización de datos de usuarios |
| HU4 | Escaneo de QR para identificación de niños |
| HU5 | Gestión de rutas y asignación de niños |
| HU6 | Envío de notificaciones a padres y conductores |
| HU7 | Gestión de antecedentes de los conductores |
| HU8 | Plataforma web para gestión de conductores y rutas |
| HU9 | Reporte de incidencias y alertas |

## **6.5 Product Backlog del Proyecto SafeKIDS**

**Técnica de Estimación**

Para estimar los puntos de las historias de usuario hemos utilizado la técnica de **Planning Poker**. Los puntos reflejan el esfuerzo y la complejidad asociados con cada historia.

| **ID** | **Historia de Usuario** |  | **Estimación** | **Prioridad** |
| --- | --- | --- | --- | --- |
| HU1 | Registro de Usuarios |  | 13 | 1 |
| HU2 | Inicio de sesión de Usuarios |  | 10 | 2 |
| HU3 | Visualización de datos de usuarios |  | 12 | 3 |
| HU4 | Escaneo de QR para identificación de niños |  | 15 | 4 |
| HU5 | Gestión de rutas y asignación de niños |  | 14 | 5 |
| HU6 | Envío de notificaciones a padres y conductores |  | 12 | 6 |
| HU7 | Gestión de antecedentes de los conductores |  | 13 | 7 |
| HU8 | Plataforma web para gestión de conductores y rutas |  | 14 | 8 |
| HU9 | Reporte de incidencias y alertas |  | 11 | 9 |

## **6.5 Planificación de Sprints**

| **Sprint 1: Planificación del Proyecto y Diseño de la Arquitectura** | | | |  |
| --- | --- | --- | --- | --- |
| **HU01: Registro de Usuarios** | |  |  |  |
| **Rol:** Padre, Conductor | |  |  |  |
| **Característica / Funcionalidad:** Registro en la aplicación móvil | | |  |  |
| **Número de escenario** | **Criterio de aceptación (Título)** | **Contexto** | **Evento** | **Resultado / Comportamiento esperado** |
| 1 | Registro exitoso | El usuario ingresa sus datos en el formulario de registro. | El usuario envía el formulario de registro. | El sistema debe enviar un correo de verificación y mostrar un mensaje de confirmación. |
| 2 | Registro con datos inválidos | El usuario ingresa datos incorrectos (por ejemplo, un correo inválido). | El usuario envía el formulario de registro. | El sistema debe mostrar un mensaje de error indicando los datos incorrectos. |
| 3 | Confirmación de correo | El usuario recibe un correo de verificación. | El usuario hace clic en el enlace de verificación. | El sistema debe activar la cuenta y permitir el inicio de sesión. |

| **HU02: Inicio de Sesión de Usuarios** | |  |  |  |
| --- | --- | --- | --- | --- |
| **Rol:** Padre, Conductor | |  |  |  |
| **Característica / Funcionalidad:** Inicio de sesión en la aplicación móvil | | |  |  |
| **Razón / Resultado:** Para que los usuarios puedan acceder a su cuenta. | | |  |  |
| **Número de escenario** | **Criterio de aceptación (Título)** | **Contexto** | **Evento** | **Resultado / Comportamiento esperado** |
| 1 | Inicio de sesión exitoso | El usuario ingresa su correo y contraseña correctos. | El usuario envía la solicitud de inicio de sesión. | El sistema debe autenticar al usuario y redirigirlo a la pantalla principal. |
| 2 | Inicio de sesión con datos incorrectos | El usuario ingresa un correo o contraseña incorrectos. | El usuario envía la solicitud de inicio de sesión. | El sistema debe mostrar un mensaje de error indicando datos incorrectos. |
| 3 | Recuperación de contraseña | El usuario ha olvidado su contraseña. | El usuario solicita la recuperación de contraseña. | El sistema debe enviar un correo con un enlace para restablecer la contraseña. |

| **Sprint 2: Desarrollo de Funcionalidades Principales** | | |  |  |
| --- | --- | --- | --- | --- |
| **HU04: Escaneo de QR para Identificación de Niños** | | |  |  |
| **Rol:**Conductor |  |  |  |  |
| **Característica / Funcionalidad:** Escaneo de QR para la identificación de niños | | | |  |
| **Razón / Resultado:** Para que los conductores puedan identificar a los niños mediante un código QR. | | | |  |
| **Número de escenario** | **Criterio de aceptación (Título)** | **Contexto** | **Evento** | **Resultado / Comportamiento esperado** |
| 1 | Escaneo exitoso | El conductor escanea el código QR del niño. | El conductor realiza el escaneo con la aplicación móvil. | El sistema debe mostrar la información del niño y su asignación de ruta. |
| 2 | Escaneo fallido | El conductor intenta escanear un código QR no válido. | El conductor realiza el escaneo con la aplicación móvil. | El sistema debe mostrar un mensaje de error indicando que el código no es válido. |
|  |  |  |  |  |
| **HU05: Gestión de Rutas y Asignación de Niños** | | |  |  |
| **Rol:** Administrador | |  |  |  |
| **Característica / Funcionalidad:** Gestión de rutas y asignación de niños | | |  |  |
| **Razón / Resultado:** Para que los administradores puedan asignar rutas a los niños. | | | |  |
| **Número de escenario** | **Criterio de aceptación (Título)** | **Contexto** | **Evento** | **Resultado / Comportamiento esperado** |
| 1 | Asignación exitosa de ruta | El administrador asigna una dirección del niño al sistema | El administrador selecciona un niño en el sistema para asignar la dirección | El sistema debe confirmar la asignación y actualizar la información. |
| 2 | Modificación de ruta | El administrador necesita cambiar la dirección asignada de un niño. | El administrador realiza un cambio en la dirección asignada. | El sistema debe enviar una notificación a los padres del niño informando del cambio. |
| **HU06: Envío de Notificaciones a Padres y Conductores** | | |  |  |
| **Rol:**Sistema |  |  |  |  |
| **Característica / Funcionalidad:** Envío de notificaciones a padres y conductores | | | |  |
| **Razón / Resultado:** Para mantener informados a los padres y conductores sobre el estado del viaje. | | | |  |
| **Número de escenario** | **Criterio de aceptación (Título)** | **Contexto** | **Evento** | **Resultado / Comportamiento esperado** |
| 1 | Notificación enviada correctamente | El sistema detecta una llegada tardía del autobús. | El sistema envía automáticamente una notificación a los padres y conductores. | Los padres y conductores deben recibir la notificación con la información correspondiente. |
| 2 | Fallo en el envío de notificación | El sistema intenta enviar una notificación, pero hay problemas de conectividad. | El sistema registra el intento de envío. | El sistema debe reintentar el envío y, si falla, registrar un error y notificar al administrador. |

| **Sprint 3: Integración y Gestión Administrativa** | | |  |  |
| --- | --- | --- | --- | --- |
| **HU07: Gestión de Antecedentes de los Conductores** | | |  |  |
| **Rol:** Administrador | |  |  |  |
| **Característica / Funcionalidad:** Gestión de antecedentes de los conductores | | | |  |
| **Razón / Resultado:** Para verificar la seguridad y antecedentes de los conductores. | | | |  |
| **Número de escenario** | **Criterio de aceptación (Título)** | **Contexto** | **Evento** | **Resultado / Comportamiento esperado** |
| 1 | Verificación de antecedentes completada | El administrador solicita la verificación de antecedentes de un conductor. | El administrador realiza la solicitud a través de la plataforma. | El sistema debe confirmar la recepción de la solicitud y mostrar los resultados una vez completada la verificación. |
| 2 | Error en la verificación de antecedentes | La plataforma encuentra un error al intentar verificar los antecedentes de un conductor. | El sistema intenta realizar la verificación de antecedentes. | El sistema debe notificar al administrador sobre el error y proporcionar opciones para resolverlo. |
| **HU08: Plataforma Web para Gestión de Conductores y Rutas** | | |  |  |
| **Rol:** Administrador | |  |  |  |
| **Característica / Funcionalidad:** Plataforma web para gestión de conductores y rutas | | | |  |
| **Razón / Resultado:** Para que los administradores puedan gestionar conductores y rutas de manera centralizada. | | | | |
| **Número de escenario** | **Criterio de aceptación (Título)** | **Contexto** | **Evento** | **Resultado / Comportamiento esperado** |
| 1 | Gestión exitosa de conductor | El administrador accede a la plataforma web para gestionar conductores. | El administrador realiza modificaciones a través de la interfaz web. | El sistema debe actualizar la información de los conductores y reflejar los cambios en tiempo real. |
| 2 | Gestión exitosa de ruta | El administrador accede a la plataforma web para gestionar rutas. | El administrador realiza modificaciones a través de la interfaz web. | El sistema debe actualizar la información de las rutas y notificar a los interesados sobre los cambios. |
| **HU09: Reporte de Incidencias y Alertas** | |  |  |  |
| **Rol:** Sistema, Administrador | |  |  |  |
| **Característica / Funcionalidad:** Reporte de incidencias y alertas | | |  |  |
| **Razón / Resultado:** Para informar a los administradores sobre incidencias y emergencias. | | | |  |
| **Número de escenario** | **Criterio de aceptación (Título)** | **Contexto** | **Evento** | **Resultado / Comportamiento esperado** |
| 1 | Incidencia reportada correctamente | Un usuario o sistema detecta una incidencia. | El usuario o sistema genera un reporte de incidencia. | El sistema debe registrar la incidencia y notificar al administrador y partes interesadas. |
| 2 | Alerta generada automáticamente | El sistema detecta un comportamiento inusual o emergencia. | El sistema genera automáticamente una alerta. | El sistema debe notificar a los administradores y activar los protocolos de emergencia establecidos. |

## **6.6 Etapas del Proyecto**

* **Planificación del proyecto**
* **Diseño de la arquitectura**
* **Desarrollo de la aplicación móvil**
* **Desarrollo de la plataforma web**
* **Integración y Pruebas de sistema**
* **Despliegue y puesta en producción**
* **Evaluación final y Cierre del proyecto**

| **Actividad** | **Fase 1** | | | | **Fase 2** | | | | | | | | | | | | **Fase 3** | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **S 1** | **S 2** | **S 3** | **S 4** | **S 5** | **S 6** | **S 7** | **S 8** | **S 9** | **S 10** | **S 11** | **S 12** | **S 13** | **S 14** | **S 15** | **S 16** | | **S 17** | **S 18** |
| Planificación del proyecto | **X** | **X** |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |
| Diseño de la arquitectura |  |  | **X** | **X** |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |
| Desarrollo de la aplicación móvil |  |  |  |  | **X** | **X** | **X** | **X** | **X** |  |  |  |  |  |  |  | |  |  |
| Desarrollo de la plataforma web |  |  |  |  |  |  |  |  |  | **X** | **X** | **X** |  |  |  |  | |  |  |
| Integración y Pruebas de sistema |  |  |  |  |  |  |  |  |  |  |  |  | **X** | **X** |  |  | |  |  |
| Despliegue y puesta en producción |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **X** | **X** | |  |  |
| Evaluación final y Cierre del proyecto |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | **X** | **X** |

# **7. Plan de trabajo**

## **7.1 Descripción del plan de trabajo**

El Plan de Trabajo para el proyecto SafeKIDS se divide en siete fases, distribuidas a lo largo de 18 semanas, siguiendo la metodología SCRUM y con la realización de sprints de 2 semanas. Cada fase se centra en un aspecto clave del desarrollo, desde la planificación inicial hasta la evaluación final y cierre del proyecto. Este enfoque estructurado permite un progreso continuo y eficiente, asegurando que cada etapa del proyecto se complete a tiempo y con la calidad requerida.

## **7.2 Tabla de plan de trabajo**

| **Fases** | **Semanas** | **Actividades** | **Entregables** |
| --- | --- | --- | --- |
| **Planificación del proyecto** | 1-2 | - Reunión de inicio del proyecto  - Análisis de requerimientos  - Creación del Product Backlog | Documentación de requerimientos  Product Backlog inicial |
| **Diseño de la arquitectura** | 3-4 | - Diseño detallado de la arquitectura del sistema  - Definición de la base de datos  - Diseño de UI/UX | Diseño de la arquitectura  Prototipo de UI/UX |
| **Desarrollo de la aplicación móvil** | 5-9 | - Implementación de la aplicación móvil  - Desarrollo de funcionalidades clave (escaneo QR, rutas, notificaciones, antecedentes de conductor)  - Pruebas unitarias y ajustes | Aplicación móvil funcional con características clave |
| **Desarrollo de la plataforma web** | 10-12 | - Implementación de la plataforma web  - Creación de CRUD para Gestión de conductores por parte del colegio  - Integración con la base de datos | Plataforma web funcional con monitoreo en tiempo real |
| **Integración y Pruebas de sistema** | 13-14 | - Integración completa de la aplicación móvil y la plataforma web  - Pruebas de sistema integradas  - Validación con usuarios finales | Sistema integrado listo para despliegue |
| **Despliegue y puesta en producción** | 15-16 | - Despliegue del sistema SafeKIDS en producción  - Capacitación a usuarios finales (conductores, padres, personal escolar | Sistema desplegado y usuarios capacitados |
| **Evaluación final y Cierre del proyecto** | 17-18 | - Revisión final y retroalimentación del proyecto  - Documentación completa del proyecto  - Reflexión y cierre del proyecto | Proyecto completado, documentado y cerrado |

# **8. Evidencias**

## **8.1 Determinación de las Evidencias**

**Documentación de Requerimientos y Product Backlog Inicial:**

* Reuniones de planificación y análisis de requerimientos.
* Product Backlog inicial con las funcionalidades y tareas priorizadas.

**Diseño de la Arquitectura y Prototipo de UI/UX:**

* Diagramas de la arquitectura del sistema (diagrama de clases, PERT, flujo de datos, etc.).
* Prototipo del diseño de la interfaz de usuario (mockups, wireframes).

**Aplicación Móvil Funcional con Características Clave:**

* Código fuente de la aplicación móvil con las funcionalidades clave implementadas (escaneo de QR, rutas, notificaciones, visualización de antecedentes del conductor).
* Resultados de pruebas unitarias y ajustes realizados.

**Plataforma Web Funcional con Monitoreo en Tiempo Real:**

* Código fuente de la plataforma web, incluyendo la implementación del CRUD para la gestión de conductores.
* Resultados de las pruebas de integración con la base de datos y la funcionalidad de monitoreo en tiempo real.

**Integración y Pruebas de sistema:**

* Resultados de las pruebas de sistema integradas.

**Despliegue y puesta en producción:**

* Registro del despliegue en producción del sistema SafeKIDS.
* Materiales y resultados de la capacitación a usuarios finales (conductores, padres y personal escolar mediante encuestas).
* **Documentación Completa del Proyecto:**
* Informe final del proyecto con toda la documentación técnica, resultados de pruebas, feedback de usuarios y reflexiones finales.
* Presentación del proyecto.

## **8.2 Justificación de las Evidencias**

La justificación de las evidencias seleccionadas responde a la necesidad de garantizar que cada etapa del Proyecto APT “SafeKIDS” esté adecuadamente documentada y validada, asegurando la entrega de un producto final que cumpla con los requisitos establecidos y que sea funcional, seguro y fácil de usar para los usuarios finales.

# **9. Tecnologías Utilizadas**

El desarrollo del proyecto SafeKIDS requerirá una combinación de tecnologías modernas que aseguren tanto la eficiencia como la seguridad en el seguimiento y monitoreo de los estudiantes. A continuación, se detallan las tecnologías principales que se emplearán:

## **1. Ionic con Angular (Aplicación Móvil)**

Ionic es un framework para el desarrollo de aplicaciones móviles híbridas, que se complementa con Angular para crear aplicaciones dinámicas y eficientes. Este será utilizado para el desarrollo de la aplicación móvil destinada a los conductores de furgones escolares y a los apoderados de los alumnos.

**Alternativa: React Native**

**Ventajas de React Native:** Utiliza JavaScript y React, lo que ofrece un rendimiento más cercano al nativo en comparación con aplicaciones híbridas y una experiencia de usuario más fluida.

**Desventajas frente a Ionic/Angular:** React Native está más orientado a aplicaciones móviles puramente nativas, lo que puede requerir más tiempo de desarrollo cuando se trata de portabilidad a diferentes plataformas web. Además, algunas funciones del sistema pueden requerir desarrollo nativo adicional (por ejemplo, módulos personalizados).

**Justificación de Ionic/Angular:** Ionic permite el desarrollo de aplicaciones móviles híbridas, lo que significa que una única base de código se puede ejecutar en iOS, Android y como una aplicación web progresiva (PWA). Esto optimiza los tiempos de desarrollo y mantenimiento, dado que Angular es un framework de alto nivel con características como el enlace de datos bidireccional y componentes reutilizables, lo que mejora la eficiencia de la aplicación.

**Alternativa: Flutter**

**Ventajas de Flutter:** Ofrece un rendimiento muy alto con su motor gráfico nativo basado en Dart, permitiendo interfaces de usuario personalizadas y fluidas. Además, permite el desarrollo para múltiples plataformas (iOS, Android, web, desktop).

**Desventajas frente a Ionic/Angular:** El ecosistema de Flutter es más reciente y aún carece de la madurez en cuanto a componentes y librerías adicionales comparado con Ionic y Angular. El aprendizaje de Dart puede ser un obstáculo para desarrolladores con experiencia en JavaScript.

**Justificación de Ionic/Angular:** Ionic utiliza tecnologías web estándar como HTML, CSS y JavaScript, lo que facilita el uso para desarrolladores con experiencia en tecnologías web y no requiere el aprendizaje de un lenguaje adicional como Dart. También se beneficia de la sólida base de Angular, lo que facilita la integración de nuevas funcionalidades y el mantenimiento a largo plazo.

## **2. HTML, CSS y JavaScript (Página Web)**

La página web, accesible solo para personal autorizado para modificar los vehículos alumnos y conductores autorizados, será desarrollada utilizando HTML, CSS y JavaScript. Estos son los estándares fundamentales para la creación de interfaces web interactivas y atractivas:

* **HTML (HyperText Markup Language):** Utilizado para estructurar el contenido de la página web.
* **CSS (Cascading Style Sheets):** Empleado para diseñar y dar estilo a la página, asegurando que la interfaz sea atractiva y fácil de usar.
* **JavaScript:** Proporciona interactividad en la página web, permitiendo realizar operaciones en el frontend sin necesidad de recargar la página.

**Alternativa: React**

**Ventajas de React:** React es una biblioteca JavaScript muy popular para construir interfaces de usuario con un enfoque basado en componentes. Ofrece un rendimiento muy alto y una comunidad activa.

**Desventajas frente a HTML/CSS/JS estándar:** React requiere una configuración más compleja (Webpack, JSX) y depende de bibliotecas externas para tareas como el enrutamiento y el manejo del estado (por ejemplo, React Router, Redux), lo que puede agregar complejidad adicional al proyecto.

**Justificación de HTML/CSS/JS estándar:** Utilizar HTML, CSS y JavaScript en su forma pura ofrece simplicidad y claridad. Para una aplicación web sencilla, sin necesidad de interactividad avanzada o manejo del estado a gran escala, las tecnologías estándar son más ligeras y rápidas de implementar, reduciendo la sobrecarga técnica y optimizando el rendimiento.

**Alternativa: Vue.js**

**Ventajas de Vue.js:** Vue es un framework progresivo que puede integrarse fácilmente en proyectos existentes y ofrece un rendimiento excelente con una curva de aprendizaje más suave que Angular.

**Desventajas frente a HTML/CSS/JS estándar:** Aunque es más simple que React o Angular, Vue sigue añadiendo una capa adicional de complejidad en comparación con el uso puro de HTML, CSS y JavaScript. Para proyectos donde no se necesiten dinámicas complejas o una estructura avanzada de componentes, esta complejidad puede ser innecesaria.

**Justificación de HTML/CSS/JS estándar:** Las tecnologías puras son más que suficientes para crear una página web estática o con interactividad ligera, especialmente si se combinan con una solución backend eficiente como Firebase para el manejo de datos. No es necesario implementar un framework adicional cuando las necesidades de la página web son más simples.

## **3. Base de Datos**

**Base de datos NoSQL:** La base de datos centralizada estará gestionada por Firebase Firestore, un sistema de base de datos NoSQL en tiempo real. Esta base de datos almacenará los registros y datos de estudiantes, conductores y vehículos. Firebase Firestore permite realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) de manera eficiente, manteniendo la integridad de los datos y sincronizando automáticamente los cambios en tiempo real entre la aplicación móvil y la plataforma web.

**Alternativa: MongoDB**

**Ventajas de MongoDB:** Es una base de datos NoSQL muy popular que permite almacenamiento y consulta de grandes volúmenes de datos, y ofrece una flexibilidad en la estructura de datos similar a Firestore. Además, está bien optimizado para integrarse con soluciones empresariales y arquitecturas basadas en microservicios.

**Desventajas frente a Firebase Firestore:** Aunque MongoDB es muy poderoso, requiere una infraestructura propia o en la nube, lo que añade complejidad en cuanto a configuración, escalabilidad y mantenimiento. También carece de características integradas de tiempo real y autenticación nativa como las que ofrece Firebase.

**Justificación de Firebase Firestore:** Es una solución completamente gestionada, lo que significa que no hay necesidad de configurar o administrar servidores. Además, tiene características **nativas de tiempo real** y se integra perfectamente con Firebase Authentication, lo que simplifica enormemente la implementación de seguridad y sincronización entre dispositivos en tiempo real. Esto lo convierte en una opción más práctica para aplicaciones móviles que necesitan actualizaciones instantáneas de datos.

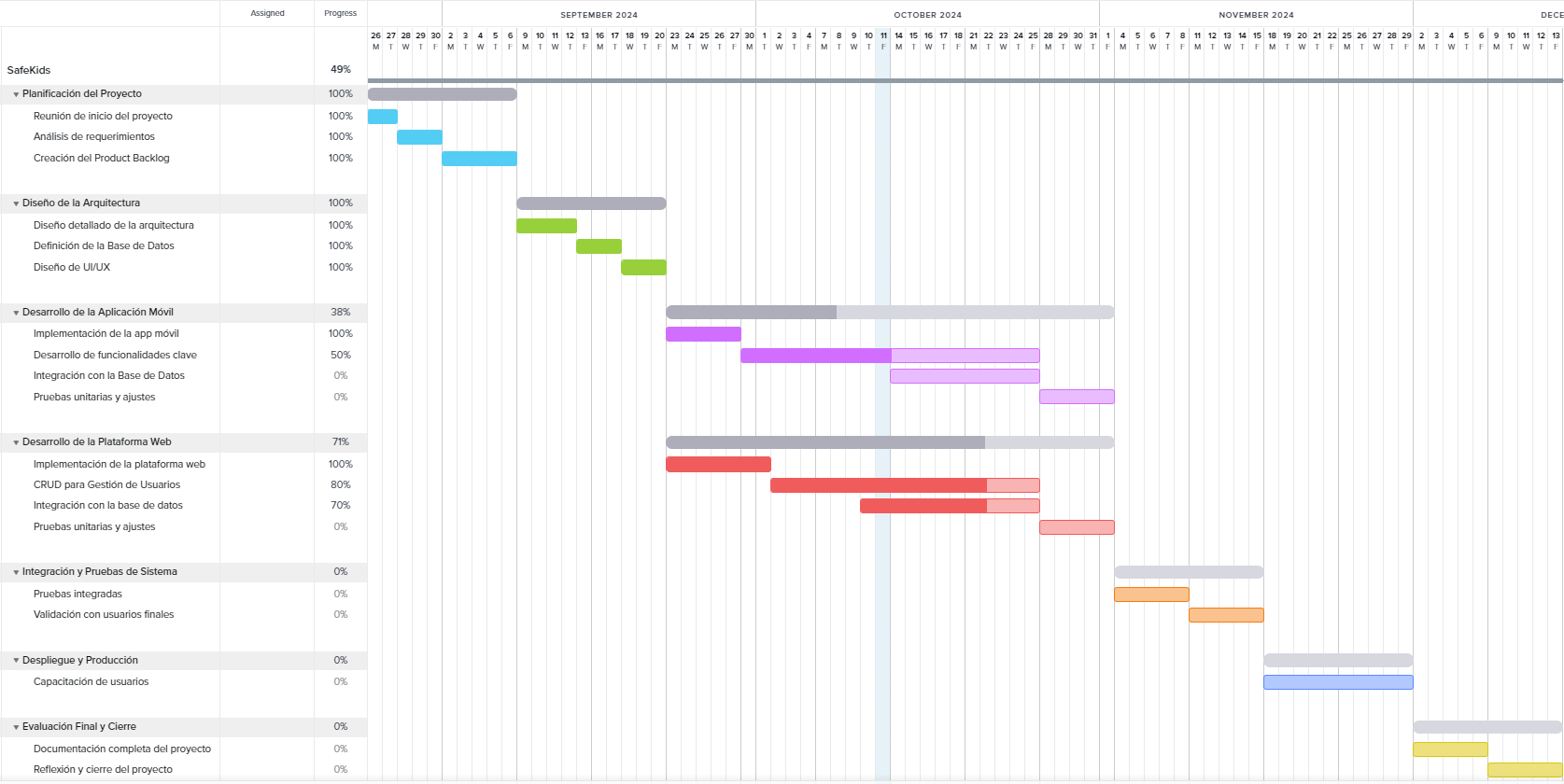
**Alternativa: MySQL**

**Ventajas de MySQL:** Es una base de datos relacional madura y ampliamente utilizada, ideal para proyectos que requieren relaciones entre datos y transacciones ACID. Ofrece una estructura de datos rígida que asegura la integridad y consistencia de los datos.

**Desventajas frente a Firebase Firestore:** MySQL, al ser relacional, requiere un esquema rígido que puede ser limitante para aplicaciones móviles que necesitan flexibilidad en la estructura de los datos. Además, MySQL no es nativo en tiempo real, lo que añade complejidad si se necesita sincronización de datos en varias plataformas.

**Justificación de Firebase Firestore:** Firebase Firestore, al ser NoSQL, ofrece una mayor **flexibilidad en la estructura de los datos** y permite escalar horizontalmente de manera más eficiente. Para una aplicación que requiere manejar datos no estructurados o con cambios frecuentes en el esquema, Firestore es más adecuado. También su soporte nativo para actualizaciones en tiempo real proporciona una experiencia de usuario más dinámica y fluida.

# **10. Carta Gantt Actualizada (11/10/2024)**

****

A día de la fecha indicada, el desarrollo del proyecto SafeKIDS se encuentra alineado con el cronograma planificado en la carta Gantt. Todas las actividades se han cumplido conforme a las fechas establecidas, lo que refleja un avance constante y dentro de los plazos. Se espera que el progreso continúe de la misma manera, asegurando que las entregas futuras se realicen a tiempo y con la calidad prevista.

# 

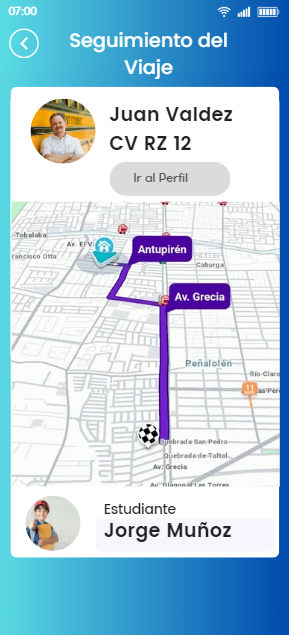
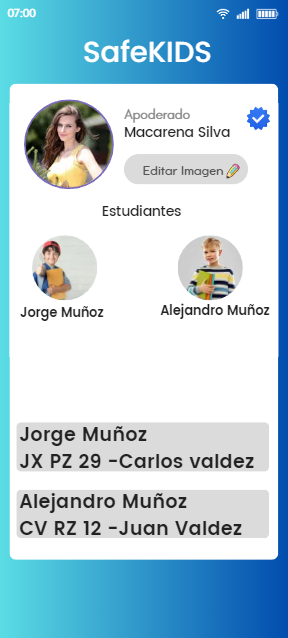
# **11. Mockups**

## **11.1 Vista Apoderado**

****

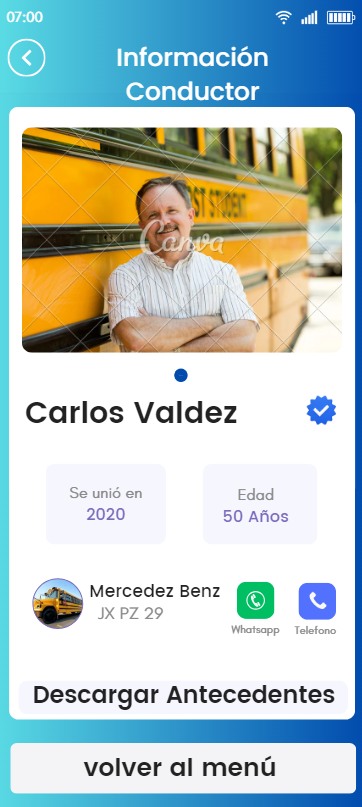
1. Vista en donde los apoderados van a poder ingresar con su correo y su contraseña a la plataforma móvil. Cuenta con las funcionalidades para, en caso de olvidar la contraseña, llevarlo a una pantalla para poder restablecer la contraseña, y en caso que no tenga una cuenta, llevarlo a la pantalla para registrarse, por último, el botón para iniciar sesión en caso que los datos sean correctos y estén registrados en el sistema.

2. Vista en donde los apoderados van a poder registrarse con todos los datos que se solicitan. Al apretar el botón, mandará un correo electrónico al sistema, y en caso que la institución dé el visto bueno a los datos, se registrará correctamente, mandando un correo indicando que ya se le ha registrado.

****

3. Vista que verán los apoderados la iniciar sesión correctamente. Se podrá editar la imagen de perfil del apoderado, seleccionando una imagen o tomando una en el momento. Se verán los alumnos asignados al apoderado, y en caso de haber viajes en curso, en la parte inferior se mostrarán los viajes actualmente activas, mostrando el nombre del alumno que va en el bus e información sobre el conductor y el bus.

4. Vista que se verá a apretar sobre algún viaje en curso, en donde se mostrará información básica del conductor, y un botón que llevaría al perfil completo del conductor. Se verá también en tiempo real en dónde se encuentra el alumno, y por último se vería la imagen del alumno y su nombre.



5. Vista que se verá al apretar en la parte inferior de la vista del seguimiento del viaje, en donde se encuentra la información del alumno. Se verá la información completa del alumno, como su nombre, dirección, el género, edad y curso actual. También se verá el nombre de su colegio junto con la dirección.

6. Vista que se verá al apretar en la parte superior de la vista del seguimiento del viaje, en donde se encuentra la información del conductor. Se verá la información completa del conductor, como su nombre, cuándo se unió al sistema, su edad, el bus vinculado a él, y dos botones para poder comunicarse con el conductor. Por último, se podrán solicitar los antecedentes en caso de ser requeridos.

## **11.2 Vista Conductor**

****

1. Vista en donde los conductores van a poder ingresar con su correo y su contraseña a la plataforma móvil. Cuenta con las funcionalidades para, en caso de olvidar la contraseña, llevarlo a una pantalla para poder restablecer la contraseña, y en caso que no tenga una cuenta, llevarlo a la pantalla para registrarse, por último, el botón para iniciar sesión en caso que los datos sean correctos y estén registrados en el sistema.

2. Vista que verán los conductores al iniciar sesión correctamente. Se podrá editar la imagen de perfil del conductor, seleccionando una imagen o tomando una en el momento. También se verán los datos básicos del vehículo asignado. Habrá un botón para iniciar un viaje, que llevará a la lista de los alumnos asignados al viaje. En la parte inferior estará la dirección del colegio y el nombre del mismo.

****

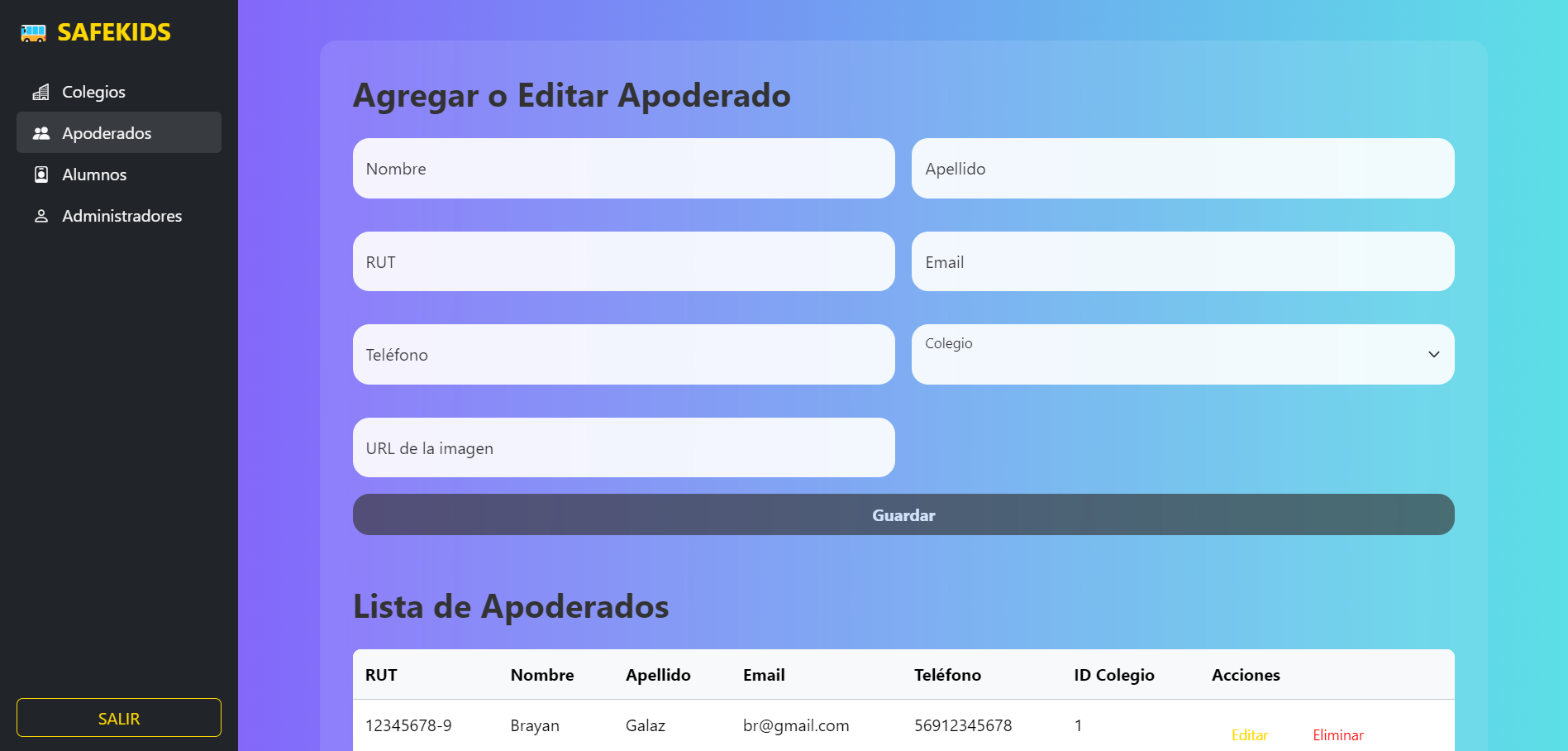
3. Vista que se verá al iniciar un viaje, en donde se verá la lista de estudiantes asignados al viaje, en donde se listará las imágenes de los alumnos y sus nombres. En la parte inferior, se verá el botón para escanear los códigos QR de los alumnos. En caso de escanear exitosamente uno, se pondrá el ticket en la parte derecha del respectivo alumno, y en caso de no escanear, quedará con una equis. Por último, el botón para poder finalizar el viaje cuando todos los alumnos hayan llegado a sus casas.

4. Vista que se verá al apretar una de las imágenes de la lista de los estudiantes, en donde se verá la información completa del alumno, como su nombre, dirección, género, edad y el curso actual. También se verá el apoderado designado y un botón para poder comunicarse en caso de alguna ocurrencia. Por último, el colegio asignado.

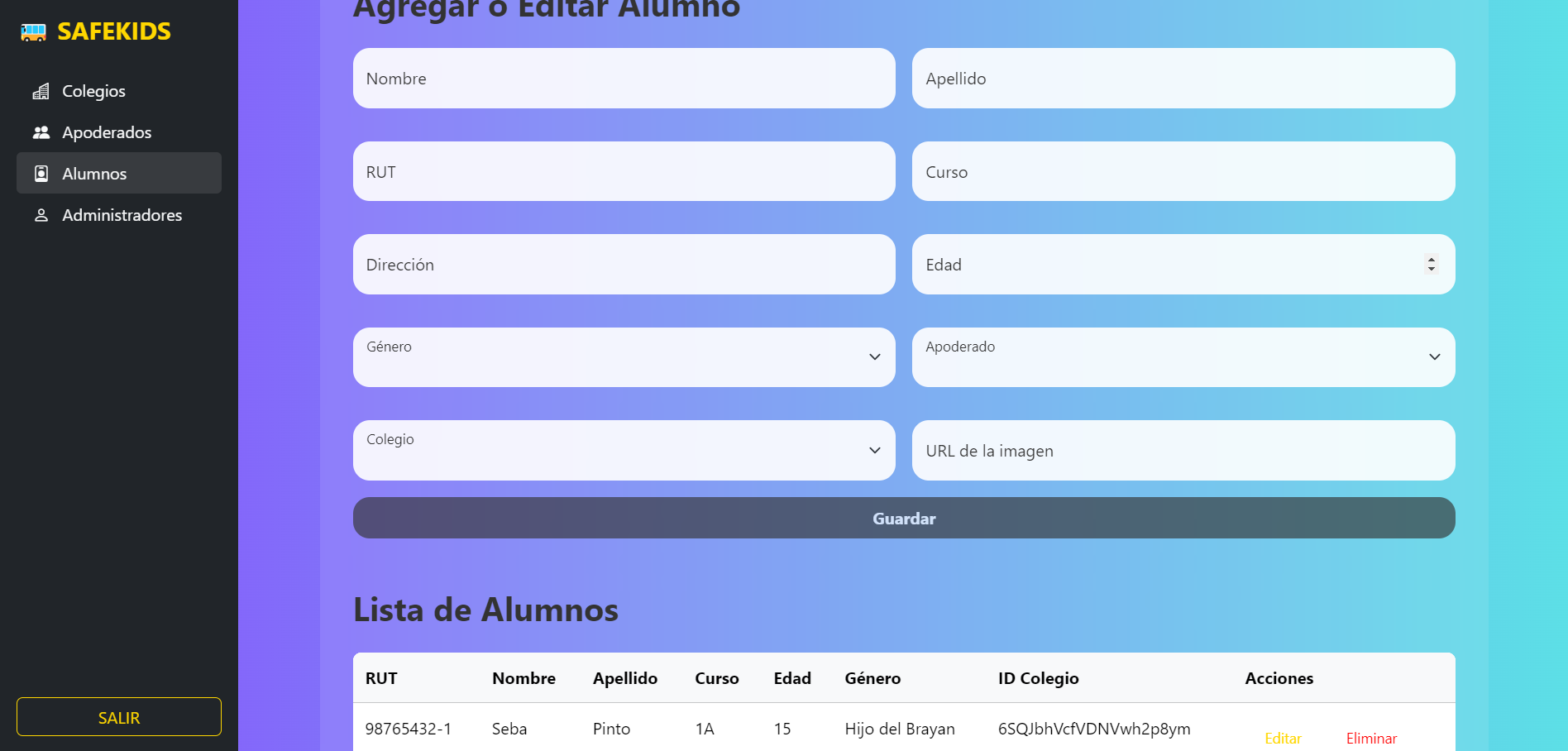
## **11.3 Vista Administrador Principal**

****

1. Vista para gestionar los colegios registrados, que incluye un panel lateral a la izquierda (que se encuentra en todas las vistas de este CRUD) con opciones de navegación como "Colegios", "Apoderados", "Alumnos" y "Administradores". En la parte superior, se muestra un formulario para agregar o editar los datos de un colegio, con campos para ingresar el nombre del colegio, dirección, email, teléfono, el nombre del director y el ID del colegio. En la parte inferior, se despliega una tabla con la lista de colegios registrados, que incluye la información de los colegios, junto con las opciones de "Editar" o "Eliminar" cada registro.

****

2. Vista para gestionar los apoderados registrados. En la parte superior, se muestra un formulario para agregar o editar los datos de un apoderado, con campos para ingresar el nombre, apellido, RUT, email, teléfono, el colegio asociado y el URL de la imagen. En la parte inferior, se despliega una tabla con la lista de apoderados registrados, que incluye la información de los apoderados, junto con las opciones de "Editar" o "Eliminar" cada registro.

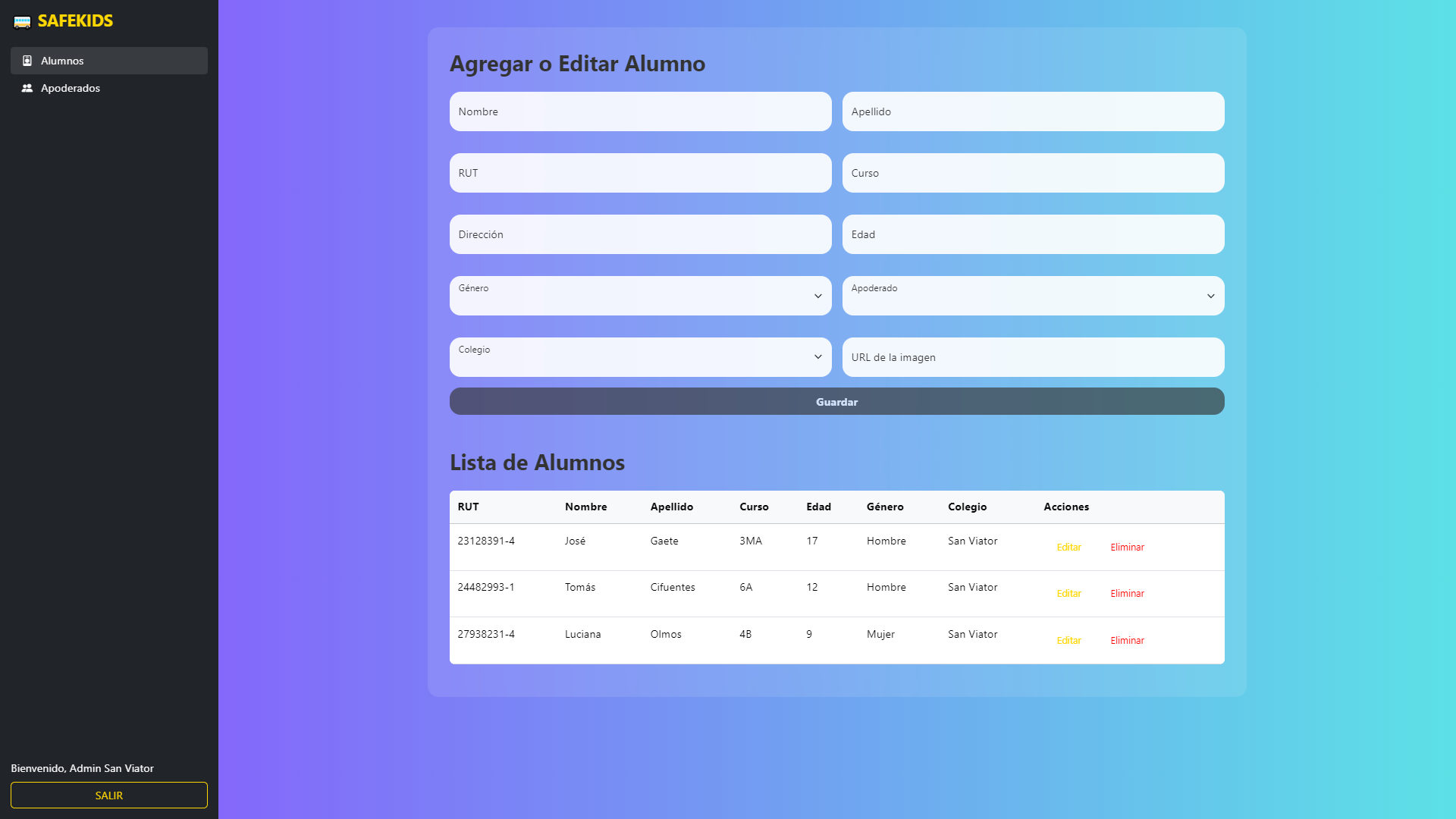
****

3. Vista para gestionar los alumnos registrados. En la parte superior, se muestra un formulario para agregar o editar los datos de un alumno, con campos para ingresar el nombre, apellido, RUT, curso, dirección, edad, género, apoderado asociado, colegio asociado y el URL de la imagen. En la parte inferior, se despliega una tabla con la lista de alumnos registrados, que incluye la información de los alumnos, junto con las opciones de "Editar" o "Eliminar" cada registro.

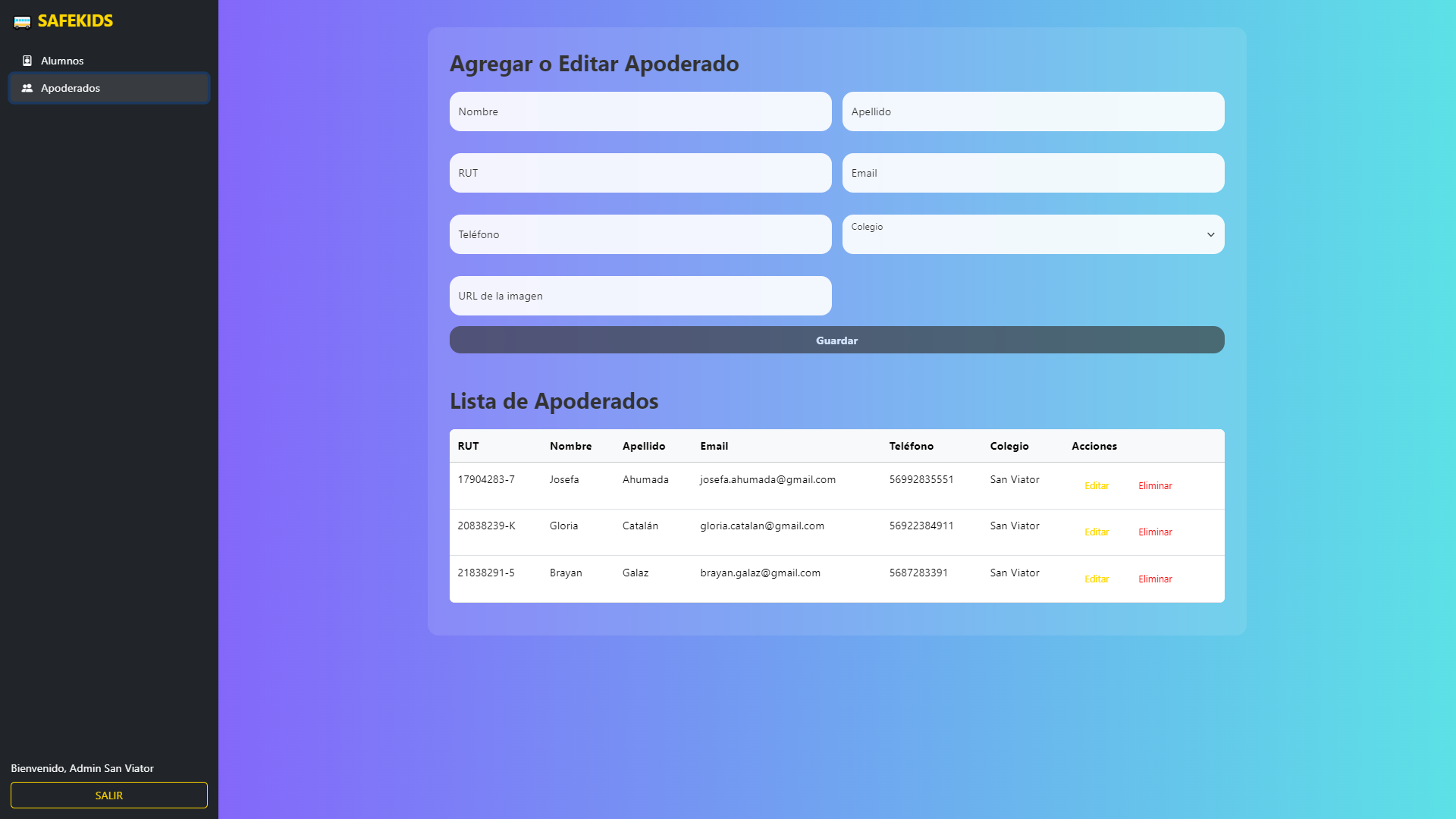
****

4. Vista para gestionar los administradores en el sistema. En la parte superior, se muestra un formulario para agregar o editar los datos de un administrador, con campos para ingresar el nombre, apellido, RUT, email, teléfono y el colegio asociado. En la parte inferior, se despliega una tabla con la lista de administradores en el sistema, que incluye la información de los administradores, junto con las opciones de "Editar" o "Eliminar" cada registro.

## **11.4 Vista Administrador Corriente**

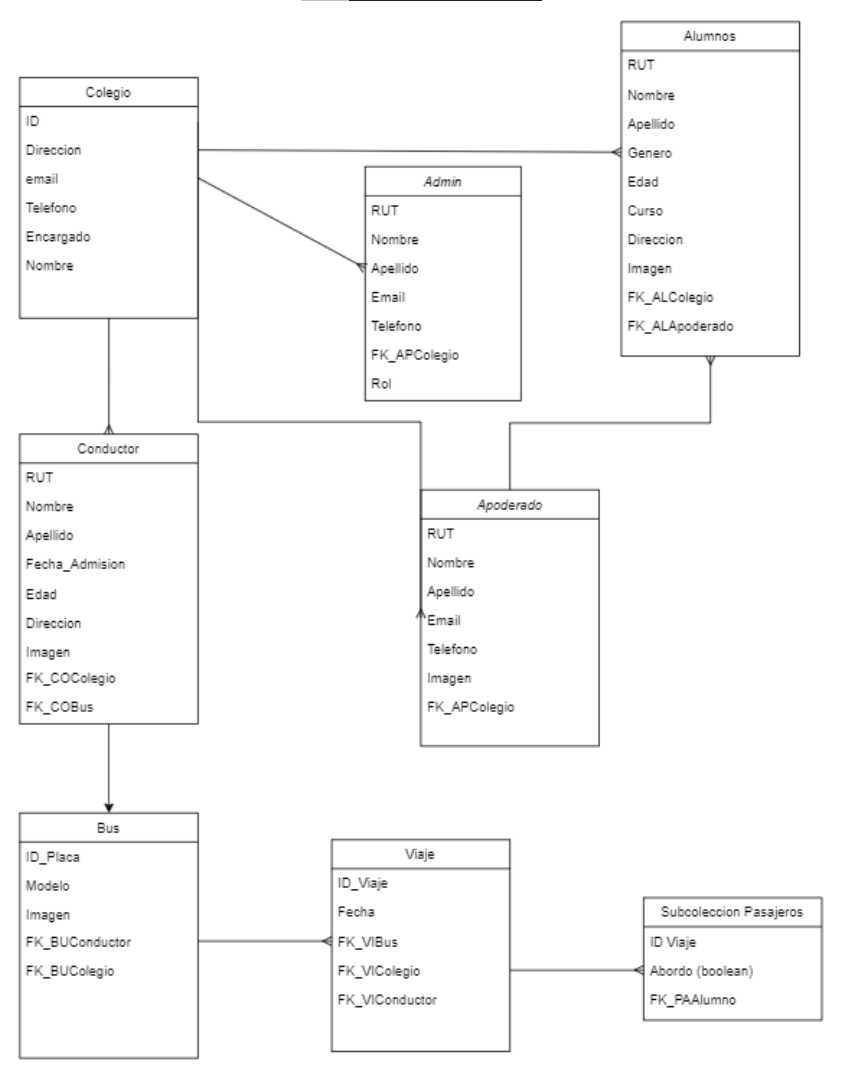
****

1. Vista para gestionar los alumnos registrados del colegio vinculado al administrador, que incluye un panel lateral a la izquierda (que se encuentran las vistas disponibles) con opciones de navegación como "Apoderados" y "Alumnos". En la parte superior, se muestra un formulario para agregar o editar los datos de un alumno, con campos para ingresar el nombre, apellido, RUT, curso, dirección, edad, género, apoderado asociado, colegio asociado y el URL de la imagen. En la parte inferior, se despliega una tabla con la lista de alumnos registrados, que incluye la información de los alumnos, junto con las opciones de "Editar" o "Eliminar" cada registro.

****

Vista para gestionar los apoderados registrados del colegio vinculado al administrador. En la parte superior, se muestra un formulario para agregar o editar los datos de un apoderado, con campos para ingresar el nombre, apellido, RUT, email, teléfono, el colegio asociado y el URL de la imagen. En la parte inferior, se despliega una tabla con la lista de apoderados registrados, que incluye la información de los apoderados, junto con las opciones de "Editar" o "Eliminar" cada registro.

# **12. Diagrama Base de Datos**

****

El sistema de gestión escolar implementado utiliza Firestore como base de datos NoSQL, aprovechando su flexibilidad y escalabilidad. La estructura de datos se organiza en torno a la colección principal del Colegio, que actúa como el punto central de referencia para las demás entidades. Esta arquitectura ofrece un diseño modular que permite una fácil expansión del sistema según las necesidades del centro educativo.

El sistema se compone de las siguientes colecciones principales:

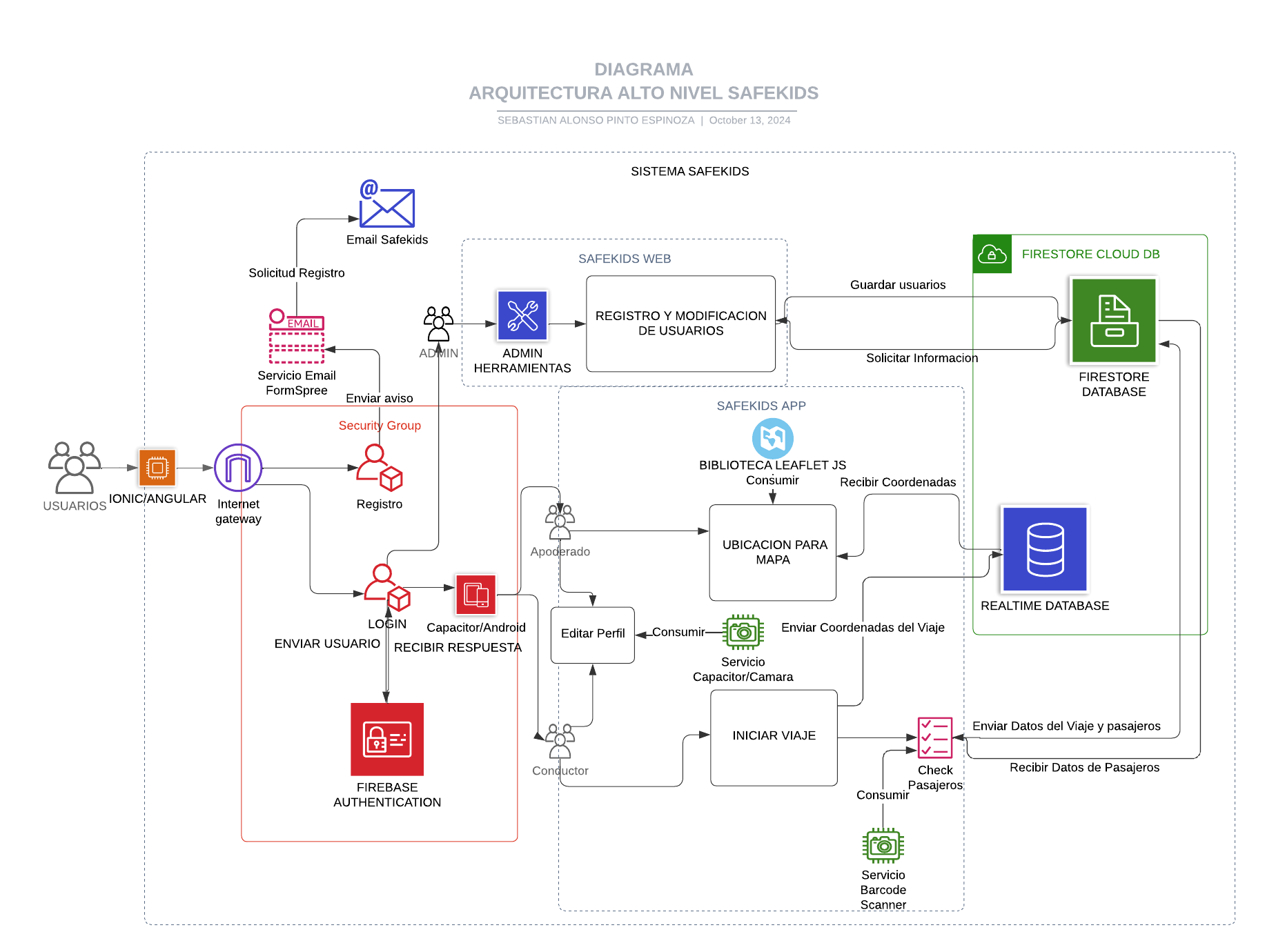
* **Colegio:** Almacena la información institucional básica como dirección, email, teléfono, encargado y nombre de la institución.
* **Alumnos:** Contiene los registros de los estudiantes, con campos como nombre, apellido, edad, curso, dirección, imagen y referencias al Colegio y al Apoderado.
* **Apoderado:** Guarda los datos de los tutores legales de los alumnos, incluyendo nombre, apellido, email, teléfono e imagen. Además, está vinculado al Colegio.
* **Conductor:** Registra la información de los choferes del transporte escolar, incluyendo nombre, apellido, fecha de admisión, edad, dirección, imagen, y referencias al Colegio y al Bus que manejan.
* **Buses:** Mantiene los detalles de los vehículos de transporte como modelo, imagen y la relación con el Conductor y el Colegio.
* **Viajes:** Almacena los recorridos realizados por los buses, incluyendo la fecha, el Bus y el Conductor asignados, así como una subcolección de Pasajeros.
* **Admin:** Contiene la información de los administradores de cada institución, con campos como nombre, apellido, email, teléfono y el rol dentro del Colegio.

Cada colección está relacionada de manera coherente mediante el uso de claves foráneas (FK) para mantener la integridad referencial y permitir consultas eficientes. Estas relaciones incluyen:

* Los Alumnos están vinculados a un Colegio y a un Apoderado, lo que asegura que cada estudiante esté asociado tanto a su tutor legal como a la institución en la que estudia.
* Los Apoderados también están asociados al Colegio, permitiendo gestionar su relación con la institución de forma centralizada.
* Los Conductores están relacionados con el Colegio y con los Buses que manejan, permitiendo un seguimiento detallado de quién conduce cada vehículo.
* Los Viajes están asociados tanto al Bus como al Conductor, y contienen una subcolección de Pasajeros, donde se lista a los Alumnos que viajan en cada recorrido.
* La colección de Admin se relaciona con el Colegio, permitiendo gestionar los roles administrativos y el control sobre la institución desde un único punto.

La implementación de esta arquitectura en Firestore ofrece ventajas significativas en términos de rendimiento y flexibilidad, permitiendo una fácil expansión del sistema según las necesidades futuras del centro educativo.

# **13. Diagrama de arquitectura de alto nivel**

****

Mediante el anterior diagrama podemos visualizar cada una de las herramientas utilizadas en el proyecto y cómo interactúan con el sistema Safekids ya sea para adquirir o enviar información por medio de servicios API, herramientas, plataformas Cloud o procesos internos.

## **13.1 Herramientas en el diagrama**

**1. Ionic**

Framework para desarrollar aplicaciones móviles híbridas utilizando tecnologías web como HTML, CSS y JavaScript. Permite crear aplicaciones que funcionan en múltiples plataformas (iOS, Android, web) con una sola base de código.

**2. Angular**

Descripción: Framework de desarrollo web para construir aplicaciones de una sola página (SPA). Proporciona una estructura robusta y herramientas para crear aplicaciones dinámicas y escalables.

**3. Firebase Authentication**

Descripción: Servicio de autenticación que permite a los desarrolladores gestionar la autenticación de usuarios de manera sencilla. Soporta múltiples métodos de inicio de sesión, como correo electrónico, Google, Facebook, entre otros.

**4. Firestore Cloud Database**

Descripción: Base de datos NoSQL en tiempo real que permite almacenar y sincronizar datos entre los clientes y la nube. Ideal para aplicaciones que requieren actualizaciones en tiempo real.

**5. Realtime Database**

Descripción: Otra opción de base de datos en Firebase que permite almacenar y sincronizar datos en tiempo real. Es útil para aplicaciones que necesitan reflejar cambios instantáneamente en la interfaz de usuario.

**6. Leaflet**

Descripción: Biblioteca de JavaScript para crear mapas interactivos. Permite integrar mapas en aplicaciones web y móviles, facilitando la visualización de datos geoespaciales.

**7. Capacitor**

Descripción: Plataforma que permite a los desarrolladores construir aplicaciones nativas utilizando tecnologías web. Facilita el acceso a las funcionalidades nativas del dispositivo, como la cámara y la geolocalización.

**8. Formspree**

Descripción: Servicio que permite gestionar formularios web sin necesidad de un backend. Facilita el envío de datos de formularios a través de correo electrónico.

**9. Barcode Scanner**

Descripción: Servicio que permite escanear códigos de barras y QR utilizando la cámara del dispositivo. Es útil para aplicaciones que requieren la lectura de códigos para acceder a información o realizar acciones.

## **13.2 Descripción de las Relaciones del Stack Tecnológico**

**1. Nivel 1: Usuario**

Los usuarios interactúan con la aplicación a través de dispositivos móviles, utilizando la interfaz proporcionada por Ionic y Angular.

**2. Nivel 2: Aplicación**

Ionic y Angular trabajan juntos para crear una experiencia de usuario fluida y dinámica. Ionic proporciona componentes UI, mientras que Angular maneja la lógica de la aplicación.

Capacitor permite el acceso a funcionalidades nativas del dispositivo, como la cámara y la geolocalización.

Leaflet se integra para mostrar mapas interactivos dentro de la aplicación.

**3. Nivel 3: Servicios**

Firebase Authentication gestiona el registro y la autenticación de usuarios, asegurando que solo los usuarios autorizados accedan a la aplicación.

Firestore Cloud Database y Realtime Database se utilizan para almacenar y sincronizar datos de usuarios y viajes en tiempo real.

Formspree permite el envío de formularios, como solicitudes de registro.

Barcode Scanner se utiliza para escanear códigos de barras y QR, facilitando la interacción del usuario.

**4. Nivel 4: Infraestructura**

Firebase actúa como la plataforma en la nube que soporta todos los servicios mencionados, proporcionando una infraestructura escalable y segura.

El hosting de Firebase permite que la aplicación esté disponible en la web y en dispositivos móviles.

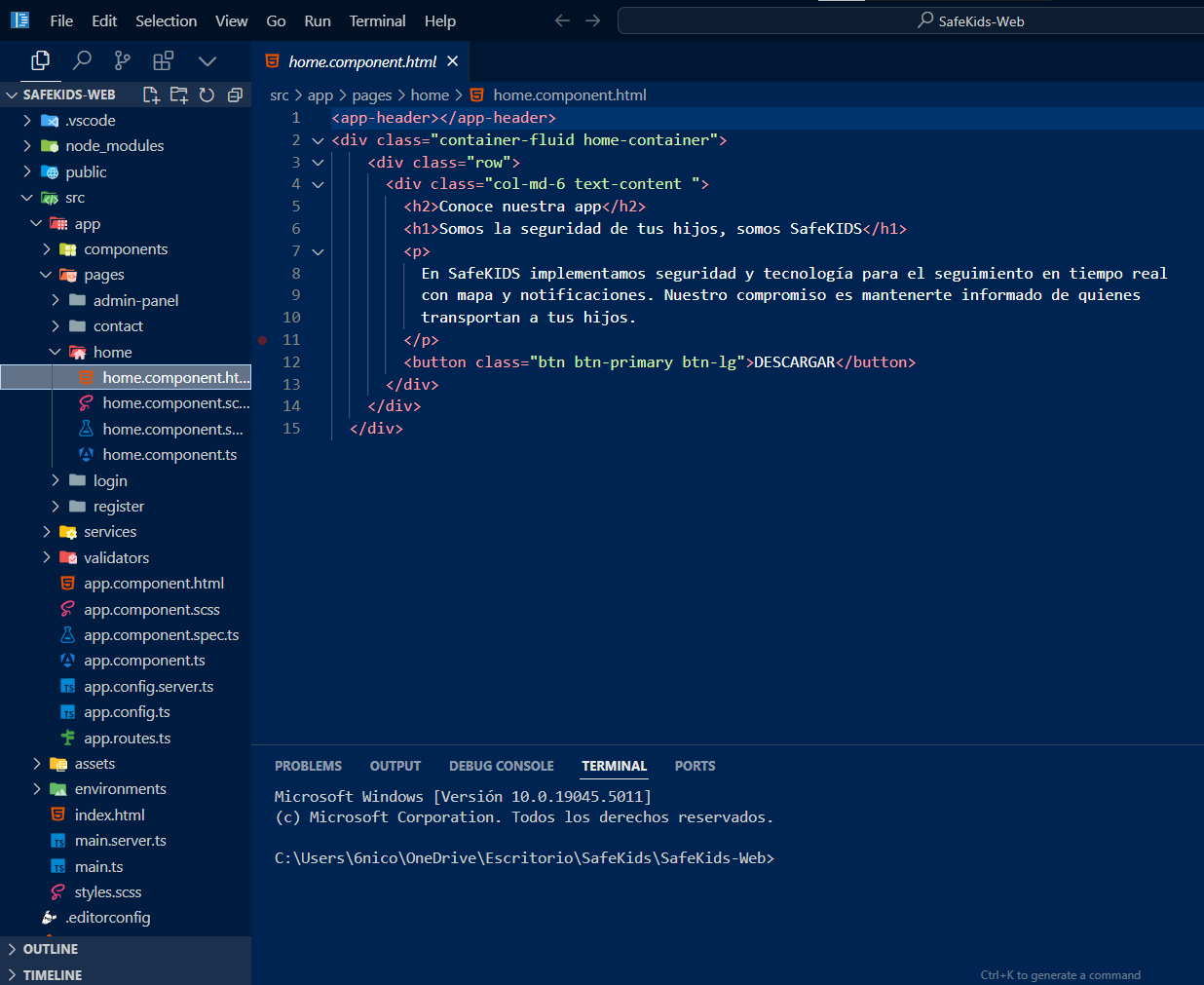
# 

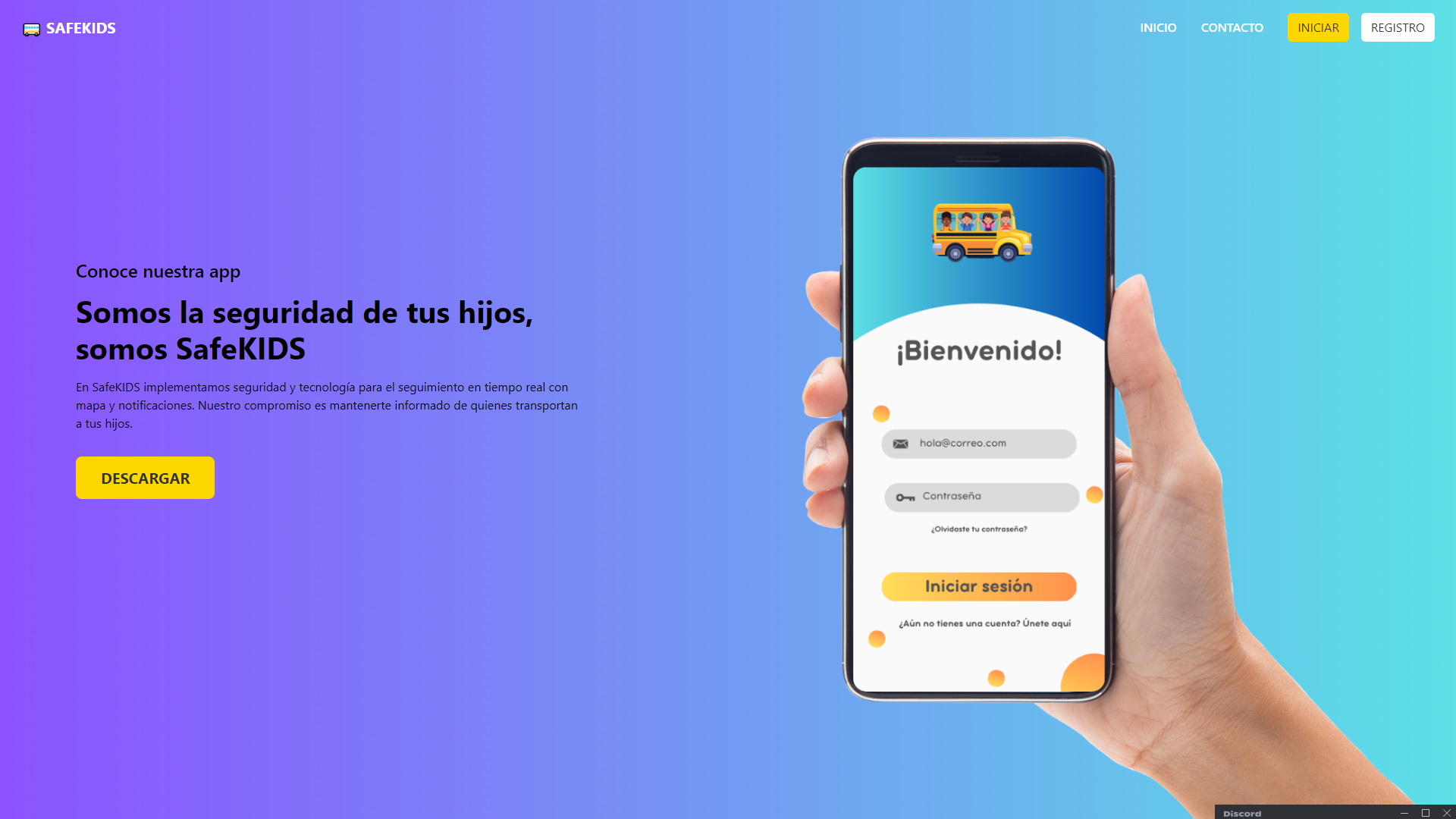
# **14. Evidencias del Desarrollo**

**Link de Github Móvil:** [**https://github.com/Nixixouwu/SafeKids-App**](https://github.com/Nixixouwu/SafeKids-App)

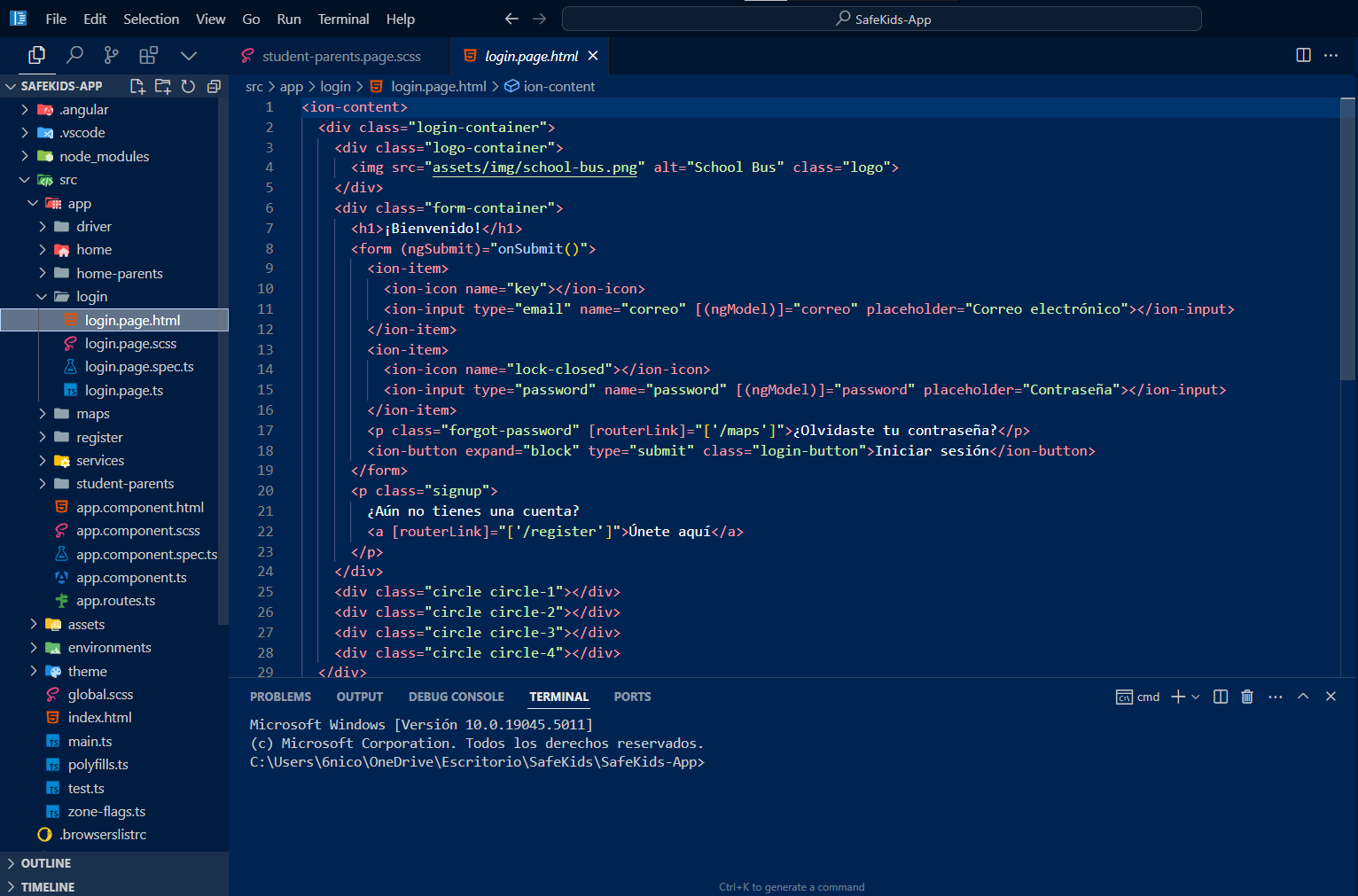
**Link de Github Web:** [**https://github.com/Nixixouwu/SafeKids-Web**](https://github.com/Nixixouwu/SafeKids-Web)

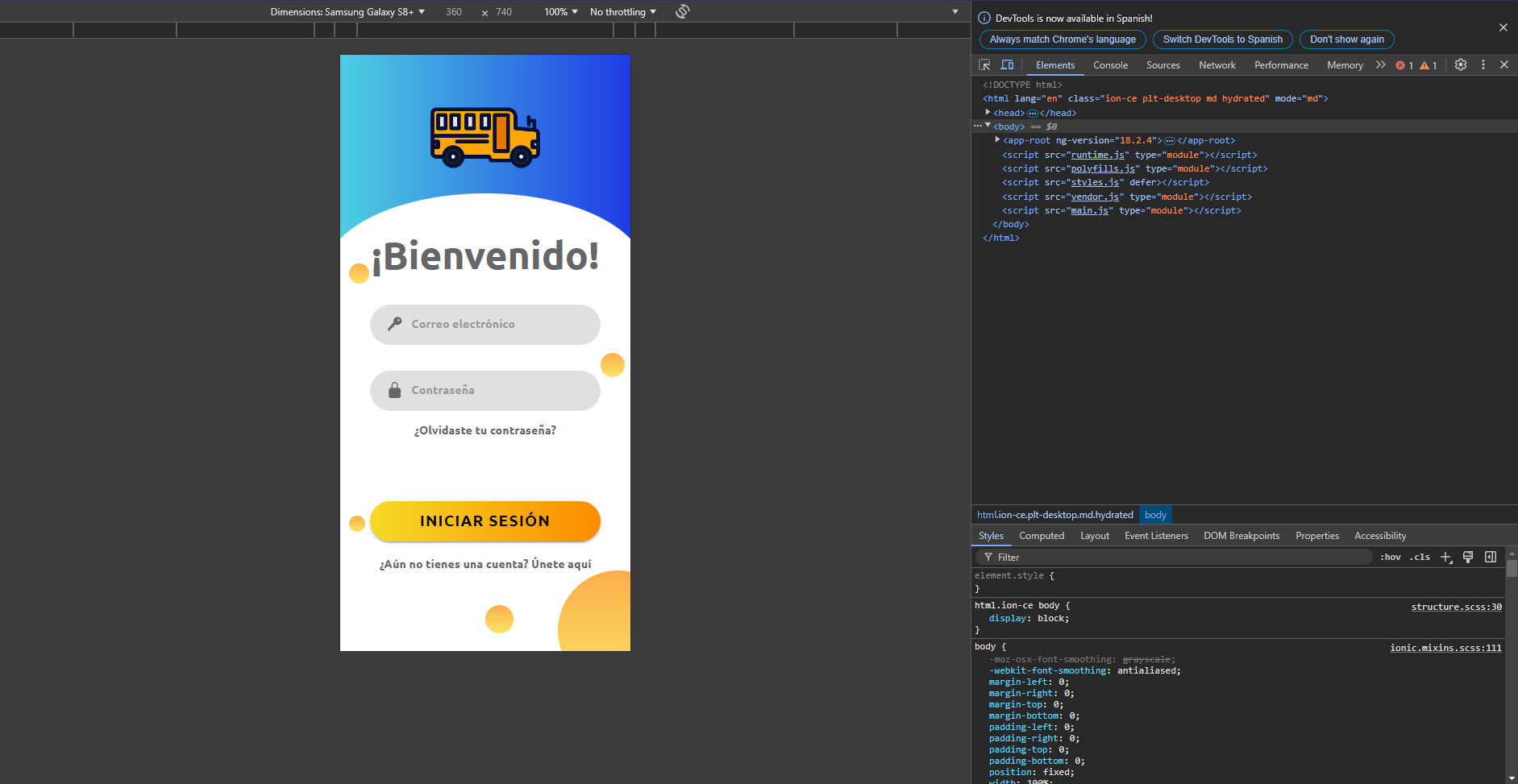
## **14.1 Código Inicio Web**

****

****

## **14.2 Código Login Móvil**

****

****

# **15. Conclusiones**

## **Nicolás Aliaga**

I’m confident in the success and future potential of the project. This system addresses a critical need for student safety, which is a priority for parents and schools alike. By leveraging modern technology to provide real-time tracking and monitoring, we are not only making school travels safer, we are also setting a base for the future. I believe that this project can be scaled to cover additional safety aspects, like monitoring in-school activities or emergency alerts. With continuous improvement, SafeKIDS has the potential to become a guide to the future of student safety.

## **Sebastián Pinto**

The SafeKIDS project is a great step in the direction of enhancing student safety, but I believe there are areas that still need improvement. To truly meet the diverse needs of parents, schools, and drivers, we must ensure that our platform is both robust and user-friendly. Continuous testing and user feedback are essential to polish these aspects. Beating these challenges would help SafeKIDS to fully realize its potential and be confident for their users.

## **Brayan Galaz**

I am satisfied with the project. We have managed to deliver on our initial goals, creating a functional and reliable system for monitoring students' journeys from school. The combination of all the features for parents and school staff provides comprehensive peace of mind. While there is always room for improvement, I believe the current state of the project meets the needs of our users.