

# **ESCUELA POLITÉCNICA NACIONAL**

## **FACULTAD DE INGENIERÍA EN SISTEMAS**

### **DESARROLLO DE MÓDULOS DE APLICATIVOS MÓVILES DE SISTEMA ERP UTILIZANDO PRÁCTICAS DEVOPS SEGURAS**

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

**JARAMILLO TORRES CHRISTOPHER LEONARDO**

christopher.jaramillo@epn.edu.ec

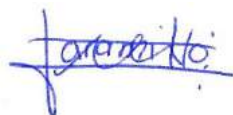
**DIRECTOR: Ing. Adrián Egüez, Msc**

adrian.eguez@epn.edu.ec

**DMQ, m**

## CERTIFICACIONES

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



---

**Christopher Leonardo Jaramillo Torres**

Certifico que el presente trabajo de integración curricular fue desarrollado por Christopher Leonardo Jaramillo Torres, bajo mi supervisión.



---

**MSc. Vicente Adrián Egüez  
Sarzosa  
DIRECTOR**

## **DECLARACIÓN DE AUTORÍA**

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

Christopher Leonardo Jaramillo Torres

Vicente Adrián Egüez Sarzosa

## DEDICATORIA

Dedico el esfuerzo y los resultados de este trabajo a mi familia, amigos y a todas las personas que me apoyaron en el proceso y confiaron en el éxito que representa este trabajo.

“Si puedes hacer un montón con todas tus ganancias  
y arriesgarlo a un golpe de azar,  
y perder, y empezar de nuevo desde el principio  
y no decir nunca una palabra acerca de tu pérdida [...]  
Tuya es la Tierra y todo lo que contiene,  
y —lo que, es más— ¡serás un Hombre, hijo mío!”

Rudyard Kipling

## **AGRADECIMIENTO**

Agradezco a mi mamá, mis hermanos y toda mi familia por estar presentes en todo mi recorrido, por apoyarme incondicionalmente y siempre darme la alegría y el entusiasmo necesario para poder culminar con este proceso.

Agradezco el acompañamiento y apoyo de mis amigos y compañeros los cuales son una parte fundamental para culminar con mi formación profesional.

Agradezco a mi universidad por la gran experiencia de vivir el proceso de convertirme en un profesional de mi carrera.

Christopher

## ÍNDICE DE CONTENIDO

1	DESCRIPCIÓN DEL COMPONENTE DESARROLLADO.....	11
1.1	Objetivo general.....	12
1.2	Objetivos específicos.....	12
1.3	Alcance.....	12
1.4	Marco teórico.....	13
2	METODOLOGÍA.....	17
2.1	DevOps.....	18
2.2	Metodología Scrum.....	26
3	PRUEBAS.....	60
4	RESULTADOS.....	71
5	CONCLUSIONES.....	74
6	RECOMENDACIONES.....	75
7	REFERENCIAS BIBLIOGRÁFICAS.....	75
8	ANEXOS.....	76

## ÍNDICE DE FIGURAS

Figura 1.	Flujo de Model-View-Whatever (MVW) [4].....	15
Figura 2.	Repositorio en Gitlab con dos ramas adicionales.....	18
Figura 3.	Servicio Git Runner.....	18
Figura 4.	Comandos para registrar un servicio de Git Runner.....	19
Figura 5.	Servicio de Git Runner en Gitlab.....	19
Figura 6.	Imagen Dokcer en archivo YML.....	20
Figura 7.	Imagen Dokcer en ejecución.....	20
Figura 8.	Comandos previos a la ejecución de un pipeline.....	20
Figura 9.	Comandos previos a la producción de una aplicación.....	21
Figura 10.	Comandos para la ejecución de un pipeline en la rama de development...	21
Figura 11.	Comandos para la ejecución de un pipeline en la rama de testing.....	21
Figura 12.	App Id de proyecto en Firebase.....	22
Figura 13.	Generación de token para identificación.....	22
Figura 14.	Commits enviados al repositorio de Gitlab.....	23

Figura 15. Ejecución de pipeline .....	23
Figura 16. Comandos ejecutados en Pipeline .....	24
Figura 17. Merge desde la rama development a la rama testing .....	25
Figura 18. Versiones de la aplicación móvil.....	25
Figura 19. Vista general de prototipo en Figma.....	30
Figura 20. Arquitectura de la aplicación móvil .....	31
Figura 21. Diseño de base de datos NoSQL .....	32
Figura 22. Flujo de navegación.....	32
Figura 23. Aplicación web en Firebase.....	35
Figura 24. Configuración en archivo Enviroment.....	36
Figura 25. Pantalla de ingreso de la aplicación móvil .....	36
Figura 26. Pantalla de registro de usuario .....	37
Figura 27. Usuarios registrados .....	37
Figura 28. Datos de usuario creado almacenados en Firebase .....	38
Figura 29. Burndown Chart del sprint 1 .....	39
Figura 30. Pantalla de filtros de mantenimientos.....	41
Figura 31. Resultado y opciones de búsqueda de mantenimiento .....	41
Figura 32. Formulario de creación y edición de mantenimientos.....	42
Figura 33. Aviso de confirmación de la opción eliminar .....	43
Figura 34. Menú de navegación lateral.....	43
Figura 35. Burndown Chart del sprint 2 .....	45
Figura 36. Búsqueda de técnicos y productos.....	47
Figura 37. Formulario de creación y edición de productos .....	47
Figura 38. Formulario de creación y edición de técnicos .....	48
Figura 39. Ejecución exitosa de los pipelines.....	49
Figura 40. Versión de archivo APK de la aplicación móvil .....	49
Figura 41. Opción de eliminación de técnicos y productos .....	49
Figura 42. Brundown Chart del sprint 3 .....	50
Figura 43. Búsqueda de planes y plantillas .....	52
Figura 44. Formulario de creación y edición de planes.....	53
Figura 45. Formulario de creación y edición de plantillas .....	53
Figura 46. Opción de eliminación de planes y plantillas.....	54
Figura 47. Brundown Chart del sprint 4 .....	55
Figura 48. Búsqueda de características por selección.....	57
Figura 49. Formulario de creación y edición de características .....	57
Figura 50. Opción de eliminación de características.....	58
Figura 51. Pipelines ejecutados con éxito .....	58

Figura 52. Versiones de la aplicación móvil.....	58
Figura 53. Brundown Chart del sprint 5 .....	59
Figura 54. Resultados de funcionalidad.....	70
Figura 55. Resultado de pipelines ejecutados.....	71
Figura 56. Versiones totales de aplicación móvil.....	72

## ÍNDICE DE TABLAS

Tabla 1. Descripción de historia de usuario 1 .....	27
Tabla 2. Descripción de historia de usuario 2.....	27
Tabla 3. Descripción de historia de usuario 3.....	28
Tabla 4. Descripción de historia de usuario 4.....	28
Tabla 5. Descripción de historia de usuario 5.....	28
Tabla 6. Descripción de historia de usuario 6.....	29
Tabla 7. Descripción de historia de usuario 7.....	29
Tabla 8. Release plan del módulo de mantenimiento .....	30
Tabla 9. Product backlog inicial .....	34
Tabla 10. Tareas para el Sprint 1.....	34
Tabla 11. Resumen del daily meeting Sprint 1 .....	35
Tabla 12. Revisión de puntuación del sprint 1 .....	38
Tabla 13. Tareas para el Sprint 2.....	40
Tabla 14. Resumen del daily meeting sprint 2.....	40
Tabla 15. Revisión de puntuación del sprint 2.....	44
Tabla 16. Tareas para el Sprint 3.....	46
Tabla 17. Resumen del daily meeting sprint 3.....	46
Tabla 18. Revisión de puntuación del sprint 3.....	50
Tabla 19. Tareas para el Sprint 4.....	51
Tabla 20. Resumen del daily meeting sprint 4.....	52
Tabla 21. Revisión de puntuación sprint 4.....	54
Tabla 22. Tareas para el Sprint 5.....	56
Tabla 23. Resumen del daily meeting sprint 5.....	56
Tabla 24. Revisión de puntuación sprint 4.....	59
Tabla 25. Cuestionario SUS para aplicación móvil .....	60
Tabla 26. Resultados de formulario de usabilidad .....	61
Tabla 27. Caso de prueba 01 .....	62
Tabla 28. Resultado para el caso de prueba 01 .....	62
Tabla 29. Caso de prueba 02.....	63



Tabla 30. Resultado para el caso de prueba 02.....	63
Tabla 31. Caso de prueba 03.....	64
Tabla 32. Resultado para el caso de prueba 03.....	64
Tabla 33. Caso de prueba 04.....	65
Tabla 34. Resultado para el caso de prueba 04.....	65
Tabla 35. Caso de prueba 05.....	66
Tabla 36. Resultado para el caso de prueba 05.....	66
Tabla 37. Caso de prueba 05.....	67
Tabla 38. Resultado para el caso de prueba 05.....	67
Tabla 39. Caso de prueba 06.....	68
Tabla 40. Resultado para el caso de prueba 06.....	68
Tabla 41. Caso de prueba 07.....	69
Tabla 42. Resultado para el caso de prueba 07.....	69
Tabla 43. Caso de prueba 08.....	69
Tabla 44. Resultado para el caso de prueba 08.....	70
Tabla 45. Resultados de pipelines.....	71

## **ÍNDICE DE ANEXOS**

ANEXO I. Enlace al prototipo de la aplicación móvil en Figma.....	76
ANEXO II. Enlace al diseño de base relacional y no relacional.....	76
ANEXO III. ANEXO III. Video de módulo de mantenimiento de sistemas Sisreco.....	76
ANEXO IV. Carpeta con documentos de historias de usuario.....	77
ANEXO V. Repositorio Gitlab con el código del módulo de mantenimiento.....	77
ANEXO VI. Formulario de usabilidad.....	77
ANEXO VIII. Formulario de funcionalidad.....	79

## RESUMEN

La integración continua permite la combinación de cambios realizados en el desarrollo de un software en un repositorio, las pruebas se realizan de manera automática en las diferentes versiones que puede tener un software. El objetivo de una integración continua es mejorar la rapidez y validación de actualizaciones de un software.

El presente trabajo de integración curricular propone una aplicación móvil, para el módulo de mantenimiento de un sistema ERP, utilizando la metodología Scrum y la integración continua, con el fin de automatizar el proceso de generación de un archivo ejecutable o archivo APK para dispositivos móviles, al igual que el despliegue o versionamiento de estos archivos mediante las prácticas DevOps, generando un modelo para la implementación de una integración continuar en el desarrollo de una aplicación móvil.

**PALABRAS CLAVE:** APK, integración continua, versionamiento.

## **ABSTRACT**

Continuous integration allows the combination of changes made in the development of a software in a repository, the tests are carried out automatically in the different versions that a software may have. The goal of continuous integration is to improve the speed and validation of software updates.

This curricular integration work proposes a mobile application for the maintenance module of an ERP system, using the Scrum methodology and continuous integration, in order to automate the process of generating an executable file or APK file for mobile devices, As well as the use or versioning of these files through DevOps practices, a model will appear for the implementation of a continuous integration in the development of a mobile application.

**KEY WORDS:** APK, continuous integration, versioning.

# 1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

El módulo de mantenimiento está especializado en la gestión de bienes, recursos o productos. La base para el módulo de mantenimiento desarrollado son los planes para los productos los cuales permiten mantener constancia sobre los mantenimientos realizados sobre un producto y los posibles planes a aplicarse sobre varios productos de la misma categoría o productos similares.

El módulo de mantenimiento, al ser desarrollado con el framework Ionic, mantiene una implementación de diseño de backend y frontend basado en un entorno web por lo cual posee elementos HTML, CSS y JS, debido a la abstracción de requerimientos esta característica permitió escalar la aplicación de manera más sencilla.

El módulo de mantenimiento utiliza herramientas propias de Firebase para poder realizar el proceso de autenticación, distribución y almacenamiento de datos para la aplicación móvil, para la creación de usuario la aplicación móvil utiliza la herramienta Authentication con un proveedor de correo y contraseña para poder utilizar la aplicación móvil y registrar productos y mantenimientos, por otro lado se utiliza Firestore para registrar la información necesaria para mantener la integridad de los datos y registros que se necesiten sobre los mantenimientos de los productos, por último se utiliza App Distribución para mantener un registro de las versiones de los archivos APK que se generen del módulo de mantenimiento según los avances que se realicen en base a la metodología scrum.

La escalabilidad del módulo se logró mediante el recurso del pipeline (CI/CD) implementado mediante Gitlab, ya que los cambios que se realizaron tanto en el diseño del frontend y backend para mejorar la funcionalidad y usabilidad de la aplicación, se pueden verificar constantemente mediante las diferentes versiones de la aplicación que se generan y cargan en Firebase.

## **1.1 Objetivo general**

Desarrollar una aplicación móvil para la gestión de mantenimientos de diferentes productos mediante el uso de la metodología SCRUM mejorando el proceso con prácticas DevOps.

## **1.2 Objetivos específicos**

1. Definir un flujo DevOps para la automatización del despliegue de una aplicación móvil.
2. Utilizar un flujo DevOps para mejorar el proceso diario de las diferentes fases de la metodología SCRUM tanto para el desarrollo y el despliegue de una aplicación móvil.
3. Mantener prácticas adecuadas para el desarrollo de la funcionalidad, usabilidad y seguridad de la aplicación móvil.

## **1.3 Alcance**

El módulo de mantenimiento el cual es el desarrollado, desplegado e implementado en el presente trabajo, permite el registro de varios aparatos, artefactos, herramientas o maquinaria denominadas como productos para una mejor facilidad de entendimiento, estos productos pueden estar sujetos a diferentes mantenimientos según las categorías a las cuales pertenezcan. Los planes de mantenimiento están sujetos a cada categoría y se pueden implementar en los diferentes productos de cada categoría. Los registros de mantenimiento mantienen una constancia sobre los productos, categorías, planes y técnicos involucrados en el registro de un mantenimiento al igual que datos relevantes como fechas, descripciones o datos de importancia para el correcto registro y referencia del mantenimiento sostenido.

Todas las colecciones utilizadas e implementadas permiten el manejo de un CRUD tradicional (crear, leer, editar y eliminar) para los datos según la conveniencia o criterio de cada usuario.

Los cuadros de despliegue también se actualizan en todo momento para mantener la constancia entre los datos creados, editados o eliminados al momento de mostrar la información para ejecutar algún proceso de consulta en la base de datos.

## **1.4 Marco teórico**

### **1.4.1 Metodología utilizada**

#### **Scrum**

Scrum es una metodología donde se aplican buenas prácticas, especialmente para trabajar en equipo, donde cada persona involucrada en un proyecto cumple una función específica la cual complementa el proyecto de la mejor manera.

Las partes de scrum son las siguientes:

- Plantificación del Sprint: se identifica un objetivo en específico el cual será desarrollado durante toda la vida del sprint.
- Etapa de desarrollo: se realizan las actividades necesarias para cumplir con el objetivo del sprint, se revisan problemas y su solución o alternativas.
- Revisión del sprint: se revisan los resultados obtenidos de manera grupal para obtener una retroalimentación para la continuidad del desarrollo.
- Retroalimentación: la retroalimentación puede provenir por parte del cliente el cual será el consumidor del producto resultante del proyecto.

En la Figura 1, se muestra las etapas de Scrum, en donde el flujo implementado permite avanzar en el proyecto de manera efectiva, siempre manteniendo una constante retroalimentación entre las diferentes partes que conforman el equipo del proyecto, de esta manera el avance del proyecto es conocido por todos y mejora una base de conocimientos en común.

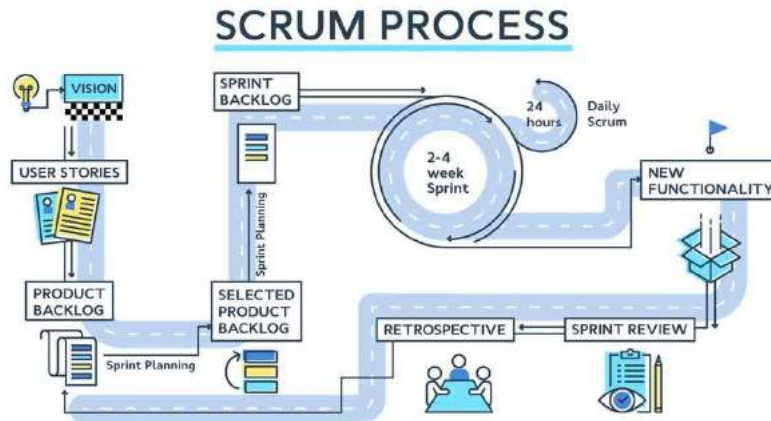


Figura 1. Etapas de la metodología Scrum [1]

### Aplicación móvil

Una aplicación móvil consiste en un software que funciona en un dispositivo móvil (teléfonos y tabletas) y ejecuta ciertas tareas para el usuario [2]. Las aplicaciones móviles pueden utilizar navegadores para poder ejecutarse utilizando recursos web como CSS, HTML y JavaScript, este enfoque se denomina aplicaciones móviles web, por otro lado, las aplicaciones nativas utilizan los recursos propios de los sistemas operativos para los cuales son desarrolladas aumentando el rendimiento de estas.

Según Javier Cuello y José Vittone las aplicaciones móviles híbridas “es una especie de combinación entre las dos anteriores.” [3], de esta manera las aplicaciones utilizan la estructura y arquitectura de aplicaciones web, pero se ejecutan o compilan en el sistema operativo del dispositivo móvil para asemejar el rendimiento a una aplicación nativa.

### Integración continua

Es una práctica que pretende automatizar la implementación o el despliegue de cambios realizados sobre un código, el fin de la integración continua es evaluar si una parte o porción de código funciona correctamente antes de ser integrado con todas las partes relacionadas o probar su funcionalidad con otros componentes externos a su desarrollo. La integración continua (CI) funciona empujando el código a una aplicación alojada en un repositorio Git. En cada push, la aplicación ejecuta un canal de scripts para construir, probar y validar los cambios de código antes de fusionarlos con la rama principal. [4]

## 1.4.2 Frameworks

### Angular

Angular es una framework JavaScript de código abierto mantenido por Google, que se utiliza para crear páginas web de tipo SPA (Single Page Application) [5].

Una SPA o aplicación de página simple, es una aplicación donde no se solicita recursos al servidor continuamente, todo el contenido de las vistas se carga una sola vez en una única dirección de un archivo HTML, sin embargo, es posible navegar entre los diferentes recursos dando la impresión al usuario de que fueran varias páginas con diferentes direcciones como normalmente se utiliza, mejorando la rapidez de la navegación.

La página puede solicitar los recursos nuevamente al servidor en caso de requerirlo, pero en la mayoría de los casos solo es necesario actualizar componentes parciales lo que ayuda al rendimiento de la aplicación.

### Modelo-Vista-Controlador en Angular

En Angular el Modelo-Vista-Controlador (MVC) presenta una particularidad de dependencia entre la vista y el controlador debido al concepto de Angular de two-way data binding, esto implica que se puede modificar el modelo desde la vista y viceversa, esto se conoce como Modelo-Vista-Whatever (MVW). En la Figura 2 se puede observar el flujo o relación de los componentes donde se aprecia la dependencia entre el modelo y la vista.

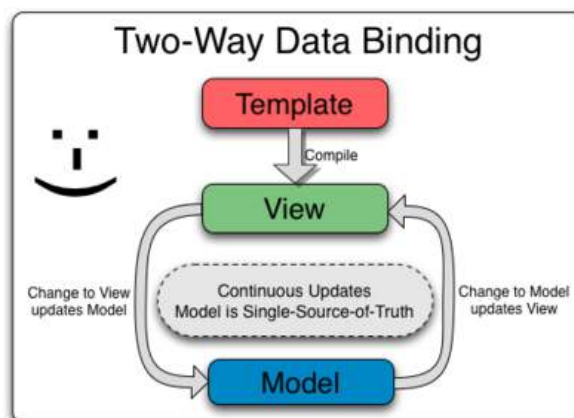


Figura 2. Flujo de Model-View-Whatever (MVW) [6]



El patrón de diseño Modelo-Vista-Controlador es uno de los más populares en la industria del desarrollo de aplicaciones empresariales [7]. Dentro de Angular el objetivo de los componentes del MVW es el mismo que el MVC:

- **Modelo:** define las reglas o lógica con la cual se van a tratar los datos, no hay que confundir al modelo con la base de datos o la manera de gestionar los datos.
- **Vista:** recibe los datos o información y la muestra mediante una interfaz o presentación simple y de fácil entendimiento para los usuarios.
- **Controlador:** recibe los eventos de la vista y los conecta a la lógica del controlador.

## **Ionic**

Ionic Framework de código abierto para el desarrollo de aplicaciones móviles híbridas, tiene como base las tecnologías HTML, CSS y JavaScript, al ser desarrollado sobre Angular JS y Cordova permite una compatibilidad con todos los sistemas operativos móviles manteniendo un enfoque web. Se integra con los principales frameworks de frontend, como Angular, React y Vue, aunque también se puede usar Vanilla JavaScript [8]. La principal ventaja es el poder utilizar el enfoque o estructura web de HTML, CSS y JavaScript para poder crear páginas, de esta manera no es necesario aprender una tecnología para poder utilizar Ionic.

### **1.4.3 Herramientas**

## **Docker**

Docker es un software que permite crear o implementar aplicaciones u otro software de manera rápida y sencilla, por lo general para poder usar una aplicación es necesario tener un sistema operativo y varias dependencias para que se ejecute correctamente, Docker permite tener un “contenedor” donde se tiene todas las características necesarias para poder ejecutar un software o aplicación de manera que se vuelve una aplicación portable utilizando los recursos del propio sistema operativo donde se está ejecutando Docker. De manera similar a cómo una máquina virtual virtualiza (elimina la necesidad de administrar directamente) el hardware del servidor, los contenedores virtualizan el sistema operativo de un servidor [9]. Uno uso común son los gestores de bases de datos como MySQL,

PostgresSql, MariaDB entre otros.

### **Gitlab**

Es un repositorio para la gestión de proyectos de software, utiliza la tecnología git el cual es un versionador de código o controlador de versiones de código, esto permite mantener constancia o registro de cada archivo que forme parte del desarrollo de un solo software. Gitlab ofrece un servicio web de control de versiones y desarrollo de software colaborativo basado en Git. Además de ser un gestor de repositorios, ofrece generación de wikis y un sistema de seguimiento de errores [10].

### **Firebase**

Es una plataforma móvil creada por Google la cual está enfocada para el desarrollo y la creación de aplicaciones móviles de manera rápida y sencilla, la plataforma está disponible para el desarrollo de aplicaciones para plataforma iOS, Android y web

- **Firestore Authentication:** gestiona el ingreso de usuario a aplicaciones móviles al igual que la creación y verificación de cuentas creadas mediante diferentes proveedores como Facebook, Gmail entre otros.
- **Firestore:** es una base de datos NoSQL que utiliza colecciones y documentos dedicada para el almacenamiento y consulta de datos para aplicaciones móviles para varias plataformas.
- **App Distribution:** permite mantener versiones de prueba o distribuir versiones de una aplicación móvil, estas versiones pueden subir de manera manual o automática.

## **2 METODOLOGÍA**

Para el desarrollo del módulo de mantenimiento se utilizó la metodología Scrum utilizando prácticas DevOps en donde se utilizaron las siguientes herramientas:

- Integración continua (CI).
- Desarrollo continuo.
- Realización de pruebas continuas.
- Puesta en marcha continua (CD).

- Supervisión continuar.

La duración de cada sprint es de 1 semana con 6 horas de trabajo diarias.

## 2.1 DevOps

### 2.1.1 Implementación de integración continuar en Gitlab

Primero se configura una cuenta de Gitlab con dos ramas como se observa en la Figura 3, una para el desarrollo nombrada “development” donde se verificará si la codificación este correcta y otra rama nombrada “testing” en donde se generará el archivo APK de la aplicación móvil y se enviara la misma a Firebase App Dsitribution, estos procedimientos se verán evidenciados mediante los pipelines en Gitlab.

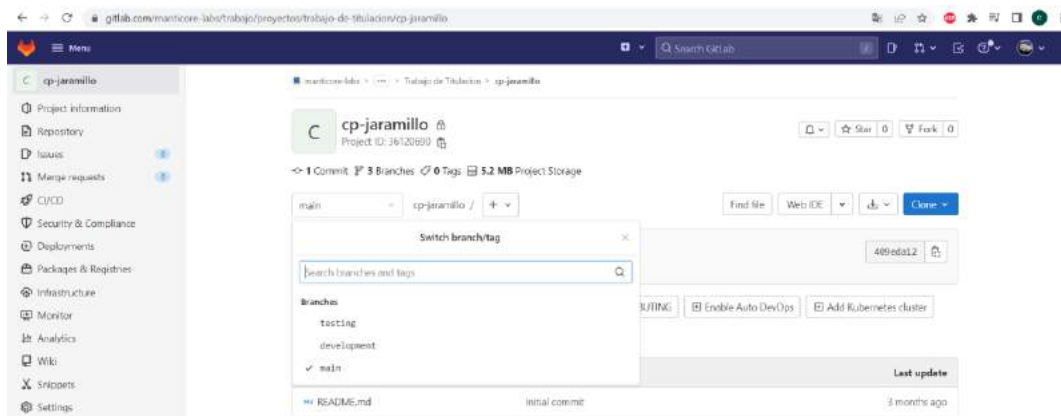


Figura 3. Repositorio en Gitlab con dos ramas adicionales

Es necesario crear un servicio en el equipo donde se va a ejecutar el desarrollo, el servicio de Git Runner permite utilizar el terminar de un servidor al cual se especifique. Se descarga el servicio desde la documentación de Gitlab, el cual es colocado en una carpeta para poder ser ejecutado, en la Figura 4 se muestra el servicio previo a su registro en Gitlab en una ruta adecuada.

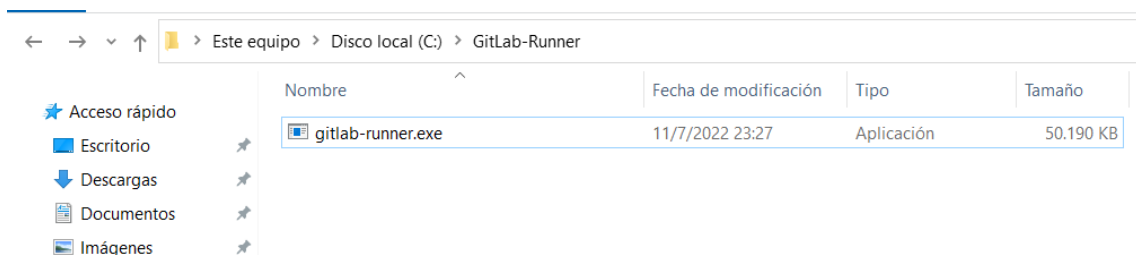


Figura 4. Servicio Git Runner

En la Figura 5 se muestran los comandos y características con las cuales se configura el servicio para que pueda utilizar una imagen en Docker. Primero se instala el servicio para que sea reconocido en Gitlab, después se registra un nuevo Git Runner para su funcionamiento. En este procedimiento se debe utilizar el token que se puede obtener desde la configuración del proyecto en Gitlab, el token y el servicio se pueden ver registrado y funcionando en la Figura 6.

```
Administrador: Símbolo del sistema - \gitlab-runner.exe register
C:\GitLab-Runner>.\gitlab-runner.exe install
Runtime platform arch=amd64 os=windows pid=4420 revision=76984217 version=15.1.0
C:\GitLab-Runner>.\gitlab-runner.exe register
Runtime platform arch=amd64 os=windows pid=10980 revision=76984217 version=15.1.0
Enter the GitLab instance URL (for example, https://gitlab.com/):
https://gitlab.com/
Enter the registration token:
GR13489411u_uyfPgSC2GixcUfiQK
Enter a description for the runner:
[DESKTOP-KB43L3A]: ModMantRunner
Enter tags for the runner (comma-separated):
Enter optional maintenance note for the runner:
Registering runner... succeeded runner=GR13489411u_uyfPg
Enter an executor: docker-windows, docker-ssh, shell, docker+machine, docker-ssh+machine, custom, parallels, ssh, virtualbox, kubernetes, docker:
windows
```

Figura 5. Comandos para registrar un servicio de Git Runner

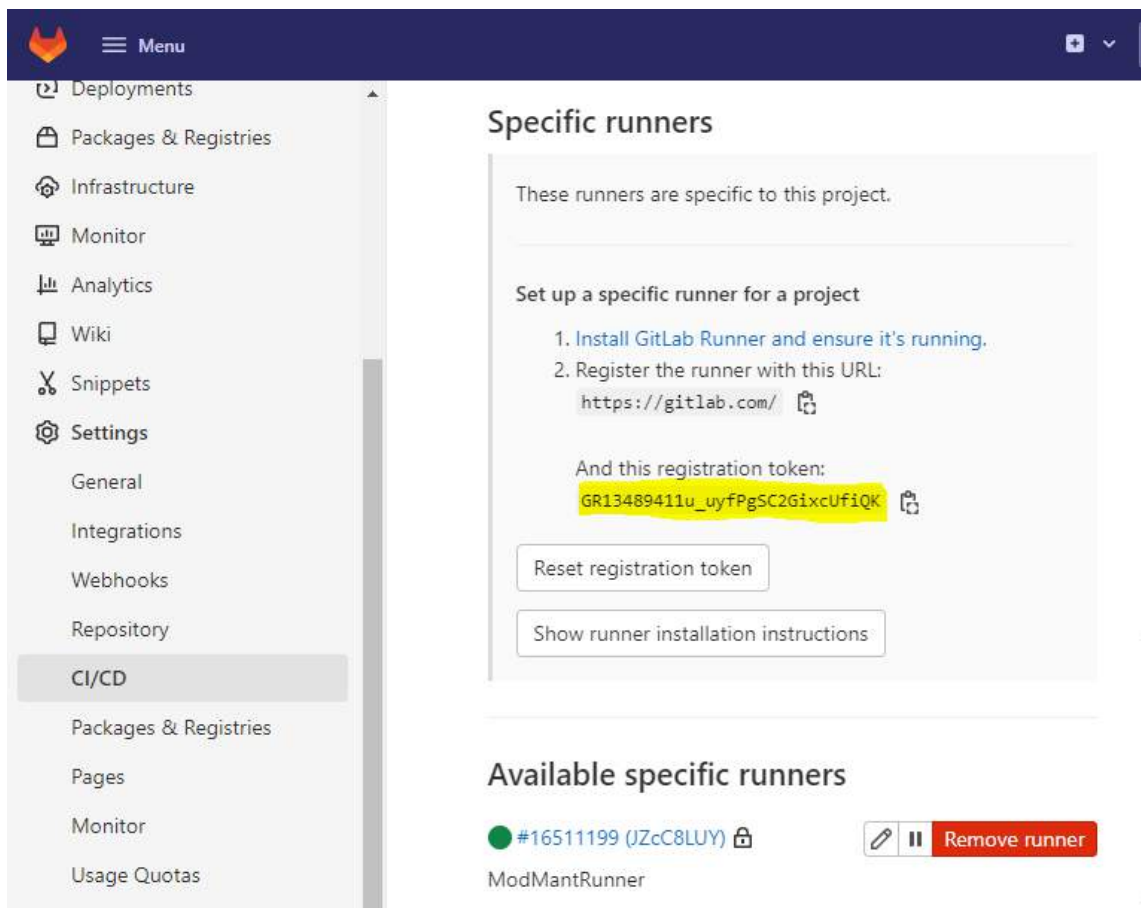


Figura 6. Servicio de Git Runner en Gitlab

Con el servicio de Git Runner en ejecución se puede configurar el archivo YML en el proyecto para ejecutar comandos desde una imagen en Docker.

En la Figura 7 se muestran la imagen que se va a utilizar desde Docker, esta imagen contiene archivos esenciales para poder generar un archivo APK mediante comando, esta opción es la más efectiva ya que no presentan inconvenientes con los elementos que se necesitan complementar el proceso de integración continuar, en la Figura 8 se muestra la imagen que se genera en Docker.

```
1 image: circleci/android:api-28-node
2
```

Figura 7. Imagen Docker en archivo YML

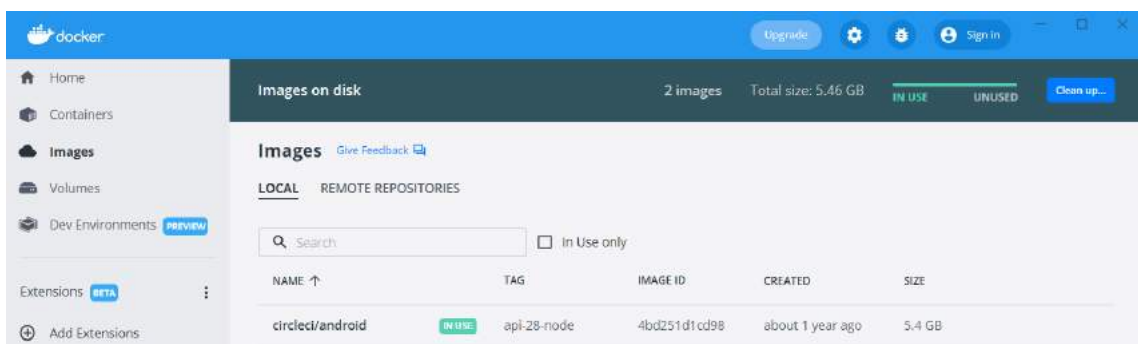


Figura 8. Imagen Docker en ejecución

En la Figura 9 se muestra los comandos a ejecutar antes de la ejecución para la verificación del código. En esta parte se incluyen las dependencias del Node JS, el framework Ionic, dependencias de angular y dependencias de firebase, se verifican las versiones de cada dependencia para verificar las dependencias con las cuales el proyecto funciona correctamente.

```
3 before_script:
4   - echo before_script...
5   - sudo apt-get update
6   - sudo apt-get install nodejs
7   - npm install
8   - sudo npm install -g @angular/cli
9   - sudo npm i -g @ionic/cli
10  - sudo npm install -g firebase-tools@10.9.2
11  - nodejs -v
12  - npm -v
13  - ionic -v
14  - echo end before_script...
```

Figura 9. Comandos previos a la ejecución de un pipeline

En la Figura 10 se muestra los dos stage con los cuales se va a trabajar, el stage llamado "build" se ejecuta cuando se realiza un commit en la rama development, mientras que el stage "test" se ejecuta cuando se realiza un commit o un merge a la rama de testing.

```

16 stages:
17   - build
18   - test
19

```

Figura 10. Comandos previos a la producción de una aplicación

En la Figura 11 se muestra la configuración del stage build, la configuración indica que este stage se ejecutara solo en la rama de development, esto para probar que el código se pueda ejecutar correctamente y se genere el archivo APK correspondiente, este procedimiento se realiza mediante comandos simples utilizando las dependencias instaladas previamente.

```

20 build-job:
21   stage: build
22   only:
23     - development
24   script:
25     - echo build-job from development branch...
26     - cd /builds/manticore-labs/trabajo/proyectos/trabajo-de-titulacion/cp-jaramillo/ModMantenimiento
27     - npm install
28     - ionic build
29     - ionic capacitor rm android
30     - ionic capacitor add android
31     - ionic capacitor sync
32     - cd android
33     - sudo chmod +x gradlew
34     - ./gradlew assembleDebug
35     - echo build-job ended...
36

```

Figura 11. Comandos para la ejecución de un pipeline en la rama de development

En la Figura 12 se muestra la configuración del stage test, la configuración indica que este stage se ejecutara solo en la rama de testing, en este stage se va a cambiar el nombre del archivo APK, esto mediante la fecha y minutos exactos en que se está ejecutando el pipeline, de esta manera se puede subir y mantener un versionamiento correcto de las versiones de la aplicación móvil.

```

37 testing-job:
38   stage: test
39   only:
40     - testing
41   script:
42     - echo testing-job from testing branch
43     - cd /builds/manticore-labs/trabajo/proyectos/trabajo-de-titulacion/cp-jaramillo/ModMantenimiento
44     - npm install
45     - ionic build
46     - ionic capacitor add android
47     - ionic capacitor sync
48     - cd android
49     - sudo chmod +x gradlew
50     - ./gradlew assembleDebug
51     - cd /builds/manticore-labs/trabajo/proyectos/trabajo-de-titulacion/cp-jaramillo/ModMantenimiento/android/app/build/
52     outputs/apk/debug/
53     - sudo mv app-debug.apk app-debug-$(date +%d-%m-%y-%M).apk
54     - dir
55     - export nombreapp=app-debug-$(date +%d-%m-%y-%M).apk
56     - echo $nombreapp
57     - firebase appdistribution:distribute $nombreapp --app 1:715669845914:android:086a00287e0b775bebac23 --token "1//
58     05R04MFPONJ3PUCgYIARAAGAU5NwF-L9IrY6wVrEQD96UE9MFORcPVMdLUF4TvPY1bMjBYJxJ6goHKcWLqD4e8Lu36_36giUK6jPQ"
59     - echo testing-job ended...
60

```

Figura 12. Comandos para la ejecución de un pipeline en la rama de testing

La app ID se la puede obtener desde el proyecto configurado en Firebase como se muestra en la Figura 13, mientras que para el token es necesario ejecutar el comando desde el equipo de desarrollo para poder obtener el mismo, este proceso se puede observar en la Figura 14, en este paso es necesario tener disponible un navegador web ya que es necesario identificar la cuenta con la cual se está generando el token que se va a utilizar, este procedimiento es propio de Firebase.

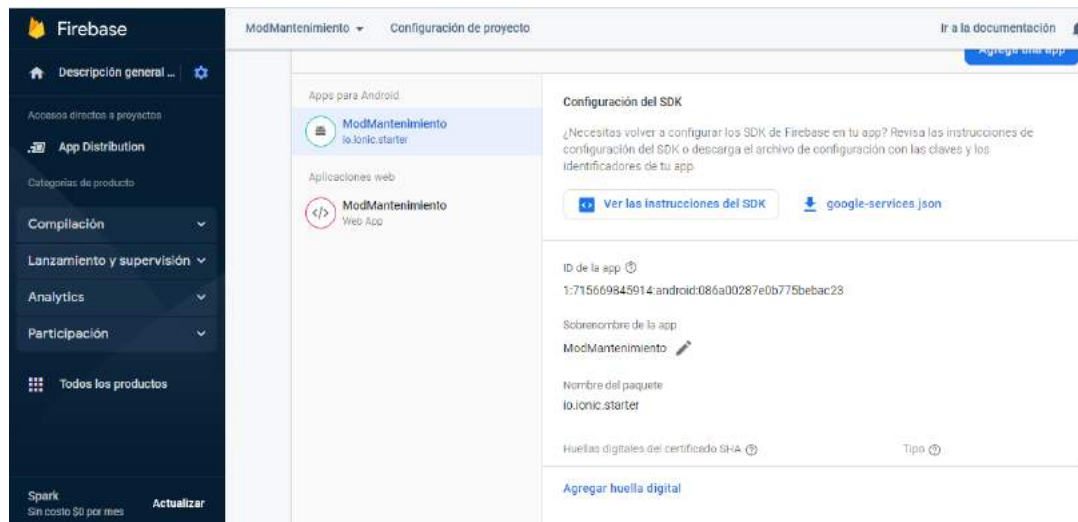


Figura 13. App Id de proyecto en Firebase

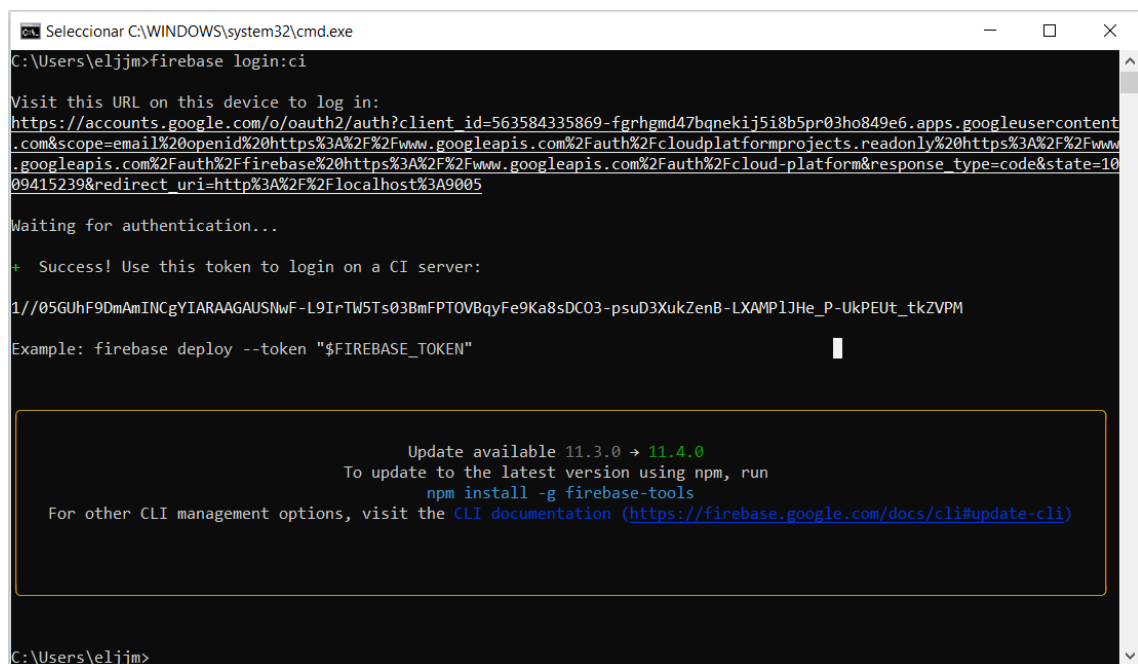


Figura 14. Generación de token para identificación

## 2.1.2 Ejecución de integración continua

Los pipelines son las ejecuciones correspondientes a cada commit que se realice en las dos ramas que se configuraron en el repositorio de Gitlab.

El proceso comienza cuando se realiza un commit en la rama de development, esto se realiza desde Git Kraken, en la Figura 15 se puede observar los diferentes commits realizados por el equipo de desarrollo.

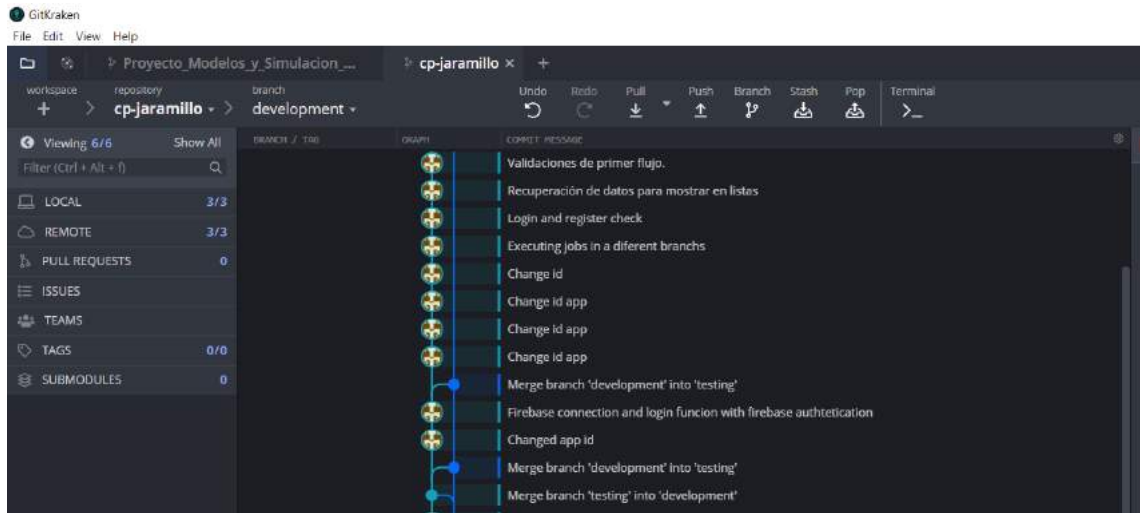


Figura 15. Commits enviados al repositorio de Gitlab

En la Figura 16 se observa la creación y ejecución de un pipeline para la rama de development, se ejecuta únicamente el stage de build, esto genera el APK con los diferentes cambios realizados durante el desarrollo, por otro lado, en la Figura 17 se puede observar una parte de los comandos ejecutados durante el pipeline.



Figura 16. Ejecución de pipeline



```
174 npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.3.2 (node_modules/firebase-tools/node_modules/fsevents):
175 npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.3.2: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"x64"})
176 npm WARN optional SKIPPING OPTIONAL DEPENDENCY: puppeteer@9.0.0 (node_modules/firebase-tools/node_modules/puppeteer):
177 npm WARN optional SKIPPING OPTIONAL DEPENDENCY: puppeteer@9.0.0 install: `node install.js`
178 npm WARN optional SKIPPING OPTIONAL DEPENDENCY: Exit status 1
179 + firebase-tools@10.9.2
180 added 1275 packages from 1200 contributors in 55.404s
181 $ nodejs -v
182 v14.17.5
183 $ npm -v
184 6.14.14
185 $ ionic -v
186 6.20.1
187 $ echo end before_script...
188 end before_script...
189 $ echo build-job from development branch...
190 build-job from development branch...
191 $ cd /builds/manticore-labs/trabajo/proyectos/trabajo-de-titulacion/cp-jaramillo/ModMantenimiento
192 $ npm install
193 npm WARN read-shrinkwrap This version of npm is compatible with lockfileVersion@1, but package
ajoy-de-titulacion/cp-jaramillo generated for lockfileVersion@2. I'll try to do my best with it!
```

Figura 17. Comandos ejecutados en Pipeline

El siguiente paso es realizar un merge desde la rama de development a la rama de testing, de esta manera aseguramos que cada cierta cantidad de sprints según la metodología Scrum, todo el desarrollo que se encuentra en la rama de development será enviado a la rama de testing para poder versionar el aplicativo de móvil. En la Figura 18 se muestra un merge realizado junto con el pipeline que se ejecutó correspondiente al stage de test en la rama de testing con los comandos correspondientes.

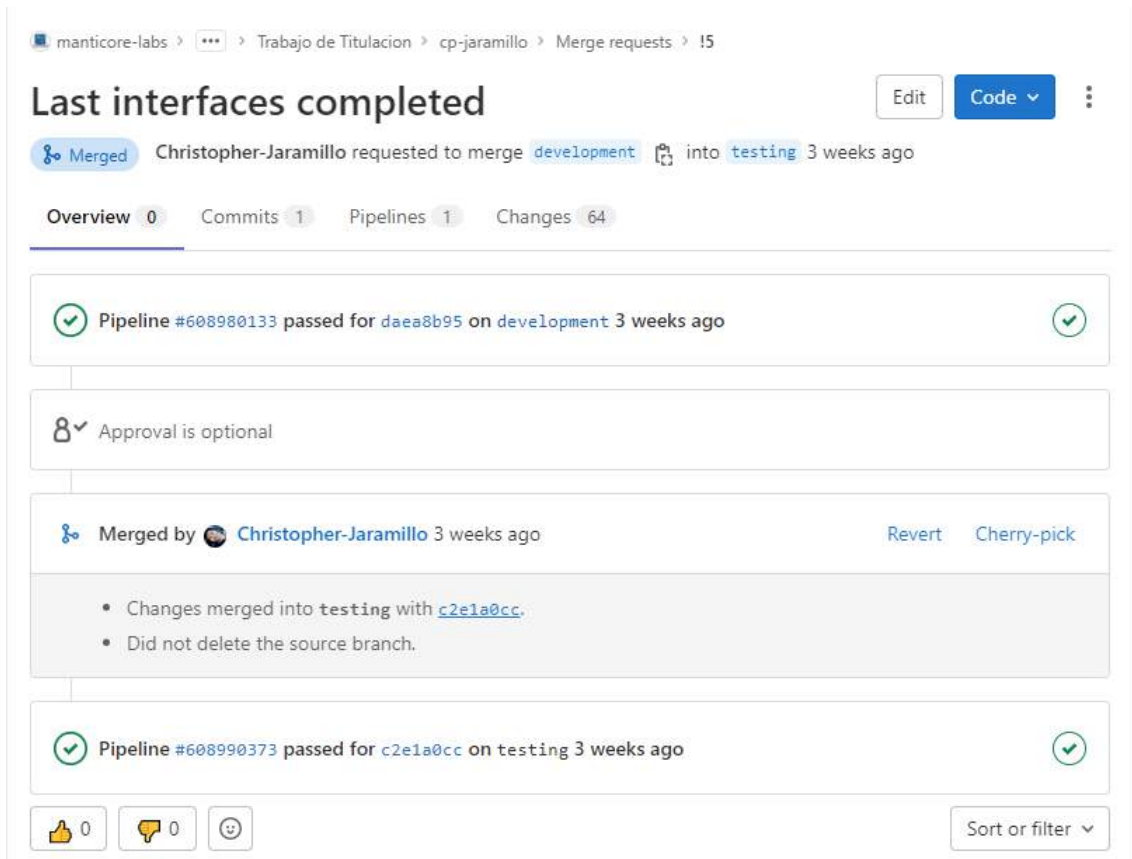


Figura 18. Merge desde la rama development a la rama testing

Por último, se puede observar el versionamiento de la aplicación móvil en Firebase App Distribution, las diferentes versiones se pueden observar en la Figura 19.

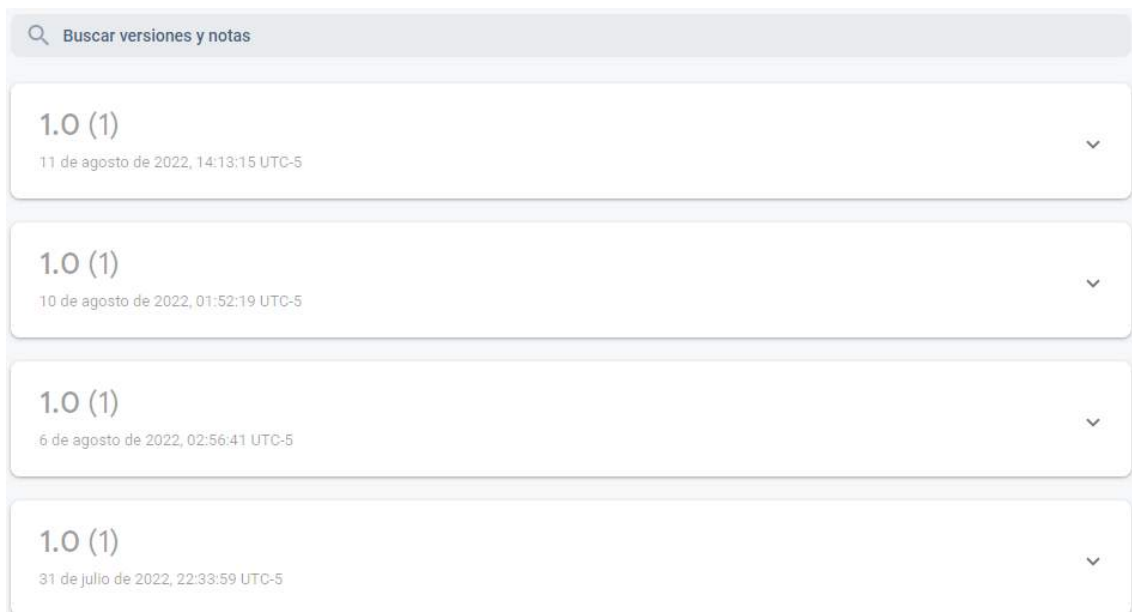


Figura 19. Versiones de la aplicación móvil

## 2.2 Metodología Scrum

### 2.2.1 Equipo Scrum

El equipo de Scrum está conformado por el Product Owner, un Scrum Master y el equipo de desarrollo, los asignados a cada parte son:

- Product Owner: Msc. Andrián Eguez.
- Scrum Master: Msc. Andrián Eguez.
- Equipo de desarrollo: Christopher Jaramillo.

### 2.2.2 Requerimientos iniciales

Para poder aplicar la integración continua primero se establece los requerimientos referentes al módulo de mantenimiento de un sistema ERP, esto se logra mediante la abstracción de requerimientos tomando como referencia el módulo de mantenimiento del sistema Sisreco creado para el Ejército Ecuatoriano.

Para poder adaptar el módulo de mantenimiento a una aplicación móvil se especifican las siguientes características.

- Gestión de usuarios (Tabla 1).
- Gestión de productos. (Tabla 2).
- Gestión de mantenimientos por producto. (Tabla 3).
- Gestión de planes. (Tabla 4).
- Gestión de plantillas. (Tabla 5).
- Gestión de técnicos. (Tabla 6).
- Gestión de características. (Tabla 7).
  - Gestión de rangos.
  - Gestión de operaciones.
  - Gestión de acciones.
  - Gestión de elementos.
  - Gestión de tipos de mantenimientos.

La especificación de un solo módulo facilita el desarrollo ya que se entiende de mejor manera el flujo que se necesita obtener al finalizar el desarrollo.

#### Historia de usuario 1: Gestión de usuarios

Permite la creación e ingreso de un usuario para el acceso de las diferentes

partes de la aplicación móvil.
<b>Escenarios</b>
Creación de usuario.
Ingreso a las funcionalidades de la aplicación móvil.
<b>Criterios de aceptación</b>
Mensajes de error y aceptación de creación de usuario.
Todos los campos son requeridos.
La contraseña debe contener mínimo 6 caracteres.

Tabla 1. Descripción de historia de usuario 1

<b>Historia de usuario 2: Gestión de mantenimientos</b>
Permite la creación, visualización, edición y eliminación de mantenimientos para cada producto.
<b>Escenarios</b>
Creación de mantenimientos.
Edición de mantenimientos.
Eliminación de mantenimientos.
Visualización de mantenimientos.
<b>Criterios de aceptación</b>
Se puede filtrar mantenimientos por: <ul style="list-style-type: none"> <li>- Nombre del producto</li> <li>- Nombre del plan</li> <li>- Fecha de inicio de un mantenimiento</li> <li>- Fecha de fin de un mantenimiento</li> <li>- Tipo de mantenimiento</li> </ul>

Tabla 2. Descripción de historia de usuario 2

<b>Historia de usuario 3: Gestión de productos</b>
Permite la creación, visualización, edición y eliminación de un producto para cada usuario.
<b>Escenarios</b>
Creación de productos.
Edición de productos.
Eliminación de productos.
Visualización de productos.

<b>Criterios de aceptación</b>
Los productos pueden ser buscados mediante el nombre.

Tabla 3. Descripción de historia de usuario 3

<b>Historia de usuario 4: Gestión de técnicos</b>
Permite la creación, visualización, edición y eliminación de un producto para cada técnico.
<b>Escenarios</b>
Creación de técnico.
Edición de técnico.
Eliminación de técnico.
Visualización de técnico.
<b>Criterios de aceptación</b>
Los técnicos pueden ser buscados mediante el nombre.

Tabla 4. Descripción de historia de usuario 4

<b>Historia de usuario 5: Gestión de plantillas</b>
Permite la creación, visualización, edición y eliminación de plantillas.
<b>Escenarios</b>
Creación de plantillas.
Edición de plantillas.
Eliminación de plantillas.
Visualización de plantillas.
<b>Criterios de aceptación</b>
Las plantillas pueden ser buscados mediante el nombre.

Tabla 5. Descripción de historia de usuario 5

<b>Historia de usuario 6: Gestión de planes</b>
Permite la creación, visualización, edición y eliminación de planes.
<b>Escenarios</b>
Creación de planes.
Edición de planes.
Eliminación de planes.
Visualización de planes.
<b>Criterios de aceptación</b>

Los planes pueden ser buscados mediante el nombre.
--

Tabla 6. Descripción de historia de usuario 6

<b>Historia de usuario 7: Gestión de características</b>
Permite la creación, visualización, edición y eliminación de características (rango, operaciones, acciones, elementos y tipos de mantenimientos)
<b>Escenarios</b>
Creación de características.
Edición de características.
Eliminación de características.
Visualización de características.
<b>Criterios de aceptación</b>
Las características pueden ser buscados mediante el nombre.

Tabla 7. Descripción de historia de usuario 7

### 2.2.3 Prototipo del sistema web

Con las historias de usuario especificadas para la gestión del módulo de mantenimiento se realiza el diseño de la aplicación mediante la herramienta Figma para posteriormente pasar al diseño y desarrollo de las características especificadas.

En la Figura 20 se pueden observar la vista general del diseño de la aplicación móvil, tomando en cuenta las herramientas o procesos ofrecidos por el framework Ionic. En el Anexo I se puede encontrar el enlace al prototipo completo en Figma.

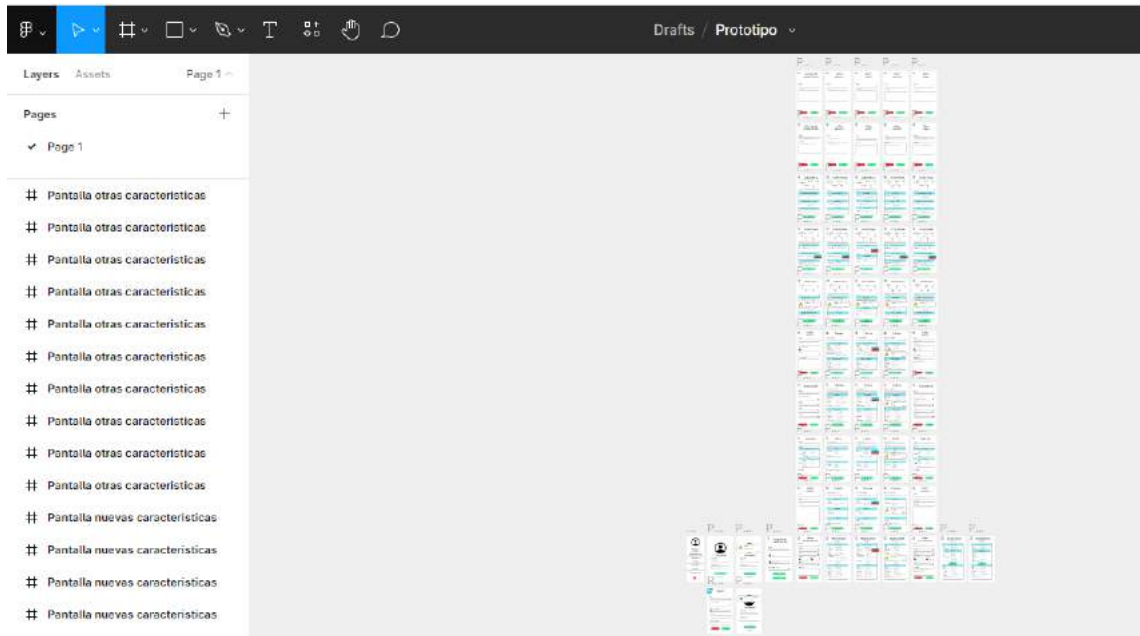


Figura 20. Vista general de prototipo en Figma

#### 2.2.4 Release Planning

Una vez definidas y aprobadas las historias de usuario con el Product Backlog junto con el diseño, se procede a planificar el desarrollo de cada una de las características especificadas para el módulo de mantenimiento. En los Sprints que se detallan en la Tabla 8 se considera cada una de las gestiones necesarias y definidas en el apartado 2.2 de la sección de Metodología.

Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5
US01	US02	US03	US05	US07
		US04	US06	

Tabla 8. Release plan del módulo de mantenimiento

#### 2.2.5 Arquitectura de la aplicación

El framework Ionic está basado en Angular, por lo cual se mantiene una arquitectura modelo-vista-controlador tradicional para la aplicación móvil. Para la parte del frontend se utiliza HTML, CSS y JS, mientras que para el backend se utiliza una estructura de base de datos NoSQL organizada por colecciones y documentos propias de Firebase.

En la Figura 21 se puede observar la interacción entre el framework Ionic y los elementos de Firebase Authentication y Cloud Firestore.

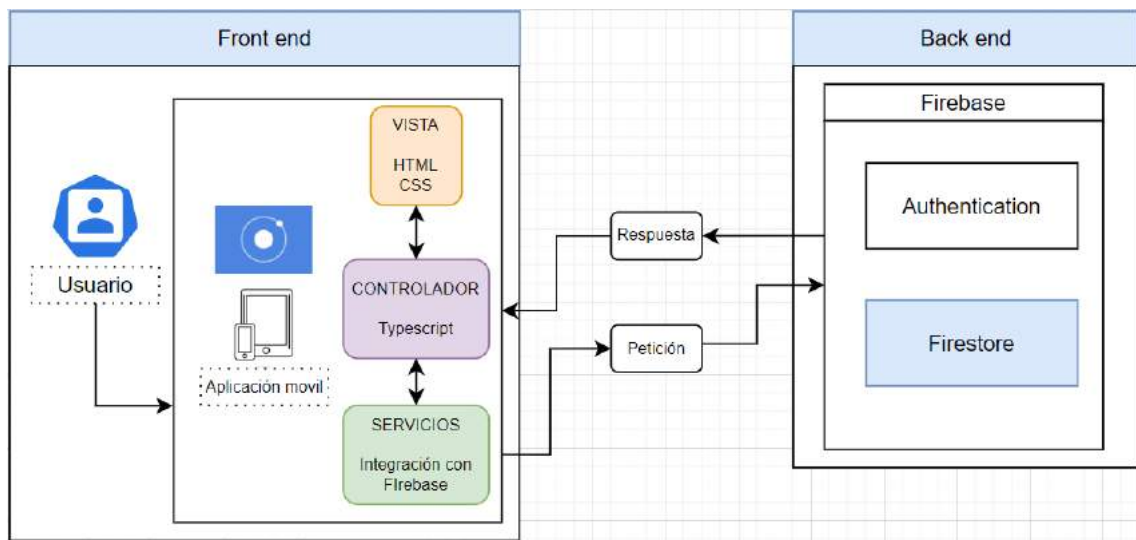


Figura 21. Arquitectura de la aplicación móvil

### 2.2.6 Modelo de base NoSQL en Firestore

Para el diseño de la base de datos no relacional primero se realizó el diseño de la base de datos en un esquema relacional para poder tomarlo como modelo y crear las colecciones y sub-colecciones respectivas según la Figura 22. En el Anexo II se puede observar los diseños correspondientes a la base de datos relacional y la base de datos no relacional utilizada.

Se obtiene un total de 12 colecciones agrupadas en 3 niveles, tomando como principal colección la correspondiente a la gestión de los usuarios ya que de la misma depende la gestión de productos junto con la creación de mantenimientos.



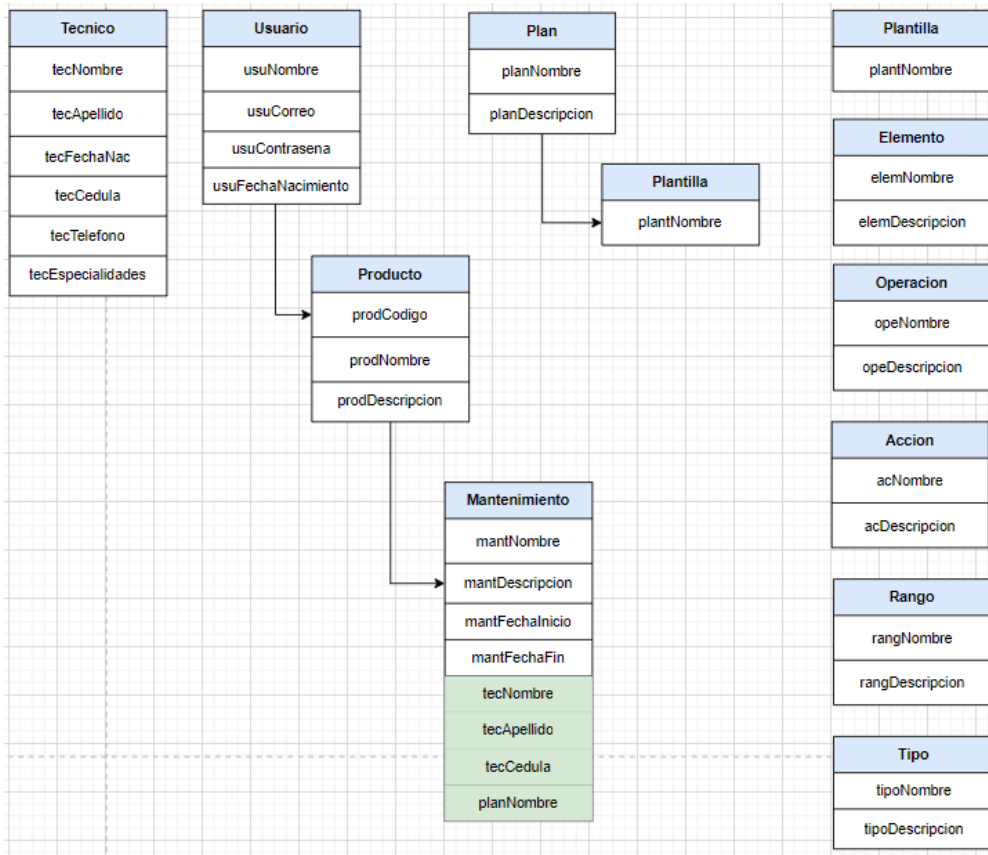


Figura 22. Diseño de base de datos NoSQL

### 2.2.7 Diagrama de navegación

En la Figura 23 se muestra el flujo de navegación para un usuario entre las diferentes partes del módulo de mantenimiento especificadas en las historias de usuario.

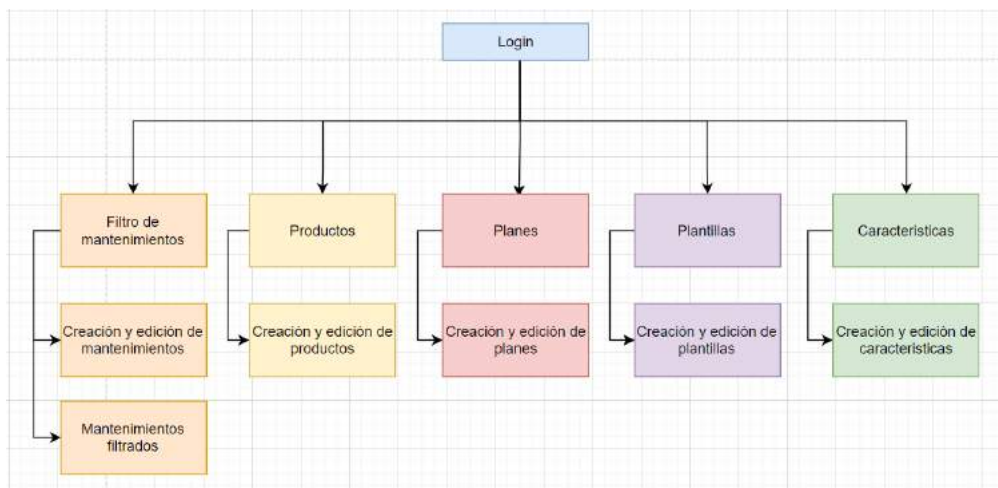


Figura 23. Flujo de navegación

### 2.2.8 Product Backlog

En la Tabla 9 se muestran los requerimientos o historias de usuario de la aplicación móvil junto con una prioridad y peso el cual fue tomado en cuenta al momento de planear los Sprints.

Código	Nombre de historia de usuario	Descripción de historia de usuario	Prioridad	Puntos (/10)
US01	Gestión de usuario	Como persona deseo poder crear un usuario.	Alta	8
US02	Gestión de mantenimientos	Como usuario quiero añadir, editar y eliminar la información de los mantenimientos de los productos a los que tengo acceso.	Alta	10
US03	Gestión de productos	Como usuario quiero añadir, editar y eliminar la información de los productos a los que tengo acceso.	Alta	9
US04	Gestión de técnicos	Como usuario quiero añadir, editar y eliminar la información de los técnicos a los que tengo acceso.	Alta	8
US05	Gestión de plantillas	Como usuario quiero añadir, editar y eliminar la información de las plantillas a los que tengo acceso.	Alta	8
US06	Gestión de planes	Como usuario quiero añadir, editar y eliminar la información de los planes a los que tengo acceso.	Alta	8
US07	Gestión de	Como usuario quiero	Alta	8

	características	añadir, editar y eliminar la información de las características a las que tengo acceso.		
--	-----------------	---	--	--

Tabla 9. Product backlog inicial

## 2.2.9 Sprints

### 2.2.9.1 Sprint 1

**Objetivo:** Implementar la gestión de usuarios mediante la autenticación de FirebaseAuth para facilitar el registro y logeo de usuarios para la aplicación móvil.

#### Sprint planning

Durante el sprint se realizará la implementación de conexión con Firebase para utilizar sus herramientas, de manera principal se realizará la gestión de usuarios y autenticación mediante Firebase Authentication. En la Tabla 10 se muestran las tareas que se van a realizar acorde a la historia de usuario US01 planificada para este sprint.

Código de HU	Tareas		
	Código de tarea	Descripción	Puntuación HU
US01	US01.1	Conexión de proyecto Ionic con Firebase.	8
	US01.2	Codificación de formulario de registro de usuario y de ingreso al sistema.	
	US01.3	Conexión con Firebase Authentication.	
	US01.4	Validación de campos de formulario y validación de ingreso a la aplicación.	
	US01.5	Almacenamiento de información de usuario en Firestore.	

Tabla 10. Tareas para el Sprint 1

## Daily meeting

Durante la ejecución del sprint se presentaron las siguientes dificultades.

Dificultades encontradas	Solución
Generación de conexión con Firebase.	Se añade una aplicación con un nuevo ID en Firebase para poder conectar el proyecto.
Generación de archivo APK añadiendo una nueva aplicación a Firebase.	Se utiliza el id de aplicación de Firebase creado para aplicaciones Android para generar el archivo de una aplicación desarrollada con un enfoque web.
Recuperación de datos o campos de cada documento.	Se mapean los datos en arreglos, dependiendo del caso se recuperan todos los documentos o solo se recupera una parte mediante consultas específicas.
Mostrar información en tarjetas y en opciones de selección.	Se presentan la información mediante cartas propias de Ionic, las opciones para editar o eliminar se realizan mediante opciones que se pueden deslizar.

Tabla 11. Resumen del daily meeting Sprint 1

## Codificación

Se configura en el ambiente del proyecto la conexión para utilizar las propiedades de Firebase mediante el ID de la aplicación web añadiendo una nueva app, en la Figura 24 se puede observar el ID una aplicación web que será utilizada junto con el comando necesario para poder implementar las dependencias.

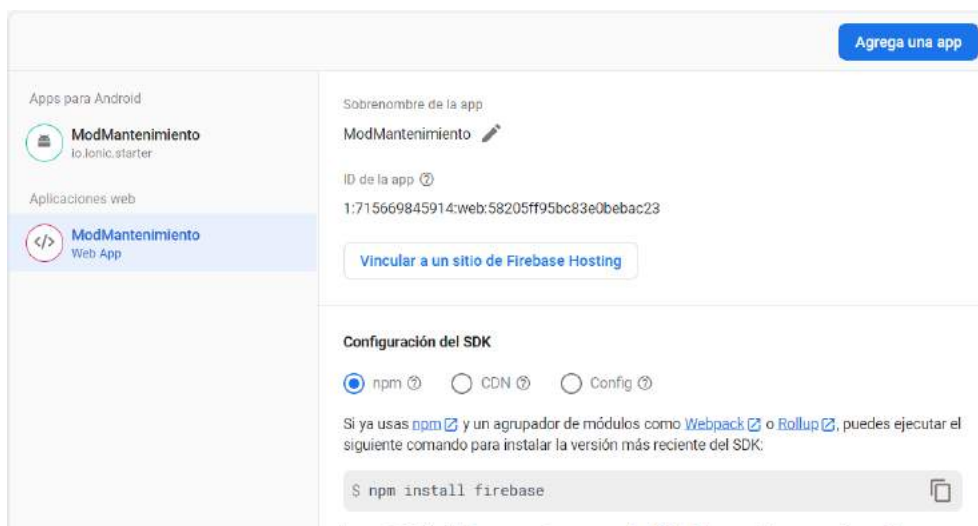


Figura 24. Aplicación web en Firebase

En el ambiente del proyecto se añade la configuración proporcionada por la

documentación de Firebase para poder identificar el proyecto que va a ser utilizado. En la Figura 25 se muestra los campos necesarios para poder conectar la aplicación con las herramientas de Firebase.

```
export const environment = {
  production: false,

  firebaseConfig: {
    apiKey: 'AIzaSyBtPcjnsGuqAVgAPhoKnP1w7mxhikwTiKI',
    authDomain: 'modmantenimiento-4e1e8.firebaseio.com',
    projectId: 'modmantenimiento-4e1e8',
    storageBucket: 'modmantenimiento-4e1e8.appspot.com',
    messagingSenderId: '715669845914',
    appId: '1:715669845914:web:58205ff95bc83e0bebac23'
  }
};
```

Figura 25. Configuración en archivo Environment

Se crean interfaces para el logeo o ingreso de un usuario registrado en la aplicación y el formulario para el registro de un nuevo usuario. En las Figura 26 se puede observar la ventana de ingreso junto con un mensaje de validación en caso de error al momento de ingresar o autenticarse en la aplicación móvil. Este procedimiento es realizado mediante Firebase Authentication. En esta primera parte se completa la primera historia de usuario (US01).

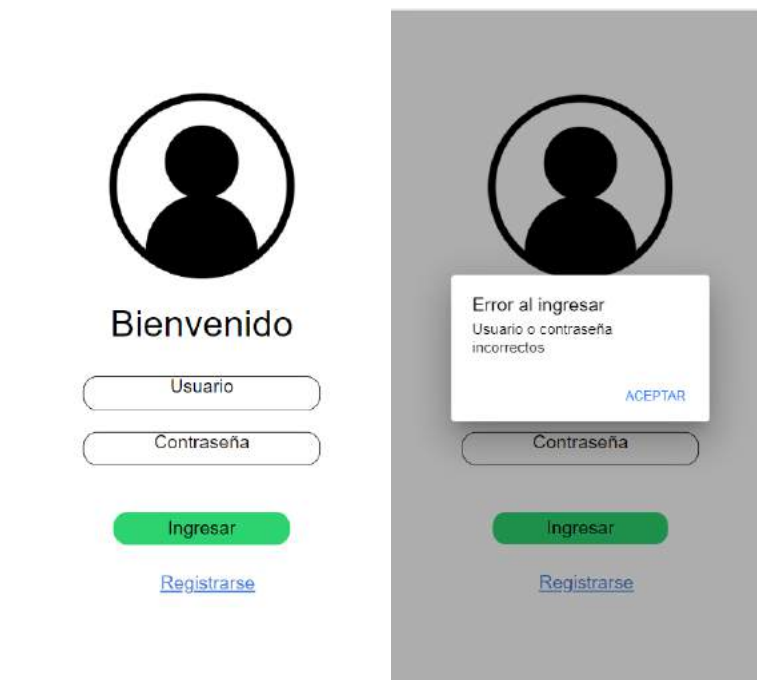


Figura 26. Pantalla de ingreso de la aplicación móvil

La ventana de registro de usuarios contiene los campos de nombre, correo, fecha de nacimiento, contraseña y confirmación de contraseña. En la Figura 27 se pueden observar los campos y el mensaje de alerta en caso de error en el registro.

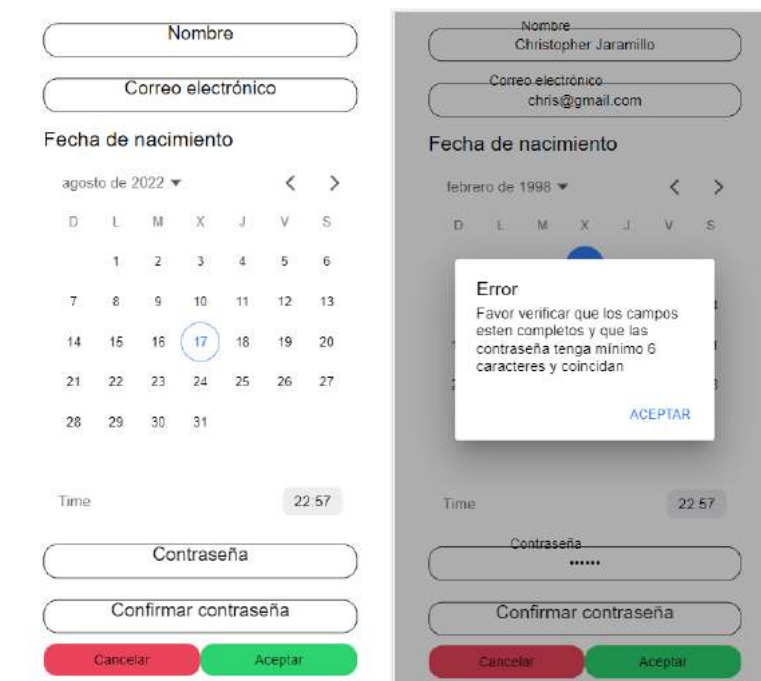


Figura 27. Pantalla de registro de usuario

En la Figura 28 se puede observar los usuarios creados en la herramienta de Firebase Authentication.

Identificador	Proveedores	Fecha de creación	Fecha de acceso	UID de usuario
leo@gmail.com	✉	2 ago 2022	2 ago 2022	sA9abo0K9TeTMI6p11ovzjuYRLQ2
chris@gmail.com	✉	1 ago 2022	4 ago 2022	0XqUzwmVikAXB0V0TH87wPR78...

Figura 28. Usuarios registrados

Para mantener una constancia de los usuarios junto con los datos de creación se registran también como documentos en la colección de “Usuario” como se observa en la Figura 29.

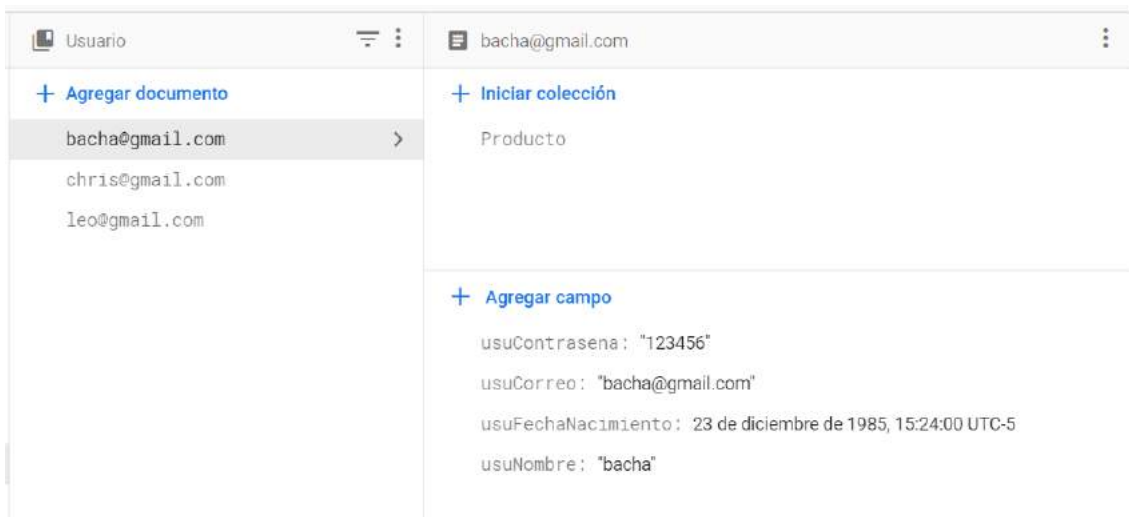


Figura 29. Datos de usuario creado almacenados en Firebase

### Sprint Review

Se cumplió con el objetivo del sprint planteado. La autenticación de usuarios funciona correctamente y al utilizar Firebase Authentication se facilitó la implementación junto con el diseño. Los datos almacenados en Firestore permite tener una constancia sobre los datos con los que se crean los usuarios permitiendo tener un respaldo o modificar en caso de necesitarlo. Al finalizar el sprint 1 se cumplieron con todas las tareas de la Tabla 10 sin ninguna observación sobre lo realizado, este respaldo se puede observar en la Tabla 12.

Código HU	Puntos estimados	Puntos reales	Observaciones
US01	8	8	Ninguna
<b>Total</b>	8	8	

Tabla 12. Revisión de puntuación del sprint 1

### Sprint Retrospective

Al momento la autenticación y registro de usuarios de la aplicación funciona correctamente, se ha implementado los métodos y servicios para mantener un estándar de programación para la aplicación móvil, los estilos aplicados mediante código CSS son valido para aplicar en los siguientes formularios que se necesiten. En el Burndown Chart del sprint correspondiente a la Figura 30, se muestra el avance obtenido con efectividad.

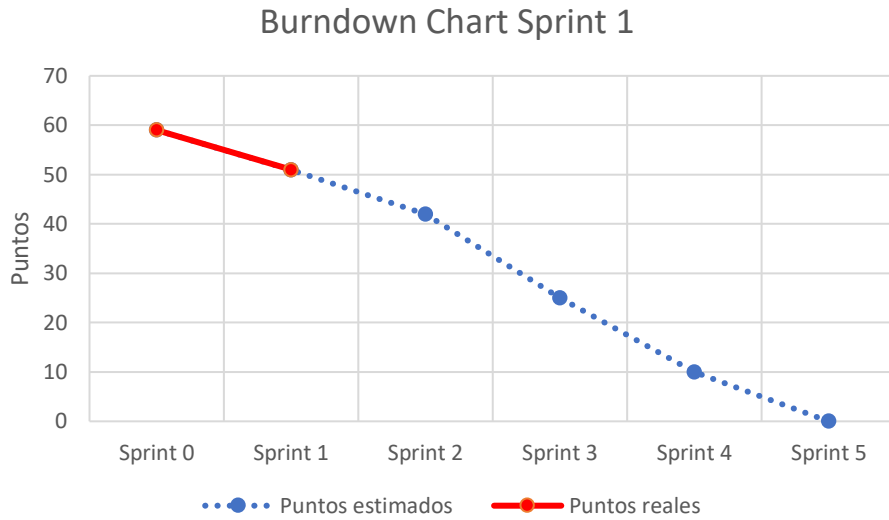


Figura 30. Burndown Chart del sprint 1

La documentación de Firebase facilitó la implementación y conexión de los diferentes componentes con el desarrollo de la aplicación, la mayor parte del tiempo de la implementación se utilizó para analizar y verificar el comportamiento de los componentes de Ionic utilizados, esto permite mantener una mejor constancia sobre los procesos y métodos que se tiene que utilizar.

### 2.2.9.2 Sprint 2

**Objetivo:** Implementar la gestión de mantenimientos para los productos creados por cada usuario, para esta parte se crearon datos de prueba en Firestore.

#### Sprint planning

Durante el sprint se realizará el desarrollo de servicios de lectura, escritura, edición y eliminación para los mantenimientos de los productos para cada usuario, en este sprint se incluyen el desarrollo para los filtros de mantenimientos y el menú de navegación, las tareas se listan en la Tabla 13.

Código de HU	Tareas		
	Código de tarea	Descripción	Puntuación HU
US02	US02.1	Codificación de formulario para mantenimientos y filtros para búsqueda	10



		de mantenimientos.	
	US02.2	Validación de formulario de mantenimientos.	
	US02.3	Codificación de menú de navegación.	

Tabla 13. Tareas para el Sprint 2

### Daily meeting

Durante la ejecución del sprint se presentaron las siguientes dificultades.

Dificultades encontradas	Solución
Los estilos del archivo CSS no se aplican de igual manera que en un proyecto de Angular.	Se crean clases e id para los diferentes elementos de Ionic para poder cumplir con el diseño planteado previamente.
Los elementos de tipo fecha o date se guardan como TimeStamp en Firestore, lo cual dificulta la manera de recuperarlos ya que este elemento se guarda en un formato TimeStamp.	Se recupera la primera parte del tipo de dato TimeStamp donde se puede recuperar la fecha exacta, también se recuperan todos los documentos de la colección en un arreglo para poder filtrarlos mediante diferentes campos.
El uso de un menú contextual es importante para mostrar opciones, en Ionic no existe un elemento predeterminado para poder generar un menú contextual.	Se utiliza el elemento ion sliding para poder mostrar las opciones de mostrar y eliminar para cada una dos resultados de consultas.

Tabla 14. Resumen del daily meeting sprint 2

### Codificación

En la Figura 31 se puede observar la pantalla de filtros para la búsqueda de mantenimientos, se valida el uso de estos para tener una respuesta más exacta.

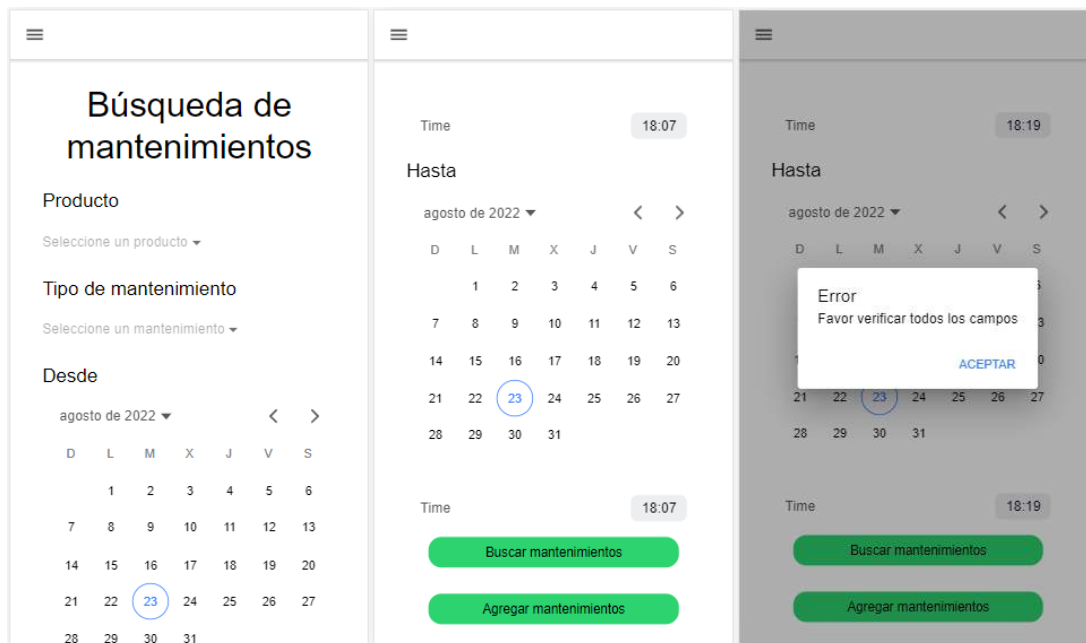


Figura 31. Pantalla de filtros de mantenimientos

Los resultados obtenidos como el mantenimiento que se muestra en la Figura 32, pueden ser editados o eliminados mediante las opciones que se muestran al deslizar sobre un resultado hacia la parte izquierda.



Figura 32. Resultado y opciones de búsqueda de mantenimiento

Para la opción de editar o crear un nuevo mantenimiento se utiliza la pantalla que se indica en la Figura 33, y se realizara la acción respectiva según lo necesitado.

The figure displays two side-by-side mobile application screens for managing maintenance records.

**Top Screen: Editar Mantenimiento**

- Form Fields:**
  - Nombre:** Input field containing "Primer Mantenimiento".
  - Producto:** Dropdown menu with "Seleccione un producto" below it.
  - Plan:** Dropdown menu with "Seleccione un plan" below it.
  - Técnico:** Dropdown menu with "Seleccione un técnico" below it.
  - Tipo de mantenimiento:** Dropdown menu with "Seleccione un tipo de mantenimiento" below it.
  - Fecha inicio:** Date picker showing "agosto de 2022" with a calendar view below it.
  - Fecha fin:** Date picker showing "agosto de 2022" with a calendar view below it. The date "23" is selected and circled in blue.
  - Time:** Two time pickers, both showing "18:45".
  - Descripción:** Input field containing "Mantenimiento por cortesía".
- Buttons:** "Cancelar" (red) and "Editar" (green).

**Bottom Screen: Crear Mantenimiento**

- Form Fields:**
  - Nombre:** Input field containing "Nombre".
  - Producto:** Dropdown menu with "Seleccione un producto" below it.
  - Plan:** Dropdown menu with "Seleccione un plan" below it.
  - Técnico:** Dropdown menu with "Seleccione un técnico" below it.
  - Tipo de mantenimiento:** Dropdown menu with "Seleccione un tipo de mantenimiento" below it.
  - Fecha inicio:** Date picker showing "agosto de 2022" with a calendar view below it.
  - Fecha fin:** Date picker showing "agosto de 2022" with a calendar view below it. The date "23" is selected and circled in blue.
  - Time:** Two time pickers, both showing "18:46".
  - Descripción:** Input field containing "Descripción".
- Buttons:** "Cancelar" (red) and "Crear" (green).

Figura 33. Formulario de creación y edición de mantenimientos

Por otro lado, la opción de eliminar muestra un aviso de confirmación previo a la eliminación de un documento de Firestore como se observa en la Figura 34.

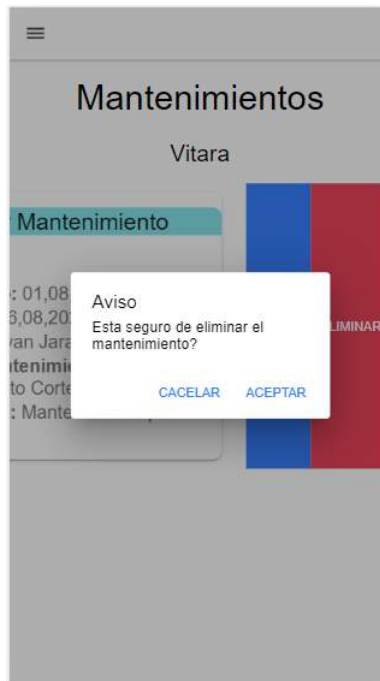


Figura 34. Aviso de confirmación de la opción eliminar

De esta manera se completa el CRUD para la parte de mantenimientos de productos afectando los documentos de las colecciones en Firestore manteniendo la integridad de la información.

Para finalizar este sprint se crea el menú lateral que se observar en la Figura 35 para poder implementar la navegación que puede tener un usuario.

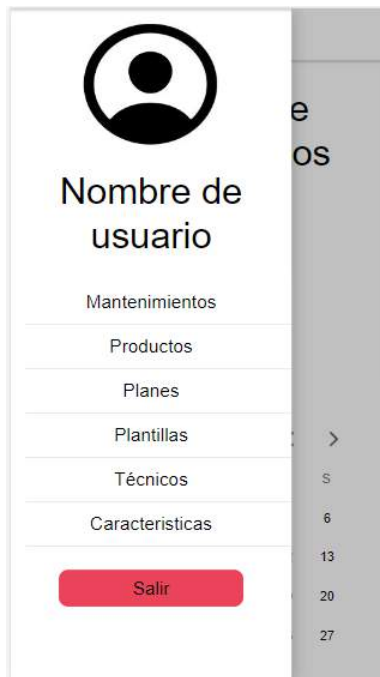


Figura 35. Menú de navegación lateral

### **Sprint Review**

Se cumplió con el objetivo del sprint planteado. Se ha implementado el módulo de mantenimiento con éxito, los filtros funcionan correctamente al igual que los componentes de Ionic implementados, uno de los principales problemas fue la implementación de un menú contextual él requiere una mayor implementación, como alternativa se utilizó opciones de sliding para la edición y eliminación de documentos en Firestore, las observaciones se presentan en la Tabla 15.

<b>Código HU</b>	<b>Puntos estimados</b>	<b>Puntos reales</b>	<b>Observaciones</b>
US02	10	9	La sustitución de un menú contextual por un sliding retraso la implementación de la búsqueda de mantenimientos por lo que aumentaron las horas de trabajo estimadas, sin embargo, el sprint se terminó en la semana determinada
<b>Total</b>	10	9	

Tabla 15. Revisión de puntuación del sprint 2

### **Sprint Retrospective**

Al momento el filtrado junto con el CRUD de mantenimientos de productos funciona correctamente mediante la implementación de servicios y métodos que mantienen un estándar de programación. En la Figura 36, se puede observar el avance hasta al avance hasta el sprint 2.

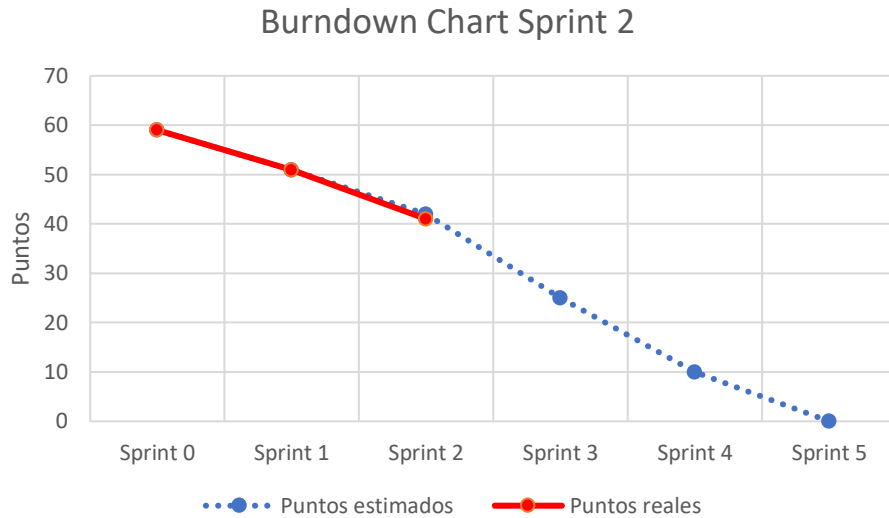


Figura 36. Burndown Chart del sprint 2

La implementación de filtros para la gestión de mantenimientos representa una pantalla la cual tuvo que ser implementada en su totalidad ya que no se puede reutilizar la pantalla de registro de usuario para esta sección, mientras que para el formulario de la creación y edición de mantenimiento se facilitó por la reutilización de la pantalla antes mencionada.

### 2.2.9.3 Sprint 3

**Objetivo:** Implementar la gestión de productos y técnicos según el diseño, para esta parte se crearon datos de prueba en Firestore.

#### Sprint planning

Durante el sprint se realizará el desarrollo de servicios de lectura, escritura, edición y eliminación para los productos y técnicos, en este sprint se incluye los filtros dinámicos para cada pantalla de consulta, las tareas para este sprint se observan y evalúan en la Tabla 16.

Código de HU	Tareas		
	Código de tarea	Descripción	Puntuación HU
US03	US03.1	Codificación de formulario para productos.	9

	US03.2	Validación de formulario de productos.	
	US03.3	Codificación de búsqueda dinámica de productos y de opciones para cada producto.	
US04	US04.1	Codificación de formulario para técnicos.	8
	US04.2	Validación de formulario de técnicos.	
	US04.3	Codificación de búsqueda dinámica de técnicos y de opciones para cada técnico.	

Tabla 16. Tareas para el Sprint 3

### Daily meeting

Durante la ejecución del sprint se presentaron las siguientes dificultades.

Dificultades encontradas	Solución
No existe un componen en Ionic que permite realizar búsqueda dinámica de manera automática y general.	Se utilizan los componentes pipes para poder generar filtros para cada una de las búsquedas que se necesiten.

Tabla 17. Resumen del daily meeting sprint 3

### Codificación

Se crean las pantallas donde se recuperan todos los productos y técnicos, en el caso de los productos la consulta depende del usuario que inicie la sesión, y en el caso de los técnicos estos son visibles para poder registrar los mantenimientos correspondientes, en la Figura 37 se pueden observar las consultas de los técnicos y los productos al igual que el funcionamiento de las consultas dinámicas.

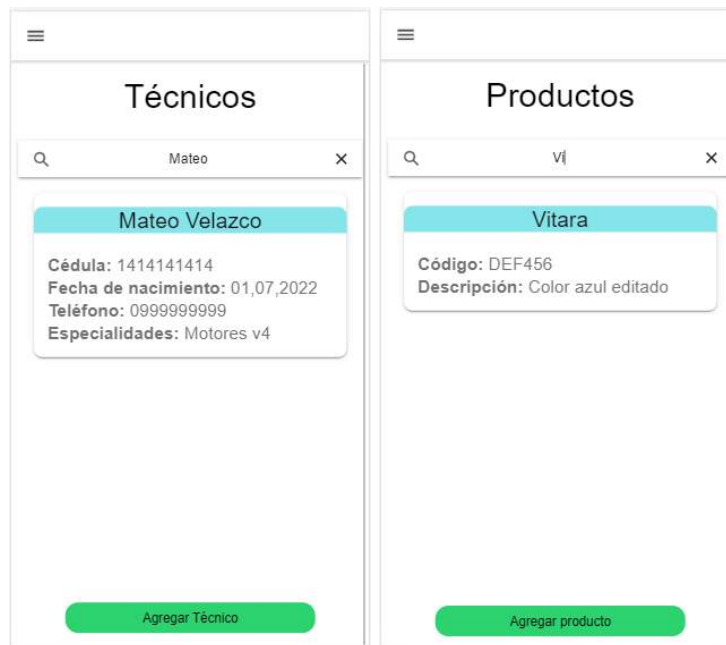


Figura 37. Búsqueda de técnicos y productos

Al igual que en el sprint 2, se utiliza una pantalla para realizar la creación y edición tanto para los técnicos y para los productos como se observa en las Figura 38 y Figura 39.

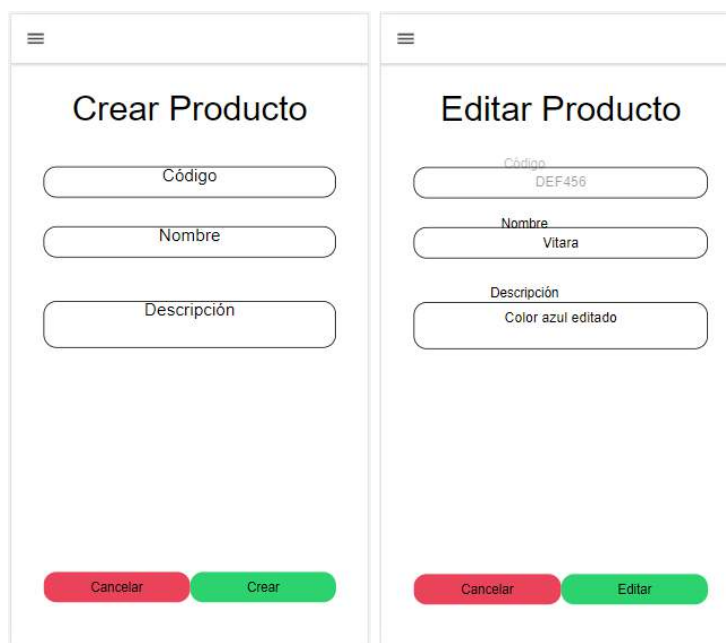


Figura 38. Formulario de creación y edición de productos



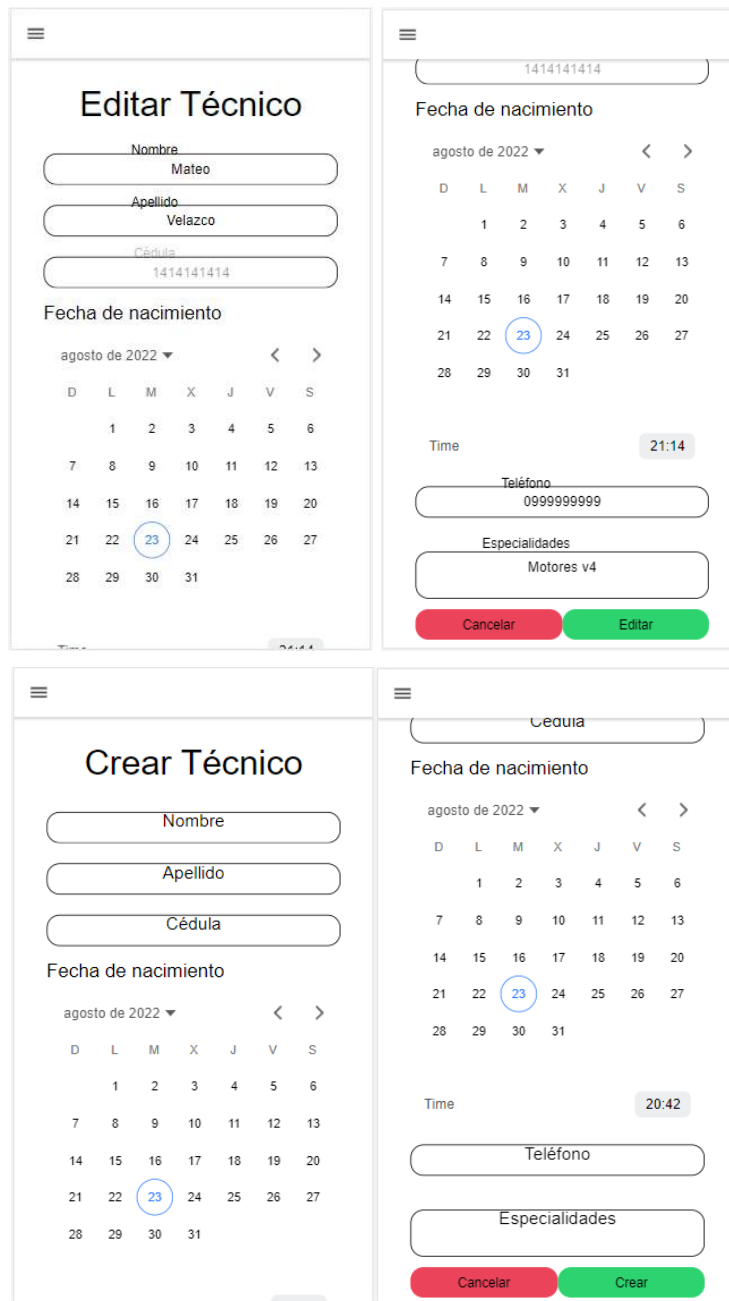


Figura 39. Formulario de creación y edición de técnicos

Para poder realizar una prueba se realiza un commit a la rama de development en Gitlab y se realiza un merge request a la rama de testing para poder generar el archivo APK y probar la aplicación, el resultado de éxito se puede observar en la Figura 40 donde se muestra la ejecución del pipeline y en la Figura 41 donde se obtiene la versión del archivo APK en Firebase App Distribution.

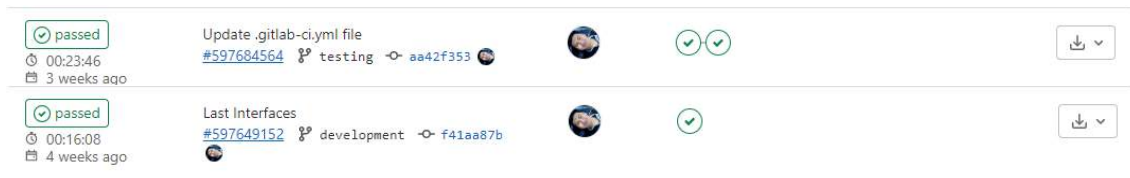


Figura 40. Ejecución exitosa de los pipelines

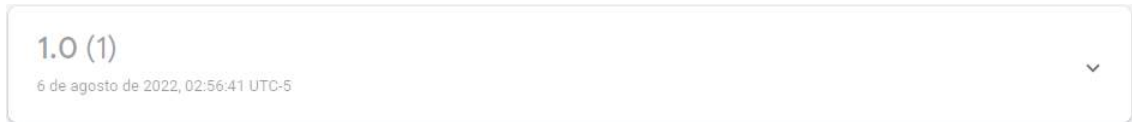


Figura 41. Versión de archivo APK de la aplicación móvil

La opción de eliminación muestra un mensaje de confirmación para los técnicos y productos como se observa en la Figura 42.

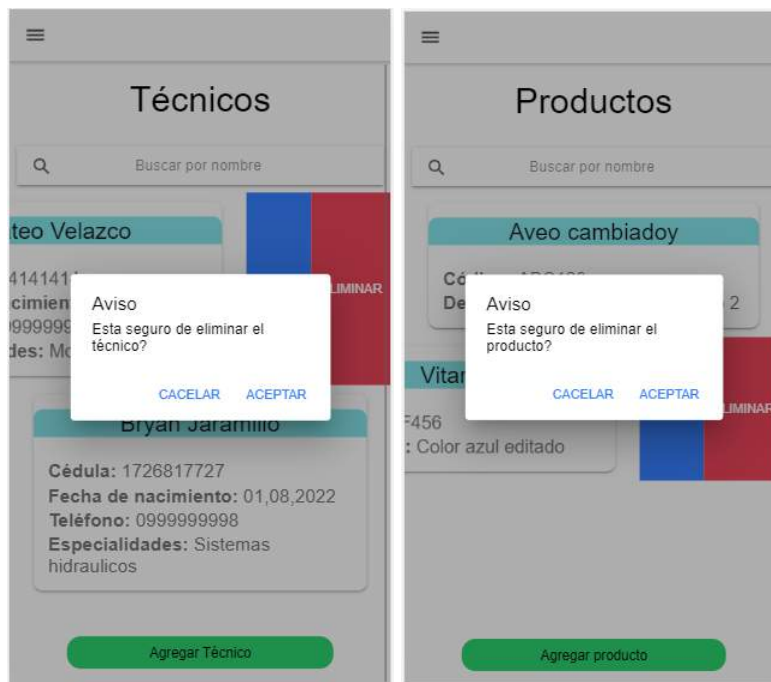


Figura 42. Opción de eliminación de técnicos y productos

### Sprint Review

Se cumplió con el objetivo del sprint planteado. Las opciones de técnicos y productos se facilitaron debido a que se puede utilizar las pantallas de mantenimiento como plantilla para las nuevas pantallas a crear. Una de las nuevas características que se necesita implementar es la búsqueda dinámica para cada pantalla donde se muestre información, esto se logró mediante pipes, uno para cada búsqueda que se necesite. En la Tabla 18 se muestra la puntuación para las historias de usuario de este sprint.

Código HU	Puntos estimados	Puntos reales	Observaciones
US03	9	8	Para la búsqueda dinámica se implementa un pipe, ya que esta implementación requiere un funcionamiento especial, esto permite utilizarlo como plantilla para las siguientes pantallas.
US04	9	8	Ninguna.
<b>Total</b>	<b>17</b>	<b>16</b>	

Tabla 18. Revisión de puntuación del sprint 3

### Sprint Retrospective

La implementación de las historias de usuario correspondientes al sprint realizado se facilitó debido a la reutilización de componentes utilizados en el sprint anterior. De igual manera las pantallas obtenidas como resultado de este sprint van a servir como nuevas plantillas por su implementación de una búsqueda dinámica, en este sprint el avance es ideal como se observa en la Figura 43.

Burndown Chart Sprint 3

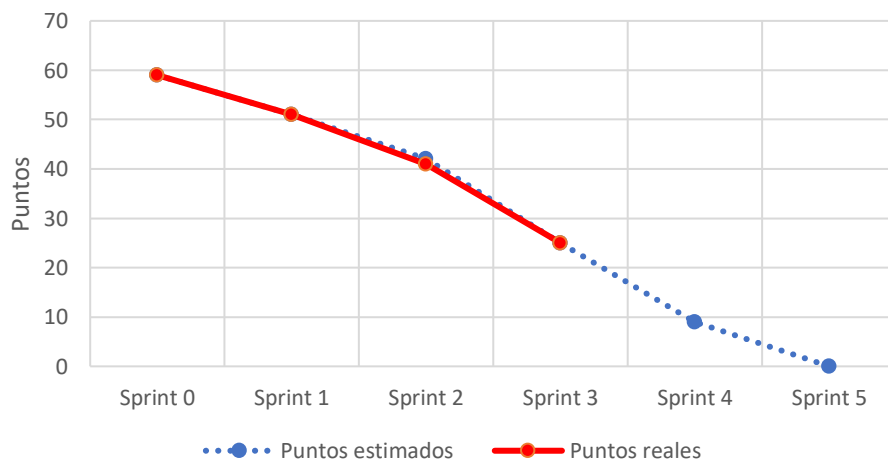


Figura 43. Brundown Chart del sprint 3

En este sprint se realizó un commit de los avances realizado hasta el sprint 3 para poder agregar una nueva versión de la aplicación al versionamiento en Firebase App Distribution, esto para evaluar los resultados tanto del desarrollo de la aplicación como la aplicación de la integración continua la cual resulto exitosa, por otro lado, el desarrollo de la aplicación funciona de la manera prevista.

#### 2.2.9.4 Sprint 4

**Objetivo:** Implementar la gestión de planes y plantillas según el diseño, para esta parte se crearon datos de prueba en Firestore.

##### Sprint planning

Durante el sprint se realizará el desarrollo de servicios de lectura, escritura, edición y eliminación para los planes y plantillas, en este sprint se incluye los filtros dinámicos para cada pantalla de consulta. En la Tabla 19 se observan las tareas que se van a realizar durante el sprint.

Código de HU	Tareas		
	Código de tarea	Descripción	Puntuación HU
US05	US05.1	Codificación de formulario para planes.	8
	US05.2	Validación de formulario de planes.	
	US05.3	Codificación de búsqueda dinámica de planes y de opciones para cada plan.	
US06	US06.1	Codificación de formulario para plantillas.	8
	US06.2	Validación de formulario de plantillas.	
	US06.3	Codificación de búsqueda dinámica de plantillas y de opciones para cada plantilla.	

Tabla 19. Tareas para el Sprint 4

## Daily meeting

Durante la ejecución del sprint se presentaron las siguientes dificultades.

Dificultades encontradas	Solución
Es necesario crear una lista de check boxes o casillas de selección para las opciones, sin embargo, no es posible identificar los check box seleccionados al realizar este procedimiento	Se utiliza una función para detectar los cambios de selección que se realice en cada check boxes y poder recuperar la información necesaria para llenar los documentos de Firestore.

Tabla 20. Resumen del daily meeting sprint 4

## Codificación

Se crean las pantallas donde se recuperan todos las plantillas y planes, en la Figura 44 se pueden observar las consultas de las plantillas y planes al igual que el funcionamiento de las consultas dinámicas.

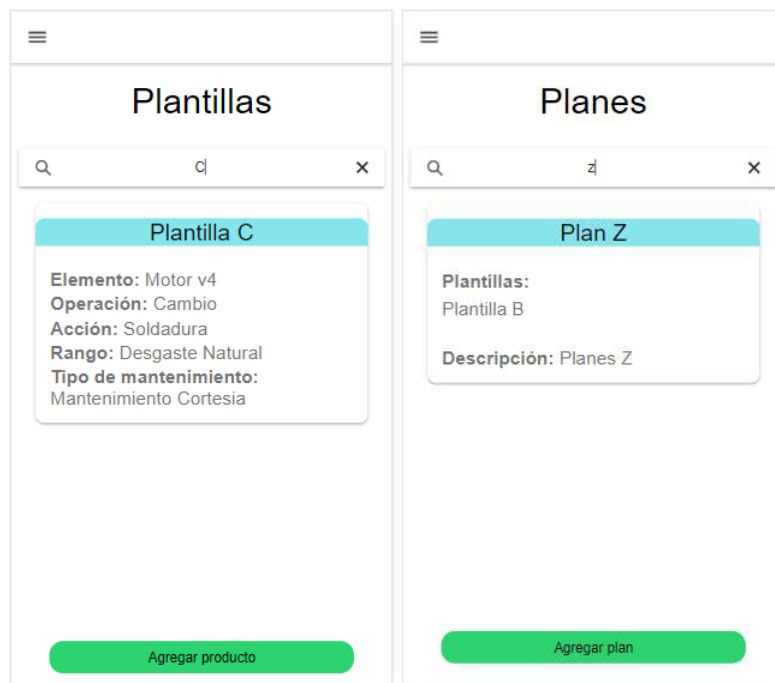


Figura 44. Búsqueda de planes y plantillas

Al igual que en el sprint 2 y 3, se utiliza una pantalla para realizar la creación y edición tanto para las plantillas y para los planes como se observa en las Figura 45 y Figura 46.

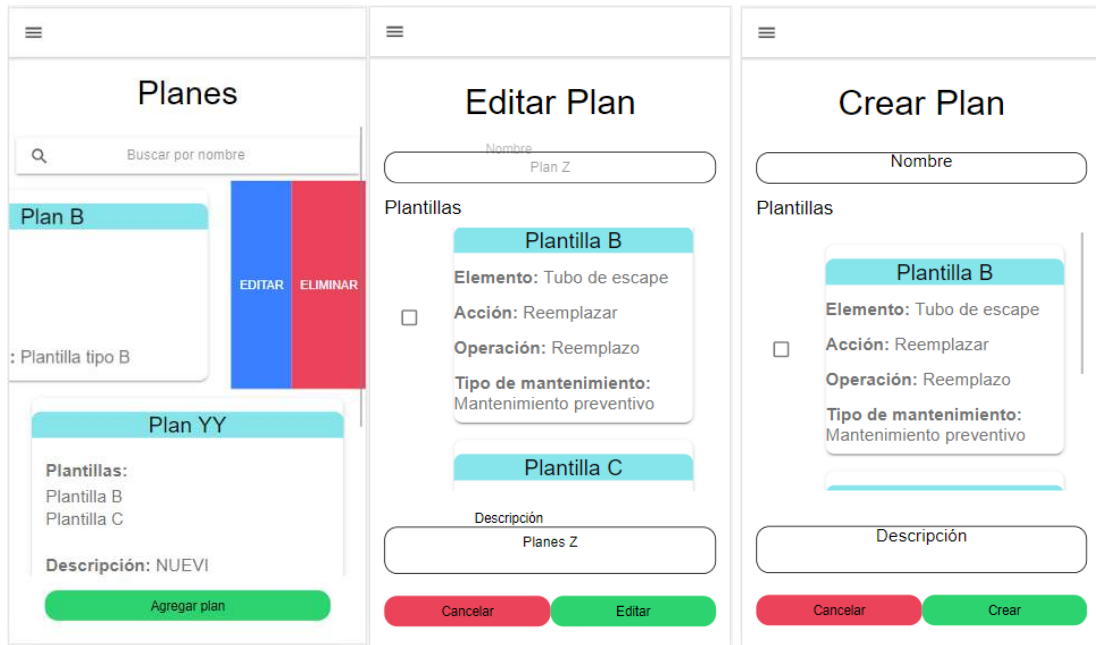


Figura 45. Formulario de creación y edición de planes

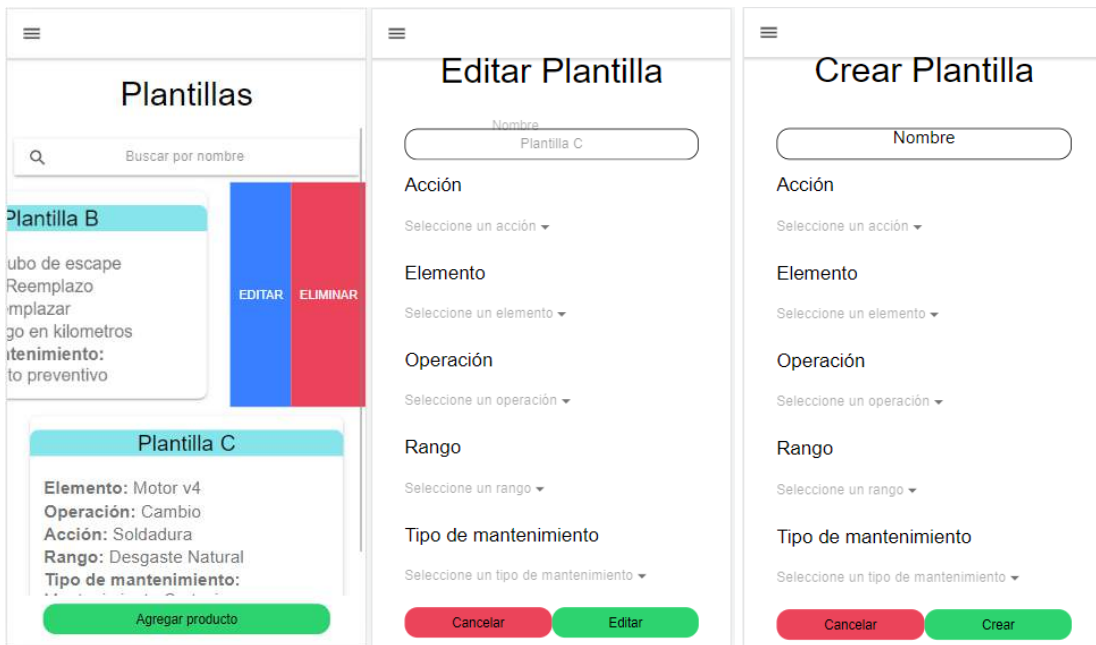


Figura 46. Formulario de creación y edición de plantillas

En la Figura 47 se puede observar los avisos de eliminación previos a borrar un documento tanto para las plantillas como para los planes.

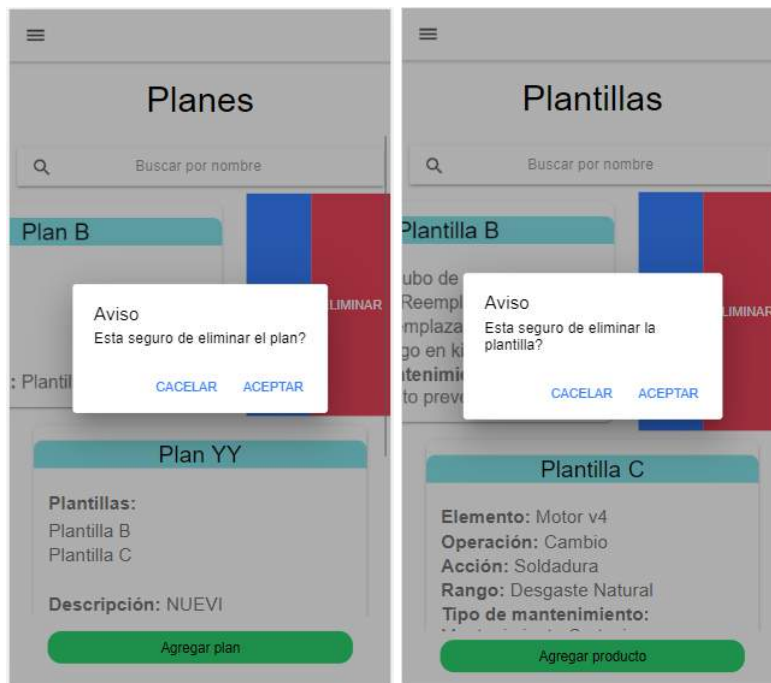


Figura 47. Opción de eliminación de planes y plantillas

### Sprint Review

Se cumplió con el objetivo del sprint planteado. Una de las características que se tuvo que implementar fue un check list a partir de una lista recuperada desde Firestore, para poder mantener las opciones seleccionadas se hizo uso de código en Typescrip para poder recuperar las opciones que se necesiten almacenar. Las pantallas implementadas funcionan correctamente tanto para la creación, edición, eliminación y búsqueda de planes y plantillas. Las observaciones se adjuntan en la Tabla 21.

Código HU	Puntos estimados	Puntos reales	Observaciones
US05	8	8	Ninguna
US06	8	7	La implementación de un método para recuperar la selección de los checks puede mejorarse o reemplazarse por otros elementos de Ionic.
<b>Total</b>	16	15	

Tabla 21. Revisión de puntuación sprint 4

## Sprint Retrospective

La implementación de las historias de usuario correspondientes al sprint realizado se facilitó debido a la reutilización de componentes utilizados en el sprint anterior, al finalizar el sprint 4 se puede notar una pequeña desviación en el Burndown Chart que se expresa en la Figura 48.

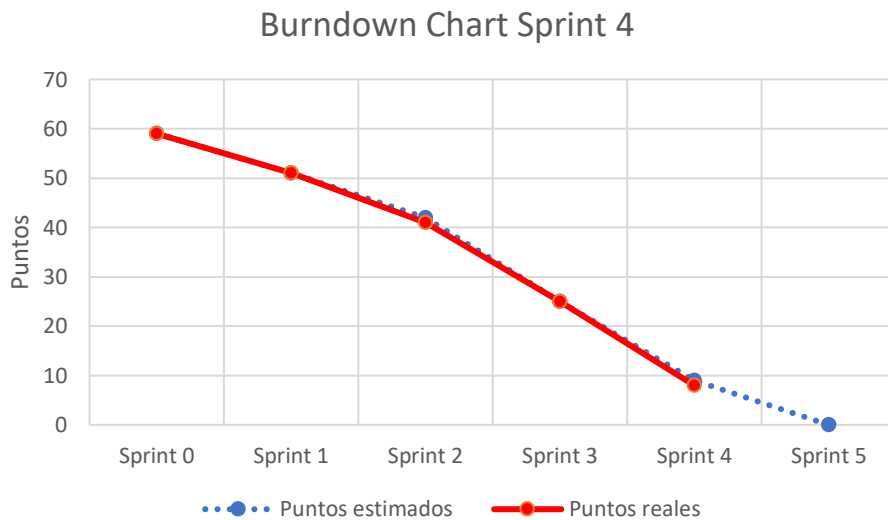


Figura 48. Brundown Chart del sprint 4

La implementación de codificación adicional para manejar procesos como la selección y des selección de una lista de checks aumento del tiempo de investigación para buscar alternativas, finalmente se optó por manejar el funcionamiento del componente mediante código en lugar de una característica propia del componente, al finalizar con sprint se obtuvo de manera exitosa el funcionamiento de las historias de usuario correspondiente.

### 2.2.9.5 Sprint 5

**Objetivo:** Implementar la gestión de características según el diseño, para esta parte se crearon datos de prueba en Firestore.

#### Sprint planning

Durante el sprint se realizará el desarrollo de servicios de lectura, escritura, edición y eliminación para las características, para la historia de usuario correspondiente se reutilizo la misma pantalla para todas las características (rango, operaciones, acciones, elementos y tipos de mantenimientos) de esta



manera la implementación minimiza el tiempo ya que solo es necesario actualizar la información cambio entre las características que se necesiten, esta historia de usuario se detalla en la Tabla Tabla 22.

Código de HU	Tareas		
	Código de tarea	Descripción	Puntuación HU
US07	US07.1	Codificación de formulario para características.	8
	US07.2	Validación de formulario de características.	
	US07.3	Codificación de búsqueda dinámica para cada característica.	

Tabla 22. Tareas para el Sprint 5

### Daily meeting

Durante la ejecución del sprint se presentaron las siguientes dificultades.

Dificultades encontradas	Solución
Ya que las características tienen los mismos formularios y vistas es mejor reutilizar la misma pantalla.	Se usa una selección dinámica para cambiar los resultados según se necesite.

Tabla 23. Resumen del daily meeting sprint 5

### Codificación

Se crea la pantalla para poder observar las características que se necesiten visualizar y modificar mediante una selección, en la Figura 49 se muestra las características como opciones y los resultados obtenidos.

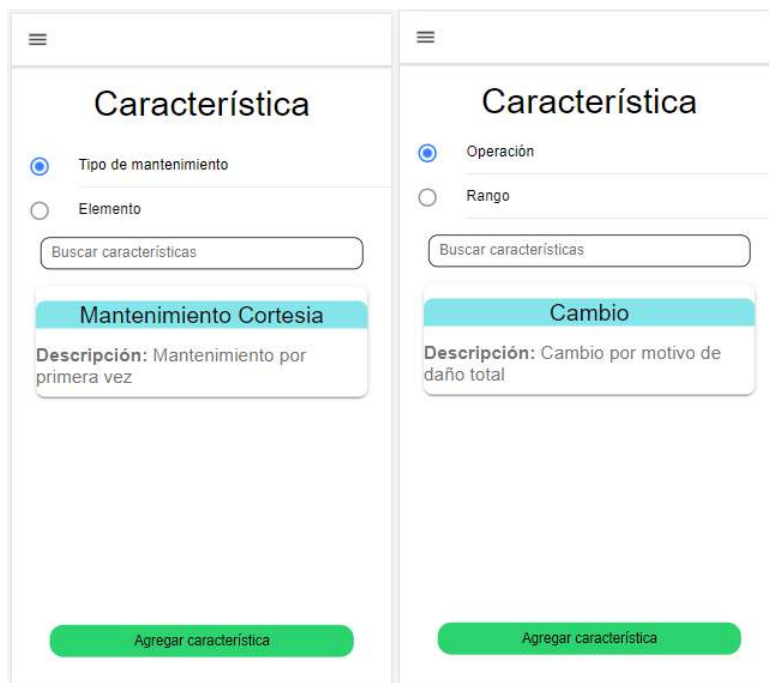


Figura 49. Búsqueda de características por selección

Al igual que en el sprint 2, 3 y 4, se utiliza una pantalla para realizar la creación y edición de todas las características siempre seleccionando la opción que se desea crear o editar como se puede observar en la Figura 50.

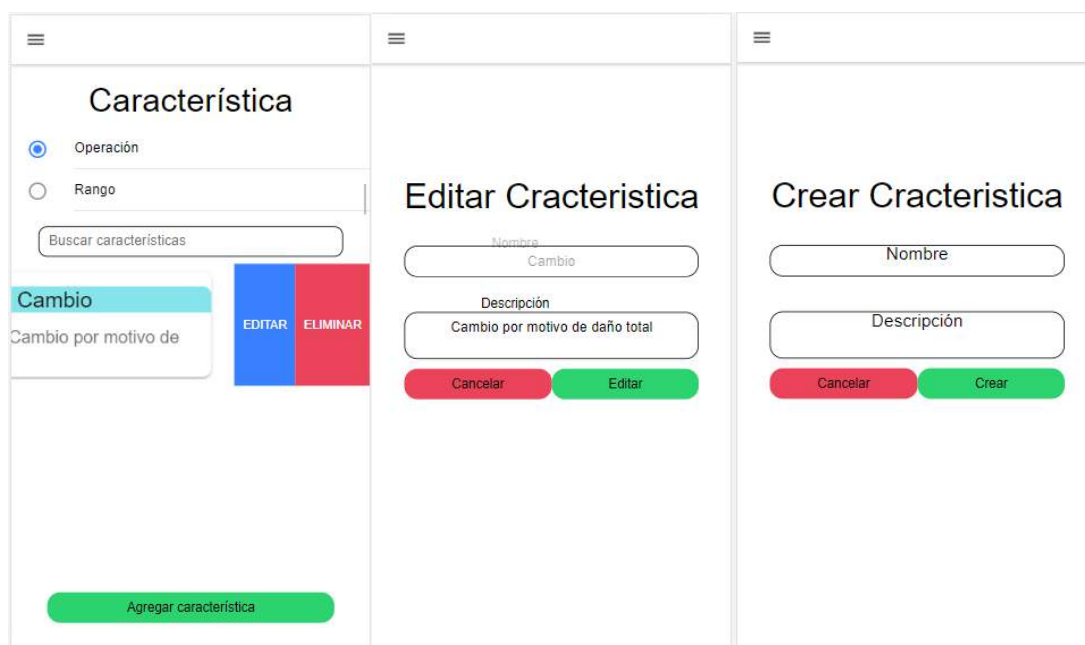


Figura 50. Formulario de creación y edición de características

La opción de eliminación muestra un mensaje de confirmación para todas las características que se consulten como se observa en la Figura 51.

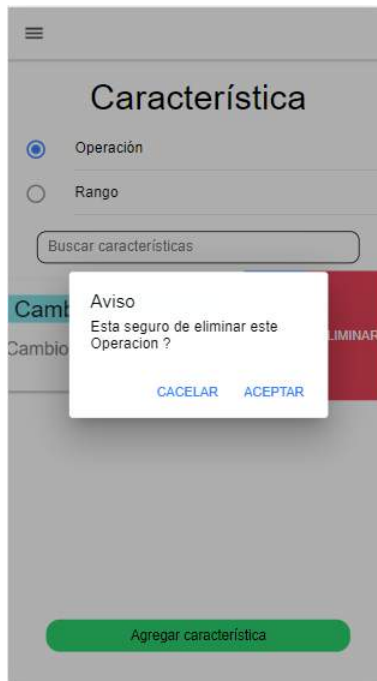


Figura 51. Opción de eliminación de características

Al finalizar con la última historia de usuario, se realiza otro commit a la rama de development y a la rama de testing para generar el archivo APK en Firebase App Distribution, en la Figura 52 se puede observar los pipelines ejecutados con éxito.

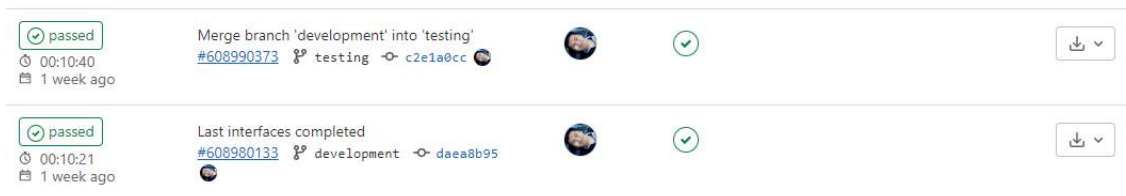


Figura 52. Pipelines ejecutados con éxito

En la Figura 53 se puede observar todas las versiones cargadas desde el inicio del proyecto, durante los sprints hasta la creación y prueba de todas las historias de usuario completadas.



Figura 53. Versiones de la aplicación móvil

## Sprint Review

Se cumplió con el objetivo del sprint planteado. Ya que las características emplean el uso de un mismo formulario y vista de consulta, se utiliza una selección para poder cambiar entre las diferentes características según la elección, de esta manera las vistas se reutilizan de manera se agilizó el desarrollo y la implementación, para este sprint no existe ninguna observación como se muestra en la Tabla 24.

Código HU	Puntos estimados	Puntos reales	Observaciones
US07	8	8	Ninguna
<b>Total</b>	8	8	

Tabla 24. Revisión de puntuación sprint 4

## Sprint Retrospective

La implementación de las historias de usuario correspondientes al sprint realizado se facilitó debido a la reutilización de componentes utilizados en el sprint anterior. Durante el último sprint se realizó el último commit a las ramas del repositorio para poder generar la versión final del APK, al final del sprint no se obtuvo ninguna desviación como se parecía en la Figura 54.

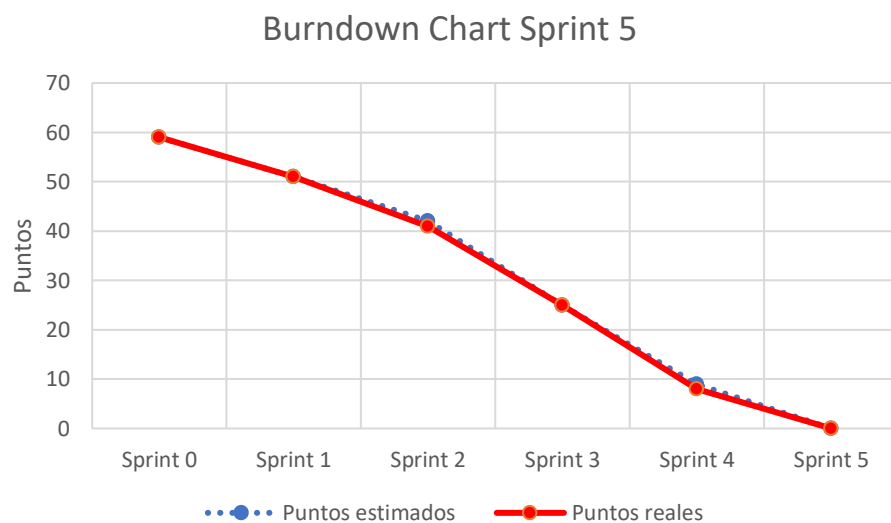


Figura 54. Brundown Chart del sprint 5

Para la implementación de todas las características se utilizó una lista de selección para poder realizar las diferentes operaciones para cada característica, de esta

manera se reutilizan las ventanas aprovechando el tiempo para validar las validaciones implementadas. Al finalizar con el sprint se obtiene la aplicación final, el objetivo del sprint se cumplió correctamente por la reutilización de pantallas.

### 3 PRUEBAS

#### 3.1 Pruebas de usabilidad

Para medir la usabilidad de la aplicación móvil se utilizó la metodología de System Usability Scale (SUS), en la Tabla 25 se detalla las preguntas implementadas para realizar el cuestionario SUS, el cuestionario tiene una medida en 1 a 5, donde 1 significa “Desacuerdo totalmente” y 5 significa “Conuerdo totalmente”.

N	Preguntas	Valoración				
		1	2	3	4	5
1	Creo que usaría esta aplicación móvil frecuentemente.					
2	Encuentro esta aplicación móvil innecesariamente compleja.					
3	Creo que la aplicación móvil fue fácil de usar.					
4	Creo que necesitaría ayuda de una persona con conocimientos técnicos para usar esta aplicación móvil.					
5	Las funciones de la aplicación móvil están bien integradas.					
6	Creo que la aplicación móvil es muy inconsistente.					
7	Imagino que la mayoría de la gente aprendería a usar esta aplicación móvil en forma muy rápida.					
8	Encuentro que la aplicación móvil es muy difícil de usar.					
9	Me siento confiado al usar esta aplicación móvil.					
10	Necesité aprender muchas cosas antes de ser capaz de usar esta aplicación móvil.					

Tabla 25. Cuestionario SUS para aplicación móvil

Para poder medir la usabilidad de manera numérica se utiliza la Ecuación 1, primero se obtienen los promedios de los resultados para cada pregunta, para las preguntas 1, 3, 5, 7 y 9 se resta 1 punto al promedio de cada una, para las

preguntas 2, 4, 6, 8 y 10 se resta de 5 el promedio obtenido, finalmente los valores obtenidos serán sumados en la ecuación descrita a continuación para obtener el resultado de la usabilidad de la aplicación móvil.

$$Puntaje\ SUS = 2.5 * \sum_{i=1}^{10} P_i$$

Ecuación 1. Fórmula para el cálculo del puntaje SUS

En la Tabla 26 se observan los resultados de la encuesta realizada, para cada encuesta se detallan las puntuaciones. Para la evaluación de la usabilidad se realizó la encuesta a personas entre 16 a 25 años, la mayoría de estas se realizó de manera virtual mediante Google Meets, mientras que otra parte se realizó de manera presencial para poder analizar de mejor manera la perspectiva de los usuarios.

Núm. Encuesta	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
1	5	2	5	2	5	1	4	1	4	3
2	4	3	4	2	4	2	4	1	5	1
3	5	1	4	1	4	1	5	1	4	1
4	4	2	4	4	4	2	4	2	4	2
5	5	1	5	1	5	1	5	1	5	1
6	3	1	5	1	4	2	5	2	5	2
7	5	1	5	5	4	1	5	2	5	3
<b>Promedio</b>	4,43	1,57	4,57	2,29	4,29	1,43	4,57	1,43	4,57	1,86

Tabla 26. Resultados de formulario de usabilidad

Aplicando la Ecuación 1 se obtiene que el resultado del puntaje SUS es de 84.64, esto indica que la usabilidad de la aplicación es satisfactoria al momento de ser utilizada por los usuarios.

### 3.2 Pruebas de funcionalidad

Para poder probar la funcionalidad del desarrollo e implementación de la aplicación móvil se ha definido casos de prueba que incluyen información relevante para poder probar partes importantes y medir la conformidad de los usuarios al momento de utilizar la aplicación móvil.

La Tabla 27 muestra el caso de prueba para registrar un nuevo usuario en el sistema, mientras la Tabla 28 indica los resultados obtenidos por los usuarios.

<b>Código</b>	CP-01		
<b>Funcionalidad</b>	Registrar nuevo usuario		
<b>Entradas</b>	Nombre: Christopher Jaramillo Correo: <a href="mailto:chris@gmail.com">chris@gmail.com</a> Fecha de nacimiento: 04/02/1998 Contraseña: 123456 Confirmación de contraseña: 123456		
<b>Prerrequisitos</b>	Ingresar a la aplicación móvil.		
<b>Acciones y/o condiciones</b>	Acceder a la aplicación móvil.		
	Dar clic en la opción de “Registrarse”		
	Llenar el formulario de registro de usuario con las entradas descritas anteriormente.		
	Dar clic en crear.		
<b>Esperados</b>	<b>Obtenidos</b>		
	<b>Si</b>	<b>No</b>	<b>Parcialmente</b>
Se muestra el mensaje de confirmación “El usuario ha sido registrado con éxito”.			
Se crea un nuevo usuario			
Se redirige al usuario a la página de Login.			

Tabla 27. Caso de prueba 01

<b>Resultados</b>				
<b>Esperados</b>	<b>Obtenidos</b>			
	<b>Si</b>	<b>No</b>	<b>Parcialmente</b>	<b>Total</b>
Se muestra el mensaje de confirmación “El usuario ha sido registrado con éxito”.	7	0	0	7
Se crea un nuevo usuario	7	0	0	7
Se redirige al usuario a la página de Login.	7	0	0	7

Tabla 28. Resultado para el caso de prueba 01

La Tabla 29 muestra el caso de prueba para ingresar al sistema, mientras la Tabla 30 indica los resultados obtenidos por los usuarios.

<b>Código</b>	CP-02		
<b>Funcionalidad</b>	Ingreso al sistema		
<b>Entradas</b>	Usuario: <a href="mailto:chris@gmail.com">chris@gmail.com</a> Contraseña: 123456		
<b>Prerrequisitos</b>	Registrarse en la aplicación móvil.		
<b>Acciones y/o condiciones</b>	Acceder a la aplicación móvil.		
	Llenar el formulario de ingreso con las entradas descritas anteriormente.		
<b>Esperados</b>	<b>Obtenidos</b>		
	<b>Si</b>	<b>No</b>	<b>Parcialmente</b>
Se redirige a la página de búsqueda de mantenimientos.			

Tabla 29. Caso de prueba 02

<b>Resultados</b>				
<b>Esperados</b>	<b>Obtenidos</b>			
	<b>Si</b>	<b>No</b>	<b>Parcialmente</b>	<b>Total</b>
Se redirige a la página de búsqueda de mantenimientos.	7	0	0	7

Tabla 30. Resultado para el caso de prueba 02

La Tabla 31 muestra el caso de prueba para crear un producto en el sistema, mientras la Tabla 32 indica los resultados obtenidos por los usuarios.

<b>Código</b>	CP-03
<b>Funcionalidad</b>	Creación de producto
<b>Entradas</b>	Código: 123 Nombre: Aveo Family Descripción: auto 2019 color gris
<b>Prerrequisitos</b>	Ingreso a la aplicación móvil con credenciales válidas
<b>Acciones y/o condiciones</b>	Acceder a la aplicación móvil.
	En el menú que se encuentra en la parte superior izquierda



	seleccionar la opción de “Productos”.		
	Dar clic en el botón de “Agregar producto” que se encuentra en la parte inferior.		
	Llenar el formulario con los campos indicados anteriormente.		
	Dar clic en crear		
Esperados	Obtenidos		
	Si	No	Parcialmente
Se registra un nuevo producto.			
Se muestra el mensaje “Producto creado con éxito”			
Se redirige a la pantalla de productos			

Tabla 31. Caso de prueba 03

Resultados				
Esperados	Obtenidos			
	Si	No	Parcialmente	Total
Se registra un nuevo producto.	7	0	0	7
Se muestra el mensaje “Producto creado con éxito”	7	0	0	7
Se redirige a la pantalla de productos	7	0	0	7

Tabla 32. Resultado para el caso de prueba 03

La Tabla 33 muestra el caso de prueba para visualizar los productos creados, mientras la Tabla 34 indica los resultados obtenidos por los usuarios.

<b>Código</b>	CP-04
<b>Funcionalidad</b>	Visualización de producto
<b>Entradas</b>	Ninguna
<b>Prerrequisitos</b>	Acceder a la aplicación móvil y seleccionar en el menú superior izquierdo la opción de “Productos”
<b>Acciones y/o condiciones</b>	Acceder a la aplicación móvil. En el menú que se encuentra en la parte superior izquierda

	seleccionar la opción de “Productos”.		
Esperados	Obtenidos		
	Si	No	Parcialmente
Se visualiza los productos existentes			
Al deslizar sobre un producto hacia la izquierda se pueden ver las opciones de “Editar” y “Eliminar”			

Tabla 33. Caso de prueba 04

Resultados				
Esperados	Obtenidos			
	Si	No	Parcialmente	Total
Se visualiza los productos existentes	7	0	0	7
Al deslizar sobre un producto hacia la izquierda se pueden ver las opciones de “Editar” y “Eliminar”	7	0	0	7

Tabla 34. Resultado para el caso de prueba 04

La Tabla 35Tabla 33 muestra el caso de prueba para editar un producto, mientras la Tabla 36 indica los resultados obtenidos por los usuarios.

<b>Código</b>	CP-05		
<b>Funcionalidad</b>	Edición de producto		
<b>Entradas</b>	Nombre: Aveo Family Descripción: auto 2019 color azul		
<b>Prerrequisitos</b>	Acceder a la aplicación móvil y seleccionar en el menú superior izquierdo la opción de “Productos”		
<b>Acciones y/o condiciones</b>	Deslizar sobre un producto creado para mostrar las opciones.		
	Seleccionar la opción de “Editar”		
	Llenar el formulario con las entradas indicadas anteriormente.		
	Dar clic en el botón de “Editar” en la parte inferior.		
Esperados	Obtenidos		
	Si	No	Parcialmente

Se visualiza el producto editado			
Se muestra el mensaje "Producto editado con éxito"			
Se redirige a la pantalla de productos.			

Tabla 35. Caso de prueba 05

<b>Resultados</b>				
<b>Esperados</b>	<b>Obtenidos</b>			
	<b>Si</b>	<b>No</b>	<b>Parcialmente</b>	<b>Total</b>
Se visualiza el producto editado	6	0	1	7
Se muestra el mensaje "Producto editado con éxito"	6	0	1	7
Se redirige a la pantalla de productos.	6	0	1	7

Tabla 36. Resultado para el caso de prueba 05

La Tabla 37 muestra el caso de prueba para crear un nuevo mantenimiento, mientras la Tabla 38 indica los resultados obtenidos por los usuarios.

<b>Código</b>	CP-06	
<b>Funcionalidad</b>	Creación de mantenimiento	
<b>Entradas</b>	Nombre: Primer mantenimiento Producto: Aveo Family Tipo de mantenimiento: Mantenimiento semestral Técnico: Bryan Jaramillo Fecha de inicio: 01/08/2022 Fecha de fin: 01/09/2022 Descripción: Mantenimiento por primera vez	
<b>Prerrequisitos</b>	Acceder a la aplicación móvil y pulsar el botón de agregar mantenimientos en la parte inferior.	
<b>Acciones y/o condiciones</b>	Acceder a la aplicación móvil,	
	Llenar el formulario con las entradas indicadas anteriormente.	
	Dar clic en el botón de "Crear" en la parte inferior.	
<b>Esperados</b>	<b>Obtenidos</b>	

	Si	No	Parcialmente
Se visualiza el producto editado			
Se muestra el mensaje "Producto editado con éxito"			
Se redirige a la pantalla de productos.			

Tabla 37. Caso de prueba 06

Resultados				
Esperados	Obtenidos			
	Si	No	Parcialmente	Total
Se muestra el mensaje "Mantenimiento creado con éxito"	6	0	1	7
Se redirige a la pantalla de mantenimientos.	6	0	1	7

Tabla 38. Resultado para el caso de prueba 06

La Tabla 39 muestra el caso de prueba para visualizar un mantenimiento, mientras la Tabla 40 indica los resultados obtenidos por los usuarios.

<b>Código</b>	CP-07		
<b>Funcionalidad</b>	Visualización de mantenimiento		
<b>Entradas</b>	Producto: Aveo Family Tipo de mantenimiento: Mantenimiento semestral Fecha de inicio: 01/08/2022 Fecha de fin: 01/09/2022		
<b>Prerrequisitos</b>	Acceder a la aplicación móvil.		
<b>Acciones</b>	Llenar el formulario con las entradas indicadas anteriormente.		
	Dar clic en el botón de "Buscar mantenimientos"		
Esperados	Obtenidos		
	Si	No	Parcialmente
Se visualiza los mantenimientos existentes			
Al deslizar sobre un producto hacia la izquierda se pueden ver las opciones de			

“Editar” y “Eliminar”			
-----------------------	--	--	--

Tabla 39. Caso de prueba 07

<b>Resultados</b>				
<b>Esperados</b>	<b>Obtenidos</b>			
	<b>Si</b>	<b>No</b>	<b>Parcialmente</b>	<b>Total</b>
Se visualiza los mantenimientos existentes	6	0	1	7
Al deslizar sobre un producto hacia la izquierda se pueden ver las opciones de “Editar” y “Eliminar”	6	0	1	7

Tabla 40. Resultado para el caso de prueba 07

La Tabla 41 muestra el caso de prueba editar un mantenimiento, mientras la Tabla 42 indica los resultados obtenidos por los usuarios.

<b>Código</b>	CP-08		
<b>Funcionalidad</b>	Edición de mantenimiento		
<b>Entradas</b>	Producto: Aveo Family Tipo de mantenimiento: Mantenimiento semestral Técnico: Bryan Jaramillo Fecha de inicio: 01/08/2022 Fecha de fin: 01/09/2022 Descripción: Mantenimiento por segunda vez		
<b>Prerrequisitos</b>	Acceder a la aplicación móvil.		
<b>Acciones</b>	Llenar las entradas de filtros.		
	Dar clic en el botón de “Buscar mantenimientos”.		
	Deslizar sobre el mantenimiento que se quiere editar hacia la izquierda.		
	Seleccionar la opción de “Editar”.		
	Llenar el formulario con la información indicada anteriormente.		
	Dar clic en el botón de “Editar” en la parte inferior.		
<b>Esperados</b>	<b>Obtenidos</b>		
	<b>Si</b>	<b>No</b>	<b>Parcialmente</b>

Se redirige a la pantalla de mantenimientos.			
Se muestra el mensaje “Mantenimiento editado con éxito”			

Tabla 41. Caso de prueba 08

<b>Resultados</b>				
<b>Esperados</b>	<b>Obtenidos</b>			
	<b>Si</b>	<b>No</b>	<b>Parcialmente</b>	<b>Total</b>
Se redirige a la pantalla de mantenimientos.	6	0	1	7
Se muestra el mensaje “Mantenimiento editado con éxito”	6	0	1	7

Tabla 42. Resultado para el caso de prueba 08

La Tabla 43 muestra el caso de prueba para la búsqueda de técnicos, mientras la Tabla 44 indica los resultados obtenidos por los usuarios.

<b>Código</b>	CP-09		
<b>Funcionalidad</b>	Búsqueda de técnicos		
<b>Entradas</b>	Nombre de técnico.		
<b>Prerrequisitos</b>	Acceder a la aplicación móvil.		
<b>Acciones</b>	En el menú de la parte superior izquierda, selección la opción “Técnicos”		
	Ingresar un nombre en la barra de búsqueda en la parte superior.		
<b>Esperados</b>	<b>Obtenidos</b>		
	<b>Si</b>	<b>No</b>	<b>Parcialmente</b>
Se muestra los técnicos acordes a la búsqueda			

Tabla 43. Caso de prueba 09

Resultados				
Esperados	Obtenidos			
	Si	No	Parcialmente	Total
Se muestra los técnicos acordes a la búsqueda	7	0	0	7

Tabla 44. Resultado para el caso de prueba 09

Las pruebas de usabilidad demuestran que la aplicación móvil permite su uso de manera eficiente para los usuarios, sin embargo, se puede interpretar los resultados como una oportunidad de mejorar la navegación y diseño, esto puede ser una pauta para buscar alternativas o mejoras a los componentes de Ionic implementados como el formato de las fechas.

Las pruebas de funcionalidad indican que los usuarios tuvieron problemas principalmente con el apartado de mantenimientos, esto puede deberse a la necesidad de utilizar filtros para los mantenimientos, lo cual puede aumentar la complejidad al momento de realizar acciones como la visualización, edición y eliminación de un mantenimiento, para una futura versión este tema puede ser un punto clave para cambiar o mejorar. En la Figura 55 se muestra los resultados obtenidos del formulario de funcionalidad aplicado sobre los usuarios.

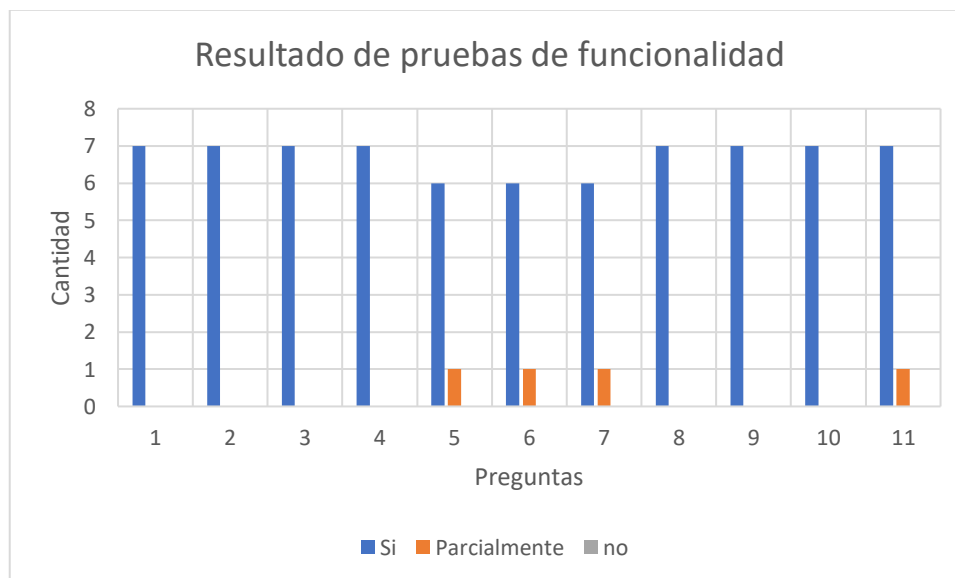


Figura 55. Resultados de funcionalidad

## 4 RESULTADOS

### 4.1 Efectividad basada en ejecución de pipelines

Para poder medir la efectividad del uso de prácticas DevOps en el desarrollo de una aplicación móvil se puede tomar como referencia los pipelines realizados tomando en cuenta los estados de estos los cuales son aprobado, fallido y rechazado. En la Figura 56 se observa la representación de porcentajes de los 3 estados resultados de todas las ejecuciones de commits a las dos ramas del repositorio en Gitlab, estos resultados se observan en la Tabla 45.

Estado de Pipeline	Cantidad
Aprobado	29
Fallido	16
Cancelado	5
<b>Total</b>	<b>50</b>

Tabla 45. Resultados de pipelines

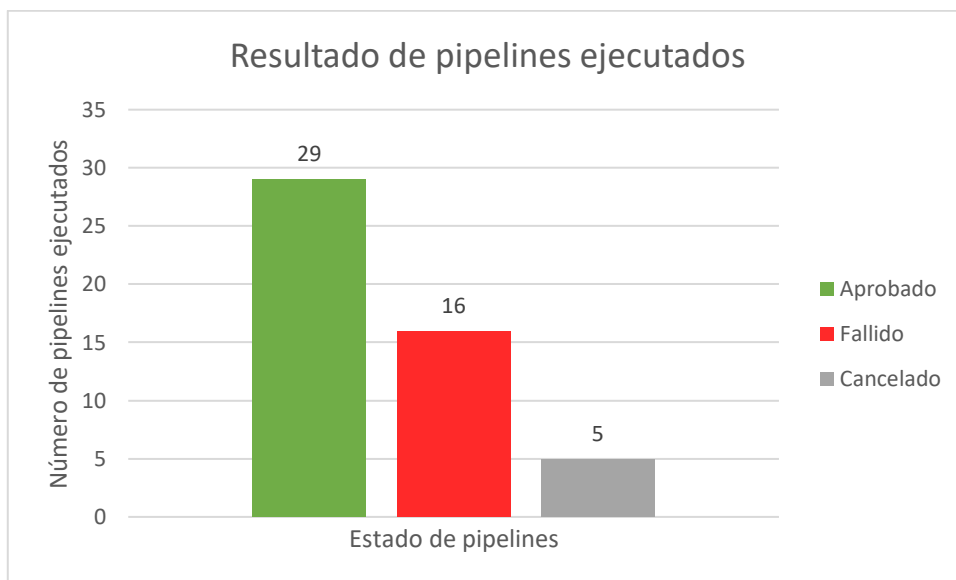


Figura 56. Resultado de pipelines ejecutados

El número de pipelines rechazados corresponde a pruebas realizadas para probar y corregir los comandos con los cuales el archivo APK se va a generar y subir de manera automática a Firebase App Distribution, las pruebas incluyen la implementación de dependencias y el cambio de nombre de archivos APK para el correcto versionamiento de la aplicación móvil.



Por otro lado, la mayor parte del número de pipelines aprobados, también corresponden a pruebas realizadas para comprobar que los comandos que se ejecutan funcionen no solo en el pipeline, si no también verificar los directorios creados, generación del archivo APK, cambio del nombre del archivo y carga del archivo en Firebase App Distribution. En menor parte los pipelines aprobados corresponden a ejecuciones correctas después de realizar commits a las diferentes ramas del repositorio durante el desarrollo de la aplicación móvil.

Por último, el número de pipelines cancelados corresponden a pruebas innecesarias debido a cambios mal configurados o errores de sintaxis al momento modificar el archivo para los pipelines.

## 4.2 Archivos en Firebase App Distribution

El versionamiento de la aplicación móvil resulto adecuado para las diferentes etapas desarrolladas durante los sprints, en la Figura 57 se pueden observar todas las versiones del archivo APK correspondiente al desarrollo realizado, de esta materia se pudo probar la aplicación desde un dispositivo móvil real para analizar configuraciones o aumentar un mejor nivel de experiencia según el uso de la aplicación simulando un ambiente real de uso.

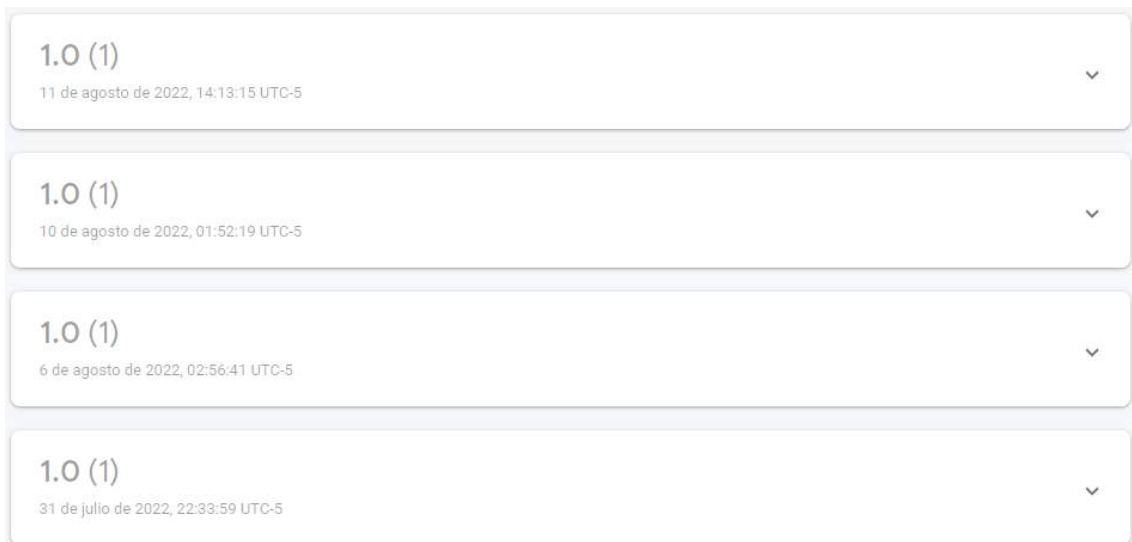


Figura 57. Versiones totales de aplicación móvil

Las pruebas se realizaron validando las características correspondientes a cada uno de los sprints desarrollados, en donde se pudo examinar de manera específica todos los procesos y características para la gestión de cada parte del módulo.

### **4.3 Aplicación móvil**

El módulo de mantenimiento funciona en su totalidad y de manera correcta en la versión final que se encuentra en Firebase App Distribution, las características que se utilizaron tanto para el backend y frontend permitieron obtener una aplicación escalable, la base de datos no relacional permite aumentar información necesaria en cualquier versión futura ya que la estructura del diseño no relacional puede aumentar colecciones y documentos sin la necesidad de diseñar nuevamente toda la base de datos en Firestore. El framework Ionic también ayuda a la escalabilidad por la facilidad de aumentar componentes de manera rápida por código HTML, aplicar estilos mediante código CSS y manejar la funcionalidad mediante JS, manteniendo un desarrollo con orientación web.

### **4.4 Modelo de integración continua**

Previo al desarrollo de la aplicación móvil se configuró un entorno de integración continua en un repositorio de Gitlab correspondiente a prácticas DevOps. La integración continua corresponde al archivo YML del proyecto en Gitlab, la configuración del archivo incluye la generación automática del archivo APK y de la carga de las versiones de estos archivos a Firebase App Distribution.

La configuración del archivo YML permite que cualquier aplicación móvil (desarrollado en Ionic) que se cargue en la rama de development en el repositorio ejecutara un pipeline donde un stage se encargara de crear el archivo APK, mientras que si se hace un commit o un merge a la rama de testing se cargara el archivo APK a Firebase App Distribution.

La imagen utilizada en Docker y los comandos para las instalar las dependencias pueden ser utilizados de manera general en la implementación de una proyecto con integración continua, por otro lado para poder conseguir enviar el archivo APK a Firebase App Distribution es necesario generar un token para poder autenticar y verificar el archivo que se va a cargar y el ID de la aplicación Android generada en Firebase para distinguir el proyecto en el cual se va a gestionar las versiones del aplicación móvil, por último es necesario verificar los directorios para poder establecer correctamente las carpetas que se van a crear y sobre las cuales se van a trabajar.

## 5 CONCLUSIONES

Al analizar el desarrollo de la aplicación móvil mediante la metodología Scrum y la implementación de una integración continua se obtuvo como resultado un modelo a seguir para poder realizar la integración continua de cualquier aplicación móvil de tipo web. Las prácticas DevOps representan una alternativa de despliegue de aplicaciones móviles altamente eficiente ya que aumenta el rendimiento de las pruebas de funcionalidad y usabilidad, al reducir el tiempo que conlleva el proceso de unión y adaptación de las diferentes partes desarrolladas, permitiendo corregir errores de adaptación y desarrollo de manera más eficiente y eficaz.

Para la implementación continua es necesario establecer la arquitectura de desarrollo y funcionamiento de manera efectiva, ya que si existe algún cambio en esta arquitectura puede ocasionar que el proceso de integración continua falle cuando se ejecute la automatización de los procesos de verificación y despliegue de la aplicación móvil, por otro lado, las dependencias para el desarrollo como complementos o librerías pueden ser añadidas fácilmente ya que al momento de la integración continua las dependencias pueden ser incluidas en cada momento.

Una manera efectiva de implementar una integración continua es probando o ejecutando los comandos que se van a utilizar de manera local en un equipo, dependiendo del sistema operativo o imagen que se va a utilizar o donde se ejecutaran los procesos de manera automática, el funcionamiento de los comandos será el mismo en un archivo YML y permitirá buscar alternativas o mejorar la eficiencia y eficacia de los comandos y su ejecución al momento de desplegar la integración continua de un proyecto.

La mejor alternativa para implementar la integración continua de prácticas DevOps es mediante un repositorio como en este caso Gitlab, ya que posee configuraciones fáciles de implementar para poder integrar fácilmente una integración continua, adicionalmente todo el proceso que se realice se puede monitorear en todo momento.

La codificación implementada en el archivo YML representa una generalidad de la aplicación de prácticas DevOps para poder gestionar la construcción y despliegue de una aplicación móvil mediante versiones, al finalizar con este proceso también se puede analizar el flujo utilizado el cual puede ayudar a mejorar mediante variaciones que mantengan la estructura de DevOps.

El desarrollo de la aplicación móvil mediante el uso de la metodología Scrum permitió que la abstracción de requerimientos pueda gestionarse y distribuirse de manera correcta para la fase de desarrollo, los sprints junto con todos sus pasos se

lograron ejecutar de manera independiente y sin afectar las prácticas DevOps implementadas, esto garantiza una efectividad al momento de utilizar Scrum y DevOps.

Al utilizar el framework Ionic basado en Angular, la aplicación móvil mantuvo un enfoque de desarrollo web, los componentes de Ionic permitieron gestionar de mejor manera el frontend ya que los componentes pueden ser implementados fácilmente mediante HTML y CSS, se puede implementar prácticas correctas para la usabilidad durante el desarrollo, el backend es gestionado mediante herramientas de Firebase y el lenguaje de programación de Typescript esto permite mantener un estándar adecuado sobre los recursos creados para el funcionamiento de la aplicación móvil.

## 6 RECOMENDACIONES

En el desarrollo de la aplicación móvil se tomó en cuenta únicamente la generación del archivo APK para Android por su facilidad de probar su funcionamiento en dispositivos móviles reales, por esta razón es eficiente complementar el desarrollo y almacenamiento de datos mediante Firebase, en caso de que se requiera generar aplicaciones para dispositivos iOS es recomendable reevaluar la integración continua.

Para el desarrollo del componente existen dos opciones para el desarrollo, Ionic y Android Studio, Ionic se utilizó para un uso de investigación de este framework en conjunto con la metodología scrum y las prácticas DevOps, sin embargo, es recomendable evaluar el diseño o las opciones que se necesitan para poder seleccionar la mejor opción ya que pueden existir limitaciones entre las dos opciones lo cual puede condicionar el desarrollo del frontend y backend.

## 7 REFERENCIAS BIBLIOGRÁFICAS

- [1] X. Cmapaña, «itconsultors,» 20 08 2019. [En línea]. Available: <https://www.itconsultors.com/metodologia-scrum>. [Último acceso: 30 08 2022].
- [2] G. Puetate y J. L. Ibarra, «MMA Spain,» [En línea]. Available: <https://mmaspain.com/wp-content/uploads/2015/09/Libro-Blanco-Apps.pdf>. [Último acceso: 2022 05 24].
- [3] J. Cuello y J. Vittone, Diseñando apps para móviles, Catalina Duquee Giraldo,

2013.

- [4] H. Bustillos, «Encora,» 31 Julio 2020. [En línea]. Available: <https://www.encora.com/es/blog/implementing-continuous-integration-and-continuous-deployment-on-gitlab>. [Último acceso: 15 Agosto 2022].
- [5] L. Puciarellu, Angular, Buenos Aires, 2020.
- [6] «fullstack,» [En línea]. Available: <https://www.fullstack.pe/blog/angular-data-binding>. [Último acceso: 24 05 2022].
- [7] P. Arizmendi, Angular, 2018.
- [8] J. M. A. Atmitim, «Profile,» 22 02 2021. [En línea]. Available: [https://profile.es/blog/que-es-ionic/#%C2%BFQue\\_es\\_ionic\\_y\\_para\\_que\\_sirve](https://profile.es/blog/que-es-ionic/#%C2%BFQue_es_ionic_y_para_que_sirve). [Último acceso: 25 05 2022].
- [9] aws, «aws.amazon,» [En línea]. Available: <https://aws.amazon.com/es/docker/>. [Último acceso: 25 05 2022].
- [10] Turingears, «Turingears,» 2022. [En línea]. Available: [https://turingears.com/que-es-gitlab/?utm\\_source=rss&utm\\_medium=rss&utm\\_campaign=que-es-gitlab](https://turingears.com/que-es-gitlab/?utm_source=rss&utm_medium=rss&utm_campaign=que-es-gitlab). [Último acceso: 25 05 2022].
- [11] M. A. Alvarez, «desarrolloweb,» 14 01 2022. [En línea]. Available: <https://desarrolloweb.com/articulos/dbeaver>. [Último acceso: 25 05 2022].
- [12] P. Romos, «styde,» 14 09 2018. [En línea]. Available: <https://styde.net/que-es-y-para-que-sirve-sql/>. [Último acceso: 25 05 2022].

## 8 ANEXOS

ANEXO I. Enlace al prototipo de la aplicación móvil en Figma

<https://www.figma.com/file/D239XeHU4mCm1KjepT0RBH/Prototipo?node-id=0%3A1>

ANEXO II. Enlace al diseño de base relacional y no relacional

[https://drive.google.com/file/d/1dYSgL9\\_IY9ny667Tele5wEbR2FXIY8Q7/view?usp=sharing](https://drive.google.com/file/d/1dYSgL9_IY9ny667Tele5wEbR2FXIY8Q7/view?usp=sharing)

ANEXO III. ANEXO III. Video de módulo de mantenimiento de sistemas Sisreco

<https://www.youtube.com/watch?v=pUBuAc8EX1s&t=935s>

ANEXO IV. Carpeta con documentos de historias de usuario

<https://drive.google.com/drive/folders/1ekFMfXpHzlkoMx2ZSZuAVZrL74JJSnUC?usp=sharing>

ANEXO V. Repositorio Gitlab con el código del módulo de mantenimiento

<https://gitlab.com/manticore-labs/trabajo/proyectos/trabajo-de-titulacion/cp-jaramillo>

ANEXO VI. Formulario de usabilidad

## Encuesta de usabilidad de aplicación móvil.

Encuesta para medir la usabilidad de una aplicación móvil para un módulo de mantenimiento de un ERP.

Correo \*

Correo válido

Este formulario registra los correos. [Cambiar configuración](#)

1. Creo que usaría esta aplicación móvil frecuentemente \*

1      2      3      4      5

Desacuerdo totalmente                        Conuerdo totalmente

2. Encuentro esta aplicación móvil innecesariamente compleja \*

1      2      3      4      5

Desacuerdo totalmente                        Conuerdo totalmente

3. Creo que la aplicación móvil fue fácil de usar \*

	1	2	3	4	5	
Desacuerdo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Conuerdo totalmente

4. Creo que necesitaría ayuda de una persona con conocimientos técnicos para usar esta aplicación móvil \*

	1	2	3	4	5	
Desacuerdo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Conuerdo totalmente

5. Las funciones de la aplicación móvil están bien integradas \*

	1	2	3	4	5	
Desacuerdo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Conuerdo totalmente

6. Creo que la aplicación móvil es muy inconsistente \*  
...

	1	2	3	4	5	
Desacuerdo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Conuerdo totalmente

7. Imagino que la mayoría de la gente aprendería a usar esta aplicación móvil en forma muy rápida \*

	1	2	3	4	5	
Desacuerdo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Conuerdo totalmente

8. Encuentro que la aplicación móvil es muy difícil de usar \*

	1	2	3	4	5	
Desacuerdo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Conuerdo totalmente

9. Me siento confiado al usar esta aplicación móvil \*

	1	2	3	4	5	
Desacuerdo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Conuerdo totalmente

10. Necesité aprender muchas cosas antes de ser capaz de usar esta aplicación móvil \*

	1	2	3	4	5	
Desacuerdo totalmente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Conuerdo totalmente

ANEXO VII. Formulario de funcionalidad



Pruebas de usabilidad \*

	Si	Parcialmente	No
¿Pude crear un usuario correctamente?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
¿Pude acceder con mi usuario y contraseña creados anteriormente?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
¿Pude crear un producto correctamente?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
¿Pude visualizar el producto recién creado?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
¿Pude editar el producto creado anteriormente?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
¿Pude crear un mantenimiento sobre el producto recién creado?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
¿Pude utilizar los filtros de mantenimientos con efectividad?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
¿Pude visualizar el mantenimiento recién creado?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

¿Pude editar el mantenimiento creado anteriormente?

¿Pude realizar la búsqueda de un técnico mediante su nombre?

¿Obtuve mensajes sobre los procesos que estaba realizando?

Enviar

Borrar formulario