



**BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA**

---

---

**FACULTAD DE CIENCIAS DE LA COMPUTACIÓN**

**“DESARROLLO DE UNA APLICACIÓN MÓVIL PARA EL  
REPORTE DE INCIDENCIAS EN LA FACULTAD DE  
CIENCIAS DE LA COMPUTACIÓN”**

**TESINA**

**PARA OBTENER EL GRADO DE:**

**LICENCIATURA EN INGENIERÍA EN CIENCIAS DE LA  
COMPUTACIÓN**

**PRESENTA:**

**LEONARDO ROMERO TOXQUI**

**DIRECTORA DE TESINA:**

**DRA. GUILLERMINA SÁNCHEZ ROMÁN**

**JUNIO DEL 2023**

## **Agradecimientos**

En primer lugar, quiero expresar mi más sincero agradecimiento a mis padres por su apoyo incondicional, paciencia, sacrificios y comprensión a lo largo de toda mi carrera y, en general, por todo lo que me han dado durante toda mi vida. Su constante respaldo ha sido fundamental para mi crecimiento y éxito.

También deseo extender mi agradecimiento a los profesores del Diplomado de Titulación, quienes hicieron posible que concluyera este proceso. En particular, quiero agradecer de manera especial a la Dra. Guillermina Sánchez Román, cuya guía, apoyo y dedicación han sido invaluable a lo largo de todo este trayecto. Sus amplios conocimientos y orientación han sido fundamentales para el desarrollo de mi tesina.

A mi amada esposa, Jacqueline, quiero expresar mi profundo agradecimiento por su apoyo, amor, compañía y paciencia durante estos últimos 9 años a mi lado. Gracias por alegrar mi vida en innumerables ocasiones. También quiero agradecerle por permitirme formar una hermosa familia junto a ti y nuestras dos maravillosas hijas.

A mis adorables hijas, Ariadne y Grecia, aunque quizás aún no sean conscientes de ello, quiero decirles que son y siempre serán lo más importante en mi vida. Son el mejor regalo que Dios me ha concedido, mi mayor tesoro y la fuente más pura de inspiración y amor. Estoy completamente agradecido con ustedes por darle sentido a mi vida, por motivarme cada día con sus sonrisas y permitirme ser un mejor padre para ustedes.

Por último, quiero agradecer a mis amigos: Beto, Erick, Lalo, Toño, Wuicho y Yael, por brindarme su amistad durante toda nuestra carrera universitaria. Desde que empezamos juntos hasta el final, su compañerismo ha hecho que esta etapa sea mucho más llevadera. Agradezco sinceramente por estar siempre ahí.

## Contenido

<b>CAPÍTULO I: INTRODUCCIÓN .....</b>	<b>4</b>
1.1 RESUMEN.....	4
1.2 PLANTEAMIENTO DEL PROBLEMA.....	5
1.3 JUSTIFICACIÓN .....	6
1.4 OBJETIVO GENERAL .....	6
1.5 OBJETIVOS ESPECÍFICOS.....	6
<b>CAPÍTULO II: ESTADO DEL ARTE .....</b>	<b>8</b>
2.1 APLICACIONES SIMILARES.....	9
<b>CAPÍTULO III: MARCO TEÓRICO .....</b>	<b>12</b>
3.1 METODOLOGÍA.....	12
3.2 APLICACIONES MÓVILES.....	13
3.3 BASES DE DATOS .....	15
3.4 PROTECCIÓN DE LOS DATOS.....	19
3.5 PRUEBAS DE SOFTWARE.....	20
<b>CAPÍTULO IV: DISEÑO Y DESARROLLO DEL SISTEMA .....</b>	<b>24</b>
4.1 ANÁLISIS DE REQUERIMIENTOS DEL SISTEMA .....	25
4.2 SPRINTS DEL PROYECTO .....	42
4.3 DISEÑO DEL SISTEMA .....	44
4.4 DESARROLLO DEL SISTEMA .....	54
<b>CAPÍTULO V: CONTROL DE RIESGOS Y SEGURIDAD .....</b>	<b>64</b>
5.1 ANÁLISIS PARA LA PROTECCIÓN DE DATOS DE LA APLICACIÓN .....	64
5.2 VULNERABILIDADES Y PLAN DE MITIGACIÓN DE RIESGOS.....	67
5.3 AVISO DE PRIVACIDAD .....	71
<b>CAPÍTULO VI: PRUEBAS Y RESULTADOS .....</b>	<b>74</b>
6.1 PLAN DE PRUEBAS .....	74
<b>CAPÍTULO VII: CONCLUSIONES Y TRABAJO A FUTURO .....</b>	<b>84</b>
7.1 CONCLUSIONES.....	84
7.2 TRABAJO A FUTURO .....	85
<b>BIBLIOGRAFÍA .....</b>	<b>87</b>

# CAPÍTULO I: INTRODUCCIÓN

## 1.1 Resumen

En el año 2021 en México, según datos del Instituto Nacional de Estadística y Geografía (*INEGI*), usuarios que contaban con un teléfono inteligente instalaron más de 43 millones de aplicaciones móviles, solamente en el estado de Puebla fueron instaladas más de 2 millones de aplicaciones. (*INEGI*, 2021)

Sumado a esto, cifras también proporcionadas por *INEGI*, la cantidad de usuarios en internet era de 84.1 millones en 2020 y se tenía un registro de 88.2 millones de usuarios que contaban con un teléfono, de los cuales, un 91.8% era un teléfono inteligente. (*INEGI*, 2021)

Tomando en cuenta estos datos, se puede observar un incremento y un uso constante de los teléfonos inteligentes por parte de la población en México, se podría decir que el uso de la tecnología forma parte del día a día de las personas cada vez con más frecuencia. Esto supone una gran brecha de oportunidades para atender problemáticas haciendo uso de las aplicaciones móviles.

En la Facultad de Ciencias de la Computación no existe un registro formal de reportes, pero, se pueden enviar solicitudes de atención a distintas problemáticas mediante el correo de la coordinación administrativa, solicitudes por escrito, mediante redes sociales o acudiendo de manera personal a la secretaría.

Observando esta problemática, lo que se pretende realizar con el proyecto es tener una aplicación móvil funcional en la cual alumnos o docentes puedan hacer reportes de problemáticas o incidencias en la Facultad de Ciencias de la Computación, por ejemplo, mobiliario dañado, árboles, lámparas u objetos pesados que puedan estar en peligro de caerse, equipos de cómputo, proyectores que no funcionen, etc. Con el objetivo de asignar tareas a áreas responsables para dar solución a las problemáticas en el menor tiempo posible.

También, funcionará para el reporte de objetos extraviados durante el día a día en la facultad y tendrá un concentrado de estos y un estatus de si ha sido devuelto y a quien fue entregado o el estatus de si aún no ha sido entregado.

## **1.2 Planteamiento del problema**

Uno de los problemas principales que afectan a la educación está relacionado con la infraestructura educativa. Sólo en CDMX, la *CNDH* intervino y detallo que entre los años 2012 y 2016 recibieron quejas de malas condiciones de los inmuebles, por lo cual visitó 29 planteles educativos y encontró varios daños en bardas, pisos, drenajes, así como equipos obsoletos y mobiliario viejo. Posterior a su visita, un año más tarde, visitó nuevamente las instituciones educativas y constató que 8 escuelas hicieron caso omiso a dichos reportes, 13 atendieron parcialmente los daños y sólo 7 solucionaron todos los inconvenientes. (El Universal, 2017)

Este seguimiento sólo se pudo dar porque se llevó a cabo una investigación y se redactó una bitácora, donde se registraron las distintas problemáticas. Pero ¿qué hubiera pasado si no se registraban los informes completos y eran otros los encargados de volver a realizar la inspección en los planteles? Tal vez no se hubieran percatado de cuales instituciones atendieron o no a los problemas reportados.

Actualmente, en la Facultad de Ciencias de la Computación el proceso de reportar alguna situación que pueda poner en riesgo a la comunidad estudiantil, para informar sobre inmobiliario dañado o para realizar reportes de objetos perdidos se lleva a cabo mediante distintos medios y es posible que no se atiendan o se den seguimiento a algunos, sin embargo, eso requiere de tiempo humano lo cual puede utilizarse en otros procesos administrativos.

Por tal motivo, se propone una solución a esta problemática, con un sistema totalmente funcional que pueda brindar un seguimiento eficaz y transparente a la comunidad universitaria de la facultad.

### **1.3 Justificación**

El objetivo principal es contar con una aplicación móvil que pueda funcionar como punto de partida para el cotejo de reportes, así como tener un listado actualizado, que esté al alcance de cualquier miembro de la comunidad en tiempo real para poder dar un seguimiento efectivo y transparente que logre conseguir que los encargados reduzcan el tiempo de atención a distintas problemáticas que pueda llegar a haber en la Facultad de Ciencias de la Computación.

Si se logra implementar correctamente, se tendrán dos grandes beneficios, tanto para administrativos como estudiantes. En un lado, podría permitir a personal del área administrativa estar informados de todos los problemas que ocurren dentro de la facultad, generar reportes de cuantos problemas se han solucionado, cuántos están en proceso de atención y cuales faltan por canalizar con el área encargada para su debida corrección; además permitiría reducir tiempos de solución a las problemáticas identificadas.

Por otra parte, los estudiantes podrían sentirse escuchados y saber que sus reportes son tomados en cuenta de manera efectiva, además que el agregar un apartado de objetos perdidos, logrará que los estudiantes fomenten los valores de honestidad y cooperación permitiendo tener mejores relaciones interpersonales y tratar de lograr un ambiente sano donde estudiar.

### **1.4 Objetivo general**

Desarrollar una aplicación móvil para reportar incidencias u objetos perdidos en la Facultad de Ciencias de la Computación de la BUAP para dar un seguimiento y atención más rápida a estas problemáticas.

### **1.5 Objetivos específicos**

- Crear una aplicación móvil capaz de tomar evidencias fotográficas de los reportes para que un administrador pueda darle un seguimiento a dichos incidentes.

- Reducir tiempos de solución a las distintas problemáticas que vayan siendo reportadas por los usuarios de la aplicación.
- Implementar una interfaz sencilla y amigable al usuario para una fácil interacción con la aplicación.
- Intercambiar información a través de reportes entre los usuarios que hayan perdido o encontrado objetos dentro de la facultad de computación.

## **CAPÍTULO II: ESTADO DEL ARTE**

En este capítulo, se abordan los antecedentes de las aplicaciones móviles haciendo especial énfasis en dos casos reales de aplicaciones de carácter ciudadano en el ámbito gubernamental y servirán de modelo para lo que se pretende desarrollar.

En la aplicación de soluciones de tecnología, en específico, las aplicaciones móviles y haciendo un énfasis en las aplicaciones gubernamentales, surgió el concepto de “*Gobierno Digital o electrónico*” que incluye el uso de herramientas tecnológicas para hacer más eficaz la gestión pública y mejorar una comunicación u ofrecer un mejor servicio entre el gobierno y sus ciudadanos. (Secretaría de la Función Pública, 2013)

En enero de 2004, durante el “*Foro de las Américas de Mejores Prácticas: Gobierno Abierto*”, algunos representantes de Brasil, Canadá y Chile expusieron sus estrategias y resultados que obtuvieron al implementar con éxito un gobierno digital en sus países. (Secretaría de la Función Pública, 2013)

Desde entonces distintos gobiernos dentro de la República Mexicana, han implementado y logrado duplicar un “Gobierno Digital” para sus ciudadanos; casos en distintos municipios de estados como Jalisco, CDMX, Hidalgo, Guanajuato, etc. (LUCERO, 2022) Han sido exitosos y cada vez más y más municipios han apostado por volverse parcial o totalmente un “Gobierno electrónico”.

Al revisar algunas aplicaciones de gobierno podemos rescatar que estas aplicaciones suelen tener una interfaz intuitiva y fácil de usar, lo que hace que sea más sencillo para los ciudadanos informar sobre problemas. Algunas aplicaciones incluso permiten a los usuarios adjuntar imágenes o videos para proporcionar una descripción más detallada del problema.

También es importante tener en cuenta que el éxito de estas aplicaciones depende de la capacidad de las autoridades para responder a los reportes de manera oportuna y efectiva. Además de que es importante asegurarse de que los reportes sean verificados y validados antes de tomar medidas, para evitar la difusión de información errónea o falsa.

## 2.1 Aplicaciones similares

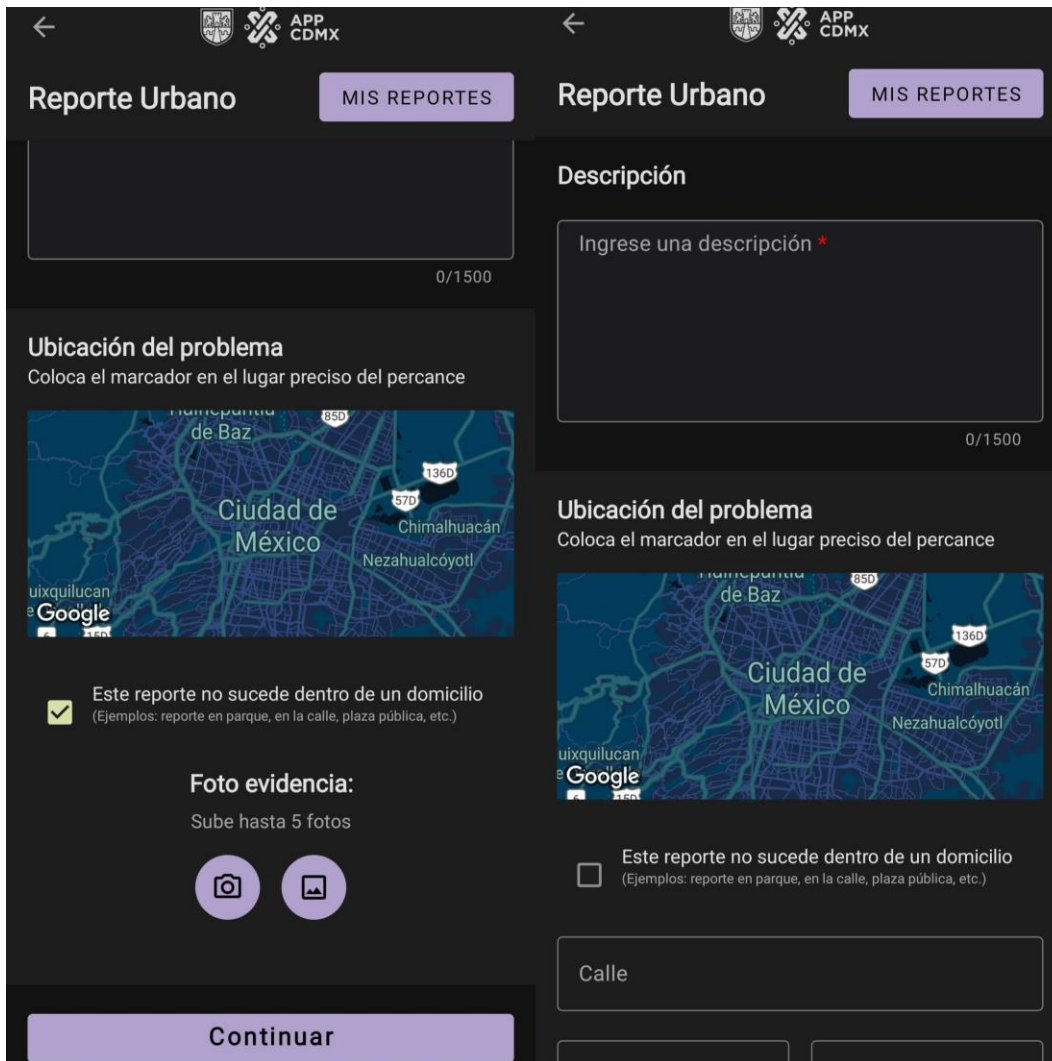
Enfocándonos en casos específicos de aplicaciones móviles que cuenten con un apartado de reportes, podemos encontrar los siguientes ejemplos.

“SACH Móvil” (Aplicación Móvil Ciudadana de San Andrés Cholula, Puebla) podemos encontrar un apartado de reportes ciudadanos, como se observa en la *Ilustración 1*, en el cual, los usuarios, pueden enviar un formulario donde se mencione el problema con evidencia fotográfica y una ubicación precisa de donde se encuentra dicha situación, para que la dependencia encargada del área en que se encuentra el reporte pueda darle solución de manera más rápida.

The image displays the 'Nuevo Reporte' (New Report) screen of the 'SACH Móvil' application. The screen is divided into two main sections. On the left is a form with the following fields: a dropdown menu for 'Área del reporte: \*', another dropdown for 'Servicio: \*', a text input for 'Ingresar una descripción \*', a text input for 'Correo \*', and a text input for 'Escribir tu teléfono'. Below the form is a camera icon and four buttons: 'TOMAR FOTO', 'SELECCIONAR FOTO', 'CANCELAR', and 'ACEPTAR'. On the right side, there is a map interface. A red location pin is placed on the map, and a text box above it says 'Mueve para indicar la ubicación del incidente'. The map shows a grid of streets with labels such as 'C. 17 Ote.', 'Av. 13 Ote.', 'Calle 2 Sur', 'C. 19 Ote.', 'C. 4 Sur', 'Av. 15 Ote.', 'C. 6 Sur', 'Av. Reforma', 'C. 23 Ote.', 'Av. 25 Ote.', 'C. 27 Ote.', 'C. 8 Sur', 'C. 7 Pte.', 'Calle 5 P.', 'C. 9 Pte.', 'C. 5 Sur', 'C. 11 Pte.', and 'C. 13 Pte.'.

*Ilustración 1 – Captura de Pantalla de “SACH Móvil”. Romero, L. (2023)*

“App CDMX” (Aplicación Móvil Ciudadana de CDMX, México) esta aplicación a diferencia de la anterior, en su apartado de reportes, tal y como se muestra en la *Ilustración 2*, los segmenta de dos formas, reportes que han ocurrido dentro de un domicilio y reportes que no han ocurrido dentro del domicilio cada caso con sus respectivos formularios para poder recolectar los datos necesarios para dar un seguimiento correcto del reporte ciudadano.



***Ilustración 2 - Capturas de Pantalla de "App CDMX". Romero, L. (2023)***

Partiendo de estos casos específicos, se puede adaptar e implementar en la Facultad de Ciencias de la Computación, como un sistema de reportes y su seguimiento para la comunidad universitaria, que puede ayudar tanto a administrativos como estudiantes a conocer problemas que podrían incluso poner

en riesgo su salud y quien haya echo el reporte darle la oportunidad de conocer el estatus de su reporte con mayor facilidad desde su dispositivo móvil sin requerir ir a verificar personalmente si se ha dado solución a su reporte.

Para concluir, a raíz de esta investigación, se pretende que con la aplicación propuesta pueda ser beneficiada la facultad con las siguientes razones:

- Reducir tiempos de solución a problemas, que si no hubiesen sido atendidos de manera rápida, podrían haber causado daños más costosos o irreparables.
- Poder obtener un listado de reportes que han sido enviados y poder conocer en que estatus se encuentra cada uno.
- Mejorar una comunicación entre la comunidad estudiantil y los administrativos de la facultad e inclusive se podría generar en un futuro dentro de la aplicación un buzón de sugerencias para aumentar el diálogo y la participación de los alumnos.
- Concientizar a directivos que puede haber cosas por reparar, sustituir, colocar en la facultad y que tal vez ellos no hayan destinado dar una atención en corto plazo, pero, para los alumnos puede ser necesidades que realmente requieran solución.
- Crear un ambiente de estudio sano y cooperativo para los estudiantes y que se sientan seguros e involucrados con hacer una mejor facultad para los futuros estudiantes.

## CAPÍTULO III: MARCO TEÓRICO

En este capítulo se sustenta la manera del cómo se desarrollará la aplicación móvil, desde la metodología elegida, *Scrum*; los fundamentos de las aplicaciones móviles y las bases de datos, los sustentos de la protección de la información mediante cifrado de datos así como la definición de las herramientas que se aplicarán a lo largo del desarrollo del proyecto.

### 3.1 Metodología

La metodología *Scrum*, forma parte de las principales metodologías ágiles está basada en la teoría de control de procesos, emplea un proceso iterativo e incremental, esta metodología es ideal para entregas de un producto funcional en corto plazo, además de que controla el riesgo de fallos en los entregables. (Jiménez, 2015)

*Scrum* está conformado por estos elementos importantes que componen cada iteración:

- **Sprint Planning Meeting:** Son reuniones con el cliente y el equipo de trabajo donde se define el proyecto, las etapas y los plazos.
- **Product Backlog:** Es una lista de funcionalidades que el cliente requiera para su producto, ordenadas de mayor a menor importancia.
- **Sprint Backlog:** Es la lista de tareas que se definió para realizar durante la iteración. Se divide en lo que hay por hacer, lo que se está haciendo y lo que se realizó.
- **Daily Scrum:** Son reuniones cortas y diarias con el equipo de trabajo donde se plantean las preguntas ¿Qué hice ayer?, ¿Qué haré hoy?, ¿Qué ayuda necesito?
- **Sprint Review;** Una vez finalizado el sprint se realiza una reunión para revisar todos los avances del proyecto, también se debe de tener en cuenta, que a este paso debe de existir un proyecto tangible para poder ser presentado al cliente.

- **Sprint Retrospective.** Es el último paso en el que se revisa lo que se hizo en el proyecto, lo que se hizo bien y los errores que se tuvieron. (Maldonado, 2018)

Además de cada elemento mencionado anteriormente, *Scrum* cuenta con varios roles a lo que se le conoce como equipo de desarrollo, estos interactúan entre ellos para llevar a cabo correctamente el desarrollo del proyecto para llegar a un buen resultado, respetando tiempos y lineamientos. El equipo está conformado por:

- **Scrum Master:** Es el líder del equipo, es el que ayuda al equipo y a los clientes a entender el funcionamiento de los procesos, además se encarga de que todo el equipo adopte correctamente la metodología *Scrum*.
- **Product Owner:** Es la persona que transmite las necesidades que requiere para su producto.
- **Team Members:** Son los responsables del desarrollo y la entrega del proyecto.
- **Users:** No forman parte del desarrollo, pero, son a quienes va dirigido el producto final a desarrollar, los que interactuarán una vez que el producto sea entregado.
- **Stakeholders:** Al igual que los usuarios no forman parte del desarrollo, pero son usuarios que están implicados indirectamente en el producto final, tales como directores, gerentes etc. (Jiménez, 2015)

## 3.2 Aplicaciones móviles

Una aplicación móvil, es un programa que puede ser ejecutado en teléfonos inteligentes, tabletas o cualquier otro dispositivo móvil; dependiendo de las necesidades de los desarrolladores pueden optar por una aplicación móvil nativa o híbrida. (Soucoup & Buck, 2022)

Estas poseen distintas características, pero, para este proyecto, sólo se hará enfoque en las aplicaciones híbridas.

### 3.2.1 Aplicaciones móviles híbridas

A diferencia de las aplicaciones nativas, estas son multiplataforma, es decir, un mismo código puede ser ejecutado en distintos sistemas, sin necesidad de hacer demasiadas modificaciones. Pueden ser desarrolladas por distintos *frameworks* y lenguajes de programación. Algunos ejemplos de tecnologías de desarrollo son *React*, *Flutter*, *Ionic*, *Xamarin*, etc.

Ventajas de las aplicaciones híbridas:

- Menor tiempo de desarrollo.
- Código base único para las distintas plataformas.
- Mayor facilidad de actualización. (Soucoup & Buck, 2022)

Desventajas de las aplicaciones híbridas:

- Las características de cada plataforma tienen características y funcionalidades distintas.
- Limitaciones al querer utilizar o interactuar con el hardware.
- Desempeño menor en relación con una aplicación nativa. (Soucoup & Buck, 2022)

### 3.2.2 Ionic

*Ionic* es un *framework* de código abierto que proporciona elementos de interfaz de usuario para desarrollar aplicaciones móviles y aplicaciones de escritorio utilizando tecnologías web, además de integrar algunos *frameworks* más populares de desarrollo web como *Angular*, *React* y *Vue*. (Ionic Team, 2014)

*Ionic* se enfoca principalmente en el desarrollo de la interfaz gráfica del usuario (UI). *Ionic* provee componentes, interacciones, gestos, animaciones, etc. Que son fáciles de aprender y pueden ser incluidas a través de *frameworks* de *TypeScript* e inclusive pueden ser implementados sin necesidad de alguno de estos usando solamente *JavaScript*. (Ionic Team, 2014)

Algunas de las ventajas de utilizar *Ionic* son:

- Multiplataforma, un solo código base para múltiples plataformas nativas *iOS*, *Android* o *PWA* (Aplicación Web Progresiva).

- Diseño limpio, simple y funcional, *Ionic* está diseñado para mostrar diseños propios nativos (por ejemplo, *material design* para Android).
- Simplicidad, *Ionic* es fácil de aprender y accesible para cualquiera que tenga experiencia con desarrollo web.
- Compatibilidad con *frameworks* basados en *JavaScript*. (Ionic Team, 2014)

### 3.3 Bases de datos

Las bases de datos son una colección de grandes cantidades de información, para poder ser manipulados, se necesita un *DBMS* (Sistema Manejador de Base de Datos) que se encarga de manejar, recuperar y almacenar la información. (Jatana, Puri, Ahuja, Kathuria, & Gosain, 2012)

Una base de datos debe también cumplir con algunas propiedades para ser considerada como tal, éstas son referidas como *ACID* (Atomicity, Consistency, Isolation, Durability)

- **Atomicity (Atomicidad):** Representa todo o nada, es decir si una parte de la transacción se queda incompleta, está se considera como fallida.
- **Consistency (Consistencia):** Asegura que antes y después de una transacción todos los datos son válidos.
- **Isolation (Aislamiento):** Asegura que se puedan ejecutar múltiples transacciones al mismo tiempo y no se afectan una a la otra mientras se ejecutan.
- **Durability (Durabilidad):** Es cuando una transacción se ha finalizado correctamente (*commit*) esta permanecerá en el mismo estado, es decir, que quedará guardada aún si el sistema fallara inmediatamente que termino la transacción. (Jatana, Puri, Ahuja, Kathuria, & Gosain, 2012)

La principal función de una base de datos es almacenar la información que se usará para y dentro de una aplicación o sistema, una vez guardados los datos podemos consultarlos de una manera rápida y ordenada.

#### 3.3.1 Tipos de Bases de Datos

Podemos considerar en esta investigación dos tipos de bases de datos las bases de datos relacionales y las no relacionales.

Cuando tenemos una colección de datos organizados con la ayuda de tablas se considera una base de datos relacional. Por este motivo es una opción para los modelos de bases de datos en forma jerárquica o de red. En cambio, si la base de datos está diseñada específicamente para datos específicos y tienen esquemas flexibles se conoce como base de datos no relacional. (Rendón, 2019)

Para determinar cuándo usar bases de datos relacionales y cuando no relacionales hay que tener en consideración nuestros requerimientos. Una guía para saber cuándo usar una u otra podría ser la siguiente.

Bases de datos relacionales:

- El crecimiento de los datos es nulo o es relativamente poco.
- Las consultas pueden ser satisfechas por un solo servidor.
- El uso por parte de los usuarios no es muy exigido. (Rendón, 2019)

Bases de datos no relacionales:

- Los datos crecen rápidamente en algunos momentos.
- Las necesidades de proceso cambian y no se pueden prevenir.
- El uso es muy exigido por momentos. (Rendón, 2019)

Para este proyecto la escalabilidad, integración con otros servicios y la reducción de tiempos de desarrollo y porque la herramienta a utilizar está basado en *NoSQL* se optó por utilizar una Base de datos no relacional, aunque también puede ser implementado en un gestor de bases de datos relacional, pero la prioridad de este proyecto a corto plazo es gestionar los tiempos de desarrollo y el utilizar *Firebase* se puede solventar la implementación de la base de datos para ser consumido por cualquier dispositivo móvil además que también provee un servicio de *notificaciones push* y puede ser usado para atender el requerimiento de avisar a los usuarios mediante este tipo de notificaciones.

### **3.3.1.1 Bases de Datos No Relacionales**

Son una clase de sistema para administrar las bases de datos, su principal diferencia con las bases de datos relacionales es que no utiliza tablas definidas. Tampoco utiliza *SQL* para realizar consultas, por lo cual, no se garantizan las propiedades *ACID*. A pesar de todo esto, el uso de estas bases de datos es muy amplio, muchas aplicaciones actualmente utilizan este tipo de base de datos. (Jatana, Puri, Ahuja, Kathuria, & Gosain, 2012)

Este tipo de bases de datos también posee una clasificación dependiendo el modo en que almacenan la información. Algunos tipos de bases de datos no relacionales son los siguientes.

- **Clave-Valor:** Es un par de datos que consta de una clave para poder acceder y el valor que es el dato que contiene la información. Con esto se elimina la necesidad de crear un modelo de datos.
- **Documentos:** Este modelo, como su nombre lo indica, está basado en documentos de un formato estándar como lo son *XML*, *BSON*, *PDF* o *Microsoft Office*. Para acceder a un documento necesitamos una ruta a la cual acceder. Para una rápida consulta de información se hace uso de una API o de un lenguaje de consulta definido.
- **Grafos:** Está basado en la estructura de datos, utiliza nodos y aristas. Los nodos representan la información mientras que las aristas relacionan mediante un nodo con otro mediante alguna propiedad definida. Para poder acceder a los datos se utiliza la teoría de grafos. (Jatana, Puri, Ahuja, Kathuria, & Gosain, 2012)

Gracias a las bases de datos no relacionales, proporciona una gran solución a la información que se encuentra en la Web, por medio de *NoSQL*, maneja de una manera óptima el crecimiento exponencial de la información y su gran rapidez para hacer operaciones de consulta. (Jatana, Puri, Ahuja, Kathuria, & Gosain, 2012)

Algunas de las ventajas de las bases de datos no relacionales son:

- Alto rendimiento de datos.
- Gran escalabilidad información.

- Los datos pueden ser modificados y guardados en cualquier momento y sin que exista algún problema.
- Se pueden ejecutar muchas consultas a la vez, puesto que, es transaccional. (Jatana, Puri, Ahuja, Kathuria, & Gosain, 2012)

También hay que considerar las siguientes desventajas:

- La búsqueda en este tipo de bases de datos puede ser ineficiente cuando está basado en múltiples criterios.
- No es muy consistente.
- Pueden existir datos duplicados (redundancia). (Jatana, Puri, Ahuja, Kathuria, & Gosain, 2012)

### **3.3.1.2 Cloud Firestore**

*Cloud Firestore* es una base de datos *NoSQL* en la nube. Está desarrollada por Google y que forma parte de la plataforma de desarrollo *Firebase*.

Al ser una base de datos no relacional, almacena la información mediante documentos, que a su vez son agrupados en colecciones, que son usados para organizar los datos y compilar consultas. El modelado de datos no es estricto y permite que se utilices cualquier estructura de datos que mejor se adapte a las necesidades del proyecto.

Una de las ventajas principales es que se pueden realizar consultas a nivel de documento sin necesidad de recuperar la información de toda una colección ni de subcolecciones anidadas.

Implementar *Cloud Firestore* es rápido y no es necesario integrarla a un servidor propio para poder utilizarlo ni hay necesidad de hacer muchas configuraciones desde consola de comandos pues provee un panel administrativo para realizar las configuraciones de seguridad como reglas de acceso, aplicaciones confiables, etc.

*Cloud Firestore* provee servicios de seguridad integrados como autenticación, reglas de seguridad de lectura y escritura, encriptación de la información en reposo y tránsito. (Google Developers, s.f.)

## 3.4 Protección de los datos

La manera más común de protección de los datos es el cifrado. El cifrado de los datos es la forma de convertir un texto plano a un texto ilegible sino se cuenta con la clave para descifrarlo. (IBM, s.f.)

El principal objetivo del cifrado es la protección de los datos, para que delincuentes cibernéticos no puedan tener acceso a la información privada de una aplicación, a esto se le conoce como cifrado en reposo. Además, es útil también en el transporte de datos, puesto que, la información puede ser monitoreada, a este tipo de cifrado se le conoce como cifrado en tránsito. (IBM, s.f.)

Para proteger los datos se usarán dos maneras, el cifrado asimétrico para el cifrado en tránsito y para proteger datos personales del usuario; mientras que para el caso de las contraseñas se utilizará funciones hash (*SHA-3 512*) para encriptarlas y no puedan ser legibles, además que dificultarán los ataques a contraseñas de ciber delincuentes.

### 3.4.1 Cifrado asimétrico

Existen dos tipos de cifrados asimétricos y simétricos, para este proyecto sólo se trabajará con un cifrado asimétrico.

Este tipo de cifrado necesita de una clave secreta para cifrar y descifrar la información. (IBM, s.f.)

Algunos tipos de cifrado asimétrico son:

- **AES (Advanced Encryption Standard):** Es el tipo de cifrado más común y a menudo se le considera como el criterio de referencia para el cifrado de datos.
- **DES (Data Encryption Standard):** Es un algoritmo de cifrado de bloques de bajo nivel que utiliza bloques de texto de 64 bits y los convierte en texto cifrado de 48 bits.
- **Triple DES:** Ejecuta el cifrado DES tres veces diferentes, cifra, descifra y vuelve a cifrar los datos nuevamente. (IBM, s.f.)

### 3.4.2 Funciones hash criptográficas

A este tipo de funciones se le provee de un texto plano que proporciona una salida conocida como “*message digest*” estos tienen una longitud de 160 a 512 bits, mientras mayor sea el tamaño y el algoritmo ocupado (*SHA-2* o *SHA-3*) más seguro y menor riesgo de colisión de hash existirá. Una ventaja de este cifrado es que es rápido a nivel computacional y no requiere de muchos recursos para realizarlo. (w3c, 2016)

Para cifrar las contraseñas se hará uso del algoritmo *SHA-3* de longitud de 512 bits.

Un ejemplo de las diferentes longitudes del cifrado entre *SHA* de 384 y *SHA* de 512 son los siguientes:

sha384-

dq2/FHsEhznzjeLeF0YKb1A1cvakYyEFm0jh89uTOTZf16X8p34jik+Lyv4Hk8vHd

sha512-

Q2bLCN8hOms2FKTDLy7eA0E3WoNr3bjZSAHFTOhEALk1T8eOsHrbV1Fu9/zCin  
RE7Af1ofPrugP2zFZvX42Fpw==

### 3.5 Pruebas de software

Las pruebas de software son la manera de evaluar y verificar que una aplicación de software realiza correctamente lo que debe hacer. (IBM, s.f.)

Realizar pruebas antes de entregar el producto al usuario final genera beneficios como:

- Prevenir errores de forma temprana.
- Reducir costos de desarrollo.
- Mejorar el rendimiento o usabilidad del software.
- Mejor satisfacción del o de los clientes. (IBM, s.f.)

Existen distintos tipos de pruebas, cada uno con un fin en específico. Se mencionarán 2 tipos de pruebas que evaluarán el funcionamiento de la aplicación y detectarán posibles errores; también se utilizarán las *heurísticas de Jakob Nielsen*,

a diferencia de las primeras pruebas, el propósito principal se enfoca en la interfaz de usuario.

### **3.5.1 Pruebas por exploración**

Este tipo de pruebas están basadas en historias de usuario que siguen una secuencia definida que conlleva a una situación de éxito y permite una documentación a medida que se va realizando la prueba. (Parmar, s.f.)

Para generar los casos de prueba que se llevarán a cabo, es indispensable contar con historias de usuario. La ventaja de este tipo de pruebas es que al ser aleatorias y no estructuradas pueden ayudar a encontrar errores que no son hallados en tipos de pruebas estructuradas. (Parmar, s.f.)

### **3.5.2 Pruebas por conjetura de fallos**

En este tipo de prueba se lleva a cabo mediante la experiencia del programador, lo que permite que se tenga identificado por parte de este que es lo que podría fallar en la aplicación. (S'Arreplec, s.f.)

Al tratarse de un tipo de prueba que busca encontrar errores que podrían pasar desapercibidos por un usuario común, depende demasiado de la experiencia e intuición del que realice las pruebas; por ejemplo usar un valor de entrada 0 para datos matemáticos que en algún momento de la ejecución incluyan divisiones. (S'Arreplec, s.f.)

### **3.5.3 Pruebas por heurísticas**

A diferencia de las pruebas anteriores este tipo de pruebas está enfocada a la usabilidad del sistema, puesto que, permite medir la calidad de la interfaz de la aplicación a través de 10 principios propuestos y definidos por “**Jakob Nielsen**”. (Fang, 2018)

Los 10 principios generales son los siguientes:

- 1. Visibilidad del estado del sistema:** Se debe mantener informado en todo momento a los usuarios sobre el estado actual del sistema y de lo que está sucediendo, mediante diálogos cortos en periodos no tan largos de tiempo.

2. **Relación entre el sistema y el mundo real:** El diseño debe de poder ser entendido en lenguaje propio de los usuarios (no usar tecnicismos ni palabras propias de un grupo u organización). Se deben de usar palabras, frases, iconos que sean reconocidos por los usuarios fácilmente.
3. **Control y libertad para el usuario:** Ofrecer al usuario una “salida de emergencia” en caso de realizar una acción no deseada o evitarla antes de que suceda.
4. **Consistencia y estándares:** No confundir a los usuarios sobre si se encuentran en lo mismo (distintos colores o distintos diseños en un mismo sitio). Además de utilizar los estándares definidos por la plataforma (web, Android, iOS, etc.) o la industria.
5. **Prevención de errores:** Evitar mediante diseño que puedan ocurrir errores. Eliminar las situaciones propensas a errores o mostrar mensajes de confirmación a los usuarios antes de realizar una acción.
6. **Minimizar la carga de memoria del usuario:** No hacer que el usuario tenga que memorizar toda la información de una interfaz, debe ser visible en todo momento o fácil de encontrar.
7. **Flexibilidad y eficiencia de uso:** El diseño debe de ser útil para usuarios inexpertos y ofrecer accesos directos para usuarios expertos. Algunos ejemplos de accesos directos pueden ser navegación por teclado o gestos en teléfonos móviles.
8. **Diseño estético y minimalista:** La información en pantalla no debe de ser irrelevante o que no sea útil, o inclusive, mostrar elementos adicionales que compitan con el objetivo principal de lo que se desea mostrar en pantalla.

**9. Ayuda a los usuarios con los errores:** Los mensajes de error deben de ser explicados con lenguaje fácil de entender (sin códigos de error) y que mencione al usuario cual ha sido el error de forma exacta, además de sugerirle una posible solución.

**10. Ayuda y Documentación:** Lo ideal en una aplicación es que no necesite una explicación de cómo funciona, pero, de no ser posible, ofrecerle al usuario una documentación que le ayude a comprender el funcionamiento del sistema. Esta ayuda debe de ser fácil de buscar así como concisa y que guie al usuario paso a paso. (Nielsen, 2020)

## CAPÍTULO IV: DISEÑO Y DESARROLLO DEL SISTEMA

Una vez definidos los objetivos generales y específicos del proyecto se iniciará la etapa de diseño y desarrollo del sistema. Para este proyecto se optó por seguir con la metodología *Scrum*, que facilitará un prototipo “funcional” desde el segundo sprint (serán mencionados más adelante, una vez que sean definidos los requerimientos del sistema); además que se adapta perfectamente con las necesidades del desarrollo de la aplicación.

En este capítulo se dividirá en dos etapas principales, análisis y diseño del sistema y desarrollo del sistema. En la primera etapa, se llevará a cabo el análisis de requerimientos funcionales y no funcionales; esto nos permitirá obtener una visión más completa de lo que se espera realice la aplicación. Posteriormente, se elaborará la especificación de casos de uso que será el punto de partida para la elaboración de los diagramas de clase y secuencia y los modelos *ER (Entidad-Relación)*, relacional y no relacional que funcionarán como un ejemplo gráfico para el modelado de la base de datos que utilizará la aplicación móvil. Por último, con toda la información obtenida en el análisis se podrá realizar el diseño de mockups de las pantallas para su continuación en la etapa de desarrollo del software.

En la etapa de desarrollo, donde el objetivo final será obtener una aplicación funcional, tomando como punto de partida el modelado obtenido del diseño del sistema, se hará uso de las siguientes herramientas tecnológicas, *frameworks*, etc.

La tecnología que será usada para el desarrollo será *Ionic*, un *framework* de desarrollo de aplicaciones móviles que utiliza tecnologías web (*HTML*, *CSS* y *JS*) y utiliza en conjunto el *framework* de *TypeScript*, *Angular*.

Con respecto a la base de datos, se optó por una del tipo no relacional y dicho gestor de base de datos *NoSQL* será *Cloud Firestore*, un servicio de *Firebase*. El gestor no utiliza tablas, sino colecciones de documentos para almacenar la información.

Existieron diversos motivos del por qué elegir *Firestore* que utiliza un modelo “no relacional” en lugar de utilizar un gestor de base de datos “relacional”, pero, algunos de los más significativos son mencionados a continuación.

- **Escalabilidad:** A lo largo del proyecto o en un trabajo a futuro se pueden ir añadiendo o eliminando campos en los documentos (tablas) según vayan surgiendo necesidades sin necesidad de volver a realizar un análisis exhaustivo y sin que se vea comprometida información almacenada previamente.
- **Integración con servicios de Firebase:** El sistema necesitará hacer uso de *notificaciones push*, que también pueden ser proveídas por *Firebase*, y el tener integrado el servicio de base de datos con *Cloud Firestore* permite una mejor integración más fácil, además que de ser necesario pueden ser implementados otros servicios de *Firebase* como “*Authentication*”, “*Cloud Storage*”, etc.
- **Tiempo de desarrollo:** El utilizar *Cloud Firestore* reduce significativamente los tiempos de desarrollo, dado que nos provee de una api totalmente funcional para obtener los datos almacenados sin necesidad de montar un servidor externo y configurarlo para que la aplicación pueda ser alimentada. (Google Developers, s.f.)

## 4.1 Análisis de requerimientos del sistema

En la siguiente sección, se presentarán los detalles del diseño del sistema, incluyendo las funcionalidades principales y la interfaz de usuario. Con ello, se espera ofrecer una solución efectiva y práctica para la gestión de incidencias y objetos perdidos.

Para llevar a cabo esta tarea, se ha realizado un análisis de las necesidades y requerimientos, así como de las características y funcionalidades que debe tener la aplicación para garantizar una experiencia de uso satisfactoria y eficiente, para ello, se hizo un listado con algunos requerimientos funcionales y no funcionales, organizados de la siguiente forma.

### 4.1.1 Requerimientos funcionales del sistema

NoReq.	Tipo de requerimiento	Nombre del requerimiento	Descripción
<b>RF1</b>	Requerimiento funcional de sistema	Iniciar Sesión	El sistema contará con un servicio de inicio de sesión con una capa de seguridad incluida.
<b>RF2</b>	Requerimiento funcional de sistema	Reportar objeto perdido	El sistema debe de contar con una sección de objetos perdidos para que el alumno pueda reportarlos. Una vez enviados, el sistema envía una notificación al resto de usuarios que tengan la aplicación instalada.
<b>RF3</b>	Requerimiento funcional de sistema	Reportar Incidencia	El sistema debe de poder generar reportes a partir de la solicitud del alumno con evidencia fotográfica.
<b>RF4</b>	Requerimiento funcional de usuario	Consultar listado de reporte de incidencias	El usuario con rol de administrador puede consultar el listado en tiempo real de los reportes que han sido enviados y registrados, para poder canalizar con el área correspondiente.
<b>RF5</b>	Requerimiento funcional de usuario	Consultar listado de objetos perdidos	Los alumnos pueden consultar que objetos perdidos han sido reportados y pueda ver detalles de cada uno de ellos sí así lo requiere.
<b>RF6</b>	Requerimiento funcional de usuario	Consultar listado de reportes de incidencias propios	El alumno puede consultar los reportes de incidencia que ha enviado y ver información detallada de cada uno de ellos o para conocer en que estatus se encuentran.
<b>RF7</b>	Requerimiento funcional de usuario	Contactar usuario	El alumno, al identificar un objeto perdido suyo, pueda contactar al usuario que lo reportó mediante los datos de contacto que han sido proporcionados al momento del registro.
<b>RF8</b>	Requerimiento funcional de usuario	Entregar objeto perdido	El alumno que encontró al propietario de un objeto perdido y que se lo ha devuelto, puede cambiar el estatus de objeto a “entregado” llenando un

			formulario con los datos del alumno al que se le fue entregado.
--	--	--	---

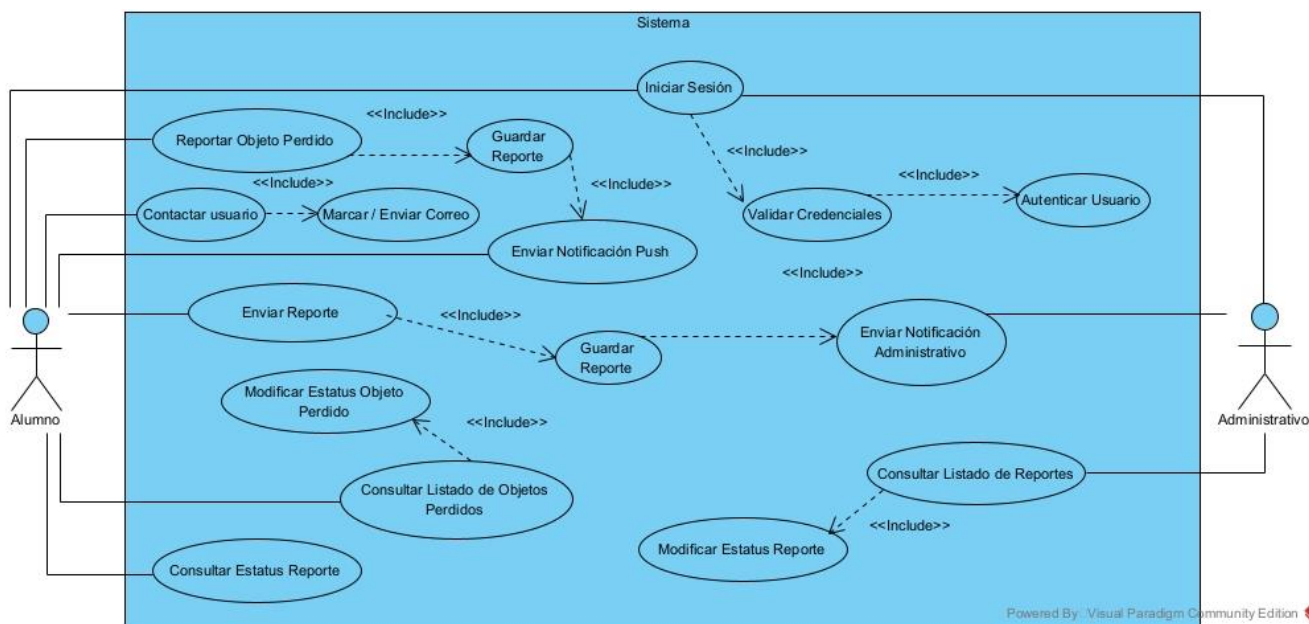
**Tabla 1 - Listado de requerimientos funcionales**

#### 4.1.2 Requerimientos no funcionales del Sistema

NoReq.	Tipo de Requerimiento	Descripción
<b>RNF1</b>	Requerimiento no funcional de Soporte	La aplicación estará disponible 24/7, exceptuando causas ajenas provocadas por el proveedor de servicios en la nube.
<b>RNF2</b>	Requerimiento no funcional de privacidad	La aplicación cumplirá con las leyes de privacidad del usuario y no compartirá sus datos con otros usuarios a menos que el usuario lo permita.
<b>RNF3</b>	Requerimiento no funcional de usabilidad	La interfaz tendrá un diseño responsivo para un correcto funcionamiento en cualquier dispositivo móvil (Teléfonos inteligentes y Tabletas).
<b>RNF4</b>	Requerimiento no funcional de usabilidad	La interfaz tendrá un diseño intuitivo y que sea fácil de usar, siguiendo las pautas de la experiencia de usuario.
<b>RNF5</b>	Requerimiento no funcional de fiabilidad	La aplicación tendrá respuesta o solución a la mayor cantidad de fallos o errores posibles que pudieran llegar a presentarse.
<b>RNF6</b>	Requerimiento no funcional de rendimiento	Lograr que la aplicación pueda atender múltiples llamadas en simultáneo y cubra la demanda de recursos por parte de los usuarios sin que haya fallas de comunicación o desempeño.

**Tabla 2 - Listado de requerimientos no funcionales**

### 4.1.3 Casos de uso del Sistema



**Ilustración 3 - Diagrama general de casos de uso, elaboración propia**

En el anterior diagrama de casos de uso, *Ilustración 3*, se muestra de manera general los actores que intervendrán en el sistema, alumno y administrativo.

El alumno, así como el administrativo, podrán iniciar sesión mediante sus propias credenciales de usuario, obtenidas previamente al momento de registrarse en la aplicación, una vez validadas y, si la información es correcta, se autenticará al usuario y se le permitirá el acceso al sistema.

El alumno, podrá reportar un objeto perdido mediante un formulario y, si la información está completa, se enviará la información para posteriormente almacenarla en la base de datos y, finalmente, se le enviará una *notificación push* al resto de usuarios que tengan la aplicación instalada y que hayan dado su consentimiento de recibir notificaciones; además que podrá ver el listado de todos los reportes de objetos perdidos enviados, y una vez que haya devuelto un objeto perdido, podrá cambiar el estatus de dicho reporte.

Si el alumno ha identificado un objeto perdido, podrá contactar al alumno que hizo el reporte mediante correo electrónico o llamada telefónica.

También, el alumno, podrá realizar reportes de incidencia, de manera similar al de un reporte de objeto perdido, y consultar solamente sus propios reportes para conocer el estatus en que cada uno se encuentra, solucionado o no solucionado.

El administrativo, no tendrá permisos para enviar reportes de ningún tipo, sólo podrá consultar los reportes de incidencia que hayan sido enviados y vincular cada reporte al área encargada de darle un pronto seguimiento, una vez solucionado, cada reporte podrá marcarse como reporte solucionado. De igual manera, el administrativo, podrá visualizar el listado de objetos perdidos enviados por los alumnos.

#### 4.1.3.1 Especificaciones de casos de uso

Caso de Uso #1	Iniciar Sesión
<b>Requerimiento relacionado</b>	RF1
<b>Objetivo</b>	Permitir al usuario iniciar sesión en el sistema
<b>Actores principales</b>	Usuario (alumno o administrativo)
<b>Actores secundarios</b>	Base de datos
<b>Precondiciones</b>	El usuario debe de estar registrado en el sistema
<b>Final Exitoso</b>	Autenticación exitosa y acceso a apartados de la app
<b>Final Fallido</b>	Error al autenticar y no permite inicio de sesión.
<b>Evento de Inicio</b>	El usuario desea acceder a la aplicación
<b>Flujo Principal</b>	<ol style="list-style-type: none"> <li>1. El usuario proporciona su usuario y contraseña.</li> <li>2. El sistema valida las credenciales proporcionadas.</li> <li>3. Si las credenciales son válidas, la capa de seguridad permite el acceso y la sesión se inicia.</li> </ol>
<b>Postcondiciones</b>	El usuario ha iniciado sesión en el sistema.
<b>Flujo Alternativo</b>	El usuario proporciona un nombre de usuario o contraseña inválido. La capa de seguridad deniega el acceso y muestra un mensaje de error.

**Tabla 3 - Caso de uso de iniciar sesión**

<b>Caso de Uso #2</b>		<b>Reportar Objeto Perdido</b>
<b>Requerimiento relacionado</b>	RF2	
<b>Objetivo</b>	Permitir al usuario enviar un reporte de objeto perdido.	
<b>Actores principales</b>	Alumno que realiza el reporte	
<b>Actores secundarios</b>	Base de datos, Resto de usuarios	
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• El alumno que reporta debe de estar registrado y haber iniciado sesión en la aplicación.</li> <li>• El alumno que reporta debe de disponer con una conexión a internet para que se pueda enviar el reporte correctamente.</li> </ul>	
<b>Final Exitoso</b>	El reporte es procesado correctamente por el sistema y notifica a todos los usuarios que se ha hecho un nuevo reporte de objeto perdido.	
<b>Final Fallido</b>	El reporte no pudo ser enviado debido a problemas de red o el servidor de base de datos no pudo completar la solicitud.	
<b>Evento de Inicio</b>	El usuario requiere enviar un reporte de objeto perdido,	
<b>Flujo Principal</b>	<ol style="list-style-type: none"> <li>1. El alumno selecciona la opción de "Reportar objeto perdido" en la aplicación.</li> <li>2. El alumno ingresa la información relevante del objeto perdido (descripción, lugar, fecha, etc.).</li> <li>3. El alumno puede incluir evidencia fotográfica.</li> <li>4. El alumno envía el reporte y es procesado por el sistema.</li> <li>5. El sistema registra el reporte en la base de datos.</li> <li>6. El sistema envía <i>notificaciones push</i> a todos los usuarios informándoles del nuevo reporte.</li> </ol>	
<b>Postcondiciones</b>	El objeto perdido se encuentra reportado y todos los usuarios han sido notificados.	
<b>Flujo Alternativo</b>	<ul style="list-style-type: none"> <li>• El alumno no cuenta con una conexión de red. El reporte no puede ser enviado y el sistema muestra un mensaje de error.</li> <li>• El gestor de base de datos no puede almacenar el reporte y el sistema muestra un mensaje con el error propio de la base de datos.</li> </ul>	

**Tabla 4 - Caso de uso de reportar objeto perdido**

<b>Caso de Uso #3</b>	<b>Reportar incidencia</b>
<b>Requerimiento relacionado</b>	RF3
<b>Objetivo</b>	Que el alumno pueda realizar un reporte para que los encargados de resolver la incidencia puedan darle atención.
<b>Actores principales</b>	Alumno
<b>Actores secundarios</b>	Administrativo, Base de datos
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• El alumno debe estar registrado y haber iniciado sesión en la aplicación.</li> <li>• El alumno debe de tener una conexión a internet.</li> </ul>
<b>Final Exitoso</b>	El reporte ha sido enviado correctamente y se ha notificado al usuario con rol administrativo o el encargado de dar seguimiento al reporte.
<b>Final Fallido</b>	El reporte no pudo ser enviado debido a problemas de red o el servidor de base de datos no pudo completar la solicitud.
<b>Evento de Inicio</b>	El alumno encontró un problema y desea reportarlo en la sección de reportes de incidencias.
<b>Flujo Principal</b>	<ol style="list-style-type: none"> <li>1. El alumno accede a la opción de "Reportar incidencia" en la aplicación.</li> <li>2. El alumno ingresa los detalles de la incidencia, incluyendo descripción y evidencia.</li> <li>3. El alumno envía el formulario del reporte.</li> <li>4. El sistema valida los datos ingresados y registra el reporte en la base de datos.</li> <li>5. El sistema envía una notificación a los usuarios con rol administrativo.</li> </ol>
<b>Postcondiciones</b>	La incidencia se ha reportado y notificado a los usuarios con rol administrativo.
<b>Flujo Alternativo</b>	<ul style="list-style-type: none"> <li>• El alumno no cuenta con una conexión de red. El reporte no puede ser enviado y el sistema muestra un mensaje de error.</li> <li>• El gestor de base de datos no puede almacenar el reporte y el sistema muestra un mensaje con el error propio de la base de datos.</li> </ul>

**Tabla 5 - Caso de uso de reportar incidencia**

<b>Caso de Uso #4</b>	<b>Consulta de reportes de incidencias</b>
<b>Requerimiento relacionado</b>	RF4
<b>Objetivo</b>	Permite al administrador consultar los reportes que han sido enviados de parte de los alumnos.
<b>Actores principales</b>	Administrador
<b>Actores secundarios</b>	Base de datos
<b>Precondiciones</b>	El administrador debe estar registrado y haber iniciado sesión previamente.
<b>Final Exitoso</b>	La aplicación muestra el listado de todos los reportes que han sido enviados hasta el momento de la consulta.
<b>Final Fallido</b>	El listado no puede ser mostrado por problemas de conexión o aún no hay ningún reporte en cola de espera.
<b>Evento de Inicio</b>	El administrador elige la opción de "consulta de reportes"
<b>Flujo Principal</b>	<ol style="list-style-type: none"> <li>1. El usuario accede a la opción de "Consultar reportes en tiempo real".</li> <li>2. El sistema solicita la información de reportes de incidencias a la base de datos y le es regresada por esta.</li> <li>3. El sistema muestra una lista de todos los reportes realizados con su estatus actual.</li> <li>4. El usuario puede seleccionar un reporte para ver más detalles.</li> </ol>
<b>Postcondiciones</b>	El usuario puede ver el seguimiento en tiempo real de los reportes
<b>Flujo Alternativo</b>	Si no hay reportes en espera, el sistema informa al usuario que no hay ningún reporte disponible en ese momento.

**Tabla 6 - Caso de uso de consulta de reportes de incidencia**

<b>Caso de Uso #5</b>	<b>Consulta del listado de objetos perdidos</b>
<b>Requerimiento relacionado</b>	RF5
<b>Objetivo</b>	Permitir al alumno consultar reportes de objetos perdidos que han sido enviados por el resto de los alumnos.
<b>Actores principales</b>	Alumno
<b>Actores secundarios</b>	Base de datos
<b>Precondiciones</b>	El alumno debe estar registrado y haber iniciado sesión en la aplicación.
<b>Final Exitoso</b>	La aplicación despliega el listado de objetos perdidos que han sido reportados.
<b>Final Fallido</b>	El listado no puede ser mostrado por problemas de conexión o aún no han sido reportados objetos perdidos.
<b>Evento de Inicio</b>	El alumno elige la opción "Consulta de objetos perdidos"
<b>Flujo Principal</b>	<ol style="list-style-type: none"> <li>1. El alumno selecciona el apartado de "Consulta de objetos perdidos"</li> <li>2. El sistema le solicita la información de objetos perdidos a la base de datos y ésta regresa la información al sistema.</li> <li>3. El sistema muestra al alumno el listado de objetos perdidos.</li> <li>4. El usuario puede seleccionar un reporte para ver más detalles.</li> </ol>
<b>Postcondiciones</b>	El usuario ha sido informado del listado de los reportes de objetos perdidos.
<b>Flujo Alternativo</b>	Aún no hay objetos reportados y no puede mostrar una información detallada, entonces el sistema muestra un mensaje con esta información.

**Tabla 7 - Caso de uso de consulta del listado de objetos perdidos**

<b>Caso de Uso #6</b>		<b>Consulta del estatus de reportes enviados</b>	
<b>Requerimiento relacionado</b>	RF6		
<b>Objetivo</b>	Permitir a los alumnos consultar el estatus de los reportes que hayan enviado previamente.		
<b>Actores principales</b>	Alumno		
<b>Actores secundarios</b>	Base de datos		
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• El alumno debe estar registrado y haber iniciado sesión en la aplicación.</li> <li>• El alumno debe haber enviado al menos un reporte previamente.</li> </ul>		
<b>Final Exitoso</b>	La aplicación muestra los reportes que ha enviado el alumno con su respectiva información adjunta.		
<b>Final Fallido</b>	El listado no puede ser mostrado ya sea por error de conexión o porque el alumno no ha hecho ningún reporte.		
<b>Evento de Inicio</b>	El alumno elige la opción "Consultar reportes"		
<b>Flujo Principal</b>	<ol style="list-style-type: none"> <li>1. El alumno selecciona la opción "Consultar reportes" en la aplicación.</li> <li>2. El sistema solicita la información correspondiente a la base de datos y recibe una respuesta.</li> <li>3. El sistema muestra una lista de los reportes previamente realizados por parte del alumno.</li> <li>4. El usuario puede seleccionar un reporte específico para ver su estatus.</li> <li>5. El sistema muestra información detallada del reporte, incluyendo su estado y cualquier actualización por parte del encargado de la atención a estos problemas.</li> </ol>		
<b>Postcondiciones</b>	El alumno ha sido informado del estatus de cada uno de sus reportes que ha enviado.		
<b>Flujo Alternativo</b>	Aún no hay reportes enviados y el sistema no puede mostrar una información detallada.		

**Tabla 8 - Caso de uso de consulta del estatus de reportes enviados**

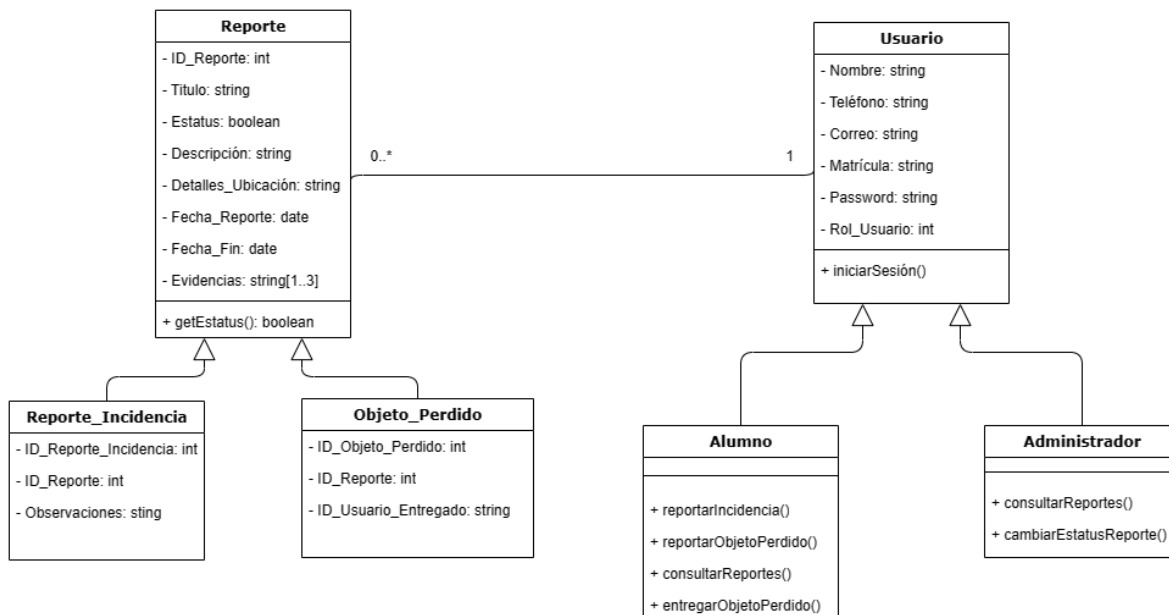
<b>Caso de Uso #7</b>		<b>Contactar a usuario que reportó objeto perdido</b>
<b>Requerimiento relacionado</b>	RF7	
<b>Objetivo</b>	El alumno puede contactar mediante la aplicación al usuario que reportó un objeto perdido que él considera que es de su propiedad.	
<b>Actores principales</b>	Alumno	
<b>Actores secundarios</b>	Base de datos, Alumno que realizó reporte.	
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• El alumno debe estar registrado y haber iniciado sesión en la aplicación.</li> <li>• El alumno debe de haber consultado la sección de objetos perdidos e identificado un objeto perdido que considere sea de su propiedad</li> </ul>	
<b>Final Exitoso</b>	El alumno se comunica exitosamente con el alumno que realizó el reporte.	
<b>Final Fallido</b>	El alumno no puede comunicarse con el alumno que realizó el reporte por motivos externos.	
<b>Evento de Inicio</b>	El alumno elige la opción “Contactar usuario”	
<b>Flujo Principal</b>	<ol style="list-style-type: none"> <li>1. El alumno accede a la sección de Objetos perdidos.</li> <li>2. El alumno detecta su objeto que ha extraviado y lo selecciona para ver detalles del objeto perdido.</li> <li>3. El sistema muestra los detalles del objeto perdido, incluyendo los datos de contacto del usuario que lo reportó.</li> <li>4. El alumno puede elegir llamar al usuario que lo reportó o enviar un correo utilizando las funciones de la aplicación para abrir llamadas o el proveedor de correo que utilice el alumno en su dispositivo móvil.</li> <li>5. El sistema realiza la acción elegida por el alumno (llamar o enviar correo electrónico)</li> </ol>	
<b>Postcondiciones</b>	El alumno ha establecido contacto con el usuario que reportó el objeto perdido y puede discutir la posible recuperación del objeto.	
<b>Flujo Alternativo</b>	<ul style="list-style-type: none"> <li>• El alumno no concedió el permiso del sistema para realizar llamadas y no puede ejecutarse dicho proceso.</li> <li>• Si no encuentra el objeto perdido que desea, el alumno procederá a cerrar la aplicación o realizar otra actividad dentro de la aplicación.</li> </ul>	

**Tabla 9 - Caso de uso de contactar a usuario**

<b>Caso de Uso #8</b>	<b>Entregar objeto perdido</b>
<b>Requerimiento relacionado</b>	RF8
<b>Objetivo</b>	El alumno entrega al usuario su objeto perdido y actualiza el estatus del objeto perdido por "entregado"
<b>Actores principales</b>	Alumno
<b>Actores secundarios</b>	Base de datos, Alumno agraviado.
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• Ambos alumnos deben estar registrado en el sistema.</li> <li>• El alumno que hizo el reporte debe haber iniciado sesión en la aplicación.</li> </ul>
<b>Final Exitoso</b>	El alumno ha devuelto ha encontrado al alumno afectado y le ha devuelto el objeto perdido.
<b>Final Fallido</b>	El alumno agraviado no está registrado en el sistema.
<b>Evento de Inicio</b>	El alumno elige la opción "Objeto devuelto"
<b>Flujo Principal</b>	<ol style="list-style-type: none"> <li>1. El alumno accede al sistema y selecciona la opción de "Objeto Devuelto".</li> <li>2. El sistema solicita que el alumno proporcione la matrícula del alumno al que se le ha sido devuelto el objeto perdido.</li> <li>3. El alumno proporciona la información requerida.</li> <li>4. El sistema verifica que el alumno al que se le ha entregado el objeto, este registrado en el sistema.</li> <li>5. Si la información es válida, el sistema cambia el estatus del objeto perdido a "entregado" además mostrará información adicional sobre el alumno al que le fue entregado.</li> <li>6. El sistema muestra un mensaje de confirmación indicando que el estatus del objeto perdido ha sido cambiado exitosamente.</li> </ol>
<b>Postcondiciones</b>	El estatus del objeto perdido ha sido actualizado a "entregado" en el sistema.
<b>Flujo Alternativo</b>	Si la información proporcionada por el alumno es inválida, el sistema muestra un mensaje de error y regresa al paso 2.

**Tabla 10 - Caso de uso de entregar objeto perdido**

#### 4.1.4 Diagrama de Clases



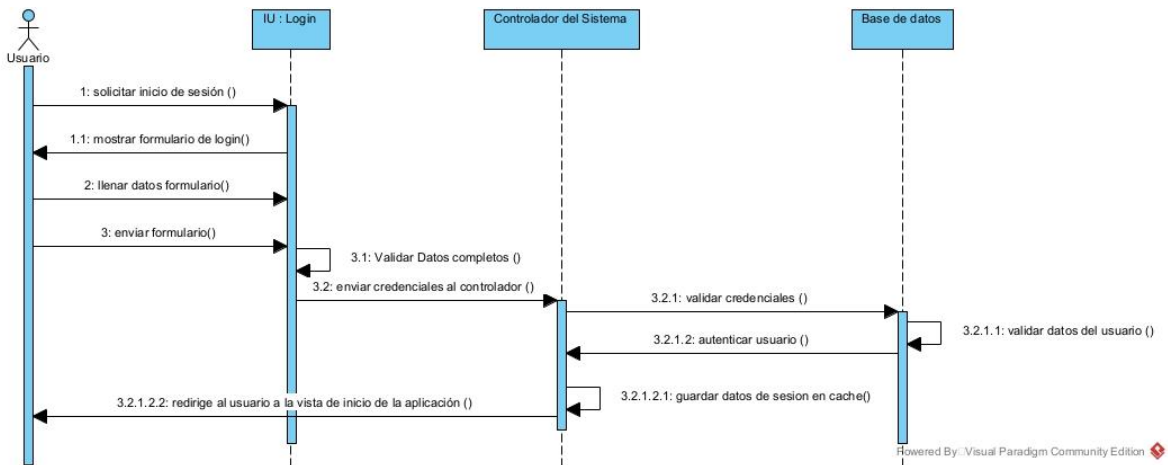
**Ilustración 4 - Diagrama de Clases, elaboración propia**

El realizar el diagrama de clases, *Ilustración 4*, permite visualizar la estructura del sistema y el cómo cada clase interactuará con la aplicación. En este caso existen 2 *clases padre* y 4 *clases hijas*. La clase **Reporte** hereda sus atributos y único método a **Reporte\_Incidencia** y **Objeto\_Perdido**. De igual manera la clase **Usuario** tiene dos clases hijas **Alumno** y **Administrador**.

#### 4.1.5 Diagramas de Secuencia

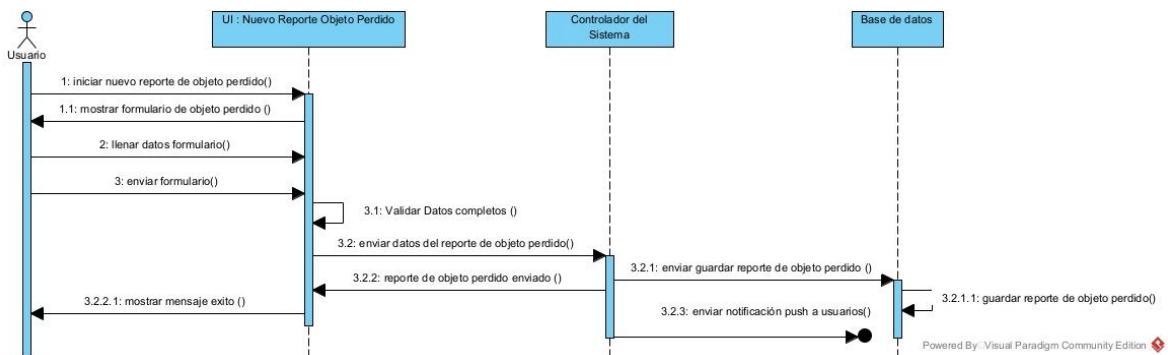
Los siguientes diagramas, indican los pasos que sigue el sistema para completar cada uno de los procesos. A continuación se describirá el flujo que realiza cada uno de ellos.

En la *Ilustración 5*, indica el proceso de inicio de sesión. El usuario se tiene que dirigir a la pantalla de inicio de sesión, posteriormente, el sistema le muestra el formulario correspondiente y le solicita las credenciales de acceso al usuario, una vez ingresadas por parte de este, el sistema valida que sean datos con un formato válido y, de ser así, las envía al gestor de base de datos con la finalidad de encontrar un usuario que coincida con dichas credenciales, y si la información es correcta, autentica al usuario y le permite entrar a la pantalla de home.



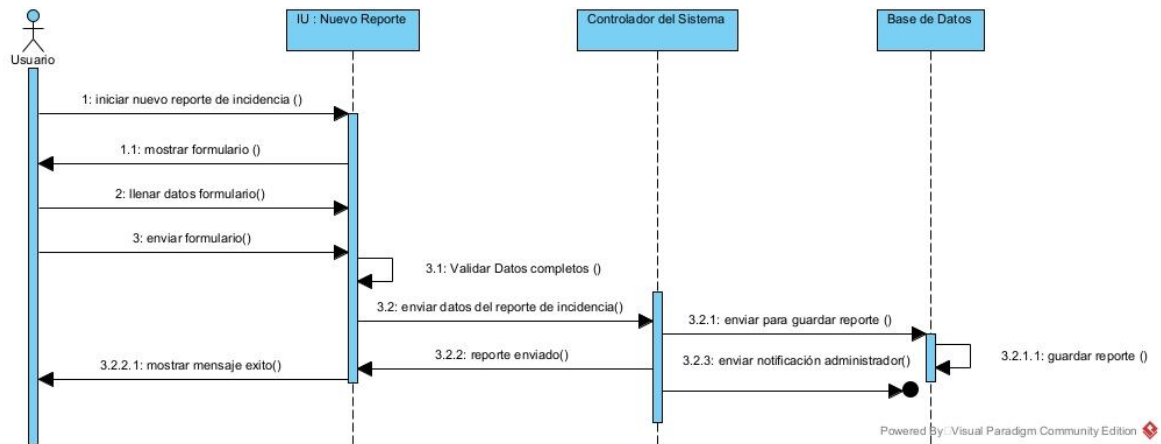
**Ilustración 5 - Diagrama de secuencia de inicio de sesión, elaboración propia**

Para el proceso de envío de reporte de objeto perdido, *Ilustración 6*, El usuario debe dirigirse a la pantalla de nuevo reporte de objeto perdido, dentro, la aplicación le mostrará el formulario con los datos que requiere para el reporte, el usuario ingresa dichos datos y, una vez completos, debe enviarlos, los datos son validados por la aplicación y, de ser correctos, se enviarán a la base de datos para ser almacenados. Si el envío del reporte ha sido correcto, le muestra un mensaje de éxito al usuario que ha enviado el reporte y finalmente el sistema envía una *notificación push* al resto de usuarios.



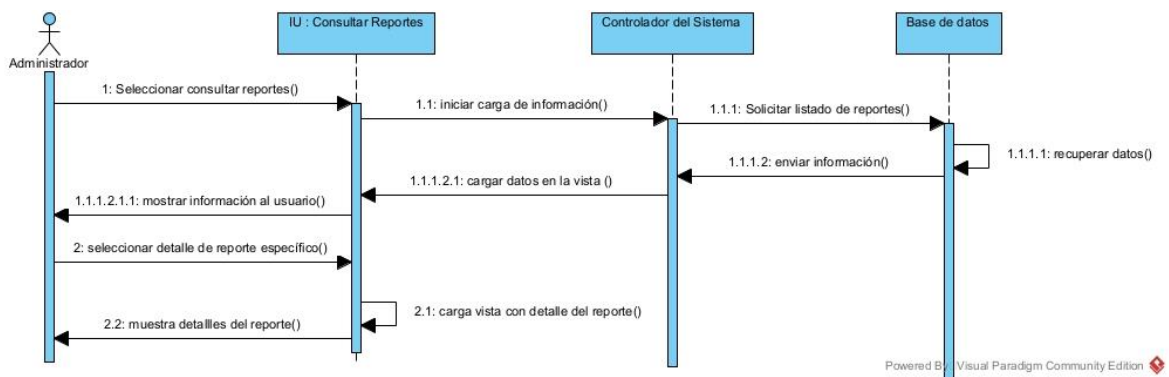
**Ilustración 6 - Diagrama de secuencia de envío de reporte de objeto perdido, elaboración propia**

En el diagrama de secuencia de reporte de incidencias, *Ilustración 7*, el proceso es similar al anterior proceso de, envío de reporte de objeto perdido, con la única diferencia de que al realizar un envío satisfactorio, sólo le muestra un mensaje de éxito al usuario que envió el reporte y no envía ninguna *notificación push*.



**Ilustración 7 - Diagrama de secuencia de envío de reporte de incidencia, elaboración propia**

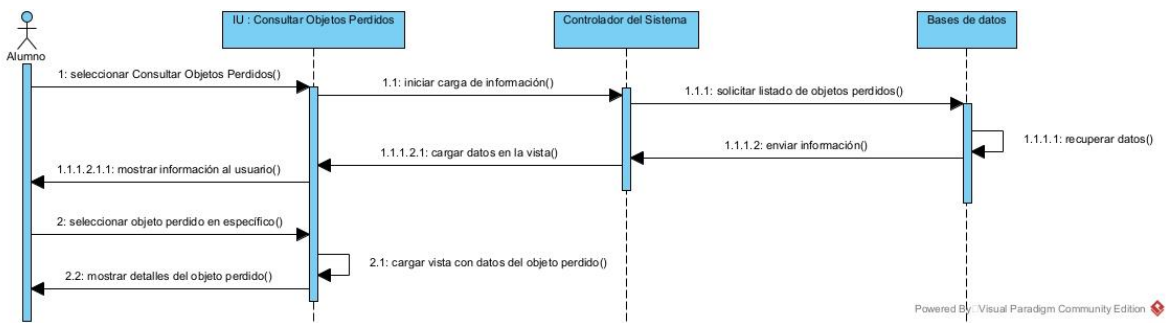
Para realizar una consulta de listado de reportes, realizado por el administrador, se ve reflejado en la *Ilustración 8*. El administrador debe dirigirse a la pantalla de listado de reporte de incidencia, una vez ahí, la aplicación le solicita a *Cloud Firestore* los reportes de incidencia enviados por los alumnos, el gestor procesa la solicitud y le regresa los datos a la aplicación para que pueda mostrar la información al usuario. Si el administrador requiere ver un reporte en específico, lo selecciona, y la aplicación le muestra la información a detalle de dicho reporte.



**Ilustración 8 - Diagrama de secuencia de consulta del listado de reportes de incidencia, elaboración propia**

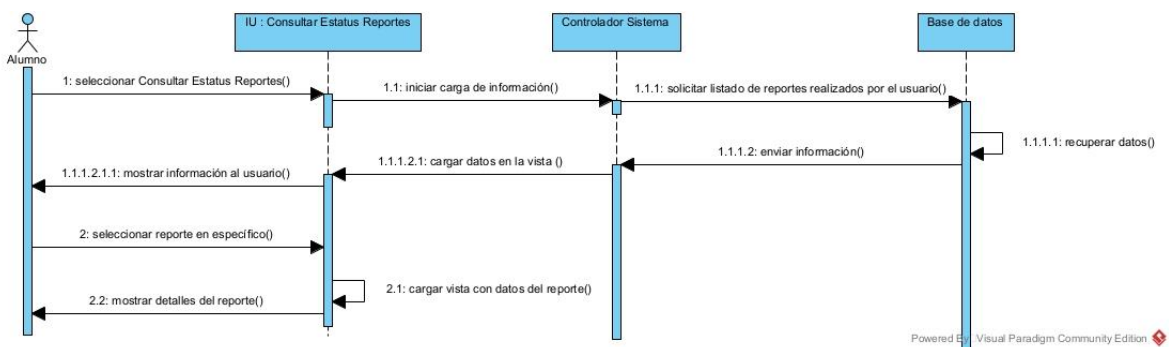
En la *Ilustración 9*, se muestra el proceso de consulta del listado de objetos perdidos, para ello, el alumno debe dirigirse al apartado del listado de objetos perdidos y, una vez allí, la aplicación se encargará de solicitar los datos al gestor de base de datos y una vez obtenida la información, mostrarle todos los reportes que

han sido enviados. Si el alumno desea ver un reporte en específico debe de seleccionarlo y la aplicación le mostrará una vista detallada de dicho reporte.



**Ilustración 9 - Diagrama de secuencia de consulta del listado de objetos perdidos, elaboración propia**

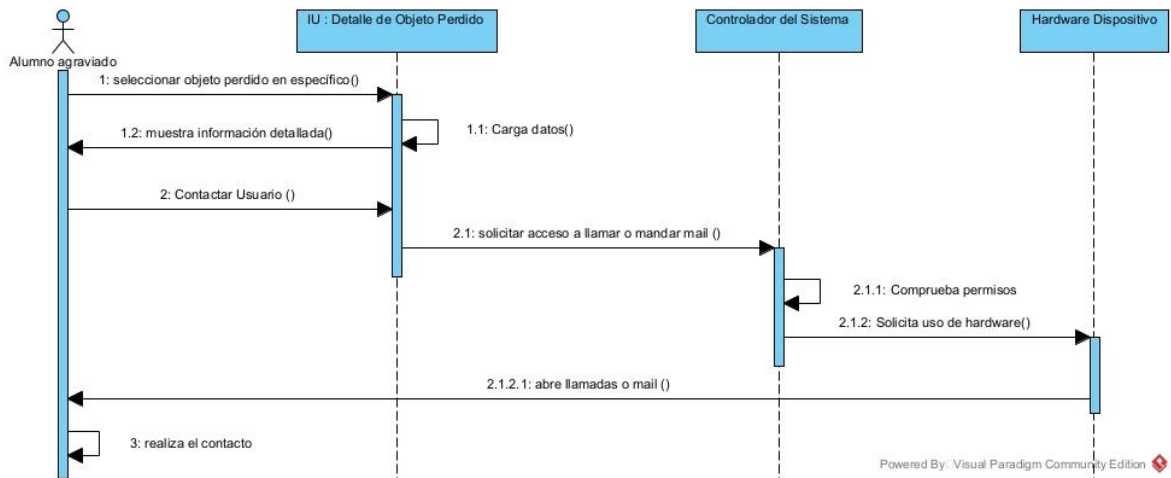
El proceso mostrado en la *Ilustración 10*, es la consulta de estatus de reportes enviados. El alumno debe dirigirse al listado de reportes de incidencias, dicho listado sólo corresponde a reportes que haya enviado, la aplicación solicitará la información almacenada en la base de datos y, en caso de contar con reportes enviado, le mostrará un listado, al seleccionar un reporte en específico podrá ver en que estatus se encuentra (solucionado o en proceso de solución). En caso de no contar con ningún reporte, la aplicación le mostrará un mensaje informándole que aún no ha enviado ninguno.



**Ilustración 10 - Diagrama de secuencia de consulta de estatus de reportes enviados, elaboración propia**

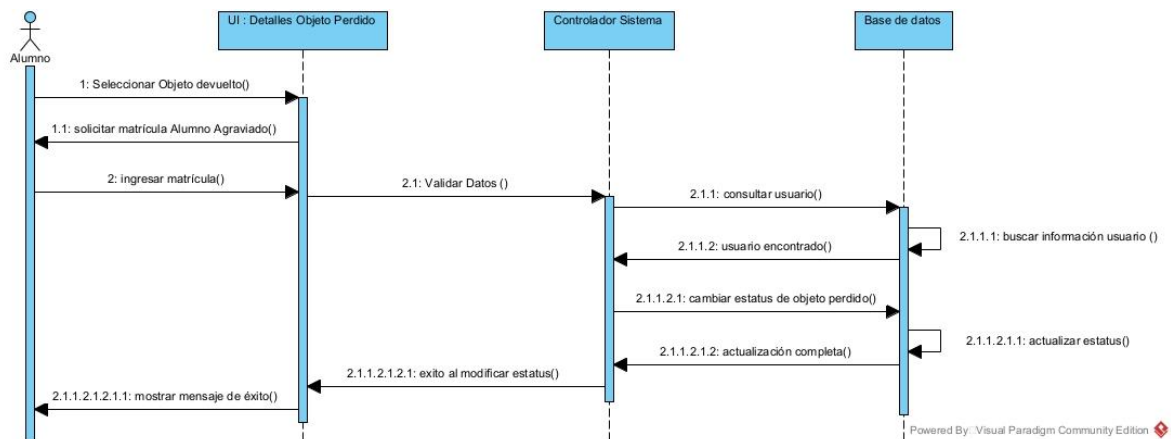
En el siguiente diagrama, *Ilustración 11*, se observan los pasos a seguir en caso de que un alumno identifique un objeto perdido suyo y desee contactar al usuario que reportó dicho objeto. Dentro de la sección de listado de objetos perdidos y,

seleccionado el reporte en específico, podrá encontrar la manera de como contactar a quien hizo el reporte, correo electrónico o llamada telefónica; una vez que el alumno haya determinado como contactarlo toca la opción deseada y se comunica con él.



**Ilustración 11 - Diagrama de secuencia de contactar a usuario que reportó objeto perdido, elaboración propia**

Finalmente, en la *Ilustración 12*, el alumno que ha hecho un reporte de objeto perdido y lo ha devuelto a su propietario, debe de dirigirse a la información del reporte que envió y, posteriormente, capturar su matrícula, si es correcta y validada por el sistema, se modifica el estatus del reporte como objeto perdido devuelto y le muestra un mensaje de éxito al usuario.



**Ilustración 12 - Diagrama de secuencia de devolver objeto perdido, elaboración propia**

## 4.2 Sprints del proyecto

Finalizada la etapa de requerimientos del sistema, se puede dar continuación a la definición de los *Sprints* en los cuales se desarrollará la aplicación móvil, desde el diseño del sistema hasta la etapa de mantenimiento del sistema.

### 4.2.1 Sprint 1

**Objetivo del sprint:** Definir el diseño que se usará para la aplicación móvil y de la base de datos.

**Sprint backlog:**

- Diseño de la base de datos.
- Elaboración de la guía de estilos.
- Diseño de la interfaz utilizando *Mockups*.
- Creación de la estructura del proyecto en *Ionic*.
- Elaboración del plan de pruebas.

### 4.2.2 Sprint 2

**Objetivo del sprint:** Definir la estructura del proyecto así como el desarrollo de las vistas a partir de los mockups realizados en la etapa anterior.

**Sprint backlog:**

- Implementación de la base de datos en *Cloud Firestore*.
- Vista de "Login".
- Vista de "Registro".
- Vista de "Configuración".
- Vista de "Acerca de".
- Vista de "Inicio".
- Vista de "Nuevo reporte de incidencia".
- Vista de "Nuevo reporte de objeto perdido".

### 4.2.3 Sprint 3

**Objetivo del sprint:** A partir de la estructura del proyecto, implementarle la funcionalidad requerida para obtener una primera versión de la aplicación.

**Sprint backlog:**

- Conexión de la aplicación con la base de datos.
- Implementación de “*Guards*” (los guards son la forma de protección de las vistas mediante sesiones de usuario en angular).
- Implementación del uso de las funciones nativas (cámara, llamadas, correo, red)
- Implementación de los servicios de *Firebase* para el uso de *notificaciones push*.
- Vista de “Listado de reportes de incidencia”.
- Vista de “Listado de objetos perdidos”.
- Vista de “Detalles de reportes de incidencia”.
- Vista de “Detalles de reportes de objetos perdidos”.
- Funcionalidad del cambio de estatus del reporte (Reporte de incidencia solucionado y objeto perdido entregado).

#### 4.2.4 Sprint 4

**Objetivo del sprint:** Validar los datos ingresados a la aplicación, así como empezar con el plan de pruebas diseñado para la aplicación.

**Sprint backlog:**

- Pruebas de envío de reportes de incidencia mediante pruebas de conjetura de errores y pruebas por exploración.
- Pruebas de envío de reportes de objetos perdidos mediante pruebas de conjetura de errores y pruebas por exploración.
- Pruebas de envío de ambos tipos de reportes en el caso de no existir una conexión de red mediante pruebas de conjetura de errores.
- Validación y pruebas del formulario de inicio de sesión mediante pruebas de conjetura de errores y pruebas por exploración.

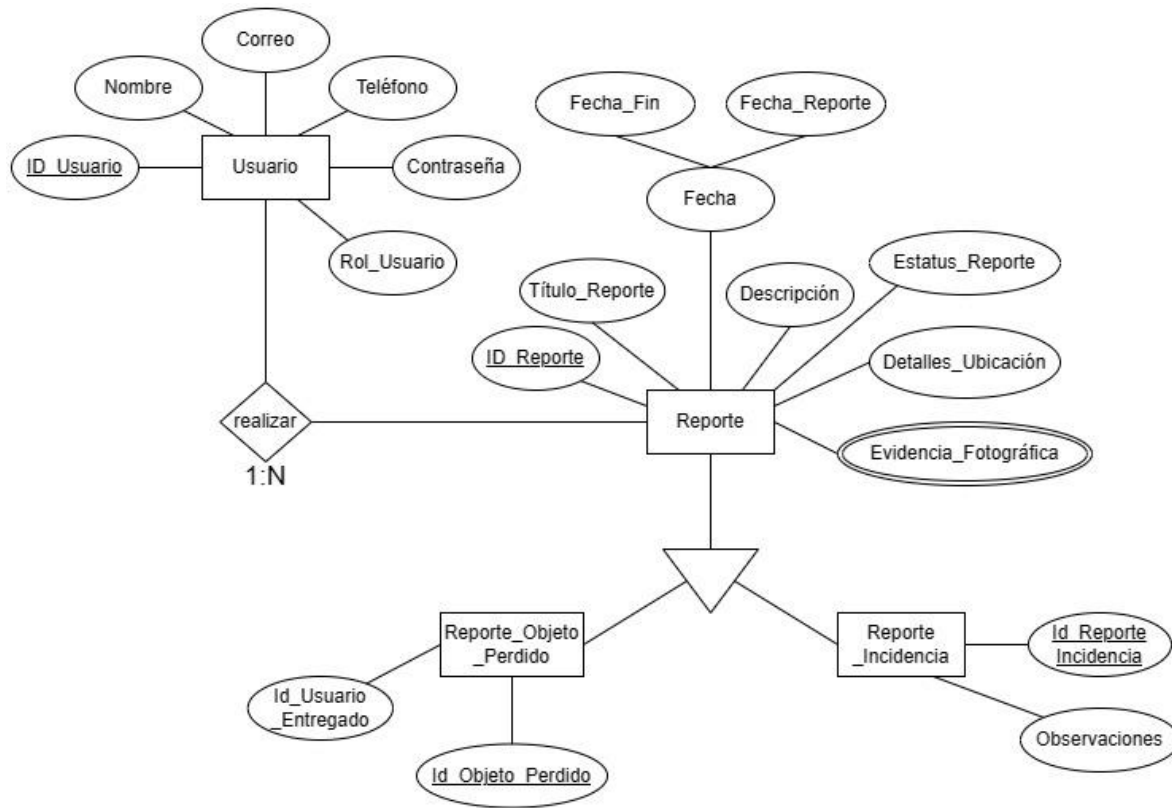
- Validación y pruebas del formulario de registro mediante pruebas de conjetura de errores y pruebas por exploración.
- Validación y pruebas de los formularios de ambos tipos de reportes mediante pruebas de conjetura de errores y pruebas por exploración.
- Validación y pruebas heurísticas de interfaz de usuario.
- Validación y pruebas heurísticas de experiencia de usuario.
- Validación y pruebas de respuesta a fallos mediante pruebas heurísticas y conjetura de errores.

## 4.3 Diseño del sistema

Una vez finalizado el análisis de los requerimientos del sistema donde se obtuvo una comprensión profunda del problema a resolver y se definieron las funcionalidades necesarias para la solución propuesta, por lo tanto, a continuación se presentará la estructura del sistema utilizando el modelo de Entidad-Relación, que es el punto de partida para diseñar la base de datos, además de la interfaz del sistema para proporcionar una idea más general de cómo se visualizará y el cómo será la aplicación en el apartado del desarrollo del sistema.

### 4.3.1 Modelo Entidad-Relación

En la *Ilustración 13* se puede observar en el modelo Entidad-Relación (*ER*) que describe como están relacionadas las entidades: **Usuario**, **Reporte**, **Reporte Incidencia** y **Reporte Objeto Perdido**. La entidad **Usuario** representa a los usuarios registrados en la aplicación y que serán los realizarán los reportes, la entidad **Reporte** representa a los reportes realizados por los usuarios, y las entidades **Reporte Incidencia** y **Reporte Objeto Perdido** heredan de la entidad **Reporte**, lo que significa que comparten los mismos atributos pero también tienen atributos específicos de cada entidad. La relación entre **Usuario** y **Reporte** es una relación de uno a muchos, lo que significa que un usuario puede realizar varios reportes, pero cada reporte pertenece a un solo usuario.



**Ilustración 13 - Modelo Entidad Relación, elaboración propia**

### 4.3.2 Modelo No Relacional

Una vez obtenido el modelo Entidad-Relación es más sencillo visualizar la información que se requerirá para poder crear un modelado de base de datos, con el cual, se alimentará la aplicación y permitir interactuar a los usuarios.

Como ya fue mencionado al principio del capítulo, la base de datos será implementada en *Cloud Firestore*, por lo tanto, no se puede implementar un modelo relacional dado que está basado en un modelo *NoSQL* así que se debe de utilizar un modelo no relacional.

Una vez analizadas e identificadas las entidades que componen la relación podemos convertirlas en los documentos con los que trabajará *Cloud Firestore*.

Al documento de Usuarios, *Ilustración 14*, se le añadió un campo compuesto de un arreglo que contendrá la información de que reportes ha realizado para hacer una búsqueda por reportes realizados por cada usuario.

Con el motivo de agilizar la búsqueda de Reportes, como se puede observar en la *Ilustración 15*, se anidó la información específica para los reportes de incidencias y de objetos perdidos agregando un campo de tipo de reporte para identificarlo.

```
Usuarios: {  
  id_usuario: {  
    "nombre": string,  
    "paterno": string,  
    "materno": string,  
    "correo": string,  
    "telefono": string,  
    "password": string,  
    "tipo_rol": string,  
    "reportes": int[],  
  }  
}
```

*Ilustración 14 - Documento Usuarios, elaboración propia*

```

Reportes: {
  id_reporte: {
    "titulo": string,
    "estatus": boolean,
    "descripción": string,
    "fecha_reporte": date,
    "fecha_fin": date,
    "detalles_ubicación": string,
    "tipo_reporte": int,
    "información_tipo_reporte": [
      //utiliza tipo reporte para saber información
      { //1 es Reporte Incidencia
        "observaciones": string,
      },
      { //2 es Objeto Perdido
        "id_usuario_entregado": string,
      }
    ]
    "evidencias": [
      {
        "evidencia_fotográfica": string
      }
      ...
    ]
  }
}

```

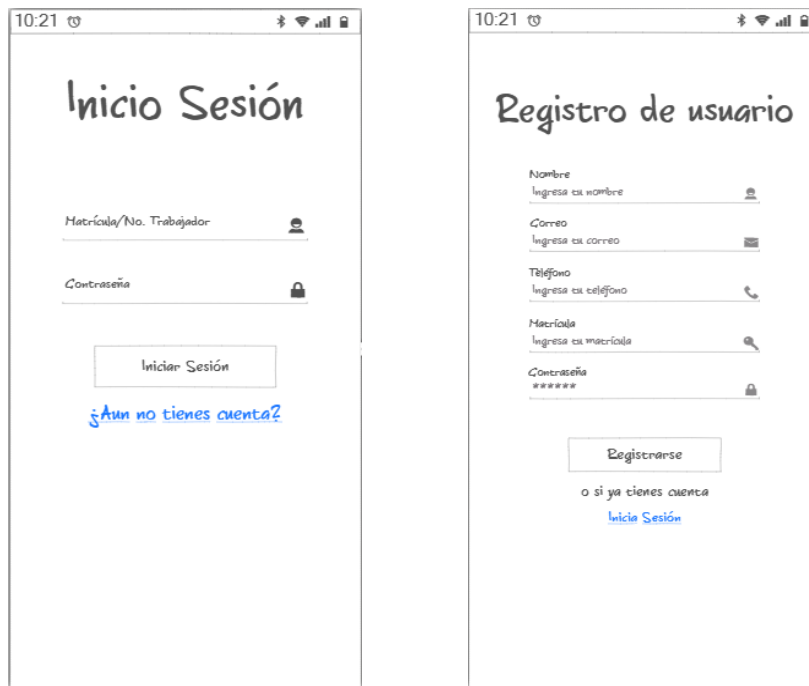
*Ilustración 15 - Documento Reportes, elaboración propia*

#### **4.3.4 Diseño de la interfaz**

En este apartado, se presentan los mockups de la interfaz de la aplicación móvil a desarrollar, los siguientes mockups son representaciones estáticas de las pantallas de la aplicación, diseñadas para mostrar cómo se visualizarán y cómo se relacionarán los diferentes elementos de la interfaz.

En la *Ilustración 16* se visualiza la forma en que los usuarios accederán a la aplicación, en caso de ser su primera vez dentro del sistema, necesitarán realizar

un registro, para esto existirá un formulario donde los usuarios podrán registrarse y posteriormente iniciar sesión.



**Ilustración 16 - Mockups de Inicio de Sesión y Registro de usuario, elaboración propia**

Para la pantalla de inicio, se consideró que la *Ilustración 17* es la ideal, dado que por motivos de experiencia de usuario se debe de poder alcanzar cada funcionalidad desde la vista principal.



**Ilustración 17 - Mockup de inicio del usuario, elaboración propia**

La *Ilustración 18* muestra los formularios de Reporte de Objeto Perdido y de Incidencia, al ser bastante similares, se puede aprovechar la reutilización de código en algunas partes para agilizar el desarrollo.

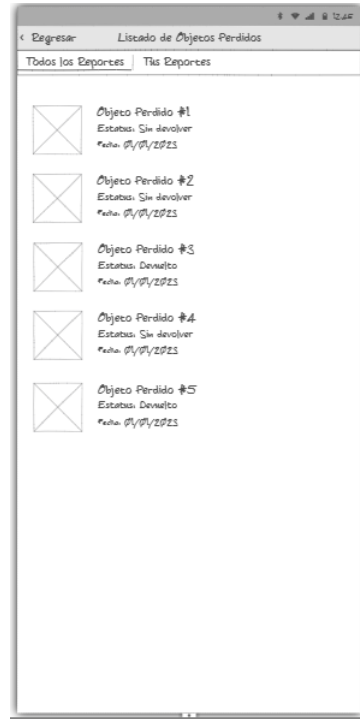
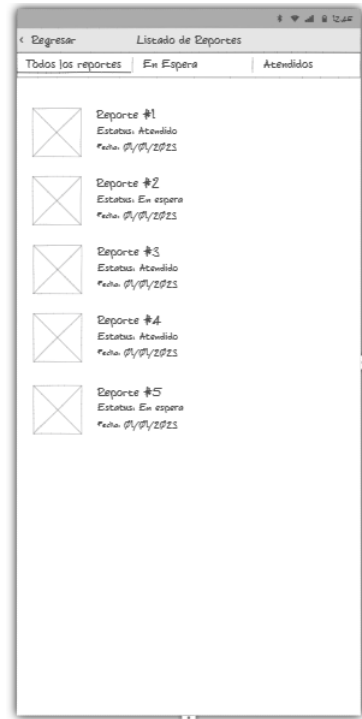


**Ilustración 18 - Mockups de Reporte de Incidencia y Objeto Perdido, elaboración propia**

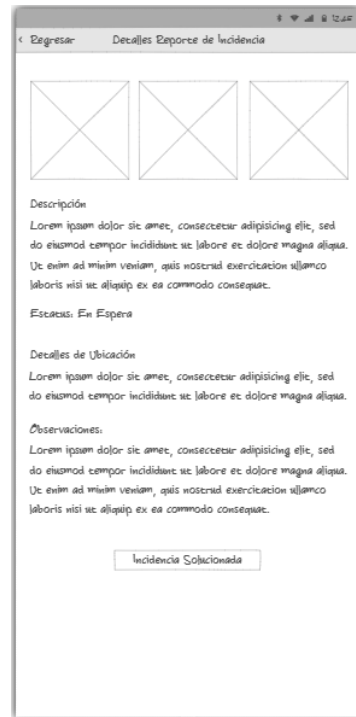
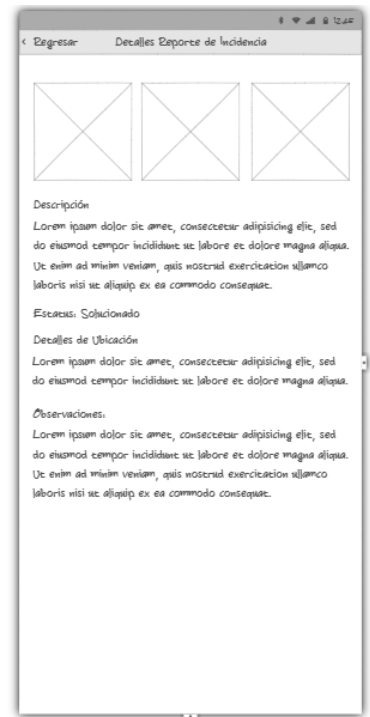
Para el listado de reportes, la manera más conveniente para mostrar la información es un lista que contenga una parte de la información más relevante del reporte *Ilustración 19*.

Para visualizar toda la información de un reporte de incidencia basta con tocar cualquiera y se mostrará todos los detalles *Ilustración 20*, para los usuarios con rol de alumnos visualizarán la ventana de la izquierda y si es rol administrador observará la ventana de la derecha.

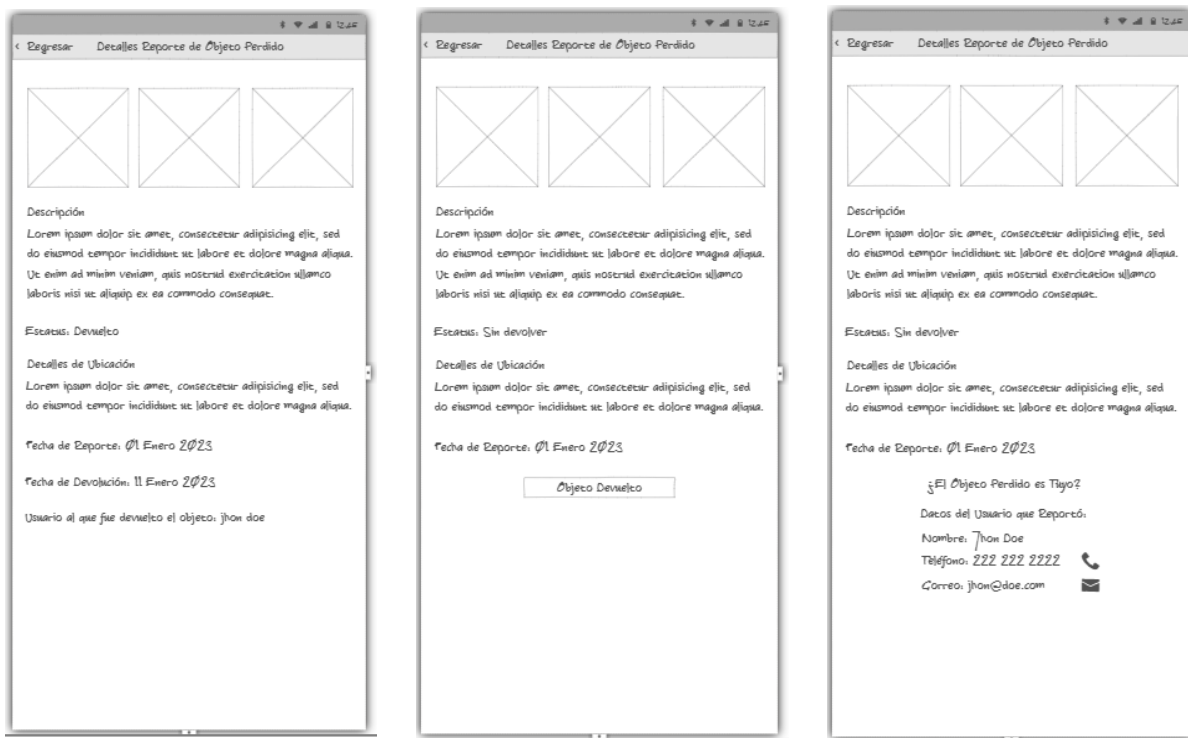
Para el caso de objetos perdidos existirán 3 casos, como lo muestra la *Ilustración 21*, si el objeto fue devuelto, la vista a desplegar será la del lado izquierdo. Si el alumno desea cambiar el estatus del reporte, que significa devolver el objeto perdido a su propietario, observará la pantalla del medio. En el caso que el alumno detecte un objeto sin devolver que no haya reportado, se le mostrará la pantalla de la derecha, donde se mostrará la información del alumno que ha hecho el reporte.



**Ilustración 19 - Mockups del Listado de Reportes, elaboración propia**



**Ilustración 20 - Mockups de Detalles de Reporte de Incidencia, elaboración propia**



**Ilustración 21 - Mockups de Detalles de Objetos Perdidos, elaboración propia**

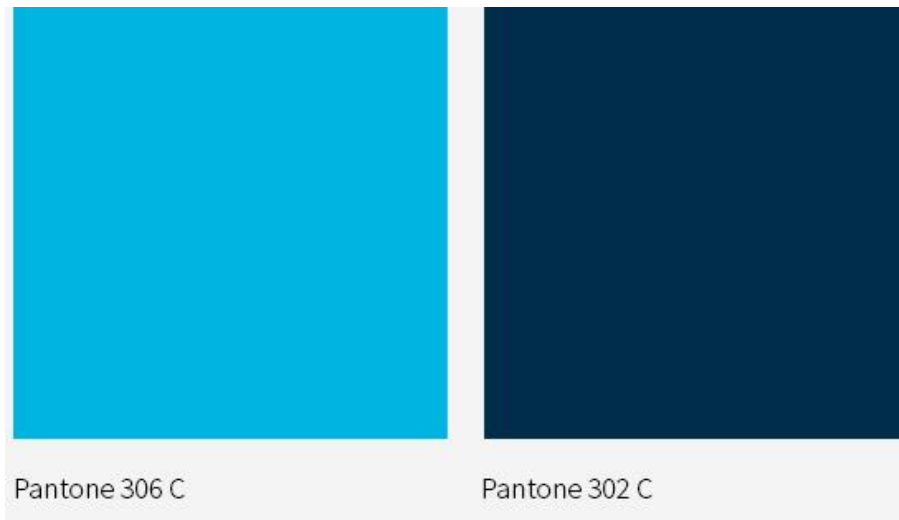
### 4.3.5 Diseño de la guía de estilos

En esta sección se definió el apartado visual, colores, tipografía e iconografía que complementará el diseño de *mockups* y hará de la aplicación móvil una presentación más atractiva para los usuarios.

#### 4.3.5.1 Paleta de colores

Al optar por una aplicación enfocada a la facultad, los colores elegidos fueron los proporcionados por el área de comunicación de la universidad.

Estos colores se encuentran en su código *Pantone*, *Ilustración 22*, pero, para ser utilizados en la aplicación móvil, se utilizaron ambos colores con su respectivo valor hexadecimal, **#00B4E1** y **#002D4C**.



**Ilustración 22 –Colores institucionales de la BUAP. BUAP. (s.f.)**

Para aplicarlos en la aplicación móvil, se utilizó a la herramienta generadora de colores de *Ionic*, como se muestra en la *Ilustración 23* e *Ilustración 24*, que facilita incorporar los colores con los componentes del *framework*.



**Ilustración 23 - Color primario de la aplicación, elaboración propia**



**Ilustración 24 - Color secundario de la aplicación, elaboración propia**

#### **4.3.5.2 Tipografía**

La tipografía seleccionada para el proyecto fue la fuente “*Source Sans Pro*” diseñada por “**Paul D. Hunt**”. La licencia de esta tipografía es *Open Font License*, esto significa que es libre de usar en proyectos o productos impresos o digitales de forma gratuita.

#### **4.3.5.3 Iconografía**

Los iconos que serán utilizados en la aplicación serán “*Ionicons*”, iconos propios de *Ionic*, además de utilizar en conjunto “*Material Symbols*” que son iconos con licencia *Open Source* de *Google*.

### **4.4 Desarrollo del sistema**

En esta etapa, fue donde se consiguió desarrollar la aplicación partiendo de las directrices de diseño obteniendo resultados satisfactorios que cumplen con los requerimientos planteados anteriormente. A continuación, se muestran las capturas de pantalla que muestran los resultados obtenidos.

Para las pantallas de inicio de sesión y registro (*Ilustración 25 e Ilustración 26*, respectivamente) se ocupó un formulario para autenticarse y registrar datos de los usuarios, esto permite usar la aplicación, si no se está registrado y autenticado previamente el alumno no puede hacer uso de la aplicación.

En la pantalla del menú *Ilustración 27*, se utilizó un menú lateral, esto nos permite navegar entre pantallas y dirigirse a la vista deseada.

La *Ilustración 28* muestra la pantalla de inicio, se agregaron las pantallas útiles del proyecto (los apartados de reportes de incidencias y objetos perdidos) en estos de igual manera se puede navegar por la aplicación e ir a la pantalla deseada.

Para poder enviar reportes, se hizo uso de formularios que solicitan la información correspondiente a los usuarios tal y como se puede observar en las *Ilustraciones 29 y 30*.

El listado de incidencias y objetos perdidos *Ilustración 31* e *Ilustración 33* se muestran los reportes enviados por los usuarios y que son devueltos por la base de datos, en caso de no contar con reportes enviados se muestran mensajes de ayuda, estos mensajes se pueden observar para el caso de reportes de incidencias en la *Ilustración 36* y para objetos perdidos en la *Ilustración 37*.

La vista detallada de un reporte de incidencia en específico, que contienen la información de dicho reporte se visualiza en la *Ilustración 32*. Para el caso de objetos perdidos, si el objeto no se ha devuelto, la pantalla correspondiente es la *Ilustración 34*. Si el objeto perdido ha sido entregado le corresponde la *Ilustración 35*.

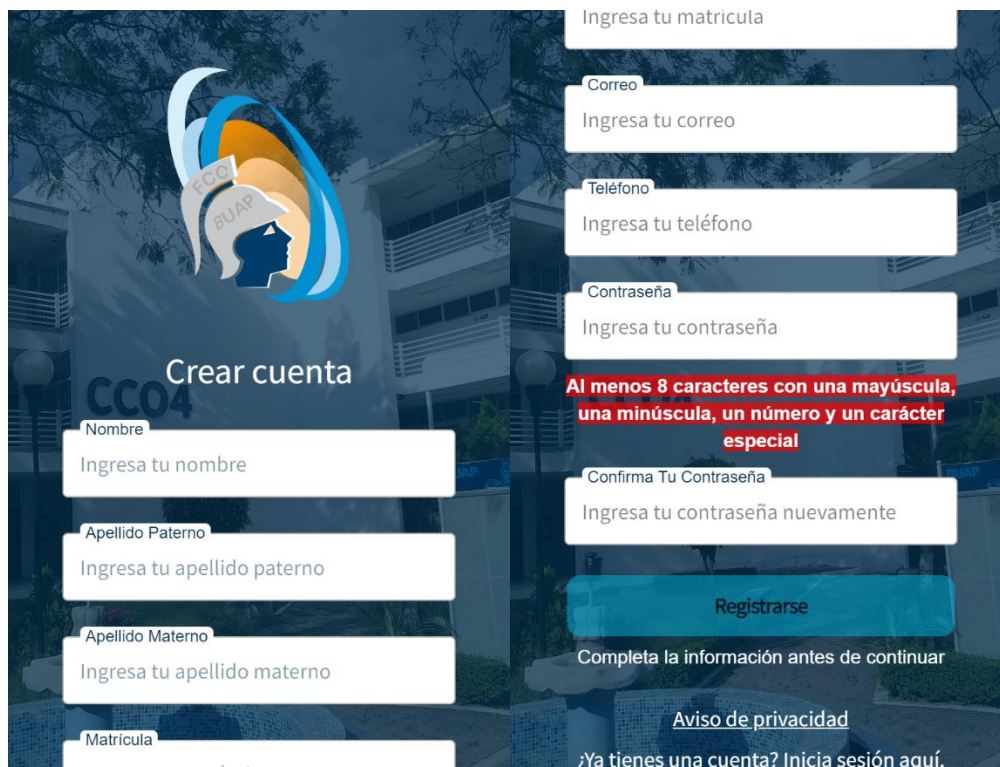
En la *Ilustración 38*, se muestra un ejemplo de una *notificación push* ejecutada tras ser enviado un nuevo reporte de objeto perdido por otro usuario.

La pantalla de configuración *Ilustración 39* muestra la información del usuario autenticado en la aplicación, además que en esta pantalla permite modificar su información personal. También, en esta pantalla, se brinda al usuario la libertad de elegir si desea recibir o no las notificaciones enviadas por el sistema.

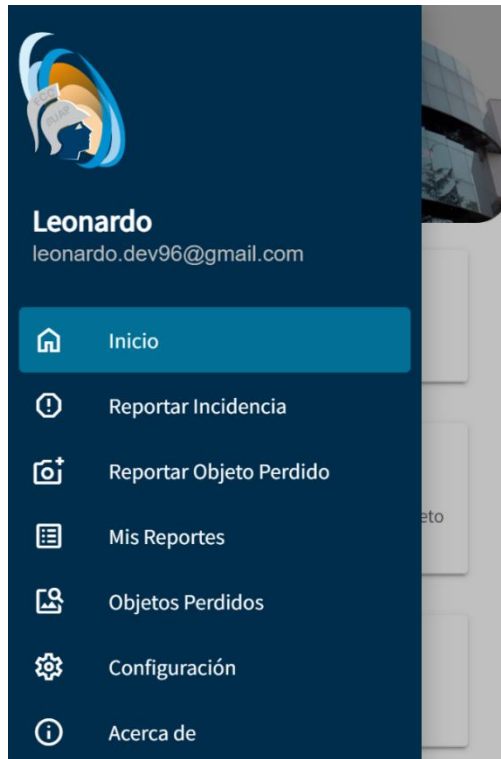
En la pantalla de acerca de, *Ilustración 40*, se muestra la información propia de la aplicación móvil, su versión de compilación y su versión actual obtenidas a través de la configuración del proyecto de Android. También en esta pantalla se puede acceder al aviso de privacidad del proyecto.



**Ilustración 25 - Pantalla de inicio de sesión, elaboración propia**



**Ilustración 26 - Pantalla de registro de usuario, elaboración propia**



**Ilustración 27 - Pantalla del menú, elaboración propia**






**Ilustración 28 - Pantalla de inicio, elaboración propia**

☰ **Nuevo Reporte de Incidencia**

Título:

Descripción:

Ubicación:

Enviar

**Ilustración 29 - Pantalla de nuevo reporte de incidencia, elaboración propia**




☰ **Nuevo Reporte de Objeto**

Título:

Descripción:

Ubicación:

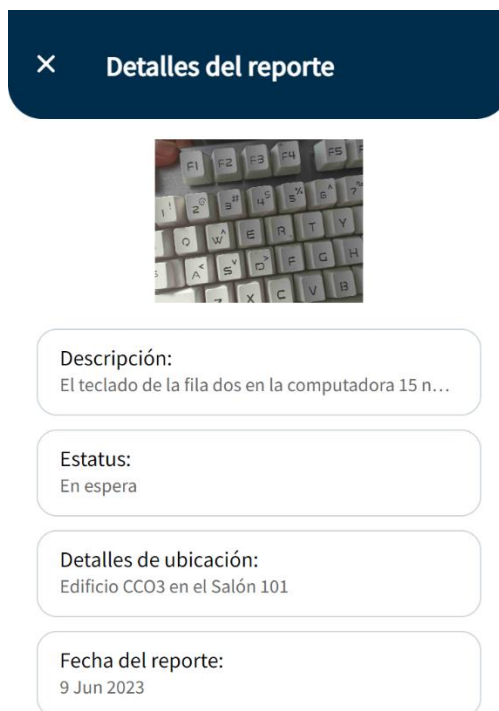
El objeto es mío

**Ilustración 30 - Pantalla de nuevo reporte de objeto perdido, elaboración propia**



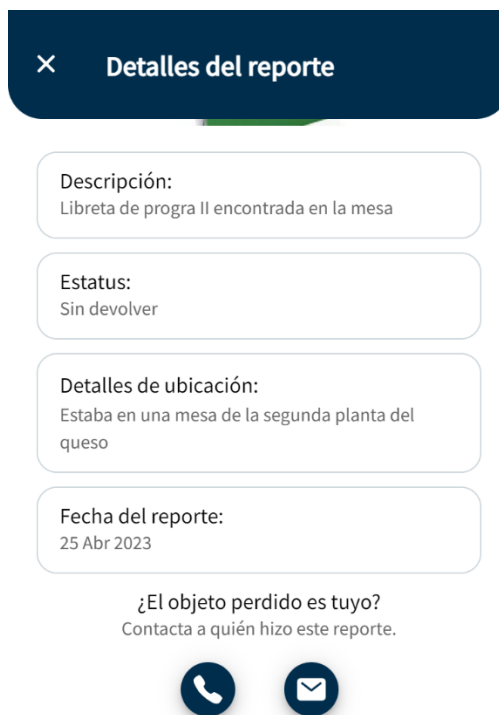
**Ilustración 31 - Pantalla de listado de reportes de incidencia, elaboración propia**



**Ilustración 32 – Pantalla de detalles de reporte de incidencia, elaboración propia**



**Ilustración 33 - Pantalla de listado de objetos perdidos, elaboración propia**



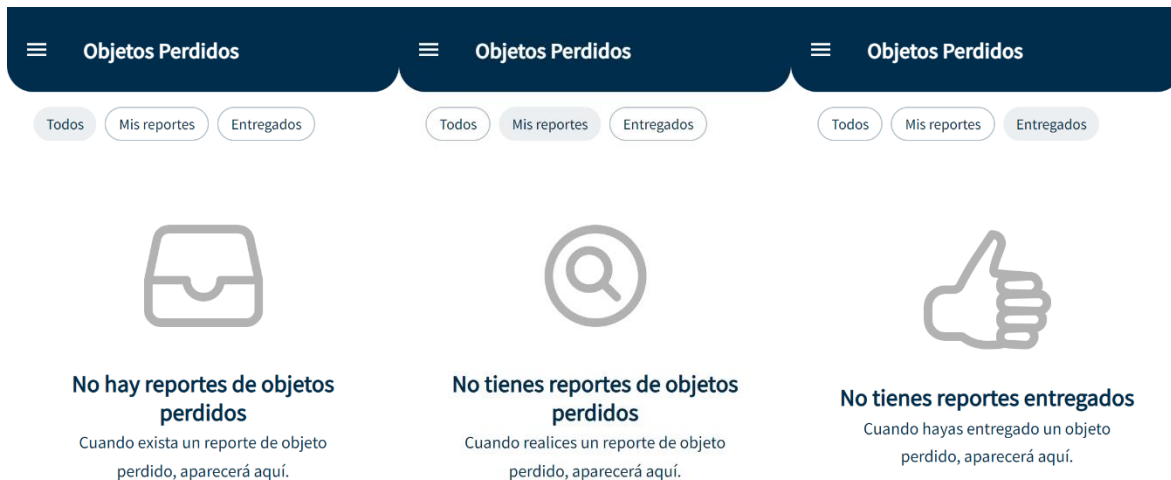
**Ilustración 34 - Pantalla de detalles de objeto perdido, elaboración propia**



**Ilustración 35 - Pantalla de reporte de objeto perdido entregado, elaboración propia**



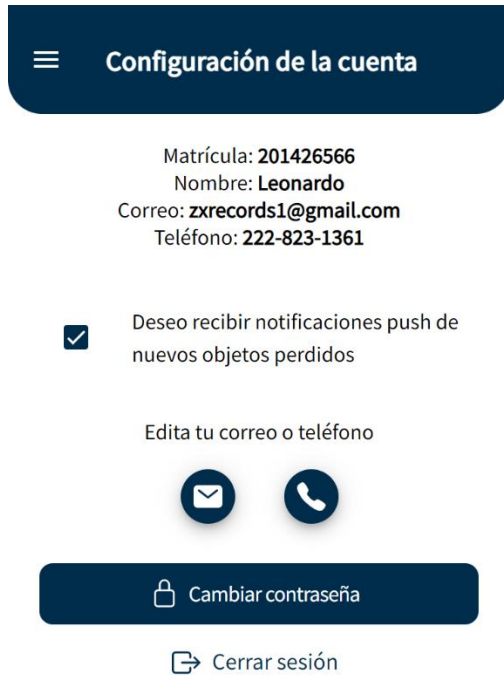
**Ilustración 36 - Pantallas de mensajes de ayuda en listado de reportes de incidencia, elaboración propia**



**Ilustración 37 - Pantallas de mensajes de ayuda en listado de objetos perdidos, elaboración propia**



**Ilustración 38 - Pantalla de notificación push lanzada tras enviarse un nuevo reporte de objeto perdido**



**Ilustración 39 - Pantalla de configuración, elaboración propia**



**Ilustración 40 - Pantalla de acerca de, elaboración propia**

## **CAPÍTULO V: CONTROL DE RIESGOS Y SEGURIDAD**

La seguridad, en el ámbito informático, es la práctica de proteger información digital de forma confidencial y segura contra los accesos no autorizados. Esta seguridad abarca la seguridad física del hardware, desde el cual se accede a los datos, además de los dispositivos donde se almacena la información, los puntos de acceso, etc. Y se complementa en la lógica del software en la cual sirve de medio para comunicar dichos elementos físicos. Cuando se aplican ciertos protocolos de seguridad y se implementa de manera correcta se protegen de manera eficaz los datos privados que pudieran ser objetivo de ciber delincuentes. (IBM, s.f.)

### **5.1 Análisis para la protección de datos de la aplicación**

Para este proyecto la prioridad a proteger son los datos personales proporcionados por los usuarios y almacenados mediante colecciones de documentos en *Cloud Firestore*.

Para el caso de la colección “**usuarios**” contienen campos a proteger que son los siguientes:

- **Contraseña**
- **Correo**
- **Matrícula**
- **Número de teléfono.**

En las colecciones de “**reportes**” y “**roles**” no se contiene información delicada o privada que pueda poner en riesgo la integridad de algún usuario. El cómo se implementarán las estrategias de seguridad para la protección de datos serán mencionadas más adelante.

#### **5.1.1 Protección contra ataques al sistema**

Para el caso de ataques de “*ingeniería social*”, donde se puede acceder al sistema mediante engaños a usuarios o dónde puedan estar comprometidas sus acciones, se plantean las siguientes estrategias:

- **Uso de contraseñas seguras:** Al momento del registro se le solicitará y el sistema validará, que el usuario introduzca una contraseña única que contenga al menos un número, una mayúscula y un carácter especial.
- **Capacitación:** Informar a los usuarios que no compartan sus contraseñas, sus cuentas y que sean cuidadosos con el origen de la información y verifiquen que provenga de parte de una fuente confiable.

Para el caso de “*Sniffer*” y lograr contrarrestar este tipo de ataque de monitoreo, el principal aliado es la encriptación en tránsito que proporciona la herramienta de *Firebase*. Los datos son ilegibles mientras viajen del cliente autorizado al servidor y del servidor al cliente autorizado. (Google Developers, s.f.)

### 5.1.2 Protección contra ataques a las contraseñas

Antes de mencionar los posibles ataques a las contraseñas, cabe señalar que las mismas no serán almacenadas en texto claro, es decir, no serán legibles para humanos. Serán almacenadas mediante su valor en hash. Esto es seguro hasta cierto punto, puesto que, algunas contraseñas puede que su hash sea idéntico al hash de otra contraseña. Para evitar esto, se agregó un *Salt* que al combinar con la contraseña del usuario evite en lo mayor posible hash similares. (w3c, 2016)

**Ataques de fuerza bruta o ataques de diccionario**, para disminuir el éxito de este ataque, como se mencionó anteriormente, el uso de contraseñas seguras es vital para que a los atacantes les tome más tiempo el poder acceder. Como medida adicional, no se dará información que pueda serles de utilidad a los atacantes, es decir, si se adivina el usuario pero la contraseña es incorrecta se mencionará en el texto de apoyo lo siguiente: **“El usuario o contraseña es incorrecto”**; para el caso de usuario y contraseña incorrectos se mencionará el mismo error.

**Ataque de tabla de arcoíris**, en este tipo de ataques se usan valores hash para reducir los tiempos de ataques y conseguir un acceso más rápido (Hill, 2021), por este motivo y como medida, se utilizó el valor *Salt* para disminuir el rango de éxito al intentar acertar la contraseña.

**Ataque de cumpleaños** este ataque usa hash que puedan ser válidos para obtener acceso de dos o más cuentas debido a una colisión hash. Al momento de generar el hash pueden existir dos o más contraseñas que arrojen el mismo valor hash, aún aplicando un Salt, por este motivo, se utilizó el algoritmo *SHA-3-512* que genera un tamaño de valor de 512 bits y que resulta casi imposible que dos o más contraseñas coincida su valor hash. (w3c, 2016)

Para añadir protección a la base de datos, sólo pueden acceder clientes autorizados, estos son registrados directamente en la consola de *Firebase*. El único cliente autorizado es la aplicación, que será instalada mediante múltiples dispositivos, por ese motivo se añadieron reglas de escritura y lectura a la base de datos. (Google Developers, s.f.)

Para poder acceder a la información debe de cumplir con las siguientes características:

- **Autenticación:** El cliente debe de entablar una conexión exitosa mediante las credenciales del usuario. Si la autenticación es exitosa entonces podrá acceder a su información de usuario almacenada en la base de datos además de que podrá realizar y consultar reportes. (Google Developers, s.f.)
- **Autorización:** El cliente sólo podrá acceder a información propia e información compartida para todos como lo son los reportes, además de datos de contacto que son vinculadas con cada reporte. No podrá modificar ni visualizar datos de otros usuarios. (Google Developers, s.f.)

### **5.1.3 Protección de los datos sensibles de los usuarios**

Como se mencionó con anterioridad, los elementos a proteger son los campos: contraseña, correo, matrícula y número de teléfono. El cómo se protegerán las contraseñas ya se ha proporcionado la información correspondiente para evitar posibles ataques. En el caso del correo, de la matrícula y del número de teléfono se cifrarán antes de ser almacenadas en *Cloud Firestore* mediante el algoritmo *AES* (Advanced Encryption Standard) (Packetizer, Inc., s.f.). Sin embargo, como medida adicional, *Firebase* cifra la información de todos los datos almacenados mediante el

algoritmo *AES*, que proporciona una segunda capa de seguridad que, en caso de ser robada mediante una vulnerabilidad de *Firebase*, los datos más críticos estarán encriptados y se necesitará otra clave secreta para poder descifrarlos y acceder a ellos. (Google Developers, s.f.)

## 5.2 Vulnerabilidades y Plan de Mitigación de Riesgos

Las vulnerabilidades del software son fallos de seguridad o dentro del mismo código fuente que puede ser utilizada y explotada por delincuentes cibernéticos. (National Institute of Standards and Technology, s.f.)

A continuación se mencionarán las diferentes vulnerabilidades que pueden afectar a la aplicación y posteriormente se describirá la manera de evitar dichas vulnerabilidades.

### 5.2.1 Vulnerabilidades tecnológicas

**Debilidad de protocolos de red:** Al compartir la información mediante *HTTPS* entre el cliente y el servidor puede haber ataques de monitoreo o de interceptación de datos.

**Debilidad del sistema:** Al usar el sistema operativo *Android*, puede haber posibles vulnerabilidades que afecten por ejemplo, la información almacenada en la cache, programas maliciosos, robo de llaveros de acceso, etc.

**Debilidades de bibliotecas o *frameworks*:** Al ser una aplicación híbrida móvil utiliza distintas bibliotecas que pueden ser vulneradas y pueden crear una puerta de acceso a la información que transite mediante ellas. De igual manera, los *frameworks* a utilizar, en este caso, *Angular*, *Ionic*, *Firebase*, pueden presentar vulnerabilidades que no se pueden prevenir en el desarrollo de la aplicación móvil.

### 5.2.2 Vulnerabilidades de configuración

**Cuentas de usuarios con contraseñas sencillas:** Los usuarios al momento de crear contraseñas, no le dan la seguridad recomendada por la aplicación (uso de mayúsculas, minúsculas, caracteres especiales y números), escriben contraseñas muy utilizadas o reciclan sus propias contraseñas, esto conlleva a que atacantes

puedan adivinar muy rápidamente sus claves de acceso y poder acceder a la aplicación.

### 5.2.3 Vulnerabilidades de políticas

**Plan de recuperación ante desastres:** Actualmente el servidor de base de datos se encuentra vinculada a una cuenta de *Gmail*. En el caso hipotético de perder el acceso y no exista forma de recuperar la cuenta ni recuperando contraseña, la consola del servidor sería inaccesible. La aplicación seguiría funcionando con normalidad pero el darles mantenimiento a los datos sería casi imposible. En cuanto al respaldo de la información, existe una manera implementada y donde cada 24 horas se exporta la información a un servicio propio en la nube.

### 5.2.4 Plan para mitigar los riesgos del proyecto

Una vez identificadas las posibles vulnerabilidades de la aplicación móvil, se presentará el plan para evitar en la mayor medida posible ser posibles candidatos a ataques indeseados o minimizar los daños en caso de que ocurran.

Para solventar las vulnerabilidades tecnológicas se proponen los siguientes pasos a seguir:

- **Cifrado de datos en tráfico:** El cifrar los datos nos ayudará a mitigar una posible monitorización o rastreo en la red de los datos que se intercambian entre el servidor y el cliente.
- **Cifrado de datos en caché:** Se cifrarán los datos del usuario almacenados en el dispositivo y esto será de suma importancia, puesto que, la presencia de algún programa malicioso pone en riesgo el robo de información que se encuentra en texto plano.
- **Actualizaciones constantes:** El estar monitoreando con frecuencia las vulnerabilidades detectadas al sistema operativo *Android*, a los *frameworks* y librerías utilizadas y que puedan poner en riesgo a la aplicación, el parchar dichos errores lo más pronto posible y liberar una actualización, logrará evitar que la aplicación pueda tener puertas abiertas a atacantes.

En el caso de vulnerabilidades de configuración las medidas a implementar son las siguientes:

- **Autenticación:** No se podrán hacer operaciones de lectura o escritura ni tener acceso a la base de datos, si no se ha llevado a cabo una verificación exitosa de la identidad del usuario mediante el inicio de sesión.
- **Autorización:** Mediante reglas de acceso de lectura y escritura en *Cloud Firestore* se especifica lo que un usuario autenticado es capaz de hacer. Un usuario que tenga un rol de alumno sólo será capaz de ver su información personal, realizar reportes de incidencias y objetos perdidos, consultar sus propios reportes de incidencias y ver el listado de reportes de objetos perdidos. El usuario con rol de administrador sólo podrá ver el listado de reportes de incidencias que han sido enviados para canalizarlos con el área correspondiente encargada de su pronta solución.
- **Contraseñas fuertes:** No se le permitirá al usuario utilizar contraseñas sencillas, para ello, el sistema solicita contraseñas que utilicen al menos una mayúscula, una minúscula, un carácter especial y un número, además de solicitar una contraseña de longitud mínima de 8 caracteres.

Por último, para las vulnerabilidades de políticas se llevará a cabo lo siguiente:

1. **Contacto de confianza:** Compartir el acceso de la consola de *Cloud Firestore* con un contacto seguro que no ponga en peligro la seguridad de los datos. Esto protegerá en caso de perder acceso a la cuenta principal de la aplicación.

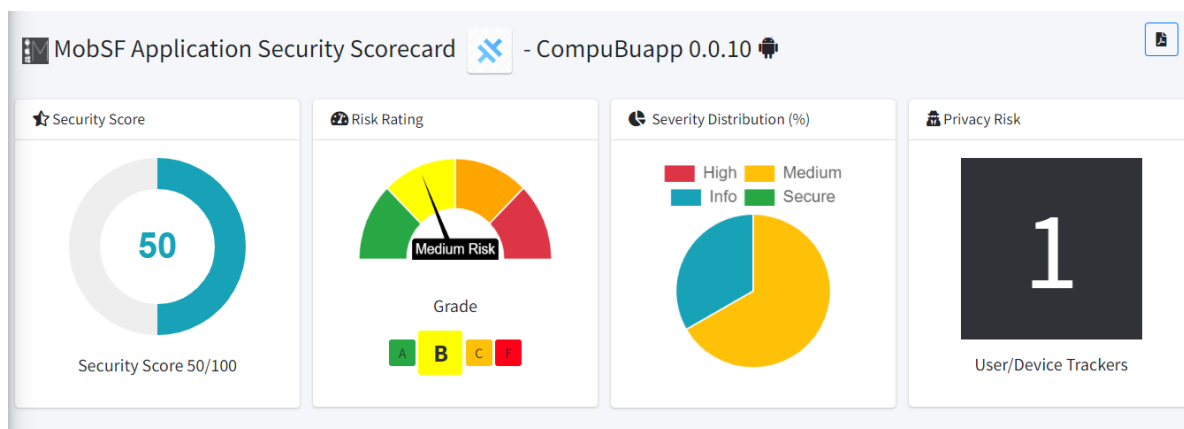
Además de estas medidas y para proteger la seguridad del versionado de la aplicación se llevará a cabo lo siguiente:

2. **Firmado de la aplicación:** Con el motivo de que no pueda ser alterada la integridad de la aplicación de manera indeseada no será posible sobrescribir la aplicación sino se cuenta con la misma clave de carga con la cuál fue firmada la aplicación original. Esto permite que sólo puedan actualizar las aplicaciones que coincidan con la misma clave de carga.

## 5.2.5 Herramienta de detección de vulnerabilidades

Con el fin de hacer una detección de vulnerabilidades automática, una vez obtenido un *APK* instalable para la aplicación, se hará uso de la herramienta “*MobSF*” que realizará un escaneo automático dinámico y, en caso de detectar alguna vulnerabilidad, se solucionará antes de tener la versión final de la aplicación.

En una última inspección (realizada el 12 de junio de 2023) con la herramienta de detección de vulnerabilidades se obtuvo la siguiente información:



**Ilustración 41 –Resultados del escaneo de vulnerabilidades generado por la herramienta “*MobSF*”. Romero, L. (2023).**

Los gráficos mostrados por la herramienta *MobSF*, como se puede observar en la *Ilustración 43*, aunque parezca de un riesgo medio, comparado con otras aplicaciones, se encuentra en el promedio puesto que tienen calificaciones similares entre 40 y 55 de calificación, también el resultado se atribuye a los permisos que utiliza la aplicación (lectura y escritura de datos, cámara, llamadas telefónicas), *MobSF* los considera de un riesgo alto y disminuye considerablemente la calificación de seguridad obtenida. En cuanto al riesgo de privacidad que menciona, hay uno rastreador identificado, se debe al rastreador de *Google Analytics* que comparte datos de fallos relacionados con *Firebase*. Se continuará investigando y tratará de reducir el riesgo antes de obtener una versión abierta al público.

### **5.3 Aviso de privacidad**

En México, existe una ley conocida como **“Ley General de Protección de Datos Personales en Posesión de Autoridades, Órganos y Organismos de Gobiernos”** en la cual se habla de que al momento de disponer de información personal, se le debe de informar al individuo por qué los recopila y cuál es el propósito de obtenerlos. (Senado de la República, 2012)

En el aviso de privacidad se debe de incluir la información, de manera clara y completa, de cómo se recopilan los datos, para qué se utilizan, con quién o quiénes se comparten, el propósito de recopilar sus datos personales, las medidas de cómo se protegerá su información, el responsable de sus datos, así como la manera de que el titular pueda ejercer sus derechos **“ARCO”** (acceder, rectificar, cancelar y oposición) de sus datos personales y el procedimiento de cómo ejercer dichos derechos.

El aviso de privacidad propuesto para esta aplicación es el siguiente:

La presente política de privacidad describe cómo se recopila, utiliza, almacena y protege la información personal de los usuarios de la aplicación móvil **“CompuBuapp”**. Al registrarse y utilizar la aplicación, usted acepta los términos y condiciones de esta política de privacidad y confirma que desea compartir su información personal como nombre, teléfono, correo con otros usuarios dentro de la aplicación. Es importante informarle que sus datos personales serán tratados y resguardados con base en los principios de licitud, calidad, consentimiento que establece la Ley Federal de Protección de Datos Personales en Posesión de los Particulares.

#### **Recopilación de Información**

La información personal que se solicita al usuario al momento de registrarse en la aplicación incluye nombre, correo, teléfono, contraseña y su matrícula. Además, los reportes de incidencia y objetos perdidos requerirán información como título del reporte, estatus, ubicación, descripción, evidencia fotográfica.

## **Uso de la Información**

La información recopilada será utilizada para permitir a los usuarios enviar y recibir reportes de incidencia y objetos perdidos dentro de la aplicación. El administrador sólo tendrá acceso a los reportes de incidencia para poder modificar su estatus de solución. La aplicación también permitirá que los usuarios puedan consultar reportes de objetos perdidos notificados por otros alumnos. No se compartirá información personal con terceros por ningún motivo, a excepción de que una autoridad competente los requiera y sólo en caso de que exista una obligación legal para hacerlo.

## **Protección de la Información**

Se implementarán medidas de seguridad para proteger la información personal recopilada. La aplicación utilizará procedimientos de autenticación para garantizar que solo los usuarios registrados tengan acceso a los reportes de incidencia y objetos perdidos. También se utilizarán medidas como cifrado de datos para salvaguardar los datos proporcionados por los usuarios para evitar el acceso no autorizado, la divulgación y el uso indebido de la información.

## **Derechos de los titulares**

Usted tiene derecho a conocer qué datos personales tenemos de usted, para qué los utilizamos y las condiciones del uso que les damos. Asimismo, es su derecho solicitar la corrección de su información personal en caso de que esté desactualizada, sea inexacta o incompleta, que la eliminemos de nuestros registros o bases de datos cuando considere que la misma no está siendo utilizada adecuadamente, así como oponerse al uso de sus datos personales para fines específicos. Estos derechos se conocen como derechos ARCO.

Para ejercer cualquiera de los derechos *ARCO*, usted deberá presentar la solicitud respectiva a través del siguiente correo electrónico: [compubuapp@gmail.com](mailto:compubuapp@gmail.com).

Dicha solicitud deberá contener: (I) nombre completo y correo electrónico del titular; (II) documentos que acrediten la identidad o, en su caso, la representación legal del titular; (III) la descripción clara y precisa de los datos personales respecto de los que se busca ejercer alguno de los derechos ARCO; y (IV) cualquier otro elemento o documento que facilite la localización de los datos personales.

Una vez recibida la solicitud, contaremos con un plazo de 20 días hábiles para dar respuesta a su solicitud y comunicarle la procedencia de esta a través del correo electrónico proporcionado por usted. En caso de ser procedente su solicitud, haremos efectiva la misma en un plazo máximo de 15 días hábiles a partir de la fecha en que le comuniquemos la procedencia de su solicitud.

Cualquier modificación a este aviso de privacidad le será notificada a través del sitio web o a través del correo electrónico que nos haya proporcionado.

## CAPÍTULO VI: PRUEBAS Y RESULTADOS

### 6.1 Plan de pruebas

Una vez finalizada la etapa de desarrollo y basándose en el plan de pruebas elaborado en el primer *Sprint*, se evaluaron los distintos componentes de la aplicación móvil. A continuación se mencionarán los resultados de los tipos de pruebas propuestos, definidos anteriormente en el *Capítulo III: Marco Teórico*, que cubrirán la mayor parte de errores que se puedan encontrar en la aplicación, así como los resultados obtenidos al aplicar dichas tareas.

#### 6.1.1 Pruebas por exploración

Las pruebas por exploración se evaluaron con las siguientes historias de usuario, además se anexan las observaciones que se identificaron en el proceso.

Historia de usuario #1	
<b>Nombre:</b>	Registro de usuario
<b>Usuario:</b>	Alumno
<b>Descripción:</b>	Comprobar que el usuario logre registrarse mediante el formulario que se encuentra en la vista de registro.
<b>Observaciones:</b>	Se validan que los datos estén completos además de que en los campos de matrícula y teléfono sólo permite ingresar números. Una vez finalizado el registro redirige al inicio de sesión para poder continuar. Los mensajes de error no son muy legibles, se recomienda aumento de tamaño de fuente o lograr un buen contraste.

*Tabla 11 - Historia de usuario del registro de usuario, elaboración propia*

Historia de usuario #2	
<b>Nombre:</b>	Inicio de sesión
<b>Usuario:</b>	Alumno / Administrador

<b>Descripción:</b>	Verificar una correcta validación de credenciales de usuario y que permita un inicio de sesión exitoso.
<b>Observaciones:</b>	Si la contraseña o matrícula no coinciden no permiten una autenticación correcta. Introduciendo las credenciales correctas redirecciona al home del usuario correctamente. Los mensajes de error no son muy legibles, se recomienda aumento de tamaño de fuente o lograr un buen contraste.

**Tabla 12 - Historia de usuario del inicio de sesión, elaboración propia**

<b>Historia de usuario #3</b>	
<b>Nombre:</b>	Envío de reporte de incidencia
<b>Usuario:</b>	Alumno
<b>Descripción:</b>	Evaluar una correcta funcionalidad de la aplicación al enviar un reporte de incidencia y verificar que sea procesado por el sistema con éxito.
<b>Observaciones:</b>	Verifica correctamente los campos, si un campo no es completado no permite el envío del reporte. Muestra mensaje de éxito al ser enviado correctamente el reporte. En caso de no conexión, muestra un mensaje de error.

**Tabla 13 - Historia de usuario del envío de reporte de incidencia, elaboración propia**

<b>Historia de usuario #4</b>	
<b>Nombre:</b>	Envío de reporte de objeto perdido
<b>Usuario:</b>	Alumno
<b>Descripción:</b>	Evaluar una correcta funcionalidad de la aplicación al enviar un reporte de objeto perdido y verificar que sea procesado por el sistema con éxito, además de comprobar que el sistema lance notificaciones a otros dispositivos que cuenten con la aplicación instalada y activa (que hayan iniciado sesión)

<b>Observaciones:</b>	<p>Verifica correctamente los campos, si un campo no es completado, no permite el envío del reporte.</p> <p>Muestra mensaje de éxito al ser enviado correctamente el reporte.</p> <p>En caso de no conexión, muestra un mensaje de error.</p>
-----------------------	---

**Tabla 14 - Historia de usuario del envío de reporte de objeto perdido, elaboración propia**

<b>Historia de usuario #5</b>	
<b>Nombre:</b>	Contactar usuario del reporte mediante correo
<b>Usuario:</b>	Alumno
<b>Descripción:</b>	Verificar que la función de contactar al usuario que hizo el reporte de objeto perdido, mediante correo electrónico en el apartado de objetos perdidos, sea correcta.
<b>Observaciones:</b>	Al tocar el elemento de contacto mediante correo, redirige en automático a la aplicación de correo del usuario (Gmail o el que el usuario elija) una vez dentro de la aplicación de correo ya se encuentra cargado el correo de quien hizo el reporte para que el usuario sólo tenga que colocar un asunto y el texto que quiera enviar como mensaje.

**Tabla 15 - Historia de usuario del contacto de usuario mediante correo, elaboración propia**

<b>Historia de usuario #6</b>	
<b>Nombre:</b>	Contactar usuario del reporte mediante llamada telefónica
<b>Usuario:</b>	Alumno
<b>Descripción:</b>	Verificar que la función de contactar al usuario que hizo el reporte de objeto perdido mediante llamada telefónica el apartado de objetos perdidos sea correcto.
<b>Observaciones:</b>	Al tocar el elemento de contacto mediante teléfono redirige a la aplicación de llamadas del teléfono, una vez dentro, comienza a marcar a quien hizo el reporte para que pueda entablar una conversación.

**Tabla 16 - Historia de usuario del contacto de usuario mediante llamada telefónica, elaboración propia**

<b>Historia de usuario #7</b>	
<b>Nombre:</b>	Entrega de objeto perdido
<b>Usuario:</b>	Alumno
<b>Descripción:</b>	Comprobar que al entregar el objeto perdido al alumno propietario, el cambiar el estatus de “no entregado” a “entregado” sea exitoso.
<b>Observaciones:</b>	Se pudo validar y se vio reflejado el cambio de estatus exitosamente.

**Tabla 17 - Historia de usuario de entrega de objeto perdido, elaboración propia**

<b>Historia de usuario #8</b>	
<b>Nombre:</b>	Editar datos de contacto
<b>Usuario:</b>	Alumno
<b>Descripción:</b>	Al intentar cambiar los datos del alumno (correo y teléfono) sea de una manera correcta y exitosa.
<b>Observaciones:</b>	Al momento de cambiar correo y teléfono son validados los datos y, si son correctos, realiza el cambio de los datos y se reflejan con éxito en la vista de configuración.

**Tabla 18 - Historia de usuario de editar datos de contacto del usuario, elaboración propia**

<b>Historia de usuario #9</b>	
<b>Nombre:</b>	Editar contraseña
<b>Usuario:</b>	Alumno
<b>Descripción:</b>	Que el cambio de contraseña, por parte del usuario, sea de manera correcta y permita el inicio de sesión sin ningún inconveniente.
<b>Observaciones:</b>	Al momento de realizar el cambio, se solicita que se introduzca la contraseña dos veces antes de continuar, si no coinciden,

	no permite el cambio, una vez coincidiendo y que tenga una composición válida, el cambio de contraseña es exitoso.
--	--

**Tabla 19 - Historia de usuario de editar contraseña, elaboración propia**

Historia de usuario #10	
<b>Nombre:</b>	Solución de reporte de incidencia
<b>Usuario:</b>	Administrador
<b>Descripción:</b>	Al cambiar el estatus de un reporte de incidencia, previamente solucionado por completo, sea de manera exitosa y se visualice correctamente.
<b>Observaciones:</b>	El cambio de estatus se ve reflejado de manera exitosa.

**Tabla 20 - Historia de usuario de solución de reporte de incidencia, elaboración propia**

### 6.1.2 Pruebas de conjetura de fallos

Para esto, se evaluaron algunas posibles fallas que pudieran encontrarse en el sistema, que no hayan sido consideradas al momento del desarrollo.

Área crítica	Suposición de fallo	Observaciones
Registro de usuario	Matrícula incorrecta (longitud inválida o no compuesta únicamente de números).	La longitud es validada a 9 dígitos y verifica que solo se encuentre compuesta de números. Cualquier otro caso muestra mensaje de error.
Registro de usuario	Correo inválido (composición de correo no válida, ejemplo: "juanito", "a@a.a" y similares).	La composición del correo debe de ser al menos "xxx@xxx.xx" (usuario debe tener una longitud de 3 caracteres, nombre de servidor de 3 y dominio de nivel superior

		mínimo de 2) en cualquier otro caso muestra un mensaje de error.
Registro de usuario	Teléfono no válido (longitud inválida o no compuesta únicamente de números).	Se verifica que la longitud sea sí o sí de 10 caracteres además de una composición exclusiva de números, en cualquier otro caso muestra un mensaje de error.
Registro de usuario	Contraseña no válida (longitud menor a 8 y no compuesta por al menos una mayúscula, una minúscula, un número y un carácter especial).	El sistema valida los datos requeridos, en cualquier otro caso muestra un mensaje de error.
Registro de usuario	Matrícula registrada (el usuario intenta crear una cuenta con una matrícula registrada en el sistema).	Si la matrícula ya se encuentra registrada, la aplicación le muestra una advertencia al usuario que la contraseña ya ha sido registrada anteriormente.
Inicio de sesión	Contraseña incorrecta (el usuario intenta acceder con una contraseña similar y logra acceder ejemplo: "#juanito1" en lugar de su contraseña registrada "#Juanito1").	Al intentar iniciar sesión con una contraseña similar, no permite, forzosamente se requiere que la contraseña sea idéntica al momento del registro, una mayúscula o

		minúscula equivocada no permite el acceso.
Envío de reportes (incidencia y objetos perdidos)	Demasiada carga de información al momento de enviar el reporte (demasiado texto, además de evidencias fotográficas con la máxima calidad del dispositivo).	Al momento de una primera prueba, se realizó con máxima calidad de una imagen de 64mpx y <i>Firestore</i> la rechazó, se comprimió la imagen en la aplicación antes de enviar (transparente para el usuario) reduciendo la calidad para un correcto envío y, en una segunda prueba, se logró enviar correctamente. También, se forzó con grandes cantidades de texto y la aplicación no mostró ningún problema.
Listado de reportes de incidencia	El alumno pueda ver reportes de incidencia que no haya realizado.	Se muestran correctamente los reportes realizados por el usuario, se realizó prueba con reportes de prueba que no fueran del usuario y no se visualizaban, por lo tanto, es correcta la funcionalidad.
Configuración de la cuenta	Cambio de datos de contacto (correo o	Al momento de intentar cambiar los datos de

	teléfono) no válidos, es decir, que al cambiar los datos, acepte valores inválidos.	contacto, son validados antes de ser enviados, si el dato es incorrecto, vuelve a solicitar la información hasta ser satisfactoria.
Configuración de la cuenta	Contraseña inválida, es decir, que no contenga al menos una mayúscula, una minúscula, un número y un carácter especial.	Al momento de intentar modificar, la contraseña es validada, si no contiene la composición requerida, vuelve a solicitar la información, en caso de ser correcta, se modifica.
Detalles de objeto perdido	Matrícula inválida al intentar devolver un objeto perdido.	Al momento de verificar la contraseña dada si no es exactamente de 9 caracteres y es exclusivamente compuesta de números, muestra un mensaje de error y no permite el cambio de estatus de entrega, en caso contrario, con una matrícula correcta, realiza el cambio de estatus correctamente.

*Tabla 21 - Bitácora de pruebas de conjetura de fallos, elaboración propia*

### 6.1.3 Pruebas heurísticas

Usando las reglas heurísticas que fueron definidas por su autor *Jakob Nielsen* y con las cuales se evaluó el sistema, se obtuvieron los siguientes resultados:

### **1. Visibilidad del estado del sistema**

En todo momento se muestra información de lo que sucede. Al cargar la información se muestran pantallas de carga y, al momento de realizar una solicitud de información, muestra un mensaje de envío correcto o de error.

### **2. Relación entre el sistema y el mundo real**

Los iconos son relacionados con los textos, además de identificar qué acción realizan.

### **3. Control y libertad para el usuario**

Mediante advertencias al usuario, si se está seguro de realizar una acción, se le informa al usuario lo que está a punto de realizar, pero, una vez aceptado, no hay una forma de corregir lo que hizo.

### **4. Consistencia y estándares**

En todo momento, existe una consistencia de la barra de navegación superior, exceptuando el inicio, pero se conserva el contexto dado, que utiliza una imagen de un edificio de la facultad, que los alumnos identifican correctamente.

### **5. Prevención de errores**

En cuanto a la prevención de errores, se realiza correctamente, por ejemplo, en los formularios de reportes, si no existe una conexión a internet, no permite enviar los formularios y muestra una advertencia al usuario de que necesita conectarse a internet, además de que los formularios necesitan completarse para poder enviarse y no permite enviar datos incompletos.

### **6. Minimizar la carga de memoria del usuario**

Se cumple, puesto que para llegar a una vista de la aplicación en específico, no hay necesidad de dar saltos, el menú lateral lleva concisamente a donde el usuario quiere llegar.

## **7. Flexibilidad y eficiencia de uso**

Al ser una aplicación móvil, no se dispone de un teclado físico en algunos dispositivos, pero ofrece opciones de navegación, tanto en la pantalla de inicio como en el menú lateral para desplazarse por las distintas pantallas.

## **8. Diseño estético y minimalista**

La información en pantalla es exactamente lo que se necesita para poder interactuar con la sección mostrada, por ejemplo, en el apartado de nuevo reporte de incidencia, no se muestra el listado de reportes y sólo muestra el formulario para realizar el reporte, además de que no hay imágenes o textos que distraigan al usuario.

## **9. Ayudar a los usuarios con los errores**

Los mensajes de error son claros y en lenguaje informal, pero en algunos puntos no se muestra una forma de solucionarlo.

## **10. Ayuda y documentación**

Para esta versión, no se encuentra un apartado de preguntas frecuentes o una documentación que ayude al usuario de cómo usar la aplicación, la solución se tomará como un trabajo a futuro.

# CAPÍTULO VII: CONCLUSIONES Y TRABAJO A FUTURO

## 7.1 Conclusiones

Al momento de realizar la captura de requerimientos del proyecto, se visualiza hasta dónde se quería cubrir el manejo de incidencias y objetos perdidos en la facultad de una manera digital, mediante la aplicación y dar por alcanzado el objetivo general, que pretendía dar un seguimiento y obtener una solución más rápida. Para lograr esto se llevará a cabo a través de una red de usuarios que usen la aplicación y podrán ser consultados en tiempo real desde un teléfono móvil, sin necesidad de ir revisando reporte a reporte recopilados en papel manualmente como se hace actualmente.

Durante el desarrollo, se decidieron las herramientas tecnológicas que permitirían un proceso más rápido y que cubrieran las necesidades que se habían recopilado y, aunque existen otras maneras de resolver el proyecto utilizando otros *frameworks* o lenguajes de desarrollo, usar una tecnología híbrida complementada de *Firebase* cumplió con lo que se esperaba en el menor tiempo posible.

Las características esenciales de la aplicación desarrollada son:

- Permite enviar reportes de incidencias y objetos perdidos de forma sencilla a través de formularios digitales.
- Poder visualizar un listado digital de los reportes de incidencias enviados por los alumnos en tiempo real, además de poder filtrar el listado de reportes mediante solucionados y no solucionados.
- Tener un registro de objetos perdidos que puede ser consultado en cualquier momento desde la aplicación por cualquier alumno.

Otros resultados que podrían obtenerse con un uso constante y un mantenimiento correcto generarían un gran impacto en el tiempo de solución de incidencias, sustituyendo la forma actual de generar reportes, además que una buena empatía entre alumnos haría del apartado de objetos perdidos una manera muy útil de reportar y recuperar objetos generando un ambiente más agradable en la comunidad escolar.

En conclusión, este proyecto y su investigación ha sido enriquecedora y gratificante y han complementado mis conocimientos que tenía antes de realizar el trabajo de tesina, espero que pueda servir como punto de partida u orientar a quienes estén interesados en temas relacionados a este.

## 7.2 Trabajo a futuro

El primer paso, antes de planear un trabajo a futuro, es difundir la aplicación y que sea utilizada por Secretaría Académica para ofrecerla a los alumnos y valorar un primer recibimiento. Si la respuesta fuese negativa, se tomarían en cuenta los comentarios de los alumnos para retroalimentar la aplicación y encaminarla a que sea recibida con una mejor aceptación.

En caso de ser bien recibida, el siguiente paso sería que la aplicación no solo se encuentre disponible para el sistema operativo *Android*, sino que pueda ser instalable en *iOS* o inclusive volverse una *PWA* para que pueda ser descargada a través de una tienda de aplicaciones, o sino que pueda usarse desde cualquier navegador.

También, como se mencionó en el capítulo de pruebas, se desea incorporar una sección de preguntas frecuentes y ayuda al usuario, así como incluir anotaciones que puedan guiar a los alumnos en caso de tener dudas de cómo realizar reportes.

Además de esto, a la aplicación se le pueden agregar nuevas funcionalidades e ir incrementando sus características poco a poco e inclusive se puede convertir en la aplicación oficial de la Facultad de Ciencias de la Computación.

Por ejemplo algunas características que se le podrían agregar serían:

1. Estar vinculada con las redes sociales (*Facebook*, *Twitter*) y al momento de realizar una publicación que la aplicación consuma mediante una *api* (*interfaz de programación de aplicaciones*) estos datos y los muestre directamente a los usuarios de la aplicación.

2. Añadir un apartado de cursos, conferencias, certificaciones, etc. Donde los alumnos puedan estar mejor enterados y puedan acceder más rápidamente que desde una red social o desde los sitios web de la Facultad.
3. Integrar una sección donde se encuentre el directorio de la facultad y aprovechando las funcionalidades de la aplicación realizada pueda ser más fácil enviar correos o marcar los números telefónicos de los docentes.

## Bibliografía

Ayuntamiento de San Andrés Cholula. (2022). SACH Móvil (versión 1.6.7).

Descargado el 05 de Enero de 2023, de:  
<https://play.google.com/store/apps/details?id=io.ibesgob.sach.appciudadana&gl=US>

Gobierno de la Ciudad de México. (2019). App CDMX (versión 3.0.17).

Descargado el 02 de Enero de 2023, de:  
<https://play.google.com/store/apps/details?id=mx.gob.cdmx.adip.apps&gl=US>

Benemérita Universidad Autónoma de Puebla. (s.f.). Colores Institucionales.

Recuperado el 23 de Abril de 2023, de:  
<https://comunicacion.buap.mx/?q=content/sistema-de-color#:~:text=Los%20colores%20institucionales%20son%20dos%20azules%20Pantone%20306,colores%20puede%20combinarse%20con%20blanco%20C%20grises%20y%20negro.>

Mobile Security Framework. (s.f). Mobile Security Framework.

Recuperado el 10 de Junio de 2023, de: <https://mobsf.live/>

El Universal. (22 de Diciembre de 2017). *El Universal*. Recuperado el 31 de Enero de 2023, de <https://www.eluniversal.com.mx/nacion/sociedad/emiten-recomendacion-sep-y-gcdmx-por-escuelas-en-mal-estado>

Fang, C. (05 de Octubre de 2018). *La evaluación heurística*. Recuperado el 01 de Junio de 2023, de Prototypr: <https://blog.prototypr.io/evaluacion-heuristica-9c8fce655759>

Google Developers. (s.f.). *Cloud Firestore*. Recuperado el 22 de Marzo de 2023, de <https://firebase.google.com/docs/firestore?hl=es-419>

Google Developers. (s.f.). *Encriptación en reposo predeterminada*. Recuperado el 23 de Abril de 2023, de <https://cloud.google.com/docs/security/encryption/default-encryption?hl=es-419>

Google Developers. (s.f.). *Encriptación en tránsito*. Recuperado el 23 de Abril de 2023, de <https://cloud.google.com/docs/security/encryption-in-transit?hl=es-419>

Google Developers. (s.f.). *Reglas de seguridad y Firebase Authentication*. Recuperado el 22 de Abril de 2023, de <https://firebase.google.com/docs/rules/rules-and-auth?hl=es-419>

Hill, M. (6 de Julio de 2021). *Rainbow tables explained: How they work and why they're (mostly) obsolete*. Recuperado el 23 de Abril de 2023, de CSO Online: <https://www.csoonline.com/article/3623195/rainbow-tables-explained-how-they-work-and-why-theyre-mostly-obsolete.html>

IBM. (s.f.). Recuperado el 14 de Junio de 2023, de ¿Qué es una prueba de software?: <https://www.ibm.com/mx-es/topics/software-testing>

IBM. (s.f.). *Protección para sus datos*. Recuperado el 24 de Abril de 2023, de <https://www.ibm.com/mx-es/topics/encryption>

IBM. (s.f.). *What is data security?* Recuperado el 09 de Abril de 2023, de <https://www.ibm.com/topics/data-security>

INEGI. (30 de Noviembre de 2021). *Encuesta Nacional sobre Disponibilidad y Uso de Tecnologías de la Información en los Hogares*. Recuperado el 30 de Enero de 2023, de [https://www.inegi.org.mx/contenidos/programas/dutih/2021/tabulados/2021\\_u2ed691.xlsx](https://www.inegi.org.mx/contenidos/programas/dutih/2021/tabulados/2021_u2ed691.xlsx)

INEGI. (22 de Junio de 2021). *Sala de Prensa del Instituto Nacional de Estadística y Geografía*. Recuperado el 30 de Enero de 2023, de <https://www.inegi.org.mx/app/saladeprensa/noticia.html?id=6606>

- Ionic Team. (01 de Marzo de 2014). *Ionic Framework Documentation*. Recuperado el 28 de Febrero de 2023, de <https://ionicframework.com/docs/>
- Jatana, N., Puri, S., Ahuja, M., Kathuria, I., & Gosain, D. (6 de Agosto de 2012). A Survey and Comparison of Relational and Non-Relational Database. *International Journal of Engineering Research & Technology (IJERT)*, 1-5.
- Jiménez, R. E. (01 de Diciembre de 2015). Metodologías Ágiles de Desarrollo de Software Aplicadas a la Gestión de Proyectos Empresariales. *REVISTA TECNOLÓGICA N° 8(6)*, 6. Recuperado el 03 de Enero de 2023, de <http://redicces.org.sv/jspui/bitstream/10972/2917/1/Articulo1.pdf>
- LUCERO, L. R. (22 de Febrero de 2022). *Transformación digital en México*. Recuperado el 18 de Enero de 2023, de El Herald: <https://heraldodemexico.com.mx/nacional/2022/2/22/transformacion-digital-en-mexico-gobierno-electronico-aun-insuficiente-380816.html>
- Maldonado, M. (12 de Abril de 2018). *Las mejores metodologías ágiles para la creación de software*. Recuperado el 30 de Enero de 2023, de Digital55: <https://www.digital55.com/desarrollo-tecnologia/mejores-metodologias-agiles-creacion-software/>
- National Institute of Standards and Technology. (s.f.). *Software Vulnerability*. Recuperado el 23 de Abril de 2023, de [https://csrc.nist.gov/glossary/term/software\\_vulnerability](https://csrc.nist.gov/glossary/term/software_vulnerability)
- Nielsen, J. (15 de Noviembre de 2020). Recuperado el 2023 de Junio de 14, de 10 Usability Heuristics for User Interface Design: <https://www.nngroup.com/articles/ten-usability-heuristics/>
- Packetizer, Inc. (s.f.). *AES Information*. Recuperado el 22 de Abril de 2023, de [https://www.aescrypt.com/aes\\_information.html](https://www.aescrypt.com/aes_information.html)
- Parmar, D. (s.f.). *Pruebas exploratorias*. Recuperado el 30 de Mayo de 2023, de Atlassian: <https://www.atlassian.com/es/continuous-delivery/software-testing/exploratory-testing>

Rendón, Y. A. (28 de Mayo de 2019). *Bases de datos relacionales vs. no relacionales*. Recuperado el 1 de Febrero de 2023, de Academia Pragma: <https://www.pragma.com.co/academia/lecciones/bases-de-datos-relacionales-vs.-no-relacionales>

S'Arreplec. (s.f.). *Conjetura de errores*. Recuperado el 31 de Mayo de 2023, de S'Arreplec: [https://sarreplec.caib.es/pluginfile.php/11333/mod\\_resource/content/8/ED03\\_Contenidos/a3\\_conjetura\\_de\\_errores.html#:~:text=La%20experiencia%20en%20la%20fase%20de%20pruebas%20indica,prueba%20a%20los%20que%20someter%20a%20los%20programas.](https://sarreplec.caib.es/pluginfile.php/11333/mod_resource/content/8/ED03_Contenidos/a3_conjetura_de_errores.html#:~:text=La%20experiencia%20en%20la%20fase%20de%20pruebas%20indica,prueba%20a%20los%20que%20someter%20a%20los%20programas.)

Secretaría de la Función Pública. (09 de Diciembre de 2013). *Gobierno Digital o Electrónico*. Recuperado el 5 de Enero de 2023, de <https://www.gob.mx/sfp/documentos/gobierno-digital-o-electronico>

Senado de la República. (2012). Recuperado el 23 de Abril de 2023, de [https://www.senado.gob.mx/comisiones/gobernacion/docs/proteccion\\_datos/Documento.pdf](https://www.senado.gob.mx/comisiones/gobernacion/docs/proteccion_datos/Documento.pdf)

Soucoup, M., & Buck, A. (21 de Septiembre de 2022). *Elección de un marco de desarrollo de aplicaciones móviles*. Recuperado el 1 de Febrero de 2023, de Microsoft learn: <https://learn.microsoft.com/es-es/azure/developer/mobile-apps/choose-mobile-framework>

w3c. (23 de Junio de 2016). *Cryptographic hash functions*. Recuperado el 23 de Abril de 2023, de <https://www.w3.org/TR/SRI/#cryptographic-hash-functions>