

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

DISERTACIÓN PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO DE SISTEMAS Y COMPUTACIÓN

DESARROLLO DE UNA APLICACIÓN WEB-MÓVIL PARA RASTREO, SEGUIMIENTO Y REPORTE DE CONTINGENCIAS DEL TRANSPORTE ESCOLAR. CASO DE ESTUDIO: UNIDAD EDUCATIVA SAN JOSÉ LA SALLE

AUTOR: FRANKLIN STEVEN ILLANES RUMIGUANO

DIRECTOR: ING. SUSANA MASAPANTA

QUITO, SEPTIEMBRE 2021

I. Aspectos preliminares e introductorios

I. Dedicatoria

Este proyecto de disertación está dedicado principalmente a mis padres, ya que ellos fueron mi fuente primordial de inspiración, y al ser ellos los que me dieron fuerza para continuar con este proceso, ya que este al ser uno de mis grandes anhelos por todos estos años necesitaba sobrepasar cualquier problema que se me presentara, por su amor, trabajo y sacrificio gracias por ayudarme a convertirme en lo que actualmente soy, ha sido un orgullo y privilegio ser su hijo.

A mis hermanos por estar siempre presentes acompañándome y apoyando moralmente cada que lo necesitaba. Dándome palabras de aliento cada vez que no podía más.

A las personas que me apoyaron para que el trabajo se realice de una forma correcta y sencilla, como son mis profesores, ya que gracias ellos me ayudarán a que se me abran las puertas del futuro y compartiera conocimientos con las demás personas.

A mis amigos que estuvieron ahí en los malos y buenos momentos, y ayudaron a seguir poniendo esfuerzo cada día en la universidad y cumplir con un logro más en nuestra vida.

A una persona muy especial en mi vida, esa persona que siempre ha estado ahí para mí en los tiempos buenos y malos, la ayuda que me ha brindado ha sido de gran importancia, ha estado para mí sobre todo en los malos momentos, siempre ayudándome. No fue fácil finalizar este proyecto con éxito no obstante siempre fue esperanzadora y motivadora en mi vida, siempre me decía que lo podía lograr.

A Dios por siempre guiarme hacia donde debía ir, por aportarme la sabiduría y ayudarme a elegir las prioridades en mi vida, y así poder alcanzar cada uno de los objetivos que me plantee.

II. Agradecimientos

Quiero agradecer a la Unidad Educativa San José La Salle por permitirme realizar este proyecto y por todo el proceso realizado, desde la toma de requerimientos hasta el correcto resultado obtenido.

A mi directora de disertación la Ing. Susana Masapanta, y a mis lectores el Ing. Alfredo Calderón e Ing. Juan Francisco Chafla que me ayudaron a lo largo de la realización de esta disertación con su asistencia y consejos para el mejor resultado.

A mi madre por el apoyo brindado durante la elaboración del proyecto con sus consejos y guías no lo podía haber logrado.

Tabla de contenido

I. ASPECTOS PRELIMINARES E INTRODUCTORIOS II

I. DEDICATORIA II

II. AGRADECIMIENTOS III

TABLA DE ILUSTRACIONES 6

1. INTRODUCCIÓN 8

1.1. JUSTIFICACIÓN 8

1.2. OBJETIVO GENERAL 8

1.3. OBJETIVOS ESPECÍFICOS 8

2. ANTECEDENTES 9

2.1. AUMENTO DE LA INSEGURIDAD EN QUITO 9

2.2. APLICACIÓN USADA PREVIAMENTE 10

3. USO DEL *SMARTPHONE* EN PADRES DE FAMILIA 11

3.1. APLICACIONES MÓVILES 11

3.2. SISTEMAS OPERATIVOS MÓVILES 12

3.2.1. iOS 12

3.2.2. Android 12

3.3. USO DEL SMARTPHONE 13

4. METODOLOGÍA USADA 15

4.1. ¿QUÉ ES SCRUM? 15

4.2. ROLES EN SCRUM 15

4.2.1. Product Owner 15

4.2.2. Equipo de Desarrollo 16

4.2.3. Scrum Master 16

4.3. COMPONENTES EN SCRUM 16

4.3.1. Product Backlog 16

4.3.2. Sprint 17

4.3.3. Sprint Backlog 18

4.3.4. Ciclo de Vida en los proyectos ágiles 18

5. DESARROLLO DE LA APLICACIÓN WEB MÓVIL 22

5.1. ETAPA 1: INICIO DEL PROYECTO 22

5.2. PLANIFICACIÓN DE FASES 22

5.3. ANÁLISIS DE REQUERIMIENTOS 25

5.4. ETAPA 2: PLANIFICACIÓN ... 26
5.4.1. Describir las historias de usuario ... *26*
5.4.2. Priorizar las historias de usuario ... *28*
5.4.3. Reunión de planificación del Sprint ... *29*
5.5. ETAPA 3: DESARROLLO DEL *SPRINT* ... 30
5.5.1. Diseño de la aplicación ... *30*
5.5.2. Desarrollar los entregables ... *31*
5.6. ETAPA 4: REVISIÓN DEL *SPRINT* ... 32
5.6.1. Reunión del Sprint ... *33*
5.6.2. Sprint 1 ... *33*
5.6.3. Sprint 2 ... *34*
5.6.4. Sprint 3 ... *34*
5.6.5. Sprint 4 ... *35*
5.6.6. Sprint 5 ... *37*
5.6.7. Sprint 6 ... *38*
5.6.8. Sprint 7 ... *40*
5.6.9. Sprint 8 ... *41*
5.6.10. Sprint 9 ... *42*
5.6.11. Sprint 10 ... *43*
5.6.12. Sprint 11 ... *45*
5.7. ETAPA 5: IMPLEMENTACIÓN ... 46
5.7.1. Planificación de la implementación ... *46*
5.8. ETAPA 6: CIERRE DEL PROYECTO ... 49

6. PRUEBAS ... 49

6.1. PRUEBAS DE USABILIDAD ... 49
6.2. PRUEBAS DE FUNCIONALIDAD ... 59

7. CONCLUSIONES Y RECOMENDACIONES ... 59

7.1. CONCLUSIONES ... 59
7.2. RECOMENDACIONES ... 61

8. BIBLIOGRAFÍA ... 62

Tabla de Ilustraciones

Imagen 1 Página inicial aplicación web (LogicFast, 2020)..... 10
Imagen 2 Página para agregar rutas (LogicFast, 2020)..... 11
Imagen 3 Comparativa del mercado en Sistemas Operativos móviles (Mena Roa, 2020) 13
Imagen 4 Evolución de porcentaje de personas que tienen un Smartphone, por área (Equipo técnico Encuesta Nacional Multipropósito de Hogares, 2020) 14
Imagen 5 Sprint (CertMind, 2019) 18
Imagen 6 Ciclo de vida SCRUM (CertMind, 2019)..... 19
Imagen 7 Arquitectura del proyecto (Fuente: el autor) 25
Imagen 8 Resultado del Sprint 1 (Fuente: el autor)..... 33
Imagen 9 Pantalla 1 Sprint 3 (Fuente: el autor)..... 34
Imagen 10 Pantalla 2 Sprint 3 (Fuente: el autor)..... 35
Imagen 11 Pantalla 1 Sprint 4 (Fuente: el autor)..... 36
Imagen 12 Pantalla 2 Sprint 4 (Fuente: el autor)..... 36
Imagen 13 Pantalla 1 Sprint 5 (Fuente: el autor)..... 37
Imagen 14 Código sobre tiempo (Fuente: el autor)..... 38
Imagen 15 Pantalla 1 Sprint 6 (Fuente: el autor)..... 39
Imagen 16 Pantalla 2 Sprint 6 (Fuente: el autor)..... 39
Imagen 17 Pantalla 1 Sprint 7(Fuente: el autor)..... 40
Imagen 18 Pantalla de Mensajes 1 Sprint 8 (Fuente: el autor)..... 41
Imagen 19 Pantalla de mensajes 2 Sprint 8 (Fuente: el autor) 42
Imagen 20 Pantalla emergencia Sprint 9 (Fuente: el autor) 43
Imagen 21 Pantalla 1 Sprint 10 (Fuente: el autor)..... 44
Imagen 22 Pantalla 2 Sprint 10 (Fuente: el autor)..... 45
Imagen 23 Pantalla Sprint 11 (Fuente: el autor)..... 45
Imagen 24 Consola Firebase (Fuente: el autor)..... 47
Imagen 25 Creación proyectos móviles (Desplegando a iOS y Android, 2019)..... 48
Imagen 26 Estructura del proyecto (Fuente: el autor) 48
Imagen 27 Banner utilizado en el sistema (Fuente: el autor) 50

Imagen 28 Banner de la pestaña Agregar Alumnos (Fuente: el autor) 50
Imagen 29 Menú lateral con palabras propias del usuario (Fuente: el autor)..... 51
Imagen 30 Caso contraseña incorrecta (Fuente: el autor) 51
Imagen 31 Caso email inexistente (Fuente: el autor) 52
Imagen 32 Estructura usada (Fuente: el autor)..... 52
Imagen 33 Home usuario no registrado (Fuente: el autor)..... 53
Imagen 34 Home Conductor (Fuente: el autor)..... 54
Imagen 35 Home Padre (Fuente: el autor)..... 54
Imagen 36 Pantalla Ruta de conductor (Fuente: el autor) 55
Imagen 37 Pantalla Ruta de conductor información (Fuente: el autor)..... 55
Imagen 38 Carga de Ruta Óptima (Fuente: el autor)..... 56
Imagen 39 Carga lista de Ruta Óptima (Fuente: el autor)..... 57
Imagen 40 Carga Notificaciones Inicio de Ruta (Fuente: el autor)..... 57
Imagen 41 Notificaciones enviadas Inicio de Ruta (Fuente: el autor) 58

Tabla 1 Porcentaje de personas que tiene Smartphone, por edades (Equipo técnico Encuesta Nacional Multipropósito de Hogares, 2020)..... 14
Tabla 2 Roles del proyecto (Fuente: el autor) 22
Tabla 3 Product Backlog (Fuente: el autor)..... 23
Tabla 4 Criterios de aceptación de tareas (Fuente: el autor) 24
Tabla 5 Cronograma de entregas (Fuente: el autor) 25
Tabla 6 Historias de usuarios Baja Prioridad (Fuente: el autor)..... 26
Tabla 7 Historias de usuarios Prioridad Media (Fuente: el autor)..... 26
Tabla 8 Historias de usuarios Alta Prioridad (Fuente: el autor)..... 27
Tabla 9 Priorización historias de usuario (Fuente: el autor)..... 28
Tabla 10 Planificación de Sprint (Fuente: el autor)..... 29
Tabla 11 Desarrollo del sprint (Fuente: el autor) 30
Tabla 12 Desarrollo de entregables (Fuente: el autor) 31
Tabla 13 Revisión del sprint (Fuente: el autor)..... 32

1. Introducción

1.1. Justificación

La motivación detrás del tema que se escogió para la disertación de grado es, que la Unidad Educativa San José La Salle ubicada en La Magdalena no cuenta con un sistema de control en el transporte escolar. Los padres de familia tienen la necesidad de realizar el seguimiento a sus hijos que asisten a educación primaria y secundaria, tanto en la entrada como en la salida del alumnado en el horario escolar establecido.

En años anteriores los estudiantes de la mencionada Unidad Educativa contaban con un sistema que trató de encargarse de esa necesidad, pero a pesar del uso de este sistema proporcionado por un tercero, existieron varios inconvenientes y se optó por dejar de lado dicho sistema, los padres de familia solicitan un sistema funcional y se pretende desarrollar una aplicación la cual cumpla con los requisitos que necesitan los padres de familia y los conductores, para así mejorar significativamente la seguridad en el plantel, dar confianza a los padres, reducir los tiempos de espera, por lo cual se plantea una implicación práctica al desarrollar, diseñar y realizar las funcionalidades necesarias para un buen uso de la tecnología.

1.2. Objetivo General

Desarrollar una aplicación web-móvil para gestionar el transporte escolar para la Unidad Educativa San José La Salle.

1.3. Objetivos Específicos

- Desarrollar una aplicación móvil para la localización ágil en el transporte escolar de la ciudad de Quito bajo estándares específicos. Cumpliendo con el análisis, diseño, desarrollo e implantación correcta de esta aplicación.
- Seleccionar la metodología adecuada para el desarrollo de aplicaciones móviles.
- Identificar un proceso de rol del control de rutas en el transporte escolar apegado a la realidad. El cual se podrá actualizar.
- Evaluar el caso de estudio mediante pruebas de funcionalidad y usabilidad.

2. Antecedentes

2.1. Aumento de la inseguridad en Quito

La inseguridad en Quito va aumentando más con el pasar de los años, podemos notarlo con las cifras oficiales que proporciona el Ministerio de Gobierno, por medio de (Notimundo, 2019) los robos hacia las personas han aumentado en un 9,43% referente a los años anteriores, lo que menciona el Alcalde Jorge Yunda es que existe un déficit de policías en el Distrito Metropolitano de Quito, alrededor de 7 mil, aunque trabajan en varios convenios para tratar de solucionar esta carencia que aquejan los quiteños.

Si bien en el año 2017 el Ecuador fue situado como el país con menor tasa de homicidios de América Latina, a partir del año 2019 esto ha ido cambiando considerablemente aumentando la delincuencia en las calles (NotiMundo, 2020). El aumento de la tasa de homicidios por cada 100.000 habitantes va en aumento conforme el pasar del tiempo, en Ecuador pasó del 5,82 al 6,82 entre el 2018 y 2019 (González, 2020). Lo cual repercute claramente en Quito, al ser capital del Ecuador y tener un gran flujo de turistas, así también de refugiados. Las justificaciones que da el Gobierno ante la inseguridad latente de la capital es que la tasa de homicidios sigue siendo baja en promedio con América Latina, aunque este cálculo puede verse afectado por no tomar en cuenta el crecimiento demográfico, como es el ingreso de ciudadanos venezolanos (González, 2020).

El presidente Lenín Moreno plantea una reforma en la Ley de Movilidad Humana, como nos menciona (González, 2020) esta ayudaría a legalizar la deportación de extranjeros que cometan delitos en el país, esta propuesta de Ley de Movilidad fue presentada el 30 de julio de 2019 y sigue siendo analizada, ya que para su aprobación se necesita al menos 70 votos. Esta ley pretende ayudar a la inseguridad de Quito, al realizar mejores controles de inmigrantes en la capital, los cuales pueden estar de forma no controlada por el gobierno. Aunque varios sociólogos nos comentan que la inseguridad está ligada fuertemente con la pobreza, desigualdad y desempleo, pero al no existir estadísticas sobre crímenes no se puede dar una conexión real a esta problemática. Así también como tratar el problema de la inseguridad en Quito como un fenómeno de inmigración y no como una temática social.

2.2. Aplicación usada previamente

La aplicación que se usaba con anterioridad prestaba algunos servicios como eran; una presentación rápida de los cursos inscritos, los pasajeros que se encontraban dentro del sistema, las unidades de transporte dentro de los sistemas, y una lista de servicios de transporte.

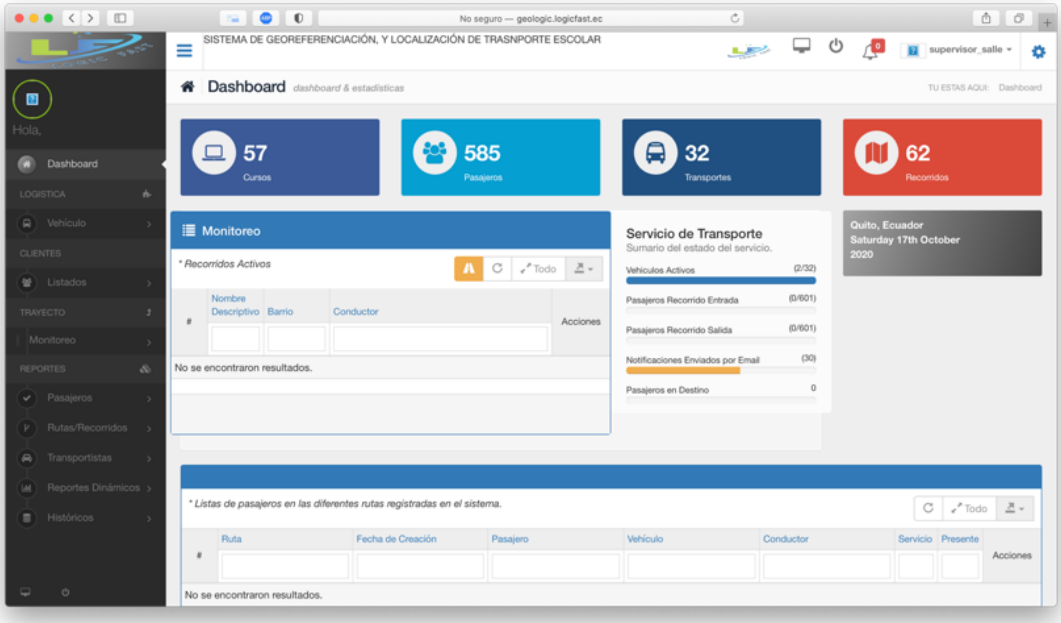


Imagen 1 Página inicial aplicación web (LogicFast, 2020)

La mayor parte de las quejas de los usuarios hacia este sistema era la deficiencia en la comunicación con los padres de familia; así como también, la poca familiarización con los transportistas con la aplicación móvil. En la página web únicamente se podía ingresar personal autorizado, lo cual a muchos padres de familia incomodaba, ya que no podían usar una aplicación por la que pagan una mensualidad dentro del sistema de transporte.

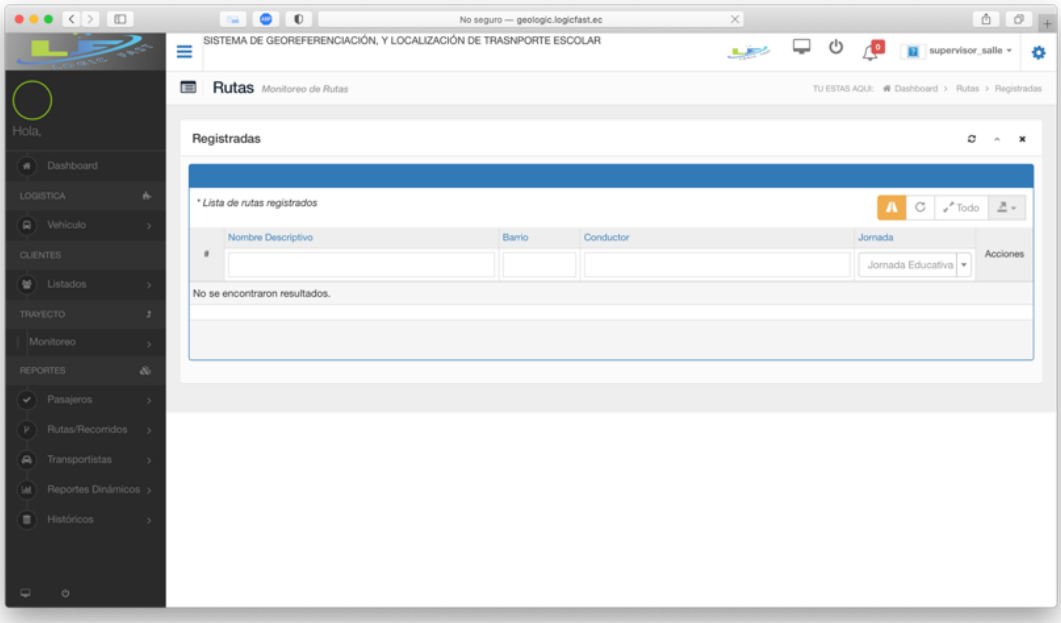


Imagen 2 Página para agregar rutas (LogicFast, 2020)

La aplicación web-móvil pretende solucionar dichos problemas, como son el acceso a usuarios del transporte escolar tanto web, como móvil. Una interfaz más amigable para transportistas y padres de familia, que posea un lenguaje mucho más familiar y una curva de aprendizaje mínima.

3. Uso del *Smartphone* en Padres de Familia

3.1. Aplicaciones móviles

Las aplicaciones o también conocidas como "*apps*" son software informático que se ejecutan en *Smartphone* o también conocidos como teléfonos inteligentes, aunque no son los únicos dispositivos los cuales son capaces de correr estas *apps* (Enriquez & Casas, 2014). Una aplicación móvil puede ser corrida sobre diferentes sistemas operativos y ser descargada de diferentes tiendas de aplicaciones. Varias de las *apps* son gratuitas lo cual no quiere decir que necesariamente sean cien por ciento gratis; para poder utilizarlas muchas veces los desarrolladores ponen publicidad dentro de estas, así también como ofrecer distintas funciones si es que se usa gratis y funciones mejoradas si se utiliza la versión paga.

3.2. Sistemas operativos móviles

Un sistema operativo o por sus siglas en inglés *OS* (*Operating System*), es aquel que permite interactuar entre el *Hardware*, *Software* y el Usuario, para realizar una acción determinada. El *OS* es el encargado de administrar todos los recursos del dispositivo inteligente, para usarlos de manera eficiente y sin interrupciones en la navegación de este (Malave Polanco & Beauperthuy Taibo, 2011).

Los sistemas operativos móviles más famosos en la actualidad son *Android* desarrollado por *Google* y *iOS* desarrollado por *Apple*, (Salazar Ospina, 2017) en un menor porcentaje *BlackBerry OS, Windows Phone* y *Symbian.*

3.2.1. iOS

Es el Sistema Operativo desarrollado por Apple, originalmente llamado *iPhone OS*. Es uno de los *OS* más utilizados en el mundo, se resalta mayormente la seguridad la cual ofrece a sus usuarios, *iOS* es de código cerrado, es decir, no puede ser usado por cualquier usuario o únicamente con la autorización de *Apple*. Posee una gran cantidad de aplicaciones que va creciendo constantemente. (Salazar Ospina, 2017)

3.2.2. Android

Es un Sistema Operativo desarrollado por *Google*, tiene un núcleo basado en *Linux*, proporciona un código abierto en el cual se pueden realizar modificaciones a cada uno de los fabricantes los cuales necesitan comprar una licencia para montar este sistema operativo en sus *Smartphone*. Ofrece una gran flexibilidad al usuario en optar por características propias para cada usuario. *Android* al ser perteneciente a *Google* tiene muy bien integrados todos los servicios, lo cual pretende ser una ventaja sobre sus competidores directos. (Salazar Ospina, 2017)

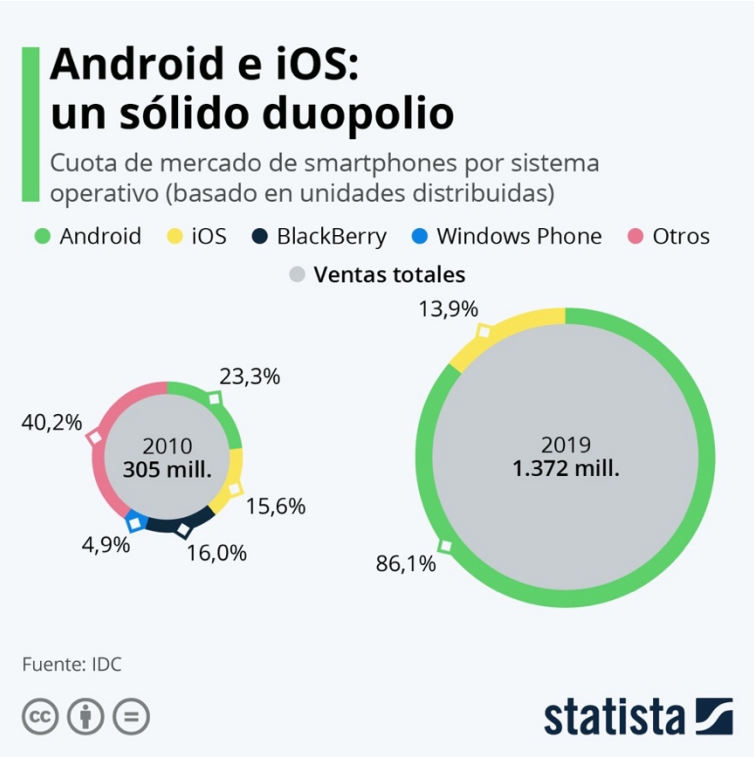


Imagen 3 Comparativa del mercado en Sistemas Operativos móviles (Mena Roa, 2020)

En la actualidad estos dos sistemas operativos están con la mayor parte de cuota de mercado, según las últimas estadísticas *Android* a final de 2019 poseía una cuota del 86% y *iOS* el 14% restante, esto indica que el resto de los sistemas operativos móviles ha tenido una gran baja dentro de sus ventas, haciendo prácticamente nulas. (Mena Roa, 2020)

3.3. Uso del Smartphone

En Ecuador va incrementando cada año el número de *Smartphone* vendidos en el mercado nacional, por consiguiente, el uso de este se cuantifica. Según el INEC es decir el Instituto Nacional de Estadística y Censos, en uno de sus últimos estudios sobre el uso de tecnologías de la información y comunicación realizada en 2018. Nos indica que para el año 2019 va aumentando el porcentaje de personas con teléfono inteligente en el área Urbana que en 2018 fue de 50% hacia una subida en 2019 al 54%. (Equipo técnico Encuesta Nacional Multipropósito de Hogares, 2020)

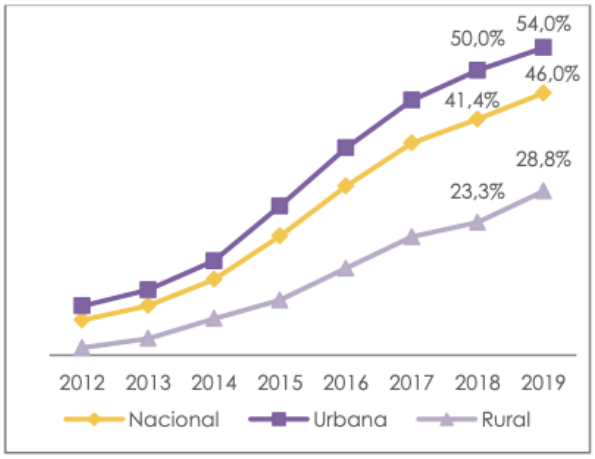


Imagen 4 Evolución de porcentaje de personas que tienen un Smartphone, por área (Equipo técnico Encuesta Nacional Multipropósito de Hogares, 2020)

Este aumento de personas con un *Smartphone* potencializa al mercado de aplicaciones para dispositivos móviles. Haciendo que exista un mayor número de competidores en esta área, ofreciendo distintos servicios y aplicaciones útiles para el usuario.

INEC además en su estudio da a conocer sobre el incremento de personas con un *Smartphone* según su edad, con una variación significativa de 2018 a 2019. (Equipo técnico Encuesta Nacional Multipropósito de Hogares, 2020)

Tabla 1 Porcentaje de personas que tiene Smartphone, por edades (Equipo técnico Encuesta Nacional Multipropósito de Hogares, 2020)

Grupos de edad	2018	2019	Variación significativa dic18/dic19
5 a 15 años	11,8%	12,2%	No
16 a 24 años	63,2%	68,9%	Si
25 a 34 años	70,2%	75,9%	Si
35 a 44 años	61,8%	67,9%	Si
45 a 54 años	49,2%	56,3%	Si
55 a 64 años	34,8%	42,0%	Si
65 y más años	13,9%	18,9%	Si

Las cifras demuestran que existió un aumento general en la población de 16 a 98 años, es decir, cada vez son más las personas que independientemente de su edad pueden utilizar y hacer un uso moderado del *Smartphone*.

4. Metodología usada

El proyecto se elaborará con SCRUM, esta metodología ayuda a implementar proyectos en corto tiempo, y da una retroalimentación constante. Se escogió por el hecho de estar orientado a trabajar en equipos productivos (Trigas Gallego, 2012). Al ser interacciones cortas que entregan parte del producto y no se basa en la completitud de este, a diferencia de metodologías tradicionales que se basan de principio a fin. (Rodriguez & Dorado, 2015)

4.1. ¿Qué es SCRUM?

Es un proceso para trabajar de una manera más productiva en el cual el equipo trabaja colaborativamente. Además, SCRUM pretende entregar a tiempo un *software* complejo, esto se realiza en forma interactiva e incremental; a las cuales llamamos interacciones o en SCRUM "*Sprints*". (Trigas Gallego, 2012)

Además, esta metodología entusiasma al cliente y lo compromete, cuando puede ver un avance real con cada una de las entregas, al ver el proyecto en etapas previas y cómo se va desarrollando puede ir realineando el *software* con los objetivos planteados inicialmente, de igual forma, se puede introducir cambios funcionales o prioritarios en cada nuevo *Sprint*. (CertMind, 2019)

4.2. Roles en SCRUM

El Equipo de SCRUM (CertMind, 2019) consiste en un *Product Owner*, el equipo de desarrollo y el *Scrum Master*. Estos equipos SCRUM son auto organizados y multifuncionales; así como también, entregan los productos de forma iterativa e incremental, esto ayuda a maximizar la retroalimentación para una mucho mejor versión potencialmente útil y mejora la funcionalidad final del producto.

4.2.1. Product Owner

Es aquel que se encarga de ser responsable de maximizar el valor entregado por el Equipo de Desarrollo. (CertMind, 2019) Este también es el representante y/o

dueño del producto, como la voz del usuario; y es quien gestiona el *Product Backlog*, presenta los informes del proyecto y aprueba cambios.

4.2.2. Equipo de Desarrollo

Es un grupo de profesionales que tienen conocimiento técnico y desarrollan el producto. Entregan un incremento de producto terminado, que se pueda poner en producción al final de cada Sprint. El tamaño óptimo es de entre 3 y 9 miembros; esto ayuda a ser ágil y completar una gran cantidad de trabajo significativo. (CertMind, 2019)

4.2.3. Scrum Master

Es aquel que promociona y apoya el proyecto. Garantiza el conocimiento y correcta aplicación del Scrum, sus prácticas y sus reglas. (CertMind, 2019) También lidera este Equipo, los guía con las reglas propuestas y ayuda en los obstáculos que impiden el desarrollo correcto del proyecto. Además de ser quienes ayudan a las personas externas a entender que iteraciones son útiles y cuáles no.

4.3. Componentes en SCRUM

4.3.1. Product Backlog

Es una pila de requisitos de los usuarios descritos de una forma que el *Product Owner* sea fácilmente entendible, esta es la única fuente de requisitos para cualquier futuro cambios que se podrían realizar. El *Product Owner* es responsable de aprobar cualquier cambio.

Nunca se considera que está completo, ya que evoluciona a medida que el producto y entorno también lo hacen. El *Product Backlog* enumera las características, funcionalidades, requisitos, futuras mejoras y correcciones sobre el producto.

Tiene como mínimos requisitos los siguientes atributos:

- Descripción (en función al valor del negocio)
- Orden (prioridad)
- Estimación
- Criterios de aceptación

Para evitar el nunca tener un fin para el proyecto es importante definir un alcance al momento de refinar el *Product Backlog*. (CertMind, 2019)

4.3.2. *Sprint*

Es una iteración con una fecha fijada con anterioridad en las que las historias del usuario se convierten en partes funcionales de un software. Se realiza de 1 a 4 semanas, donde se va incrementando el producto "terminado" utilizable y potencialmente desplegable. Se realiza inmediatamente después de haber terminado un Sprint anterior. Se realizan las tareas de planificación del Sprint, trabajo de desarrollo y su respectiva revisión.

En el Sprint:

- No se realizan cambios que pueda afectar el objetivo del Sprint.
- No disminuyen los objetivos de calidad
- Entre el *Product Owner* y Equipo de Desarrollo puede aclararse y renegociarse el alcance.

Cada uno de los *Sprint* puede considerarse como un proyecto con un objetivo, también habilitan la predictibilidad al asegurar la inspección y adaptación del progreso al menos en cada mes calendario. (CertMind, 2019)

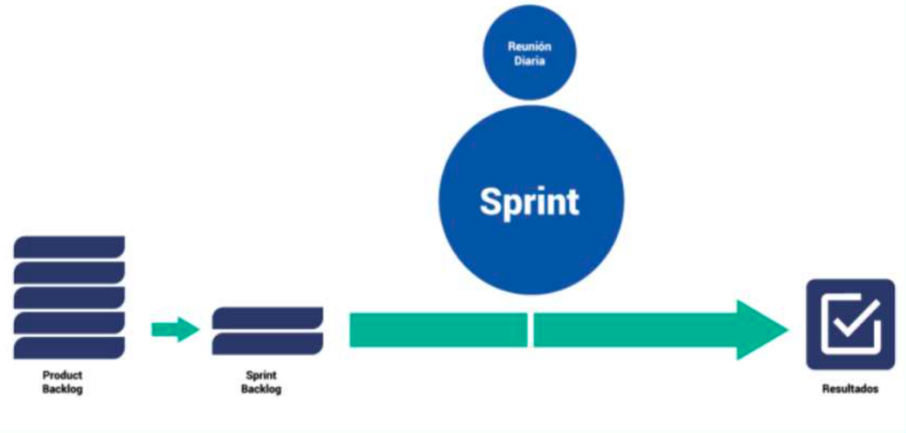


Imagen 5 Sprint (CertMind, 2019)

4.3.3. *Sprint Backlog*

Son los elementos seleccionados del *Product Backlog* los cuales serán usados en el *Sprint*, más un plan para entregar un incremento del producto y conseguir el objetivo del *Sprint*. Si es que se necesita un nuevo trabajo en el proyecto el equipo de desarrollo lo adiciona a esta sección. Y a medida que se completa se actualiza la estimación de tiempo de trabajo restante, solo el equipo de desarrollo puede cambiar este *Sprint Backlog* durante un *Sprint* (CertMind, 2019). Aquí se incluye al menos una actividad la cual puede mejorar los procesos.

4.3.4. Ciclo de Vida en los proyectos ágiles

SCRUM presenta 6 etapas definidas y cada una de estas tiene un conjunto de prácticas que las hacen en total 17. (CertMind, 2019)

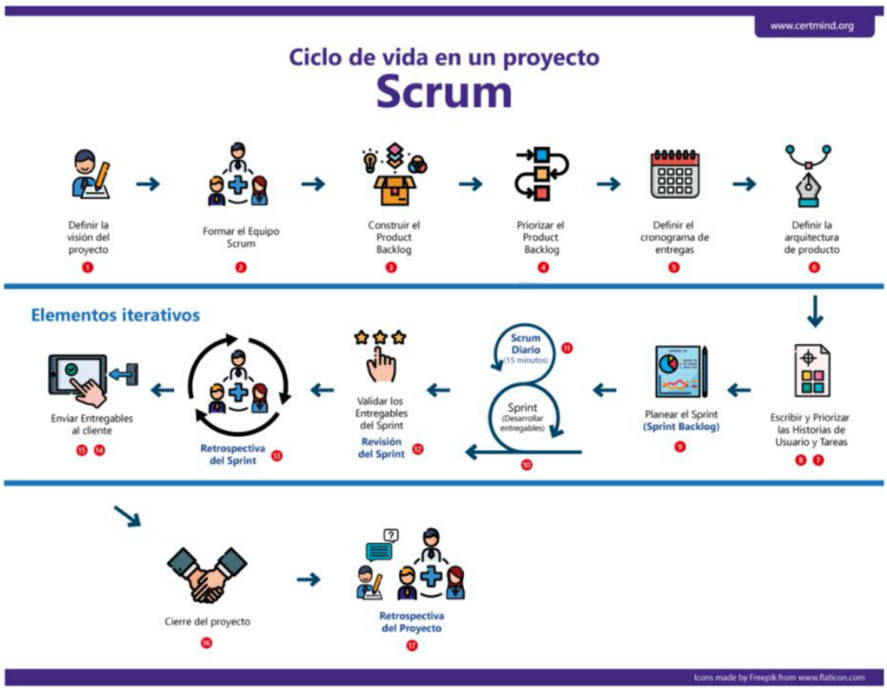


Imagen 6 Ciclo de vida SCRUM (CertMind, 2019)

4.3.4.1 Etapas de SCRUM

A continuación, se describe cada una de las etapas del SCRUM se toma como referencia lo indicado por *(CertMind, 2019)*.

Etapa 1: Inicio del proyecto

a) **Definir la visión del proyecto**

Se estructura la visión del proyecto, explicando las necesidades empresariales y el alcance; así como el tiempo, el presupuesto y la calidad

b) **Formar el Equipo SCRUM**

Se eligen los miembros del equipo SCRUM.

c) **Construir y priorizar el Product Backlog**

Se construye el *Product Backlog* con los elementos más importantes como son: Historias de usuario, errores y tareas.

Para priorizar el *Backlog* se toman en cuenta varios factores dentro del producto como son: el esfuerzo, tiempo, dependencias, opinión del equipo.

d) Definir el cronograma de entregas

Se determina la duración de todos los *Sprint* del proyecto, así como una descripción de estos que se deben cumplir con los tiempos que se establecieron. Este cronograma lo realiza el *Product Owner* con el resto del Equipo SCRUM.

e) Definir la Arquitectura del proyecto

Se establece un diseño técnico, en donde se puede crear de forma iterativa los componentes de los *Sprint*.

Etapa 2: Planificación

a) Describir las historias de usuario

Gracias a los usuarios se definen los requerimientos del sistema, el "cómo", "quiero" y "para" ayudan a definir estas historias.

b) Priorizar las historias de usuario

El *Product Owner* selecciona cuales son las historias de usuarios más prioritarias para el siguiente *Sprint*, esto se realiza con la planificación en paralelo. Y es necesario realizarlo antes de la Reunión de planificación del *Sprint* para que se optimice el tiempo.

c) Reunión de planificación del *Sprint*

El *Product Owner* presenta el objetivo a lograr y los elementos del *Product Backlog* los cuales se deben cumplir en ese *Sprint*.

Etapa 3: Desarrollo del *Sprint*

a) Desarrollar los entregables

Se hacen los manuales o toda la documentación relacionada siempre pensando en la definición de un producto terminado.

Etapa 4: Revisión del *Sprint*

a) **Reunión de revisión del *Sprint***

Se realiza una demostración de cómo ha ido incrementando el producto al *Product Owner* con el fin de obtener su aprobación. Se realizan varias actividades en esta práctica.

- Presentar el incremento del producto
- Probar el incremento del producto (Validación)
- Aprobación del incremento de producto
- Actualizar el *Product Backlog*

Etapa 5: Implementación

a) **Planificación de implementación**

Se prepara todo para la implementación exitosa de los incrementos generados en el *Script*, esto ayuda a la disminución de riesgos en las operaciones de los clientes.

b) **Implementación de entregables**

Pone a disposición de los usuarios los incrementos finalizados y utilizables desarrollados por el Equipo SCRUM.

Etapa 6: Cierre del proyecto

Se deja un registro de cierre y el motivo, en donde puede existir un acta de cierre del proyecto. Así como poder existir varios motivos para el cierre de un proyecto como:

- Cierre exitoso del proyecto
- Cierre parcial del proyecto o aplazamiento
- Cancelación del proyecto

5. Desarrollo de la aplicación web móvil

5.1. Etapa 1: Inicio del proyecto

5.1.1. Definir la visión del proyecto

Las 3 restricciones principales del proyecto son (CertMind, 2019):

- **Tiempo:** Es primordial entregar el proyecto antes de la culminación de la carrera, como plazo máximo a finales de enero del 2021.
- **Presupuesto:** Depende del estudiante que desarrolla el proyecto.
- **Alcance:** Se desarrollarán los módulos de rastreo sobre las unidades de transporte, seguimiento de dichas unidades y reporte de contingencias hacia los padres de familia.

5.1.2. Formar el Equipo SCRUM

Tabla 2 Roles del proyecto (Fuente: el autor)

Colaborador	Rol
Ing. Susana Masapanta	SCRUM *Owner*
Franklin Steven Illanes	Grupo de Trabajo

5.2. Planificación de fases

5.2.1 Construir y priorizar el *Product Backlog*

En la siguiente tabla se detallan las tareas que se realizarán en el *Product Backlog*, tomando en cuenta su prioridad y estimación de desarrollo.

Tabla 3 Product Backlog (Fuente: el autor)

Product Backlog			
Nº	**Tarea**	**Prioridad**	**Estimación**
1	Registro de usuarios.	Media	3 días
2	Ingreso a la aplicación.	Baja	3 días
3	Uso API Google.	Alta	7 días
4	Realizar el perfil de usuario.	Baja	4 días
5	Realizar recuperación o actualización contraseñas.	Media	5 días
6	Creación de *BackEnd*.	Alta	9 días
7	APIs para el consumo de base de datos en *FrontEnd*.	Alta	10 días
8	API para la correcta comunicación entre GPS y el aplicativo.	Alta	12 días
9	API para envío de mensajes.	Media	6 días
10	API para difusión masiva.	Media	4 días

Cada una de las tareas del *Product Backlog* deben tener un criterio para poder ser aceptadas (CertMind, 2019), estos se presentan a continuación:

Tabla 4 Criterios de aceptación de tareas *(Fuente: el autor)*

Criterios de aceptación de tareas	
Tarea	**Descripción**
Registro de usuarios.	Control sobre *nickname*, correo electrónico, número máximo de teléfono, ciudad de residencia, referencia del domicilio, foto referencia, coordenadas en el mapa, contraseña.
Ingreso a la aplicación.	Control del formato de correo y contraseña, que coincida con la base de datos.
Uso API Google.	Control de ubicación en tiempo real, con una exactitud suficiente para ubicar la calle que se transita.
Realizar el perfil de usuario.	Control sobre el uso de datos de usuarios y su correcta observación.
Realizar recuperación o actualización de contraseñas.	Control en caso de olvido o actualización de contraseñas, para una correcta restauración de datos.
Creación de *BackEnd*.	Control del servidor de aplicaciones en el que correo el proyecto.
APIs para el consumo de base de datos en *FrontEnd*.	Control de APIs creadas y su correcto funcionamiento, de acuerdo con el inicio del proyecto.
API para la correcta comunicación entre GPS y el aplicativo.	Control de una API para la comunicación entre el aplicativo y el GPS.
API para envío de mensajes.	Control de API para envío simultáneo de mensajes, dependiendo la ruta.
API para difusión masiva.	Control de envío total a los usuarios del aplicativo.

5.2.2 Definir el cronograma de entregas

Tabla 5 Cronograma de entregas (Fuente: el autor)

Tareas	Plazo	Semanas								
	(Semanas)	1-dic	8-dic	15-dic	22-dic	29-dic	5-ene	12-ene	19-ene	26-ene
Registro de usuarios	1									
Ingreso a la aplicación	1									
Uso API Google	2									
Realizar el perfil de usuario	1									
Realizar recuperacion o actualización contraseñas	1									
Creación de BackEnd	2									
APIs para el consumo de base de datos en FrontEnd	2									
API para la correcta comunicación entre GPS y el aplicativo	2									
API para envió de mensajes	1									
API para difusión masiva	1									
Pruebas	1									

5.3. Análisis de Requerimientos

5.3.1 Definir la Arquitectura del proyecto

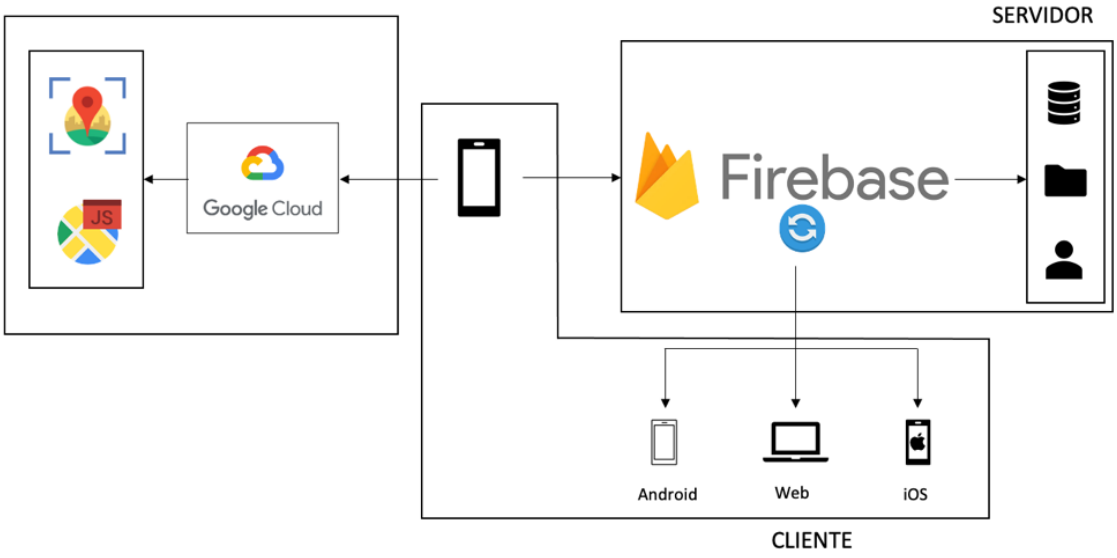


Imagen 7 Arquitectura del proyecto (Fuente: el autor)

5.4. Etapa 2: Planificación

5.*4*.1. Describir las historias de usuario

En la tabla 6 se describen las historias de usuario con una prioridad baja.

Tabla 6 Historias de usuarios Baja Prioridad (Fuente: el autor)

Historias del usuario del Product Backlog		
Tarea	**Historia de Usuario**	**Descripción**
Ingreso a la aplicación.	Como cliente debe tener un correcto ingreso a la aplicación, para usar todas las funcionalidades.	Se deberá poder ingresar sin inconvenientes, validando a la base de datos su correo y contraseña.
Realizar el perfil de usuario.	Como cliente debo ver todos mis datos correctos.	Se deberá observar los datos ingresados por el usuario.

En la tabla 7 se describen las historias de usuarios con una prioridad media.

Tabla 7 Historias de usuarios Prioridad Media (Fuente: el autor)

Historias del usuario del *Product Backlog*		
Tarea	**Historia de Usuario**	**Descripción**
Registro de usuarios.	Como cliente quiero registrarme de una manera fácil para tener acceso a las funcionalidades completas.	Se deberá registrar el usuario con los campos: correo, nombres, teléfono, ciudad, sector, contraseña.

Realizar recuperación o actualización de contraseñas.	Como cliente deseo recuperar mi contraseña en caso de pérdida o cuando crea necesario.	Se deberá tener una forma de cambiar o recuperar la contraseña de un usuario mediante correo electrónico.
API para envío de mensajes.	Como cliente quiero conocer el estado de los estudiantes.	Se deberá enviar mensajes constantes de los estados.
API para difusión masiva.	Como cliente quiero saber en caso de emergencia que paso con el estudiante.	Se deberá enviar mensajes en caso de una contingencia.

En la tabla 7 se describen las historias de usuarios con una prioridad alta.

Tabla 8 Historias de usuarios Alta Prioridad (Fuente: el autor)

Historias del usuario del Product Backlog		
Tarea	**Historia de Usuario**	**Descripción**
Uso API Google.	Como cliente quiero observar la ubicación exacta de una ruta para poder saber donde se encuentra el alumno a mi cargo.	Se deberá observar por medio de un mapa la ubicación en tiempo real de la ruta, con su dirección.
Creación de *BackEnd*.	Como cliente debe poder observar toda la información localizada desde una base de datos.	Se deberá crear todos los instrumentos necesarios para una buena interacción con la base de datos.
APIs para el consumo de base de datos en *FrontEnd*.	Como cliente debe poder usar la aplicación en cualquier lugar con acceso a internet.	Se deberá utilizar un servidor de aplicaciones que pueda utilizar

		BackEnd para consumir todas las APIs creadas.
API para la correcta comunicación entre GPS y el aplicativo.	Como cliente quiero visualizar dónde se encuentra el alumno en tiempo real.	Se deberá sincronizar en tiempo real la ubicación del medio de transporte escolar.

5.*4*.2. Priorizar las historias de usuario

En la tabla 9 se muestra como están priorizadas las historias de usuarios, desde baja, media y alta.

Tabla 9 Priorización historias de usuario (Fuente: el autor)

Priorización de Historias de Usuarios		
Nº	**Descripción**	**Prioridad**
1	Como cliente debe tener un correcto ingreso a la aplicación, para usar todas las funcionalidades.	Baja
2	Como cliente debo ver todos mis datos correctos.	Baja
3	Como cliente quiero registrarme de una manera fácil para tener acceso a las funcionalidades completas.	Media
4	Como cliente deseo recuperar mi contraseña en caso de pérdida o cuando crea necesario.	Media
5	Como cliente quiero conocer el estado de los estudiantes.	Media
6	Como cliente quiero saber en caso de emergencia que paso con el estudiante.	Media

7	Como cliente quiero observar la ubicación exacta de una ruta para poder saber donde se encuentra el alumno a mi cargo.	Alta
8	Como cliente debe poder observar toda la información localizada desde una base de datos.	Alta
9	Como cliente debe poder usar la aplicación en cualquier lugar con acceso a internet.	Alta
10	Como cliente quiero visualizar dónde se encuentra el alumno en tiempo real.	Alta

5.4.3. Reunión de planificación del *Sprint*

En la tabla 10 se presenta la planificación de los *Sprints* tomando en cuenta lo descrito en el *Product Backlog*, priorización de historias de usuario y lo expuesto al *Product Owner*.

Tabla 10 Planificación de Sprint *(Fuente: el autor)*

Nº	***Sprint***	**Planificación del *Sprint***
1	Uso API Google.	1. Controlar que en el mapa se pueda observar la posición de la ruta.
2	Creación de *BackEnd*.	1. Tener relación entre *BackEnd* y *FrontEnd*, se pretende tener avances cada 5 días. 2.Tener las funcionalidades completamente disponibles, tiempo estimado para cada avance 5 días.
3	APIs para el consumo de base de datos en *FrontEnd*.	1. Controlar todas las funcionalidades las cuales se puedan llegar a necesitar para que la aplicación tenga un correcto funcionamiento, se podrá realizar en una aproximado de 12 días.

4	API para la correcta comunicación entre GPS y el aplicativo.	1. Controlar el tiempo de refresco de la página con la ubicación de la ruta.
5	Registro de usuarios.	1.Controlar que el usuario ingrese de una correcta forma todos los datos los cuales sirvan para que se registre, entrega cada 5 días.
6	Realizar recuperación o actualización de contraseñas.	1. Controlar que el usuario pueda actualizar su información, se realiza cada 5 días.
7	API para envío de mensajes.	1. Controlar el envío de mensajes con el estado actual.
8	API para difusión masiva.	1.Controlar la comunicación en caso de contingencias.
9	Ingreso a la aplicación.	1.Controlar que el usuario pueda acceder al aplicativo, se lo realiza cada 5 días.
10	Realizar el perfil de usuario.	1.Controlar que se puedan observar los datos de forma correcta hacia el usuario, se lo realiza cada 5 días.

5.5. Etapa 3: Desarrollo del *Sprint*

5.5.1. Diseño de la aplicación

En la tabla 11 se resume todas las actividades de la etapa 3 del desarrollo y diseño de la aplicación.

Tabla 11 Desarrollo del sprint (Fuente: el autor)

Etapa 3: Desarrollo del *sprint*		
Nº	ACTIVIDADES	RESPONSABLE

1	Desarrollar los entregables.	*Product Owner, Scrum Master* y Equipo de desarrollo.

5.5.2. Desarrollar los entregables

Tabla 12 Desarrollo de entregables (Fuente: el autor)

Desarrollo de entregables		
Nº	***Sprint***	**Objetivo del entregable**
1	(Uso API Google) Controlar que en el mapa se pueda observar la posición de la ruta.	El usuario podrá visualizar la posición actual, con la posibilidad de poner una nueva en caso de ser necesario.
2	(*BackEnd*) Tener relación entre *BackEnd* y *FrontEnd*.	El desarrollador deberá controlar todo lo referente al funcionamiento correcto del *BackEnd*.
3	(*BackEnd*) Tener las funcionalidades completamente disponibles.	Se utilizará un *hosting* y dominio para el correcto funcionamiento del aplicativo móvil.
4	(APIS *FrontEnd*) Controlar todas las funcionalidades las cuales se puedan llegar a necesitar para que la aplicación tenga un correcto funcionamiento.	El desarrollador deberá realizar la creación de APIs propias para el correcto funcionamiento del aplicativo móvil.
5	(APIs GPS y aplicativo) Controlar el tiempo de refresco de la página con la ubicación de la ruta.	El desarrollador deberá tener en cuenta los recursos disponibles para no malgastarlos en tiempo de espera.
6	(Registros) Controlar que el usuario ingrese de una correcta forma todos los datos los cuales sirvan para que se registre.	El usuario deberá ingresar los datos necesarios para poder registrarse en el aplicativo y tener una mejor funcionalidad de este.

7	(Recuperación) Controlar que el usuario pueda actualizar su información.	El usuario podrá actualizar su información personal en el momento que lo desee mediante el ingreso al perfil.
8	(Envío mensajes) Controlar el envío de mensajes con el estado actual.	El usuario podrá recibir mensajes de notificación conforme se complete o avance en la ruta.
9	(API difusión) Controlar la comunicación en caso de contingencias.	El usuario recibirá un mensaje de notificación si existe algún inconveniente en la ruta.
10	(Ingreso) Controlar que el usuario pueda acceder al aplicativo.	El usuario podrá acceder al aplicativo mediante sus credenciales ingresadas durante el registro.
11	(Perfil) Controlar que se puedan observar los datos de forma correcta hacia el usuario.	El usuario podrá visualizar todos los datos personales ingresados mediante el registro.

5.6. Etapa 4: Revisión del *Sprint*

En la tabla 13 se resume todas las actividades que se realizará en esta etapa.

Tabla 13 Revisión del sprint (Fuente: el autor)

Etapa 4: Revisión del sprint		
Nº	ACTIVIDADES	RESPONSABLE
1	Reunión de revisión del *sprint*.	*Product Owner, Scrum Master* y Equipo de desarrollo.

5.6.1. Reunión del *Sprint*

A continuación, se presentan todos los *Sprints* y su correspondiente avance.

5.6.2. *Sprint* 1

Descripción: Controlar que en el mapa se pueda observar la posición de la ruta.

Incremento del producto: En la Imagen 8 se muestra un mapa con el indicador rojo sobre la posición en donde se encuentra la ruta, se puede acercar o alejar del mismo para tener un mejor punto de vista de donde se encuentra.



Imagen 8 Resultado del Sprint 1 (Fuente: el autor)

Validación: Aprobado

Existe incremento: No

Aprobación del *Product Backlog*: Aprobado

5.6.3. *Sprint* 2

Descripción: Tener relación entre *BackEnd* y *FrontEnd*

Incremento del producto: Aprobado, no aplica con visualización de una imagen.

Validación: Aprobado

Existe incremento: No

Aprobación del *Product Backlog*: Aprobado

5.6.4. *Sprint* 3

Descripción: Tener a las funcionalidades completamente disponibles

Incremento del producto: En la Imagen 9 y 10 se puede ver el dominio y el correcto funcionamiento de este para el aplicativo móvil.

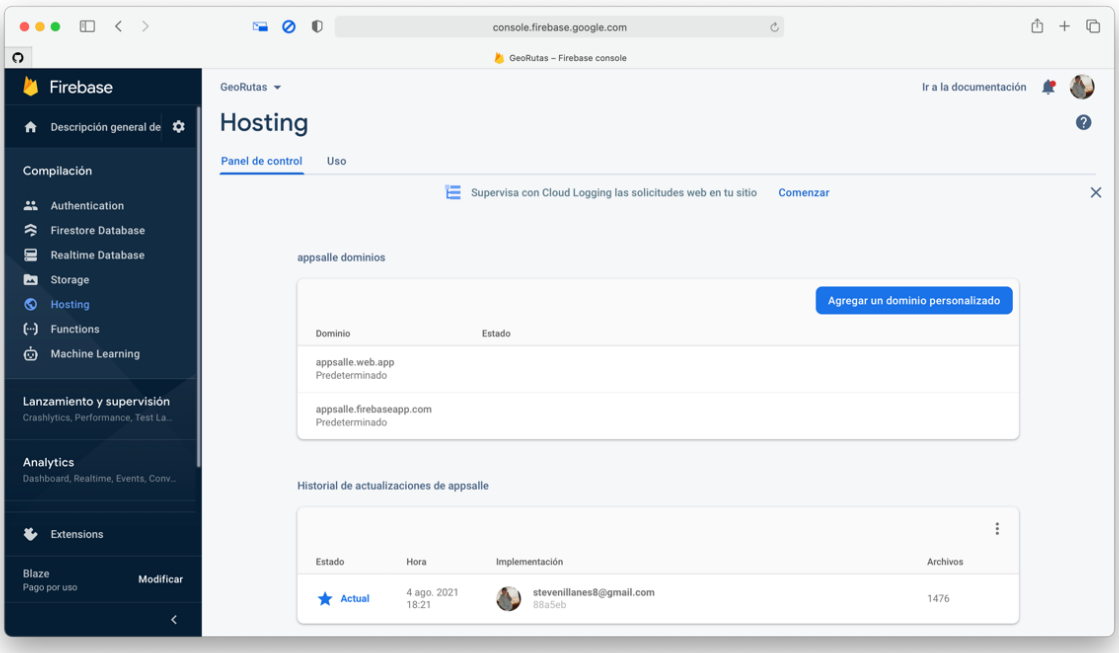


Imagen 9 Pantalla 1 Sprint 3 (Fuente: el autor)

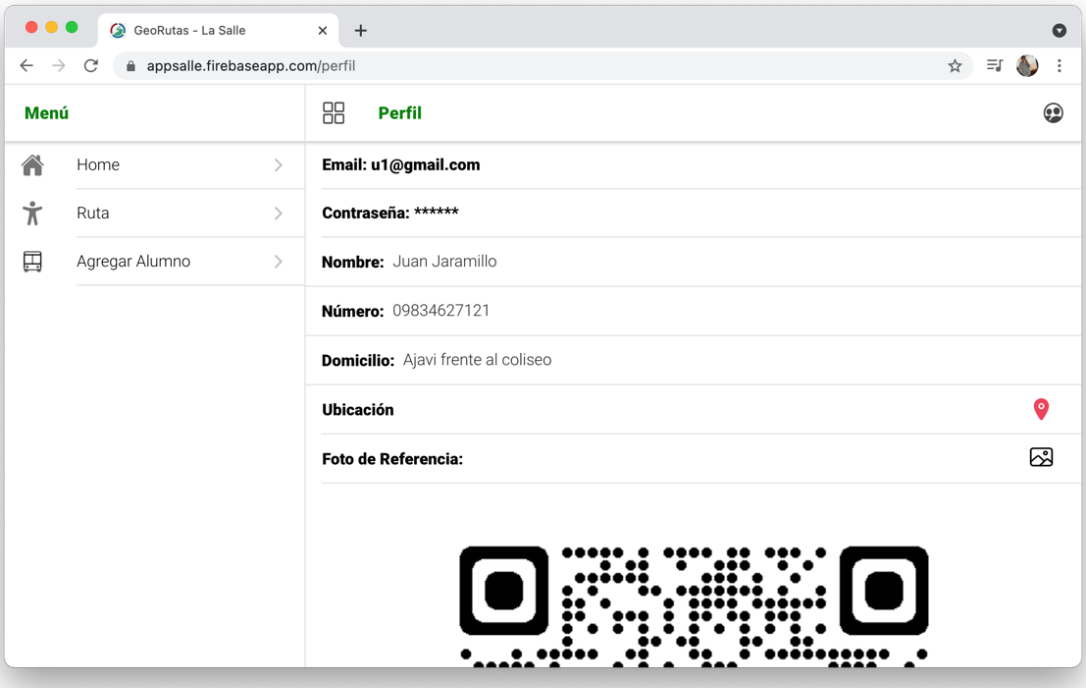


Imagen 10 Pantalla 2 Sprint 3 (Fuente: el autor)

Validación: Aprobado

Existe incremento: No

Aprobación del *Product Backlog*: Aprobado

5.6.5. *Sprint* 4

Descripción: Controlar todas las funcionalidades, las cuales se puedan llegar a necesitar para que la aplicación tenga un correcto funcionamiento.

Incremento del producto: En las imágenes 11 y 12 se muestra como se realizaron funciones que se necesitaron para el uso correcto uso del aplicativo.

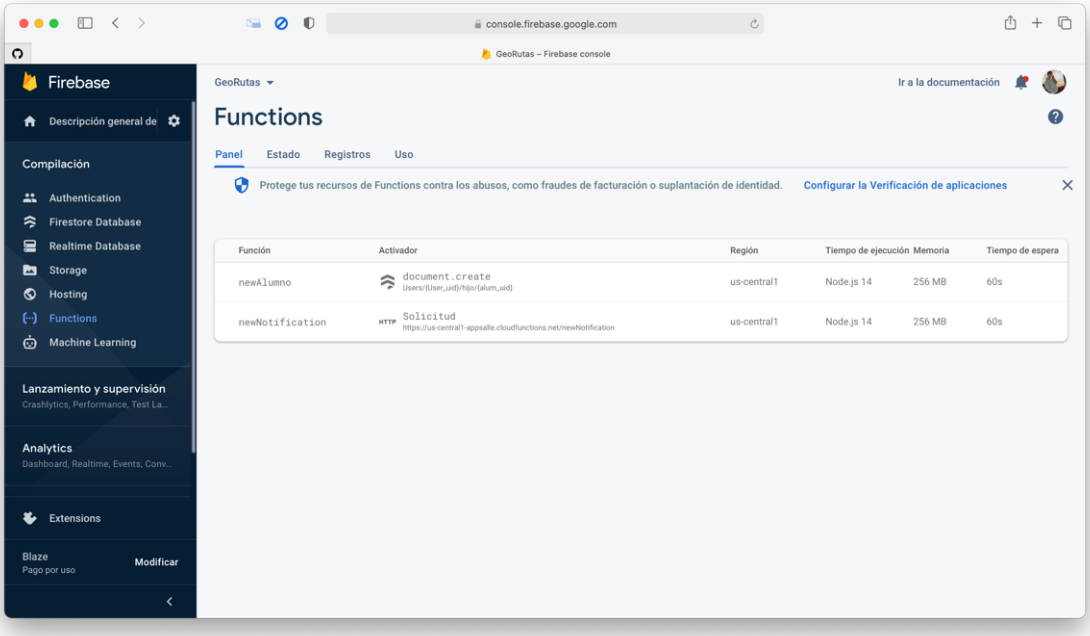


Imagen 11 Pantalla 1 Sprint 4 (Fuente: el autor)

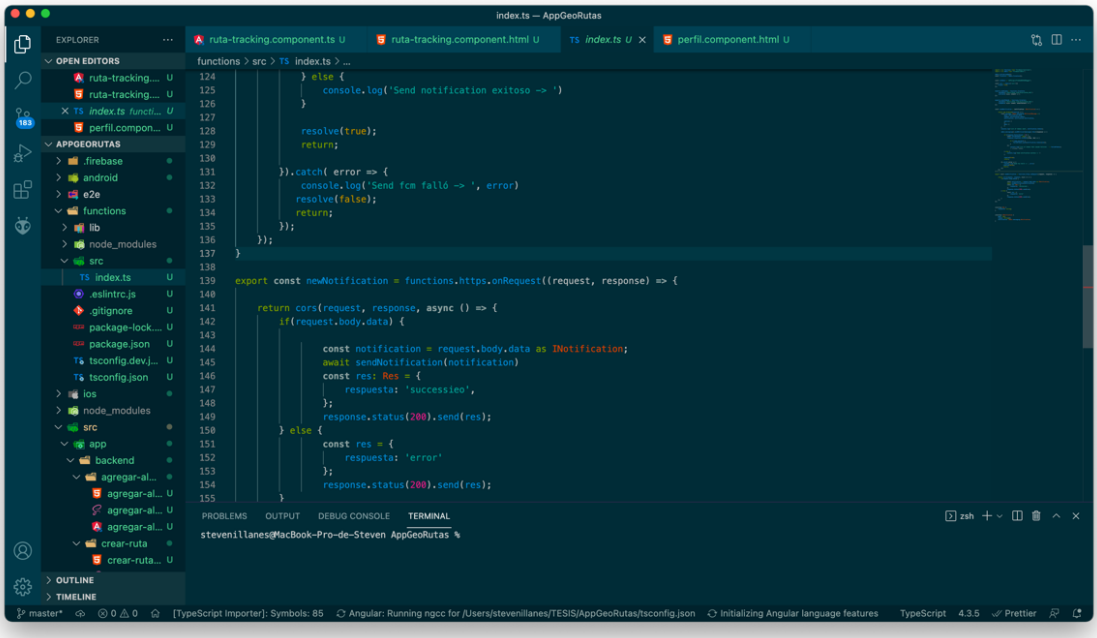


Imagen 12 Pantalla 2 Sprint 4 (Fuente: el autor)

Validación: Aprobado

Existe incremento: No

Aprobación del *Product Backlog*: Aprobado

5.6.6. *Sprint* 5

Descripción: Controlar el tiempo de refresco de la página con la ubicación de la ruta.

Incremento del producto: En la Imagen 13 y 14 se muestra como fue la implementación sobre el tiempo de refresco dentro del tracking en la ruta, así como también el código que indica cuánto tiempo le tomará refrescar y presentar la ubicación en tiempo real.

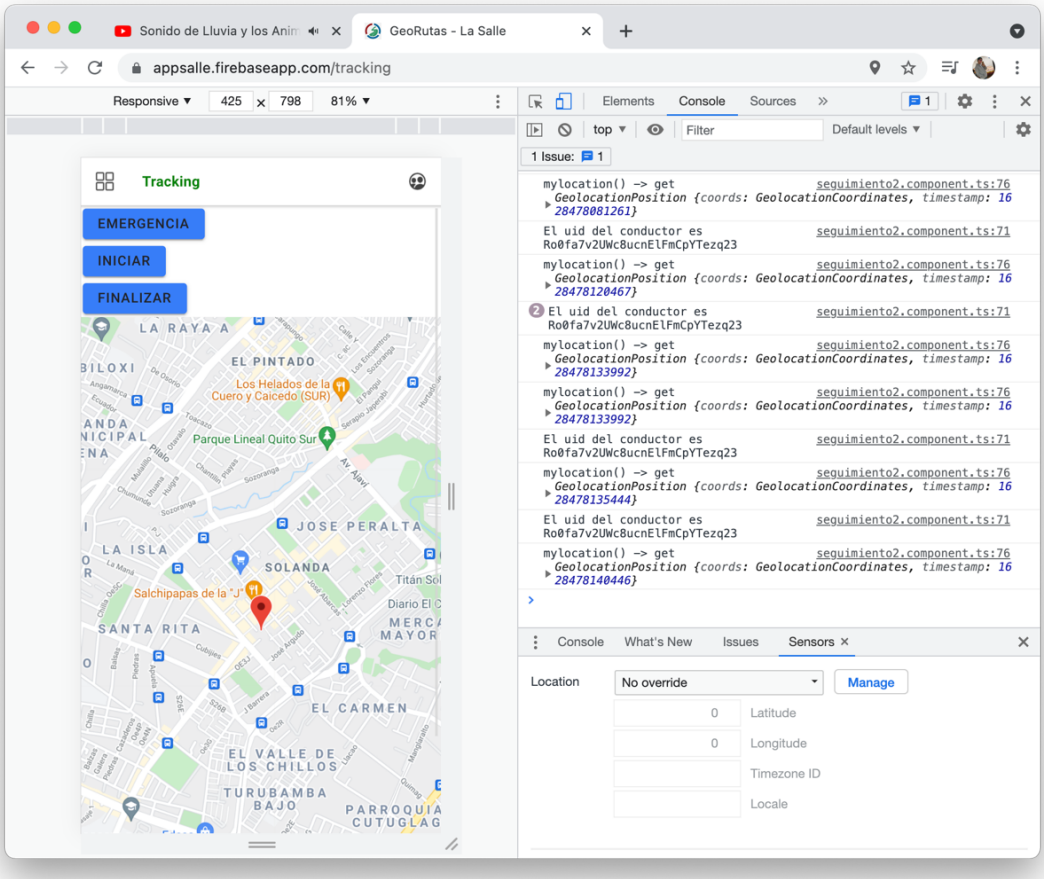


Imagen 13 Pantalla 1 Sprint 5 (Fuente: el autor)

```
suscriConductor(id_Cond:string){
  const pathCo ='Users/';

  this.TimerLocation = setInterval(() => {
    this.firestoreService.obtenerDocumento<any>(pathCo,id_Cond).subscribe(res =>{
      this.position = {
        lat: res.lat,
        lng: res.lng,
      }
      this.addMarker(this.position);
    });

  }, 5000);
}
```

Imagen 14 Código sobre tiempo (Fuente: el autor)

Validación: Aprobado

Existe incremento: No

Aprobación del *Product Backlog*: Aprobado

5.6.7. *Sprint* 6

Descripción: Controlar que el usuario ingrese de una correcta forma todos los datos los cuales sirvan para que se registre

Incremento del producto: En la Imagen 15 y 16 se muestra el resultado del *sprint*, donde los datos necesarios para la validación del registro como son, email, contraseña, domicilio, entre otros. Tienen el formato correcto para su correcto uso dentro de la base de datos del aplicativo.

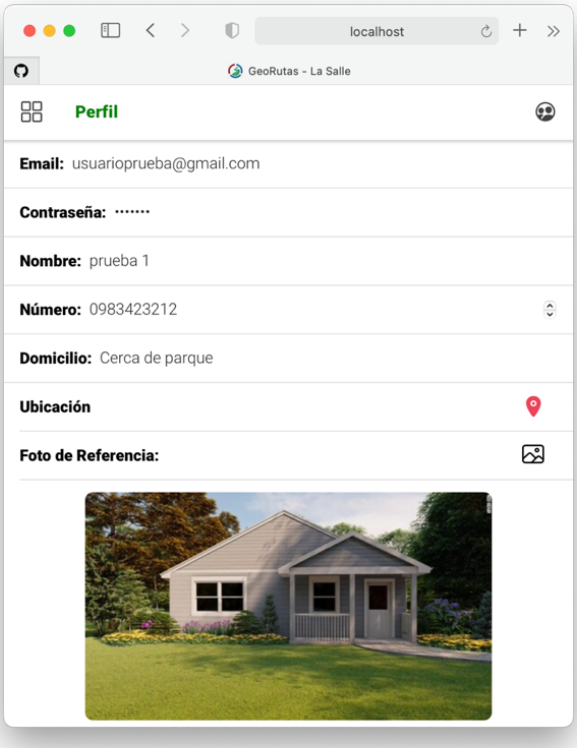


Imagen 15 Pantalla 1 Sprint 6 (Fuente: el autor)



Imagen 16 Pantalla 2 Sprint 6 (Fuente: el autor)

Validación: Aprobada

Existe incremento: No

Aprobación del *Product Backlog*: Aprobado

5.6.8. *Sprint* 7

Descripción: Controlar que el usuario pueda actualizar su información

Incremento del producto: En la Imagen 17 se muestra cómo puede actualizar el usuario su información personal, y se muestra como campos no cambiables, el *email* y contraseña de este ya que por seguridad. No se permite el cambio de estos campos.

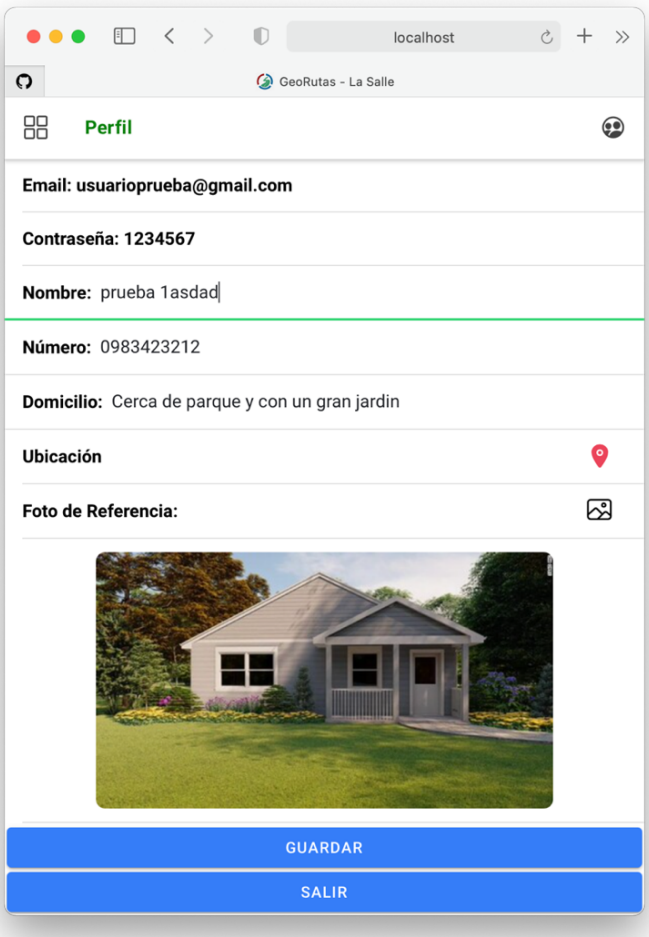


Imagen 17 Pantalla 1 Sprint 7(Fuente: el autor)

Validación: Aprobada

Existe incremento: No

Aprobación del *Product Backlog*: Aprobado

5.6.9. *Sprint* 8

Descripción: Controlar el envío de mensajes con el estado actual.

Incremento del producto: En la Imagen 18 y 19 se muestra cómo llegan las notificaciones sobre el estado actual de la ruta escolar, si se inició o finalizó la misma

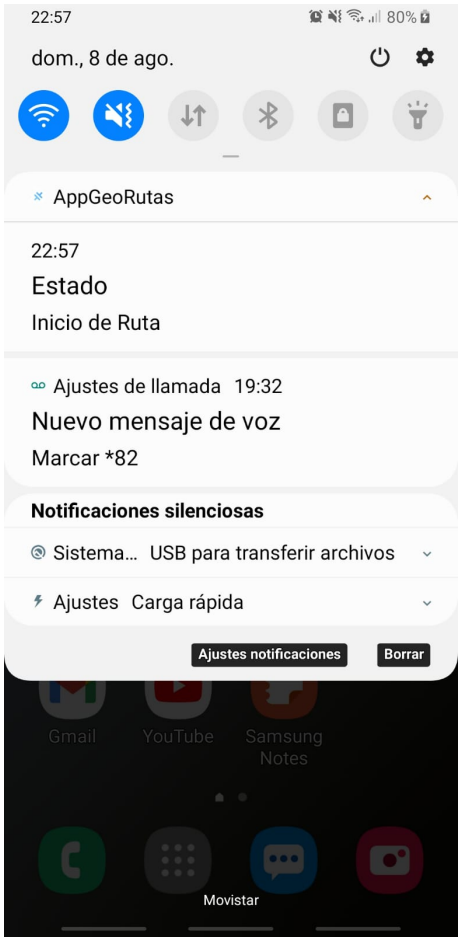


Imagen 18 Pantalla de Mensajes 1 Sprint 8 (Fuente: el autor)

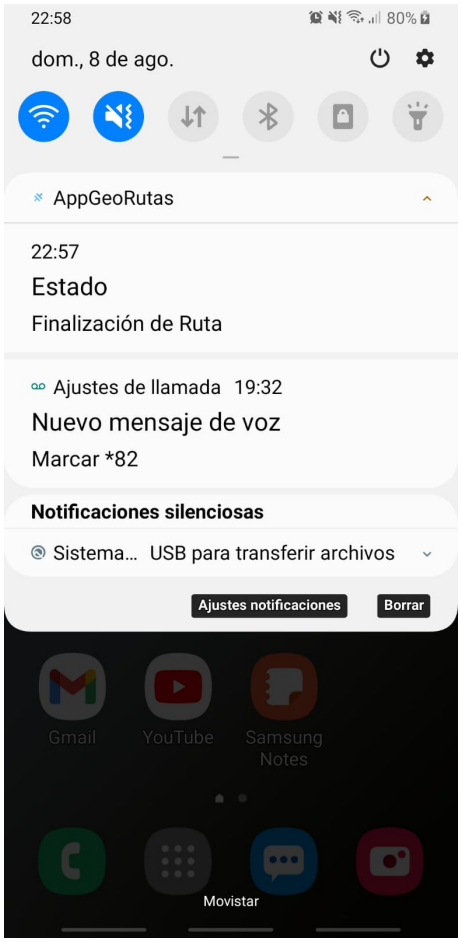


Imagen 19 Pantalla de mensajes 2 Sprint 8 (Fuente: el autor)

Validación: Aprobada

Existe incremento: No

Aprobación del *Product Backlog*: Aprobado

5.6.10. *Sprint* 9

Descripción: Controlar la comunicación en caso de contingencias.

Incremento del producto: En la Imagen 20 se muestra el resultado del *Sprint*, permitiendo comunicarse en caso de catástrofes con los padres de los estudiantes.

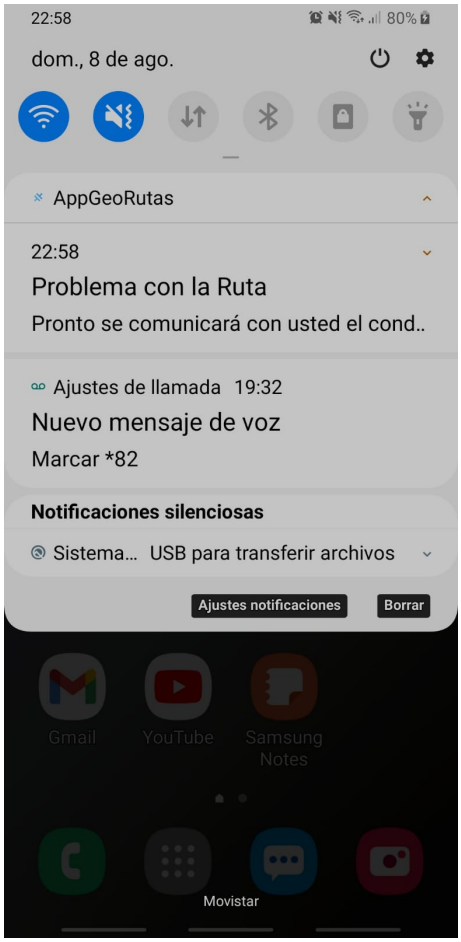


Imagen 20 Pantalla emergencia Sprint 9 (Fuente: el autor)

Validación: Aprobada

Existe incremento: No

Aprobación del *Product Backlog*: Aprobado

5.6.11. *Sprint* 10

Descripción: Controlar que el usuario pueda acceder al aplicativo

Incremento del producto: En la Imagen 21 y 22 se muestra el resultado de este *Sprint*, permitiendo el ingreso de una forma correcta por medio de correo electrónico y contraseña para acceder al inicio de sesión.

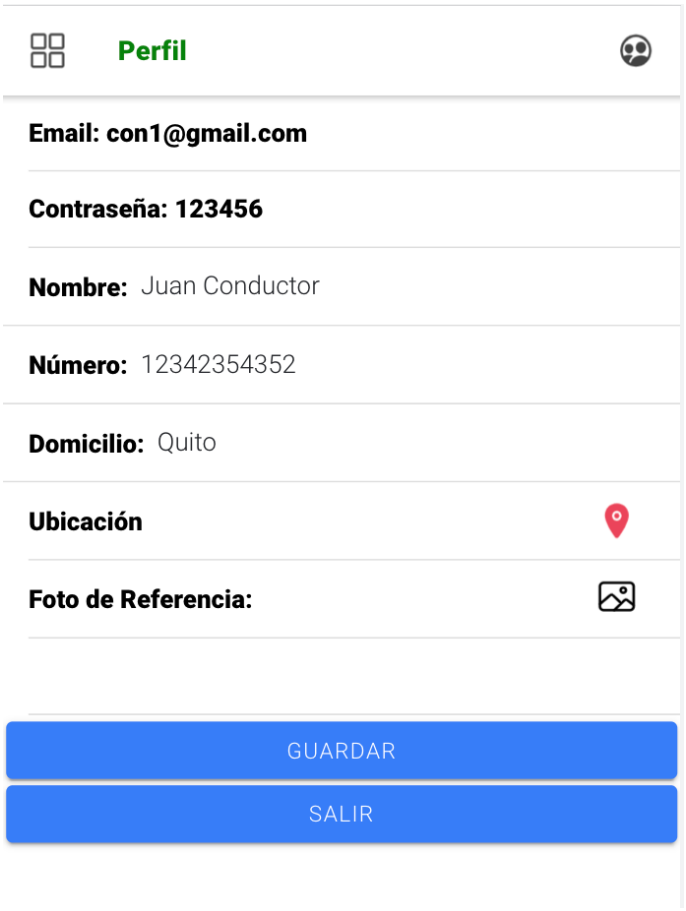


Imagen 21 Pantalla 1 Sprint 10 (Fuente: el autor)

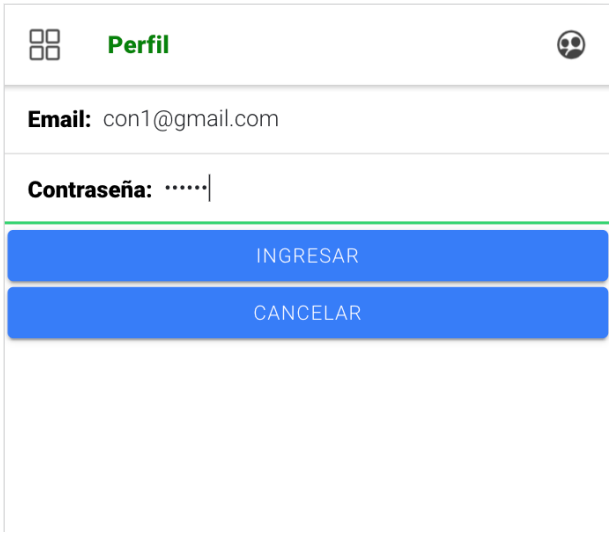


Imagen 22 Pantalla 2 Sprint 10 (Fuente: el autor)

Validación: Aprobada

Existe incremento: No

Aprobación del *Product Backlog*: Aprobado

5.6.12. *Sprint* 11

Descripción: Controlar que se puedan observar los datos de forma correcta hacia el usuario.

Incremento del producto: En la Imagen 23 se muestra cómo observa los datos ingresados por el usuario con anterioridad, así mismo, los datos proporcionados por la base de datos.

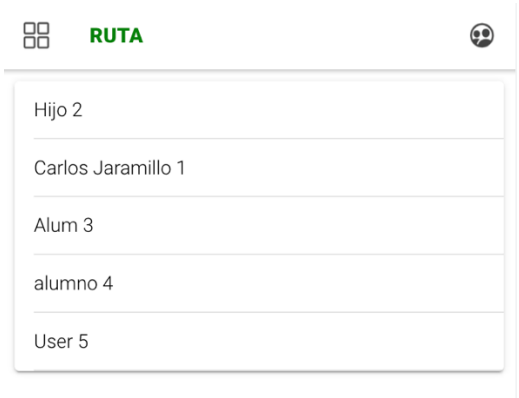


Imagen 23 Pantalla Sprint 11 (Fuente: el autor)

Validación: Aprobada

Existe incremento: No

Aprobación del *Product Backlog*: Aprobado

5.7. Etapa 5: Implementación

En esta fase únicamente se realizará la actividad de Planificación de implementación y no la actividad de Implementación de entregables, por el hecho de no existir cambios por parte de los usuarios al final de cada uno de los 11 *Sprints*.

5.7.1. Planificación de la implementación

Con la finalización de la etapa anterior y la retroalimentación dada con cada uno de los *Sprints* aprobados, se comenzó con la etapa de implementación donde se pondrán en práctica los aspectos necesarios para que el aplicativo web-móvil sea puesto en producción, y así se pueda interactuar con el cliente final. Algunos de los aspectos necesarios para la entrada a producción son:

5.7.1.1. Implementación del *BackEnd* y BDD

Para la realización de este requerimiento se implementó *Firebase* como *BackEnd* y también para la base de datos, se utilizó esta plataforma en la nube por el hecho de tener integrado varios servicios que fueron útiles en la realización de este proyecto, como son toda su plataforma de *BackEnd* y su base de datos "*Firestore Database*", gracias a estos servicios la integración con *Ionic* se realizó de una forma fácil y rápida. Además de utilizar librerías genéricas y otras concretas para determinados servicios se utilizó estas de manera independiente entre ellas.

- Para su integración, se creó un nuevo proyecto desde la página de *Firebase* y se configuró el SDK dentro de *Ionic*, copiando el siguiente comando:

```
const firebaseConfig = {

 apiKey: "AIzaSyBObry6cspY7nAKDBztxhpglpaOuA73Tac",
```

```
authDomain: "appsalle.firebaseapp.com",

projectId: "appsalle",

storageBucket: "appsalle.appspot.com",

messagingSenderId: "190605353305",

appId: "1:190605353305:web:f19dc99ccdd00e982a2c48",

measurementId: "G-8PXMM48VXP"

};
```

- Al crear la base de datos dentro de *Firebase*, y al no existir un esquema de base de datos, únicamente al ser una combinación de pares clave / valor. No existe una estructura definida de datos.
- Se utiliza la consola de *Firebase*, la cual nos ayuda a ver, agregar, editar y borrar datos. Asimismo, nos deja crear y actualizar las reglas de seguridad dentro de *Cloud Firestore*. Como se puede ver en la Imagen 24 la consola ayuda a filtrar documentos de una colección en este caso filtra los documentos dentro de la colección Alumnos. (Google, 2021)

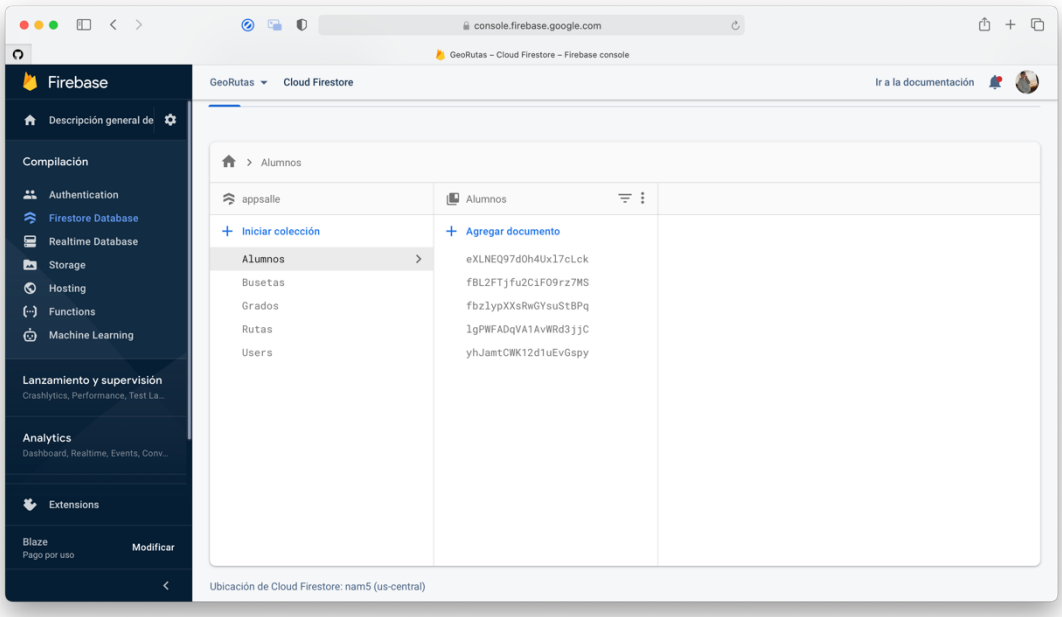


Imagen 24 Consola Firebase (Fuente: el autor)

5.7.1.2. Generación de las Apps para dispositivos móviles

En la Imagen 25 tenemos los comandos que se utilizaron para agregar las plataformas de *iOS* y *Android* a nuestro proyecto, ayudando de forma significativa a cumplir con esta implementación.

```
<code>
$ ionic cap add ios
$ ionic cap add android
</code>
```

Imagen 25 Creación proyectos móviles (Desplegando a iOS y Android, 2019)

Para la creación de las distintas plataformas se debe tener instalado previamente:

- *"Node.js"*
- *"npm"*
- *"ionic"*
- *"capacitor"*

Al utilizar *capacitor* para la creación de estos proyectos, se crean carpetas independientes para el desarrollo separado en los distintos sistemas operativos, como se puede ver en la Imagen 26 se crean las carpetas *Android* y *iOS*, para cada uno de los sistemas operativos. (Lynch, 2021)

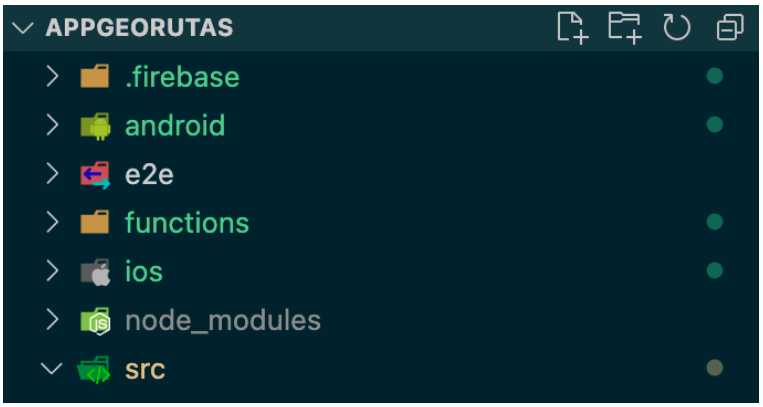


Imagen 26 Estructura del proyecto (Fuente: el autor)

> Cada vez que se realiza un cambio dentro del directorio del proyecto es necesario compilarlo, para que el directorio web se actualice y así todos los cambios hechos se puedan ver reflejados también en los proyectos nativos y no únicamente en la web. (Desplegando a iOS y Android, 2019)

5.8. Etapa 6: Cierre del Proyecto

El proyecto concluye con un **cierre exitoso**, ya que al momento todos los entregables que se planificaron en un comienzo tuvieron la recepción correcta y con el paso de todos los criterios de aceptación fueron aprobados.

En un futuro el proyecto podría tener una mejora, así como también la realización de otras implementaciones de nuevas funcionalidades que se obtendrán de una mejor manera con el paso del tiempo, y el uso del proyecto actual. (CertMind, 2019)

Para el presente proyecto lo que se busca en un futuro es una ampliación dentro de las funcionalidades, como el control de estudiantes presentes o ausentes dentro de la ruta de transporte escolar, un mejoramiento en la presentación de los mapas referentes al seguimiento de rutas, la comunicación directa sobre el administrador y los padres de familia dentro de la aplicación, entre algunas otras, las cuales se podrán obtener con el paso del tiempo y el uso que se dé a este sistema.

6. Pruebas

6.1. Pruebas de Usabilidad

Para realizar las pruebas de usabilidad es necesario saber primero qué se realiza en estas pruebas, aquí se ve la capacidad que tiene un producto de software de ser aprendido y entendido, así como cuán atractivo es para el usuario que lo utilizará. (Layedra Larrea, Ramos Valencia, Hidalgo Ponce, & Samaniego Orozco, 2019) Una de las formas de probar la usabilidad es mediante la observación de las personas mientras utilizan este producto y su futura evaluación de desempeño en el uso.

Para probar la usabilidad se utilizó la evaluación heurística, por el hecho de que esta evaluación ayuda a estimar de manera más rápida las necesidades que presenta la interfaz hacia el usuario final. Se pretende cumplir con los principios heurísticos de Jakob Nielsen, como nos dice (Tepliski, 2005) en su resumen de estos principios, no son ni únicos ni excluyentes.

- **Visibilidad del estado del sistema:** Para poder validar este punto se propuso poner un banner que indique en qué parte del sistema está, ayudando de esta manera al usuario a ubicarse dentro del sistema.

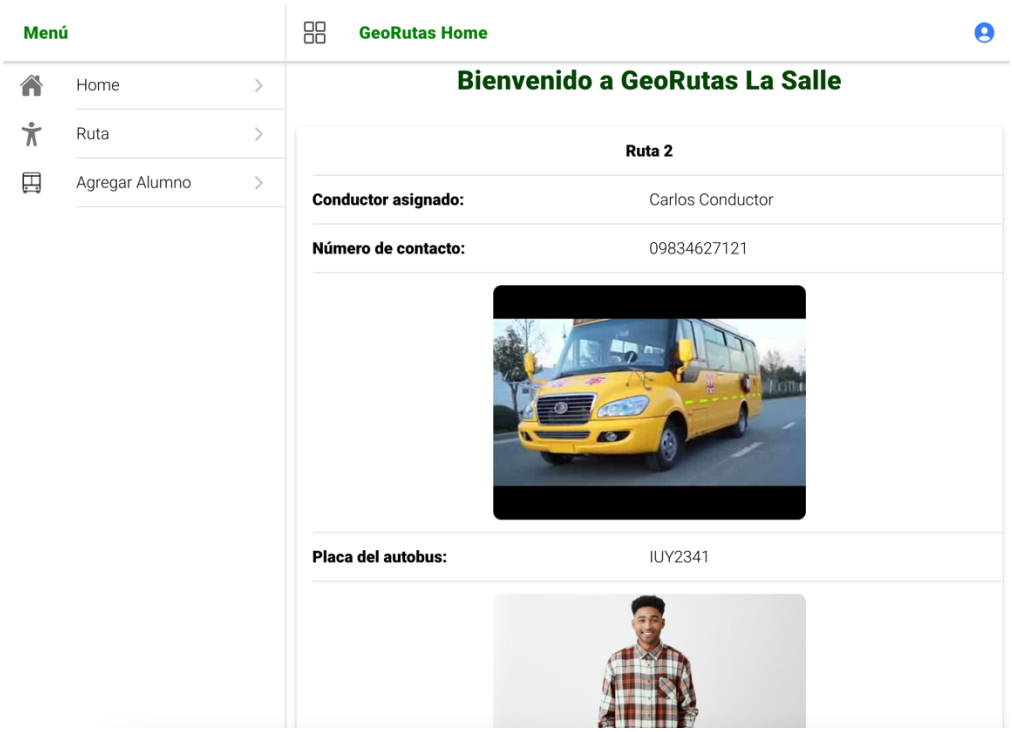


Imagen 27 Banner utilizado en el sistema (Fuente: el autor)

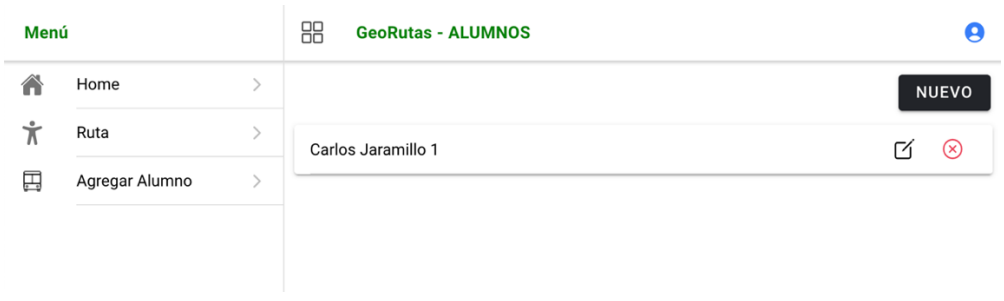


Imagen 28 Banner de la pestaña Agregar Alumnos (Fuente: el autor)

- **Coincidencia entre el sistema y el mundo real:** Para una mejor familiarización del usuario con el sistema se usan palabras naturales o familiares para él, así evitando el uso de términos complejos y que se sientan mucho más acostumbrados.

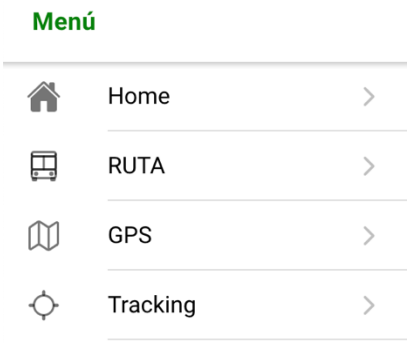


Imagen 29 Menú lateral con palabras propias del usuario (Fuente: el autor)

En lugar de usar palabras propias del sistema como son "Geolocalización", los usuarios están más acostumbrado a llamarlo GPS, así también como el uso de "*Tracking*" el cual ya estaban acostumbrados anteriormente.

- **Control y libertad del usuario:** En caso que el usuario se equivoque al ingresar o seleccionar un dato importante para el sistema, este puede darle la oportunidad de rectificar en caso que suceda, para así evitar fallos innecesarios dentro del sistema.

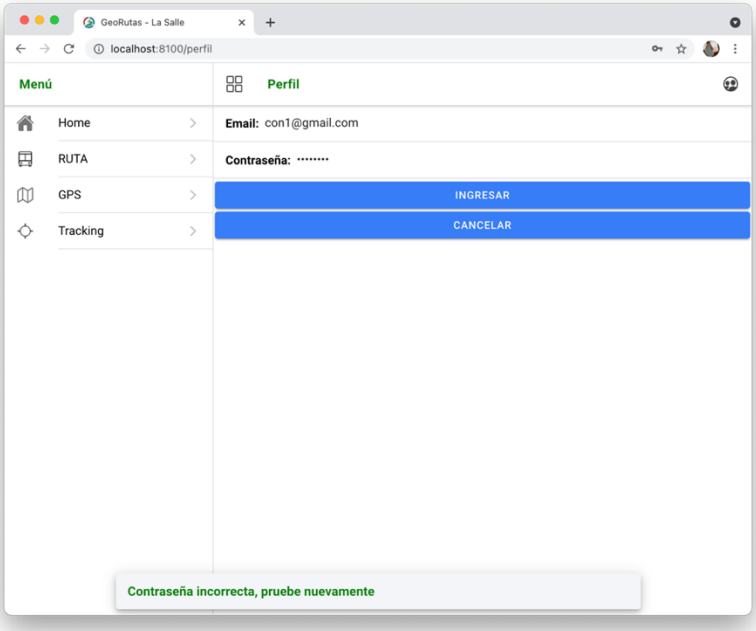


Imagen 30 Caso contraseña incorrecta (Fuente: el autor)

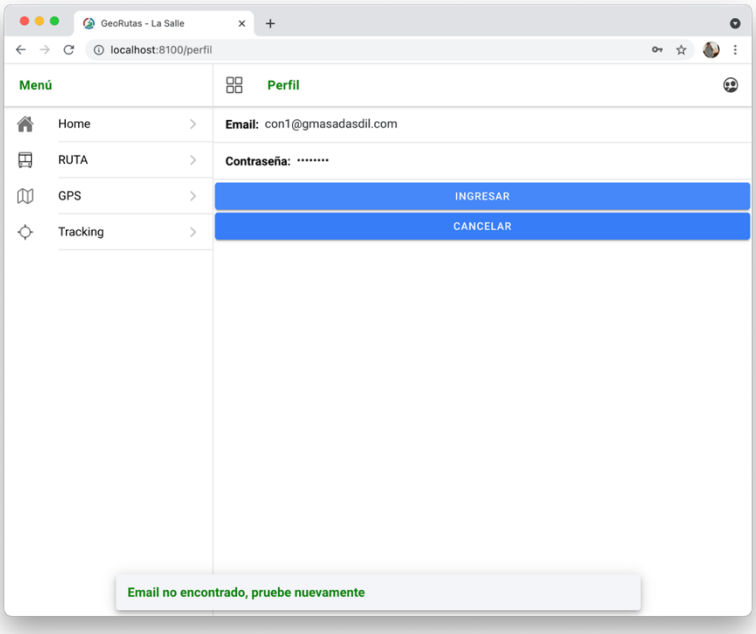


Imagen 31 Caso email inexistente (Fuente: el autor)

- **Coherencia y estándares:** Los patrones que se utilizan para que los usuarios se familiaricen se mantienen durante todo el sistema.

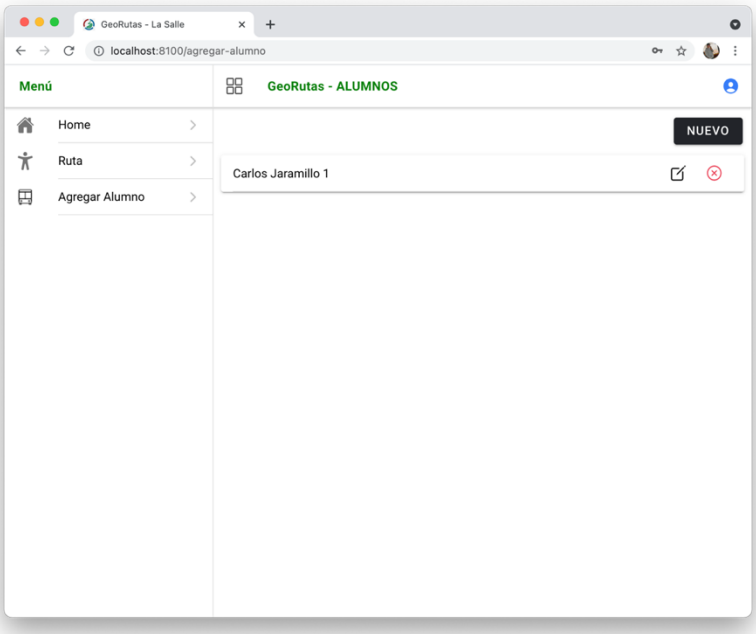


Imagen 32 Estructura usada (Fuente: el autor)

Por ejemplo, la integración de un menú lateral en todo el sistema, así como también un banner en el cual siempre está disponible un botón que los lleva de vuelta a su perfil.

- **Flexibilidad y eficiencia de uso:** Para diferenciar los usuarios habituales a los visitantes por primera vez, lo que se pretende es usar la misma interfaz y únicamente adaptar diferencias para que exista una eficiencia dentro del uso.

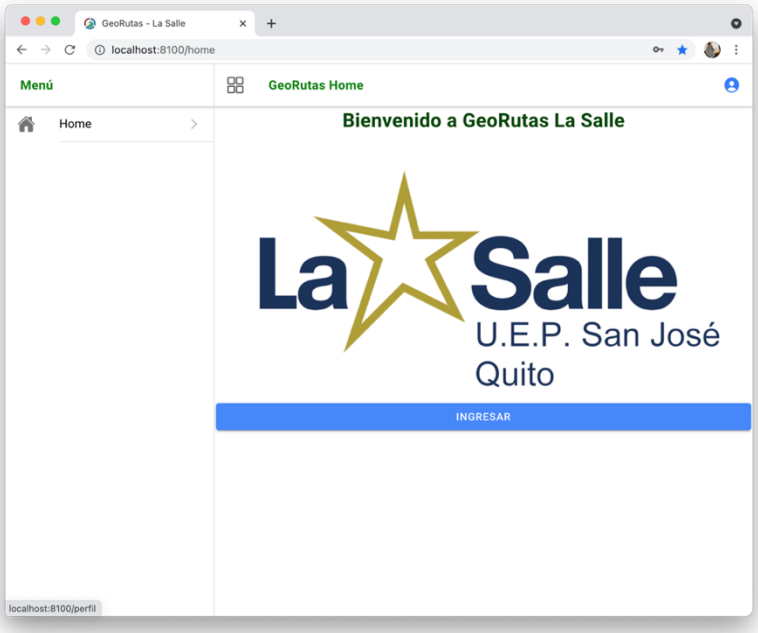


Imagen 33 Home usuario no registrado (Fuente: el autor)

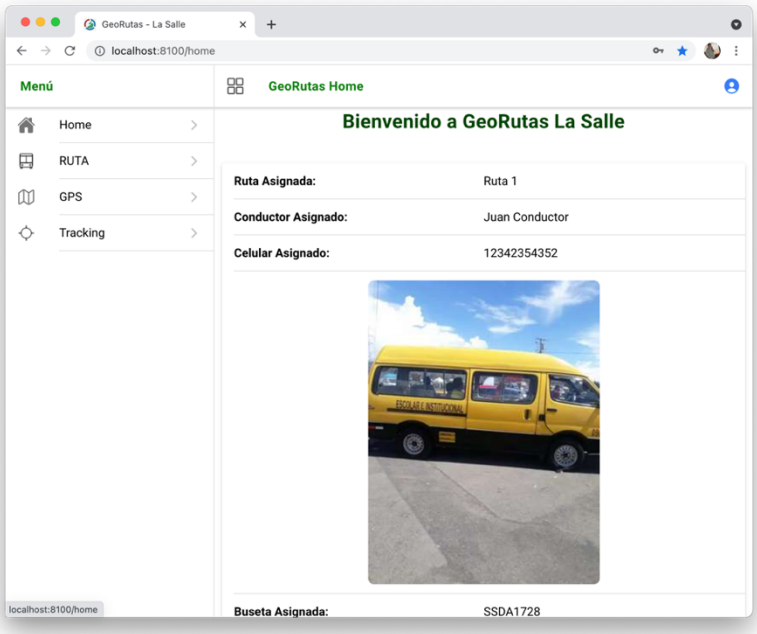


Imagen 34 Home Conductor (Fuente: el autor)

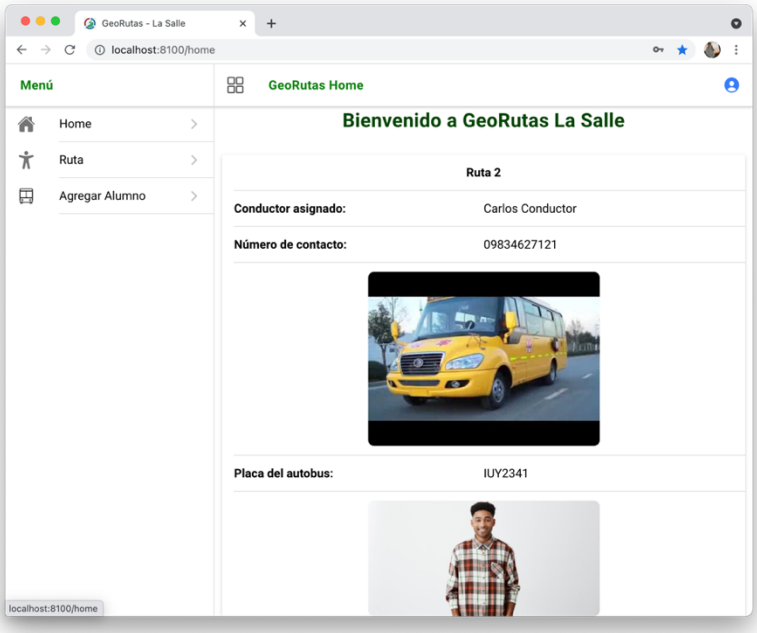


Imagen 35 Home Padre (Fuente: el autor)

- **Estética y diseño minimalista:** Para evitar una sobresaturación de información en la pantalla, se incluye únicamente lo necesario, pero en caso de necesitar una mayor información esta se presenta sin ser excesiva para el usuario, en un diseño simple y minimalista.

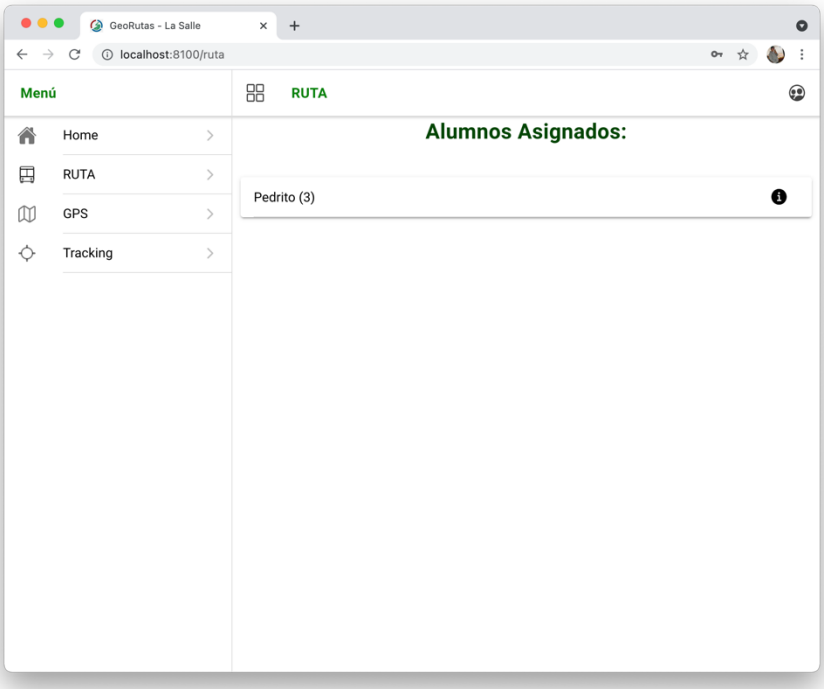


Imagen 36 Pantalla Ruta de conductor (Fuente: el autor)

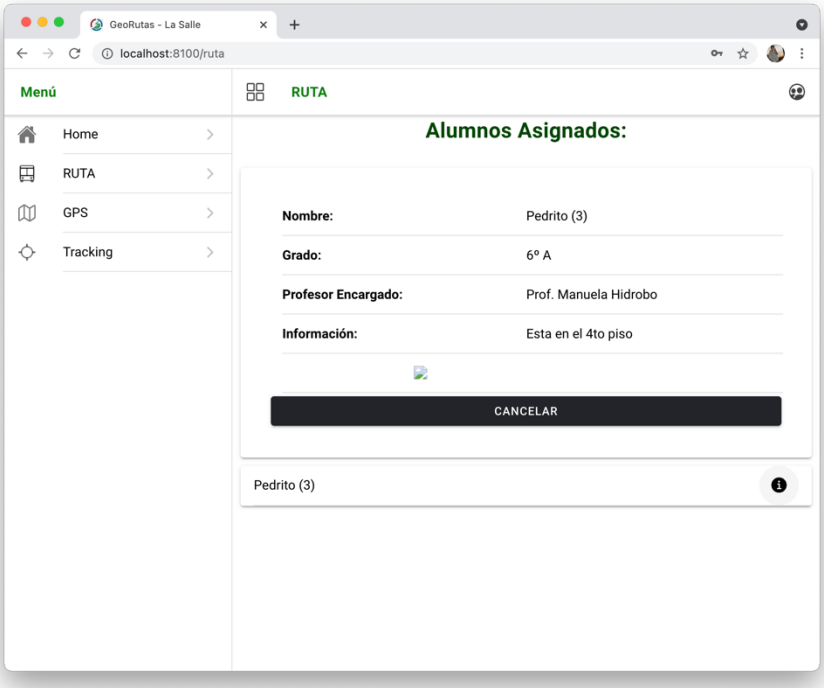


Imagen 37 Pantalla Ruta de conductor información (Fuente: el autor)

Como se puede observar en la Imagen 36 existe un diseño minimalista en donde no es necesario presentar una interfaz saturada de información, si es que es necesario se puede ampliar dicha información como se observa en la Imagen 37.

- **Ayudar al usuario a reconocer errores, diagnosticarlos y recuperarse de ellos:** En caso de existir problemas dentro del sistema, el usuario debe mantenerse informado de la situación en la que se encuentra. Si existe algún inconveniente él debe estar instruido sobre qué está pasando.

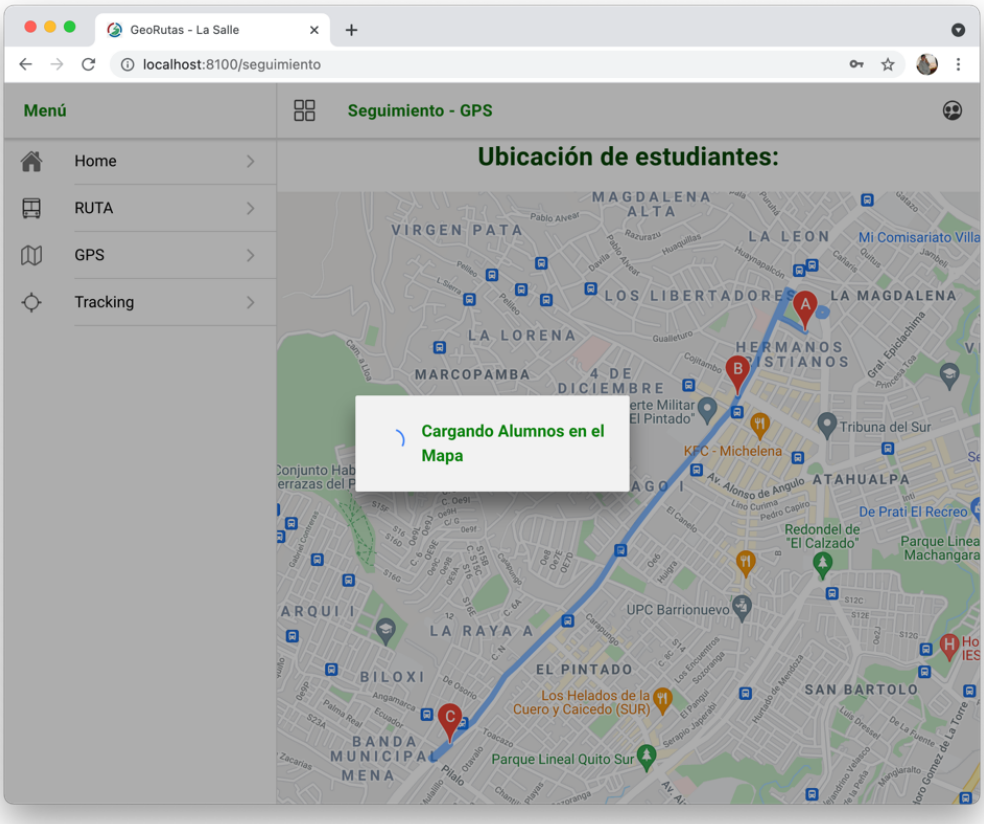


Imagen 38 Carga de Ruta Óptima (Fuente: el autor)

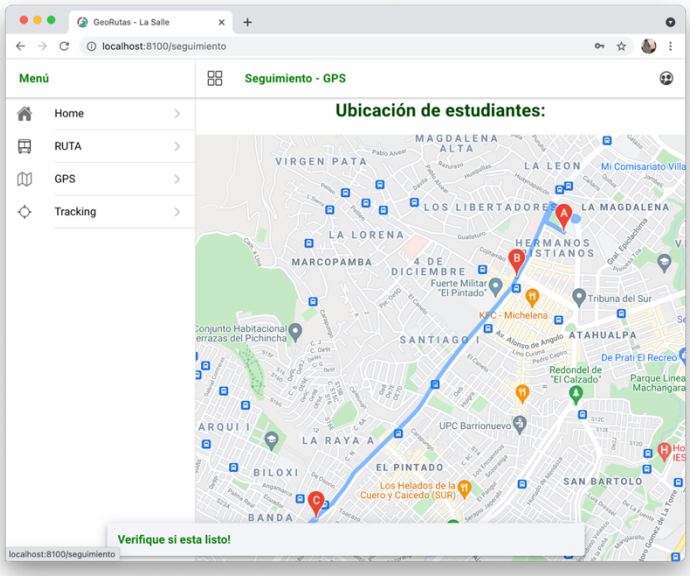


Imagen 39 Carga lista de Ruta Óptima (Fuente: el autor)

En la Imagen 38 se puede ver el *Loading* antes de la verificación correcta de este para la presentación del *Toast* que se observa en la Imagen 39. Así facilitando lo que está sucediendo siempre al usuario.

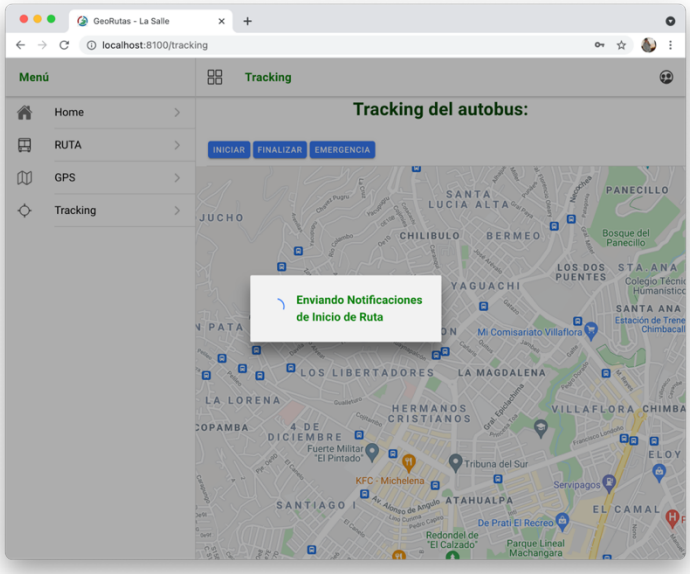


Imagen 40 Carga Notificaciones Inicio de Ruta (Fuente: el autor)

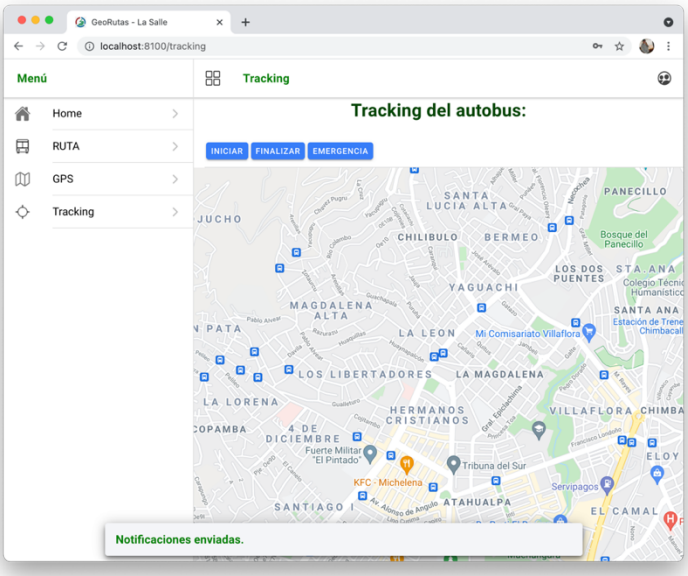


Imagen 41 Notificaciones enviadas Inicio de Ruta (Fuente: el autor)

En la Imagen 40 se puede ver el *Loading* antes de la verificación correcta de este para la presentación del *Toast* que se observa en la Imagen 41. Indicando al usuario sobre el envío correcto de notificaciones.

- **Ayuda y documentación:** Al no ser un sistema complejo de usar, no es necesario la construcción de una documentación, ya que toda la estructura utilizada dentro de este sistema es intuitiva y fácil de localizar. Como nos dicen (Rodríguez Martín, 2020) el uso de una iconografía y minimalismo dentro de la aplicación es indispensable para que pueda ser considerada intuitiva, ya que al estar centrado en el usuario se busca que la aplicación sea sencilla.

  Un ejemplo para saber si esta interfaz está diseñada de esta manera, es dando clic sobre algún botón, este debe indicar al usuario que se hizo clic sobre él. También los autores nos dicen que para que una aplicación sea fácil al navegar debemos lograr que el usuario conserve la atención necesaria y evite distraerse hacia otras cosas. En el proyecto se logra lo mencionado, evitando poner objetos que distraigan al usuario de forma innecesaria. Además, para lograr una mejor experiencia intuitiva hacia el usuario lo que se hace es que cada pantalla que se diseñó soporta una sola acción de valor real para el usuario, al hacer esto ayuda a

que sea más fácil de aprender, más fácil de usar y más fácil de agregar (Álvarez Muñoz, 2017).

6.2. Pruebas de Funcionalidad

Como nos mencionan los autores (Layedra Larrea, Ramos Valencia, Hidalgo Ponce, & Samaniego Orozco, 2019) para este tipo de pruebas se encargan de verificar si las funcionalidades desarrolladas en el sistema cumplen con sus especificaciones. Este tipo de pruebas se basan en los requerimientos funcionales. Para este tipo de pruebas se utilizarán las pruebas de caja negra por considerarlo como comportamiento externo del producto de software. (KIBERNUM CHILE, 2015)

Las pruebas de caja negra es una técnica en la cual la funcionalidad se verifica sin tomar en cuenta la estructura interna del código, detalles de implementación o escenarios de ejecución internos del software. (Layedra Larrea, Ramos Valencia, Hidalgo Ponce, & Samaniego Orozco, 2019)

7. Conclusiones y Recomendaciones

7.1. Conclusiones

- Las pruebas realizadas de usabilidad y de funcionalidad, permitió llegar a la conclusión de que la aplicación se desarrolló de manera satisfactoria, al usar tanto móvil como web para una gestión correcta del transporte escolar de la Unidad Educativa San José La Salle.
- La metodología escogida, SCRUM resultó ser adecuada para el desarrollo del proyecto, porque al permitir un desarrollo ágil se logró satisfacer las necesidades, así como los requerimientos dados por el cliente. Esta metodología ayudó mucho con la retroalimentación continua y la adaptabilidad que se logró con la aplicación, hizo que los entregables sean de una gran calidad. Además de analizar las necesidades que tenía el cliente se logró asignar de manera correcta el rol de los usuarios que intervienen dentro del transporte escolar.
- Se concluye que el cliente es una parte fundamental en el desarrollo del proyecto, dado que la retroalimentación, sobre todo en la fase de requerimientos ayudaron

a completar el proyecto de manera satisfactoria. Con un sistema previamente usado, se tenía la experiencia necesaria y las funcionalidades que carecía el anterior sistema y las que se necesitaban el sistema que se iba a desarrollar claras, esto ayudó a tener una mayor rapidez al completar el proyecto.

- El uso de *Firebase* fue de mucha utilidad, porque se necesitó compartir datos en tiempo real, como en el momento de enviar la ubicación en tiempo real de las unidades, así como, la escalabilidad que existe dentro de este *BackEnd* que fácilmente se adaptan a las necesidades del proyecto desarrollado. *Firebase* también cuenta con varios servicios que se usaron para facilitar el desarrollo como son el *hosting* y su base de datos.
- El uso de las APIs de Google, sobretodo en la API de Google *Maps* dentro del proyecto permitió comprobar lo poderosa que puede ser esta API, sobre todo en las principales funcionalidades como: la utilización de marcadores, asignación de ruta por trayectos óptimos y trazo de sectores dentro de un mapa. Se concluye que esta API ayudó a una mejor eficiencia dentro del proyecto, por la implementación de los marcadores, para el uso correcto de la ubicación del usuario, así como también del trayecto que se genera por las rutas óptimas dentro del perfil del conductor.
- El uso de Firebase ayuda que, en un futuro con un posible crecimiento del sistema, se pueda escalar de una forma fácil sin afectar el rendimiento de este. Sin la necesidad de cambiar por completo el diseño del sistema. Únicamente se debe analizar dentro de los planes que ofrece *Ionic* cual sería el correcto para el uso simultáneo de usuarios y el espacio de almacenamiento que ellos usarán.
- El uso de *Ionic* para la creación de aplicaciones multiplataforma fue una de las mejores elecciones para un desarrollo rápido, únicamente se tuvo en cuenta con qué lenguaje se desarrollará, ya qué *Ionic* utiliza tecnologías web como son *Angular, ReactJS y Vue*; y está basado tanto en *Typescript, ES6, HTML y SCSS*. Si tiene experiencia con algunos de los lenguajes previamente mencionados se puede avanzar mucho más rápido. *Ionic* ayudó con una implementación en distintas plataformas como son *iOS* y *Android* permitiendo reducir tiempos y costes del desarrollo, pues se necesitó solo la programación en un solo lenguaje para la creación de ambas plataformas y la web. Toda la documentación necesaria para su correcto uso se encuentra en la página oficial de *Ionic*.

7.2. Recomendaciones

- Se recomienda el uso de un único perfil por estudiante. Para evitar los sobrecostes de archivos multimedia al momento de subir las mismas fotografías de referencia de la ubicación del hogar.
- Se recomienda que la Unidad Educativa San José La Salle ubicada en la Magdalena digitalice la información de todos sus estudiantes, para así agilizar todos los procesos que se llevan a cabo para el correcto funcionamiento del aplicativo.
- Para el correcto uso del sistema, se recomienda usar navegadores web actualizados a su última versión para evitar posibles incompatibilidades sobre todo con el diseño.
- Se recomienda para futuras implementaciones del API de Google *Maps* y en general toda API que sea proporcionada por Google, buscar documentación dentro de la página oficial de APIs, dado que muchas veces en otras páginas existe información desactualizada y con el continuo desarrollo de las APIs, se puede llegar a perder tiempo valioso en la búsqueda de esta información.
- Para evitar incompatibilidades dentro del desarrollo en dispositivos móviles y la web, se recomienda usar el módulo de *JavaScript* dentro del API de Google *Maps* y el módulo de Geolocation, así como también tener una *key* dentro del perfil de desarrollo que tenga habilitados estos módulos para que así no existan problemas a futuro.
- Como trabajo futuro se pretende analizar de una manera exhaustiva todas las ventajas que presenta el API de Google *Maps*, puesto que en el presente proyecto se usaron funcionalidades elementales de esta API, pero se pueden usar funcionalidades mucho más avanzadas, pero a su vez más complejas, para lograr mejorar la eficiencia de estas dentro de los procesos.

8. Bibliografía

Notimundo. (15 de Agosto de 2019). *Niveles de inseguridad de Quito*. Obtenido de NotiMundo: https://notimundo.com.ec/niveles-de-inseguridad-de-quito/

NotiMundo. (5 de Febrero de 2020). *Quito registra 647 delitos en lo que va del 2020*. Obtenido de NotiMundo: https://notimundo.com.ec/quito-registra-647-delitos-en-lo-que-va-del-2020/

Malave Polanco, K., & Beauperthuy Taibo, J. L. (2011). "ANDROID" GOOGLE'S OPERATING SYSTEM FOR MOBILE DEVICES. *Revista Científica Electrónica Ciencias Gerenciales*, 79-96.

Salazar Ospina, Y. S. (2017). *Análisis comparativo entre los sistemas operativos de dispositivos móviles con mayor demanda en el mercado 2015-2016.* Universidad Tecnológica de Pereira, Facultad de Ingenierias: Electrica, electrónica, física y ciencias de la computación, Pereira.

Mena Roa, M. (30 de Julio de 2020). *Android e iOS dominan el mercado de los smartphones*. Obtenido de Statista: https://es.statista.com/grafico/18920/cuota-de-mercado-mundial-de-smartphones-por-sistema-operativo/

González, M. A. (12 de Febrero de 2020). *Las deficiencias del Gobierno contribuyen a la inseguridad*. Obtenido de Primicias: https://www.primicias.ec/noticias/politica/gobierno-inseguridad-migracion-tapar-deficiencias/

Aponte Gomez, S., & Davila Ramirez, C. (2011). *SISTEMAS OPERATIVOS MOVILES: FUNCIONALIDADES, EFECTIVIDAD Y APLICACIONES UTILES EN COLOMBIA.* Tesis previo a Ingenieria en Sistemas, Bogota.

Enriquez, J. G., & Casas, S. I. (2014). *Usabilidad en aplicaciones móviles.* Universidad Nacional de la Patagonia Austral. Argentina: ICT.

Equipo técnico Encuesta Nacional Multipropósito de Hogares. (Agosto de 2020). *INEC.* Obtenido de Tecnologías de la Información y Comunicación-TIC: https://www.ecuadorencifras.gob.ec/documentos/web-inec/Estadisticas_Sociales/TIC/2019/201912_Boletin_Multiproposito_TIC.pdf

Rodriguez, C., & Dorado, R. (2015). ¿Por qué implementar Scrum? *Ontare, 3*, 125-144.

Trigas Gallego, M. (2012). *Desarrollo detallado de la fase de aprobación de un proyecto informático mediante el uso de metodologías ágiles.* TFC.

CertMind. (2019). SCRUM An Agile Approach to Manage Successful Projects. Miami, Miami, Estados Unidos.

LogicFast. (16 de Noviembre de 2020). *LogicFast Ecuador*. Obtenido de Logic Fast: http://geologic.logicfast.ec

Desplegando a iOS y Android. (23 de Enero de 2019). *Ionic Framework.* Obtenido de Docs - Ionic Framework: https://ionicframework.com/jp/docs/es/angular/your-first-app/6-deploying-mobile

Lynch, M. (19 de Mayo de 2021). *Announcing Capacitor 3.0*. Obtenido de Ionic: https://ionicframework.com/blog/announcing-capacitor-3-0/?__hstc=213394507.1afb644526060533038575f5e200f310.1629171579597.1629171579597.1629171579597.1&__hssc=213394507.2.1629171579597&__hsfp=3064399151&_gl=1*163zhos*_ga*ODg2NjUzNzU1LjE2MjkxNzE1Nzk.*_ga_R

Tepliski, F. A. (2005). Las claves de la usabilidad, los gurúes Nielsen y Krug. En F. A. Tepliski, *Las claves de la usabilidad, los gurúes Nielsen y Krug* (págs. 100-500). Chile: Biblioteca del Congreso Nacional de Chile.

Layedra Larrea, N. P., Ramos Valencia, M. V., Hidalgo Ponce, B. F., & Samaniego Orozco, A. E. (Septiembre de 2019). *Análisis de la aplicación de pruebas funcionales y pruebas de usabilidad de software en el desarrollo de sistemas web.* Obtenido de Ciencia Digital: https://doi.org/10.33262/cienciadigital.v3i3.4.845

KIBERNUM CHILE. (2015). *¿Por qué son importantes las pruebas funcionales?* Obtenido de ¿Por qué son importantes las pruebas funcionales?: https://www.kibernum.com/2020/08/20/por-que-son-importantes-las-pruebas-funcionales/

Rodríguez Martín, C. (15 de Junio de 2020). *Vegan Land - Diseño UX/UI de una App.* Obtenido de Repositorio Institucional O2: http://hdl.handle.net/10609/119046

Álvarez Muñoz, C. (30 de Marzo de 2017). *OpenWebinars*. Obtenido de Principios de diseño de interfaz de usuario: https://openwebinars.net/blog/principios-de-diseno-de-interfaz-de-usuario/

Google. (26 de Agosto de 2021). *Administra Cloud Firestore con Firebase console*. Obtenido de Firebase: https://firebase.google.com/docs/firestore/using-console?hl=es