



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL**

**CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES E
INFORMÁTICOS**

Tema:

**APLICACIÓN MÓVIL DE APOYO A LA SEGURIDAD BARRIAL PARA
ENVÍO Y LOCALIZACIÓN DE ALERTAS DE AUXILIO MEDIANTE
NOTIFICACIONES PUSH EN LA PARROQUIA SANTA ROSA DE LA
CIUDAD DE AMBATO.**

Trabajo de Titulación Modalidad: Proyecto de Investigación, presentado previo
la obtención del título de Ingeniero en Sistemas Computacionales e Informáticos

ÁREA: Desarrollo de Software

LÍNEA DE INVESTIGACIÓN: Aplicaciones para dispositivos móviles

AUTOR: Christian Paul Chasi Chango

TUTOR: Ing. PhD. Julio Enrique Balarezo López.

Ambato – Ecuador

marzo - 2022

APROBACIÓN DEL TUTOR

En calidad de tutor del Trabajo de Titulación con el tema: APLICACIÓN MÓVIL DE APOYO A LA SEGURIDAD BARRIAL PARA ENVÍO Y LOCALIZACIÓN DE ALERTAS DE AUXILIO MEDIANTE NOTIFICACIONES PUSH EN LA PARROQUIA SANTA ROSA DE LA CIUDAD DE AMBATO , desarrollado bajo la modalidad de Proyecto de Investigación por el señor Christian Paul Chasi Chango, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 15 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y el numeral 7.4 del respectivo instructivo.

Ambato, marzo 2022.

Ing. PhD. Julio Enrique Balarezo López

TUTOR

AUTORIA

El presente Proyecto de Investigación titulado: APLICACIÓN MÓVIL DE APOYO A LA SEGURIDAD BARRIAL PARA ENVÍO Y LOCALIZACIÓN DE ALERTAS DE AUXILIO MEDIANTE NOTIFICACIONES PUSH EN LA PARROQUIA SANTA ROSA DE LA CIUDAD DE AMBATO es absolutamente original, auténtico y personal. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, marzo 2022.



Christian Paul Chasi Chango

C.C. 1804898755

AUTOR

APROBACIÓN TRIBUNAL DE GRADO

En calidad de par calificador del Informe Final del Trabajo de Titulación presentado por el señor Christian Paul Chasi Chango, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado **APLICACIÓN MÓVIL DE APOYO A LA SEGURIDAD BARRIAL PARA ENVÍO Y LOCALIZACIÓN DE ALERTAS DE AUXILIO MEDIANTE NOTIFICACIONES PUSH EN LA PARROQUIA SANTA ROSA DE LA CIUDAD DE AMBATO**, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 17 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y al numeral 7.6 del respectivo instructivo. Para cuya constancia suscribimos, conjuntamente con la señora Presidenta del Tribunal.

Ambato, marzo 2022.

Ing. Pilar Urrutia, Mg.

PRESIDENTA DEL TRIBUNAL

Ing. Hernán Naranjo, Mg.

PROFESOR CALIFICADOR

Ing. David Guevara, Mg.

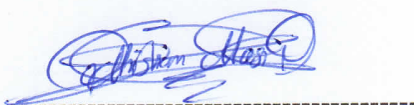
PROFESOR CALIFICADOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación en favor de la Universidad Técnica de Ambato, con fines de difusión pública. Además, autorizo su reproducción total o parcial dentro de las regulaciones de la institución.

Ambato, marzo 2022.



Christian Paul Chasi Chango

C.C. 1804898755

AUTOR

DEDICATORIA

A mi mamita Mari y papá Jorge, quienes con su esfuerzo, sacrificio y amor me han apoyado en todos los sentidos para alcanzar esta meta, guiándome en cada uno de mis pasos durante este largo camino. Son mis padres mi más grande orgullo y mi mayor inspiración.

A mis abuelitos, Francisco y María, mis segundos padres que me brindaron su cariño, apoyo y amor incondicional. Son ellos quienes con respeto y humildad me formaron para ser un hombre de grandes valores.

A Pame y Jessi que me acompañaron durante todos estos largos años de arduo trabajo y momentos difíciles mostrándome su cariño, sus consejos y dándome ánimos hasta alcanzar mi objetivo.

A mi familia en general y a esa personita especial que es un pilar importante en mi vida y en mi corazón.

AGRADECIMIENTO

A Dios, por darme una vida llena de bendiciones y lecciones que me sirvieron para culminar esta meta junto a mis seres queridos.

A mis padres que no permitieron rendirme y me han dado lo más valioso de la vida, su amor.

A mi familia, quienes en los buenos y malos momentos fueron mi fortaleza y mi motivación. Agradezco por inculcarme los valores que hoy en día me caracterizan.

A mis docentes, por impartirme sus conocimientos y guiarme por este camino formándome como un buen profesional.

A mi tutor el Ing. Julio Balarezo que además de docente ejemplar demostró ser un gran amigo dentro y fuera de la institución, ayudándome en este camino con sus consejos y conocimientos que me permitieron culminar mi proyecto de investigación.

A mis amigos, con quienes compartí los mejores momentos durante esta gran etapa de mi vida llena de aventuras y grandes vivencias que hicieron pleno mi paso por la Universidad.

INDICE GENERAL DE CONTENIDOS

| | |
|--|-------|
| APROBACIÓN DEL TUTOR..... | ii |
| AUTORIA..... | iii |
| APROBACIÓN TRIBUNAL DE GRADO..... | iv |
| DERECHOS DE AUTOR..... | v |
| DEDICATORIA..... | vi |
| AGRADECIMIENTO..... | vii |
| RESÚMEN EJECUTIVO..... | xviii |
| ABSTRACT..... | xix |
| CAPITULO I: MARCO TEÓRICO..... | 1 |
| 1.1. Tema de Investigación..... | 1 |
| 1.2. Antecedentes investigativos..... | 1 |
| 1.2.1. Contextualización del problema..... | 3 |
| 1.2.2. Fundamentación Teórica..... | 5 |
| 1.3. Objetivos..... | 18 |
| 1.3.1. Objetivo General..... | 18 |
| 1.3.2. Objetivos Específicos..... | 18 |
| CAPITULO II: METODOLOGÍA..... | 19 |
| 2.1. Materiales..... | 19 |
| 2.1.1. Humanos..... | 19 |
| 2.1.2. Institucionales..... | 19 |
| 2.1.3. Otros..... | 19 |
| 2.2. Métodos..... | 20 |
| 2.2.1. Modalidad de la investigación..... | 20 |
| 2.2.2. Población y muestra..... | 20 |

| | | |
|--|---|-----|
| 2.2.3. | Recolección de la información..... | 21 |
| 2.2.4. | Procesamiento y Análisis de Datos | 22 |
| 2.2.5. | Análisis e Interpretación de los resultados..... | 31 |
| CAPITULO III resultados y discusión..... | | 32 |
| 3.1. | Análisis y discusión de los resultados. | 32 |
| 3.1.1. | Entes que participan en los planes de control y seguridad barrial | 32 |
| 3.1.2. | Metodologías de Desarrollo | 33 |
| 3.1.3. | Servicios de Notificaciones Push | 42 |
| 3.1.4. | Análisis de tecnologías Frontend | 48 |
| 3.1.5. | Análisis de tecnologías Backend..... | 56 |
| 3.1.6. | Análisis de Bases de Datos | 60 |
| 3.2. | Desarrollo de la propuesta | 67 |
| 3.2.1. | Propuesta general del funcionamiento del sistema y sus actores. | 67 |
| 3.2.2. | Definición del modelo de proceso..... | 69 |
| 3.2.3. | Fase I: Exploración | 69 |
| 3.2.4. | Fase II: Inicialización..... | 72 |
| 3.2.5. | Fase III: Producción | 93 |
| 3.2.6. | Fase IV: Estabilización | 122 |
| 3.2.7. | Fase V: Pruebas..... | 123 |
| 3.2.8. | Publicación en la Tienda de aplicaciones..... | 136 |
| CAPITULO IV Conclusiones y Recomendaciones | | 137 |
| Bibliografía | | 139 |
| ANEXOS..... | | 144 |
| 3.3. | Anexo 1 | 144 |
| 3.4. | Anexo 2 | 148 |
| 3.5. | Anexo 3 | 155 |
| 3.6. | Anexo 4 | 155 |

INDICE DE TABLAS

| | |
|--|----|
| Tabla 1.2.1 Ventajas y Desventajas de las aplicaciones web móviles | 6 |
| Tabla 1.2.2 Ventajas y Desventajas de las aplicaciones web nativas | 7 |
| Tabla 1.2.3 Ventajas y Desventajas de las aplicaciones híbridas | 8 |
| Tabla 1.2.4 Comparativa aplicaciones web, nativas e híbridas..... | 9 |
| Tabla 1.2.5 Comparativa de aplicaciones tradicionales y ágiles..... | 13 |
| Tabla 1.2.6 Métodos de generación y recolección de datos de geolocalización..... | 14 |
| Tabla 2.1.1 Recursos económicos..... | 20 |
| Tabla 2.2.1 Población y muestra | 21 |
| Tabla 2.2.2 Recolección de la Información | 22 |
| Tabla 2.2.3 Nivel de seguridad en el barrio | 22 |
| Tabla 2.2.4 Sistemas de Seguridad en el Hogar..... | 23 |
| Tabla 2.2.5 Sistemas de seguridad comunitaria..... | 24 |
| Tabla 2.2.6 Su barrio es organizado y colaborativo..... | 25 |
| Tabla 2.2.7 De acuerdo que los barrios se protejan | 26 |
| Tabla 2.2.8 Frecuencia de delitos en el barrio | 27 |
| Tabla 2.2.9 Delitos más frecuentes en el barrio | 27 |
| Tabla 2.2.10 Importancia de un botón de pánico | 28 |
| Tabla 2.2.11 Tecnología ayuda a combatir la inseguridad..... | 29 |
| Tabla 2.2.12 Mejorar la seguridad del barrio | 30 |
| Tabla 3.1.1 Roles XP | 36 |
| Tabla 3.1.2 Roles Scrum | 38 |
| Tabla 3.1.3 Comparativa metodologías ágiles | 41 |
| Tabla 3.1.4 Ventajas y desventajas de las metodologías ágiles | 42 |
| Tabla 3.1.5 Comparativa de Servicios Notificaciones Push | 47 |
| Tabla 3.1.6 Ventajas y desventajas con proveedores de servicios push | 48 |
| Tabla 3.1.7 Ventajas y desventajas de Ionic | 50 |

| | |
|--|----|
| Tabla 3.1.8 Ventajas y desventajas de React Native..... | 52 |
| Tabla 3.1.9 Ventajas y Desventajas de Flutter..... | 54 |
| Tabla 3.1.10 Comparativa frameworks de aplicaciones móviles..... | 55 |
| Tabla 3.1.11 Ventajas y desventajas de Laravel | 57 |
| Tabla 3.1.12 Ventajas y desventajas de Node.js y Express | 58 |
| Tabla 3.1.13 Ventajas y desventajas de Ruby on Rails..... | 58 |
| Tabla 3.1.14 Comparativa Bases de Datos relacionales | 62 |
| Tabla 3.1.15 Diferencias entre SQL y No SQL | 63 |
| Tabla 3.1.16 Comparativa de costos y servicios de Bases de Datos en línea | 66 |
| Tabla 3.2.1 Descripción de la propuesta general del diagrama..... | 69 |
| Tabla 3.2.2 Tecnologías para el desarrollo del proyecto – Fase I..... | 72 |
| Tabla 3.2.3 Requerimientos funcionales aplicación móvil – Barrio Seguro..... | 78 |
| Tabla 3.2.4 Requerimientos no funcionales | 79 |
| Tabla 3.2.5 Otros Requerimientos | 80 |
| Tabla 3.2.6 Módulos de la aplicación móvil – Barrio Seguro | 81 |
| Tabla 3.2.7 Módulos aplicación móvil – Monitoreo de alertas..... | 82 |
| Tabla 3.2.8 Plantilla de una historia de usuario | 83 |
| Tabla 3.2.9 Historia de usuario – Registro de usuarios..... | 84 |
| Tabla 3.2.10 Historia de usuario – Autenticación de usuario | 84 |
| Tabla 3.2.11 Historia de usuario –Alertas de auxilio..... | 85 |
| Tabla 3.2.12 Historia de usuario – Autenticación de usuario con rol ciudadano..... | 86 |
| Tabla 3.2.13 Historia de usuario – Historial de alertas | 86 |
| Tabla 3.2.14 Historia de usuario – Contactos de emergencia | 86 |
| Tabla 3.2.15 Historia de usuario – Registro de usuarios con rol policía | 87 |
| Tabla 3.2.16 Historia de usuario – Autenticación de usuarios..... | 87 |
| Tabla 3.2.17 Historia de usuario – Gestión de alertas de auxilio..... | 88 |
| Tabla 3.2.18 Historia de usuario – Autenticación de usuario con rol policía | 88 |

| | |
|---|-----|
| Tabla 3.2.19 Planificación de Fases – Fase II..... | 90 |
| Tabla 3.2.20 Story Card – Registros de usuarios con rol ciudadano | 93 |
| Tabla 3.2.21 Story Card – Autenticación de usuarios con rol ciudadano | 97 |
| Tabla 3.2.22 Story Card – Envío de alertas de auxilio..... | 100 |
| Tabla 3.2.23 Story Card – Contactos de emergencia | 105 |
| Tabla 3.2.24 Story Card – Historial de alertas | 108 |
| Tabla 3.2.25 Story Card – Perfil de Usuario..... | 110 |
| Tabla 3.2.26 Story Card – Registro de usuarios con rol policía | 113 |
| Tabla 3.2.27 Story Card – Autenticación de usuarios con rol policía..... | 116 |
| Tabla 3.2.28 Story Card – Alertas de auxilio..... | 119 |
| Tabla 3.2.29 Story Card – Configuración y perfil de usuario..... | 120 |
| Tabla 3.2.30 Prueba Unitaria – Validación de formularios | 124 |
| Tabla 3.2.31 Prueba Unitaria – Mensaje de aviso de alerta de auxilio | 125 |
| Tabla 3.2.32 Prueba Unitaria – Ubicación mediante GPS | 125 |
| Tabla 3.2.33 Prueba Unitaria – Sonidos de las notificaciones push | 126 |
| Tabla 3.2.34 Prueba Unitaria – Escritura y lectura de información en el dispositivo | 127 |
| Tabla 3.2.35 Prueba Unitaria – Restablecer contraseña..... | 128 |
| Tabla 3.2.36 Prueba Unitaria – Recepción de la notificación push | 128 |
| Tabla 3.2.37 Resumen de Pruebas Unitarias..... | 129 |
| Tabla 3.2.38 Prueba de aceptación – Registro de Usuarios con rol ciudadano | 129 |
| Tabla 3.2.39 Prueba de aceptación – Autenticación de usuarios con rol ciudadano | 130 |
| Tabla 3.2.40 Prueba de aceptación – Alertas de auxilio | 131 |
| Tabla 3.2.41 Prueba de aceptación – Perfil de usuario | 132 |
| Tabla 3.2.42 Prueba de aceptación – Historial de alertas | 132 |
| Tabla 3.2.43 Prueba de aceptación – Contactos de emergencia | 133 |
| Tabla 3.2.44 Prueba de aceptación – Registro de Usuarios con rol policía..... | 134 |

| | |
|--|-----|
| Tabla 3.2.45 Prueba de aceptación – Autenticación de usuarios con rol policía..... | 134 |
| Tabla 3.2.46 Prueba de aceptación – Alertas de auxilio | 135 |
| Tabla 3.2.47 Prueba de aceptación – Perfil de usuario | 135 |

INDICE DE GRÁFICOS

| | |
|---|----|
| Figura 1.2.1 Arquitectura APIs | 11 |
| Figura 1.2.2 Formatos XML y JSON..... | 11 |
| Figura 1.2.3 Diferencias entre Servicios web y APIs | 12 |
| Figura 1.2.4 Proceso de envío de notificaciones push | 17 |
| Figura 2.2.1 Resultados Nivel de Seguridad en el barrio..... | 22 |
| Figura 2.2.2 Resultados Sistemas Seguridad en el Hogar..... | 23 |
| Figura 2.2.3 Resultados Sistemas de Seguridad..... | 24 |
| Figura 2.2.4 Resultados Barrio organizado y colaborativo..... | 25 |
| Figura 2.2.5 Resultados De acuerdo que los barrios se protejan | 26 |
| Figura 2.2.6 Resultados Frecuencia de delitos en el barrio..... | 27 |
| Figura 2.2.7 Resultados Delitos más frecuentes en el barrio | 28 |
| Figura 2.2.8 Resultados Importancia de un botón de pánico | 29 |
| Figura 2.2.9 Resultados Tecnología ayuda a combatir la inseguridad..... | 30 |
| Figura 2.2.10 Resultados Mejorar la seguridad del barrio | 31 |
| Figura 3.1.1 Ciclo de desarrollo XP | 34 |
| Figura 3.1.2 Fases de la metodología XP..... | 35 |
| Figura 3.1.3 Ciclo de desarrollo Scrum | 37 |
| Figura 3.1.4 Ciclo de desarrollo Mobile-D | 39 |
| Figura 3.1.5 Ciclo de vida de la metodología Mobile-D..... | 40 |
| Figura 3.1.6 Arquitectura Push FCM..... | 43 |
| Figura 3.1.7 Arquitectura Push de OneSignal..... | 44 |
| Figura 3.1.8 Página oficial Wonder Push..... | 45 |
| Figura 3.1.9 Diagrama que representa el funcionamiento de Ionic | 49 |
| Figura 3.1.10 Renderizado WebView vs Renderizado ReactNative | 51 |

| | |
|--|-----|
| Figura 3.1.11 Arquitectura de Flutter..... | 53 |
| Figura 3.1.12 Frameworks más utilizados por desarrollares en el 2021 | 59 |
| Figura 3.1.13 Lenguajes y entornos de desarrollo preferido por los desarrolladores | 60 |
| Figura 3.1.14 Almacenamiento de datos SQL y No SQL..... | 62 |
| Figura 3.1.15 Rendimiento consultas de bases de datos con 10.000 registros | 63 |
| Figura 3.1.16 Rendimiento consultas de bases de datos con 100.000 registros | 63 |
| Figura 3.1.17 Resultados del rendimiento de Bases de Datos | 64 |
| Figura 3.1.18 Bases de Datos preferidas por desarrolladores | 65 |
| Figura 3.2.1 Diagrama general del sistema..... | 68 |
| Figura 3.2.2 Diagrama de procesos del sistema para el envío de alertas de auxilio .. | 69 |
| Figura 3.2.3 Proyecto en OneSignal | 73 |
| Figura 3.2.4 Configuración final de OneSignal | 74 |
| Figura 3.2.5 Proyecto en Firebase..... | 74 |
| Figura 3.2.6 Estructura de directorios de la aplicación Barrio Seguro y Monitoreo de Alertas | 75 |
| Figura 3.2.7 Requerimientos funcionales aplicación móvil – Monitoreo de alertas.. | 79 |
| Figura 3.2.8 Arquitectura de la aplicación | 91 |
| Figura 3.2.9 Base de Datos - Colección ciudadanos | 92 |
| Figura 3.2.10 Base de datos - Colección roles | 92 |
| Figura 3.2.11 Base de Datos - Colección policías | 93 |
| Figura 3.2.12 Iteración 1 – Codificación de registro de usuario (Frontend)..... | 94 |
| Figura 3.2.13 Iteración 1 – Codificación de registro de usuario (Backend) | 95 |
| Figura 3.2.14 Iteración 1 – Interfaz registro de usuario | 96 |
| Figura 3.2.15 Iteración 2 – Codificación de autenticación de usuario (Frontend)..... | 97 |
| Figura 3.2.16 Iteración 2 – Codificación de autenticación de usuario (Backend)..... | 98 |
| Figura 3.2.17 Iteración 2 – Interfaz autenticación de usuario..... | 99 |
| Figura 3.2.18 Iteración 3 – Codificación OneSignal en el proyecto..... | 101 |

| | |
|---|-----|
| Figura 3.2.19 Iteración 3 – Codificación de envío de alertas de auxilio (Frontend) | 101 |
| Figura 3.2.20 Iteración 3 – Codificación de envío de alertas (Backend)..... | 104 |
| Figura 3.2.21 Iteración 3 – Interfaz de envío de alertas de auxilio..... | 105 |
| Figura 3.2.22 Iteración 4 – Codificación contactos de emergencia (Frontend)..... | 106 |
| Figura 3.2.23 Iteración 4 – Codificación contactos de emergencia (Backend) | 107 |
| Figura 3.2.24 Iteración 4 – Interfaz contactos de emergencia | 108 |
| Figura 3.2.25 Iteración 5 – Codificación historial de alertas | 109 |
| Figura 3.2.26 Iteración 5 – Interfaz de historial de notificaciones..... | 110 |
| Figura 3.2.27 Iteración 6 – Codificación del perfil de usuario (Frontend) | 111 |
| Figura 3.2.28 Iteración 6 – Codificación perfil de usuario (Backend)..... | 112 |
| Figura 3.2.29 Iteración 6 – Interfaz perfil de usuario | 113 |
| Figura 3.2.30 Iteración 7 – Codificación de registro de usuario (Frontend)..... | 114 |
| Figura 3.2.31 Iteración 7 – Codificación de registro de usuario (Backend) | 115 |
| Figura 3.2.32 Iteración 7 – Interfaz de registro de usuarios..... | 115 |
| Figura 3.2.33 Iteración 8 – Codificación de autenticación de usuario (Frontend)... | 117 |
| Figura 3.2.34 Iteración 8 – Codificación de autenticación de usuario (Backend)... | 118 |
| Figura 3.2.35 Iteración 8 – Interfaz autenticación de usuario..... | 118 |
| Figura 3.2.36 Iteración 9 – Codificación para mostrar las alertas de auxilio..... | 119 |
| Figura 3.2.37 Iteración 9 – Interfaz alertas de auxilio | 120 |
| Figura 3.2.38 Iteración 10 – Codificación perfil de usuario (Frontend) | 121 |
| Figura 3.2.39 Iteración 10 – Codificación perfil de usuario (Backend)..... | 122 |
| Figura 3.2.40 Iteración 10 – Interfaces de configuración y perfil de usuario | 122 |
| Figura 3.2.41 Aplicaciones publicadas en la Google Play Store | 136 |
| Figura 3.3.1 Exposición del proyecto a los moradores del barrio las Lajas..... | 144 |
| Figura 3.3.2 Socialización de la aplicación con los moradores de La Ciudadela 12 de Noviembre..... | 145 |
| Figura 3.3.3 Aplicación de encuestas a los moradores del barrio Las Lajas | 145 |

| | |
|--|-----|
| Figura 3.3.4 Socialización de la aplicación a los moradores del barrio las Lajas | 145 |
| Figura 3.3.5 Exposición del aplicativo (con el Ing. Balarezo, Tutor del Proyecto) en la policía..... | 146 |
| Figura 3.3.6 Reuniones a la que asiste la policía | 146 |
| Figura 3.3.7 Reunión Virtual con el barrio Guayaquil – Parroquia Sta. Rosa..... | 147 |

RESÚMEN EJECUTIVO

Este proyecto de investigación se enfoca el desarrollo de una aplicación móvil que ayude a acometer una problemática de contexto social, como lo es la inseguridad ciudadana, la cual afecta a varias zonas o barrios de la parroquia Sta. Rosa de la ciudad de Ambato. Los ciudadanos son los más afectados debido a que le resulta difícil realizar sus actividades cotidianas con normalidad ya que siempre están en peligro ellos, sus familiares o sus bienes materiales.

En la actualidad se están dando muchas soluciones a diferentes problemáticas a través de la implementación de sistemas o apps. Un claro ejemplo es el auge de las aplicaciones para dispositivos móviles que están al alcance de muchos usuarios simplificando sus tareas y recursos como tiempo y dinero. En ese sentido es preciso la implementación de una aplicación móvil tipo botón de pánico, enfocada principalmente a los vecinos de los barrios para que puedan alertar entre su comunidad cualquier situación de riesgo o de emergencia. El propósito de este proyecto es apoyar a los barrios de la parroquia Sta. Rosa para que puedan estar preparados y prevenidos ante el aumento de delincuencia. Por ende, estos deben ser organizados, colaborativos y participativos, con el fin de que puedan brindar ayuda al ciudadano en peligro y así dar un mensaje de advertencia a los delincuentes.

Durante el análisis y desarrollo de las tecnologías para la implementación del aplicativo se realizaron comparaciones con el objetivo de determinar las herramientas libres y gratuitas que ayuden a obtener un software de calidad y que evite incurrir en gastos para quienes utilizarán el aplicativo. Las herramientas más importantes que se utilizaron durante el desarrollo fueron Ionic y Angular para el Frontend, Node.js y MongoDB para el Backend, también se utilizó la metodología ágil de desarrollo Mobile-D ya que se adapta fácilmente a este tipo de proyectos que no requieren de muchos módulos ni de mucho tiempo para su implementación.

Palabras clave: Aplicación móvil, frameworks, notificaciones push, seguridad

ABSTRACT

This research project focuses on the development of a mobile application that helps to address a problem of social context, such as citizen insecurity, which affects several areas or neighborhoods of the Parish of Sta. Rosa in the city of Ambato. Citizens are the most affected because it is difficult for them to carry out their daily activities normally since they, their relatives or their material goods are always in danger.

Currently, many solutions are being given to different problems through the implementation of systems or apps. A clear example is the rise of applications for mobile devices that are within reach of many users simplifying their tasks and resources such as time and money. In this sense, it is opportune to implement a mobile application type panic button like the one used by the police, focused mainly on the neighbors of the neighborhoods so that they can alert among the neighborhood community any situation of risk or emergency. The purpose of this project is to support the neighborhoods of The Parish of St. Rose so that they can be prepared and prevented from the increase in crime. Therefore, these must be organized, collaborative and participatory, so that they can provide help to the citizen in danger and thus give a warning message to criminals.

During the analysis and development of the technologies for the implementation of the application, comparisons were made with the aim of determining the free and free tools that help obtain quality software and avoid incurring expenses for those who will use the application. The most important tools that were used during the development were Ionic and Angular for the Frontend, Node.js and MongoDB for the Backend, the agile Mobile-D development methodology was also used since it easily adapts to this type of projects that do not require many modules or a lot of time for their implementation.

Keywords: Mobile app, frameworks, push notifications, security

CAPITULO I: MARCO TEÓRICO

1.1. Tema de Investigación

APLICACIÓN MÓVIL DE APOYO A LA SEGURIDAD BARRIAL PARA ENVÍO Y LOCALIZACIÓN DE ALERTAS DE AUXILIO MEDIANTE NOTIFICACIONES PUSH EN LA PARROQUIA SANTA ROSA DE LA CIUDAD DE AMBATO.

1.2. Antecedentes investigativos

Para apoyar la importancia del presente proyecto de investigación se examinó el repositorio de la Facultad de Ingeniería en Sistemas Electrónica e Industrial y posterior a ello se revisó las bases de datos de los diferentes centros de educación superior, donde se obtuvo información relacionada con el tema y obteniendo como antecedentes investigativos lo siguiente:

Según Lenin Alcívar Gordillo Quizhpe, en su proyecto de titulación denominado “Revisión Sistemática sobre Arquitectura Push en el Análisis de Aplicabilidad de Ambientes Móviles”, trabajo realizado como tesis de la Universidad Nacional de Loja en el año 2019 [7]. Menciona aspectos importantes tales como:

- Las notificaciones push mejoran el proceso de notificación al usuario permitiendo enviar cada notificación en el momento pertinente, sin requerir instalar y ejecutar aplicaciones adicionales.

- Lo que más destaca de las notificaciones push es su inmediatez, ya que no hace falta estar ejecutando la aplicación para que nos llegue. Aunque la tengamos apagada o en segundo plano, cada vez que el servidor reciba una información nueva nos avisará de su existencia, es decir, las notificaciones push despiertan al móvil esté o no ejecutando la aplicación.

- Los resultados revelan que existe un crecimiento considerable desde el año 2016 por investigar esta arquitectura, con tendencias al alza dirigidas a aplicar sus ventajas en ambientes móviles, los artículos revisados demuestran que existe notable estudios y aplicativos que hacen uso de esta solución, como una forma de contar con la información en tiempo real de manera oportuna.

Según Soldado Zuña Alex Fernando, en su proyecto de titulación denominado “Análisis, diseño e implementación de un sistema con notificaciones push para plataforma Android”, trabajo realizado como tesis de la Universidad Central del Ecuador en el año 2018 [8]. Menciona aspectos importantes tales como:

- Al ser la Tecnología Push una herramienta poco conocida y no muy explotada en el país, es importante saber sobre su implementación, y de esta manera fomentar el uso de esta tecnología y así lograr un verdadero avance en el desarrollo de aplicaciones móviles.
- La principal ventaja de esta tecnología es la inmediatez con la que se reciben las notificaciones push. Esto es debido a la posibilidad de enviar y recibir notificaciones de un sistema a otro, tanto si las aplicaciones están siendo usadas en ese mismo instante, si están ejecutándose en segundo plano, si todavía no han sido iniciadas, o incluso si el dispositivo está en modo reposo.

Según Vila Ruiz, en su proyecto de titulación denominado “Centro de gestión de notificaciones Push para dispositivos móviles basados en iOS y android”, trabajo 7 realizado como tesis de la Universidad del País Vasco en el año 2015 [9]. Menciona dos aspectos importantes tales como:

- Las notificaciones Push es un sistema que mediante mensajes cortos realiza la comunicación utilizando los dispositivos móviles, actualmente en el mercado están los teléfonos celulares Android en los cuales recibimos la información.

- Con lo que respecta a los diferentes sistemas operativos de los Smartphone y Tablet la ventaja es que la tecnología push puede ser aplicada sin ningún problema, cada uno tiene una manera distinta de implementar.

1.2.1. Contextualización del problema

A nivel mundial, en las últimas décadas la inseguridad ciudadana ha sido uno de los principales problemas con los que las personas tratan de lidiar. Sobre todo, porque esta aumenta aún con la implementación de leyes y políticas, las mismas que en muchos de los casos son cuestionadas debido a la deficiencia en la lucha por el crimen organizado. Toda esta problemática ha sido evidente en la mayor parte de países tercermundistas quizás por condiciones sociales o políticas lo que no les ha permitido manejarla como corresponde [1].

En América latina la inseguridad sigue prevaleciendo, aunque no todos los países de la región sufren los mismos niveles de victimización. Sin embargo, esta percepción hace hincapié en el desarrollo de diversos mecanismos de protección ciudadana. Adoptando iniciativas de seguridad privada como instalación de cámaras de vigilancia, contratación de servicios de seguridad hasta iniciativas comunitarias como la formación de comités o asociaciones barriales, entre otros [1].

El avance tecnológico en materia de seguridad pública no se ha desarrollado con la misma rapidez o, al menos, con el mismo impacto, pues aún se observa que muchos cuerpos policiales, principalmente de la región centroamericana, siguen utilizando los mismos medios e instrumentos convencionales para prevenir el delito y combatir el crimen [2].

No es el caso de países como México, Argentina, Chile y Uruguay que en los últimos años se han apoyado de la innovación tecnológica para contrarrestar los niveles de inseguridad, impulsados por el desarrollo de sistemas integrados de información, sistemas de información geográfica, cámaras de reconocimientos facial, centros de mando integrados y otros avances tecnológicos que han sido plenamente

aprovechados por los cuerpos policiales para mejorar su actuación en la prevención del delito [2].

Ecuador, un país en proceso de desarrollo no se ha quedado rezagado respecto a tecnología, según las cifras del Instituto Nacional de Estadísticas y Censos (INEC) el analfabetismo digital en el 2020 disminuyó 1.2% a nivel nacional, 3.2% a nivel rural y 0.3% a nivel urbano con relación al año 2019 [3]. Ampliando este contexto estas cifras evidencian el interés de la población por hacer mayor uso de las Tecnologías de la Información y la Comunicación (TIC). El uso correcto de éstas en el ámbito social ayudaría a resolver varios problemas, entre ellos combatir la delincuencia y violencia que afecta a gran parte de los ciudadanos y su estilo de vida [3].

Las cifras de la Fiscalía General del Estado (FGE) con respecto a los delitos más comunes en el periodo 2020 y 2021 se observa un notable aumento en robos a domicilios con el 8.9%, a bienes, accesorios y autopartes de vehículos con el 26.2%, robos correspondientes a personas con el 5.7%, entre otros [4].

En Tungurahua, mediante datos estadísticos del INEC se constata que el índice de prevalencia del delito es de un 18,3 % superando a Pichincha que tiene 17,2% y al resto de provincias. Además la inseguridad en los barrios comprende un 46.4% según una media de personas encuestadas [5]. A su vez existen otros tipos de delitos reportados por el Sistema Integrado de Seguridad ECU 911 en el que se recalca los casos de violencia intrafamiliar indicando que Ambato es el cantón donde más se dan este tipo de llamadas seguido por Pelileo, Píllaro y Baños de Agua Santa [6].

Del mismo modo en Sta Rosa, una parroquia ubicada al suroeste de la ciudad se han reportado un alto incremento de la inseguridad, ocasionando incertidumbre y preocupación a sus moradores. A pesar de las denuncias, aún siguen notando la presencia de personas sospechosas pidiendo más colaboración por parte de las autoridades [7].

No obstante es importante señalar que la seguridad no depende solo de los esfuerzos de la policía nacional, sino también del ciudadano en cada hogar, barrio o parroquia siendo participes al denunciar y colaborar con las instituciones responsables del cumplimiento de la ley [8].

1.2.2. Fundamentación Teórica

1.2.2.1. Aplicaciones Móviles

Son aplicaciones informáticas desarrollados para proporcionar a los usuarios una variedad de funciones, y todo lo que necesitan para usarlos es un dispositivo móvil, como un teléfono celular o tableta que en el mayor de los casos necesita una conexión a internet para descargarlos [9].

Las aplicaciones están diseñadas y desarrollados en base a los requerimientos y necesidades de los usuarios apoyándose de la integración de herramientas y funcionalidades de los dispositivos tales como: cámara, Sistema de Posicionamiento Global (GPS), contenido multimedia, sensor de movimiento, entre otros, con la finalidad de aprovechar al máximo las capacidades y cualidades del teléfono móvil de desarrollo en base a las necesidades y requerimientos del proyecto y de la empresa. Para lo cual se debe seleccionar la mejor opción entre los 3 tipos de apps móviles que existen, y son [10]:

- Aplicaciones web.
- Aplicaciones nativas.
- Aplicaciones híbridas.

1.2.2.2. Tipos de Aplicaciones Móviles

- **Aplicaciones web móviles**

Este tipo de aplicaciones están diseñadas para dispositivos móviles por lo tanto se adaptan fácilmente. En la mayoría de los casos son implementadas con tecnologías Lenguaje de marcado de hipertexto (HTML), Hojas de estilo en cascada (CSS) y JavaScript. Una de sus principales ventajas es que son multiplataforma, es decir, pueden ser ejecutadas en diversos dispositivos como: ordenadores de escritorio, laptops, tabletas, teléfonos inteligentes y demás [11].

Es importante destacar que al estar desarrolladas con HTML, CSS y JavaScript no necesitan ser instaladas o compiladas en el dispositivo de destino, ya que necesitan solo de un navegador para ser ejecutadas. Además permite a los usuarios interactuar con el contenido en tiempo real, sobre todo mejorando su experiencia [12].

| Ventajas | Desventajas |
|---|--|
| Los desarrolladores solo deben conocer una tecnología | No son compatibles con navegadores antiguos. |
| Desarrollo, distribución y pruebas sencillas. | Su rendimiento es menor con respecto a las aplicaciones nativas. |
| Ampliamente soportado por la mayoría de los dispositivos móviles. | No se puede acceder a todos los recursos hardware del teléfono. |

Tabla 1.2.1 Ventajas y Desventajas de las aplicaciones web móviles

Elaborado por: Christian Chasi

- **Aplicaciones Nativas**

Son aplicaciones desarrolladas para una sola plataforma o sistema operativo como iPhone OS (iOS), Android, BlackBerry, Windows Phone, entre otros. Aprovechan el máximo potencial del dispositivo, mejorando la experiencia del usuario [11].

Son capaces de hacer uso de todos los recursos del dispositivo como la cámara, GPS, acelerómetro, agenda y muchos otros más. Sin embargo, si se desea crear el mismo

aplicativo para otras plataformas se deberá generar un nuevo código fuente para ellas, por lo tanto, su desarrollo requiere de más tiempo, conocimiento y dinero [13].

| Ventajas | Desventajas |
|---|--|
| Máximo Rendimiento | Funcionan en una sola plataforma o sistema operativo para el que fue creado. |
| Ofrecen gran experiencia de usuario por su diseño, velocidad y recursos hardware disponibles. | El desarrollo, testeo y mantenimiento son más costosos |

Tabla 1.2.2 Ventajas y Desventajas de las aplicaciones web nativas

Elaborado por: Christian Chasi

- **Tecnologías para desarrollo de aplicaciones nativas en:**

1. **Android**

Para el desarrollo de estas aplicaciones se utiliza comúnmente el entorno de Android Studio. Utiliza los lenguajes de programación JAVA o también Kotlin que ha estado creciendo en los últimos años [14].

2. **iOS**

Para esta plataforma el entorno de desarrollo más común es XCode compatible solo con dispositivos Apple. Los lenguajes más utilizados son Objective-C y Swift, este último es el más popular creado por Apple en el año 2014 [14].

- **Aplicaciones Híbridas**

Las aplicaciones híbridas combinan lo mejor de las aplicaciones nativas y web. La mayor parte de ellas son construidas con tecnologías web como HTML, CSS Y JavaScript sin embargo estas aplicaciones si pueden hacer uso de un mayor porcentaje de recursos hardware del dispositivo[13].

A diferencia de las aplicaciones nativas, se puede reutilizar el código para múltiples plataformas y también se pueden descargar en las tiendas de aplicaciones móviles. El inconveniente con estas aplicaciones es que su rendimiento depende del grado de complejidad de la misma [13].

| Ventajas | Desventajas |
|---|---|
| Tiempo de desarrollo más rápido. | Su rendimiento es inferior al de una aplicación nativa. |
| Adaptable a varias plataformas con un solo código fuente. | Acceso limitado a los recursos hardware del teléfono. |
| Costos de desarrollo, testeo y mantenimiento inferiores. | En el caso de cambios de código o diseño debe probarlo en todos los dispositivos para evitar errores. |

Tabla 1.2.3 Ventajas y Desventajas de las aplicaciones híbridas

Elaborado por: Christian Chasi

- **Tecnologías para desarrollo de aplicaciones híbridas**

Como se analizó anteriormente las aplicaciones híbridas están construidas con tecnologías HTML, CSS y Javascript o Frameworks que son muy utilizados en la actualidad, a continuación, se detallan brevemente los más populares [15]:

- 1. Ionic**

Este framework de aplicaciones móviles fue diseñado para trabajar con otras tecnologías, entre ellas apache cordova y Angular. Permite el desarrollo de aplicaciones móviles Android e iOS sin problemas en el rendimiento. Cuenta con varios elementos ya diseñados como vistas, formularios, menús y demás para la construcción de interfaces [15].

- 2. React Native**

Desarrollado por Facebook, es un framework de código abierto que se caracteriza por poseer componentes reutilizables, integración de componentes de terceros y GUI específico de frontend [15].

- 3. Flutter**

Es el framework más reciente desarrollado por Google y el que más se acerca al desarrollo nativo. Tiene una gran capacidad para desarrollar interfaces gráficas basados en widgets [15].

- 4. Xamarin**

Este framework creado por Microsoft con el objetivo de ser una herramienta de desarrollo para aplicaciones móviles multiplataforma. Utiliza el lenguaje de programación C# y tiene amplios componentes para el desarrollo [15].

- **Comparativa de los tipos de aplicaciones móviles**

| Criterio | Web | Nativa | Híbridas |
|---|-----------------------|---|---|
| Compatibilidad con diferentes plataformas | ➤ Navegador es web | ➤ Android ➤ iOS ➤ Windows Phone ➤ Otros | ➤ Android ➤ iOS ➤ Windows Phone ➤ Otros |
| Costos de Desarrollo | ➤ Bajo ➤ Medio | ➤ Alto | ➤ Bajo ➤ Medio |
| Testing | ➤ Bajo ➤ Medio | ➤ Alto | ➤ Bajo ➤ Medio |
| Mantenimiento | ➤ Bajo | ➤ Alto | ➤ Bajo ➤ Medio |
| Acceso a las capacidades del dispositivo | ➤ Bajo | ➤ Alto | ➤ Medio |
| Tiempo de Desarrollo | ➤ Bajo | ➤ Alto | ➤ Medio |
| Rendimiento y experiencia de Usuario | ➤ Medio | ➤ Muy Alto | ➤ Medio ➤ Alto |

Tabla 1.2.4 Comparativa aplicaciones web, nativas e híbridas

Elaborado por: Christian Chasi

1.2.2.3. Servicios Web y API

- **Servicios Web**

Es un conjunto de protocolos utilizado en el mundo de la informática para el intercambio de información entre sistemas o aplicaciones. Se hace hincapié a que los servicios web están diseñados exclusivamente para trabajar a través de redes informáticas [16].

Características:

1. Funcionalidad síncrona y asíncrona.
2. Capacidad de admitir llamadas a procedimientos remotos.
3. Soporta el intercambio de documentos.

Tipos de Servicios Web

Simple Object Access Protocol (SOAP). – Uno de los primeros protocolos creados para intercambiar información de forma segura [16].

Representational State Transfer (REST). – Arquitectura diseñada para trabajar con archivos, documentos, objetos, entre otros. Estos se caracterizan porque utilizan los verbos HTTP como GET, POST, PUT, DELETE para intercambiar información entre los componentes [16].

- **API (Interfaz de Programación de Aplicaciones)**

Es un conjunto de reglas o interfaces que permiten la interacción o transferencia de datos entre dos o más aplicaciones. Una característica de estas es que proporcionan comunicación entre productos o servicios que para el usuario final no es necesario saber cómo se implementan [17].

Según RedHat empresa proveedora de soluciones empresariales de TI en su página web define a las API como “...contratos, con documentación que representa un acuerdo entre las partes: si la parte 1 envía una solicitud remota estructurada de una manera particular, así es como responderá el software de la parte 2” [17].

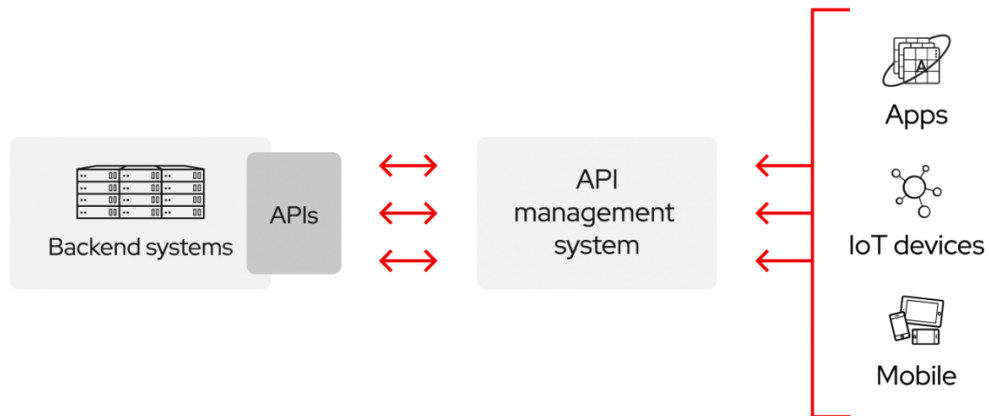


Figura 1.2.1 Arquitectura APIs

Elaborado por: [17]

Las APIs intercambian información o mensajes por dos métodos, mediante Extensible Markup Language (XML) o Javascript Object Notation (JSON) esto debido a que estos dos formatos son fáciles de utilizar por otras aplicaciones [17].



Figura 1.2.2 Formatos XML y JSON

Elaborado por: [44]

Cabe destacar que las APIs que se acercan a las características de la arquitectura REST se consideran RESTful APIs.

- **Diferencias entre Servicios Web y API**

Para tener claro el contexto de estas dos temáticas se debe tener en cuenta que todos los Servicios web son APIs, pero no todas las APIs son servicios web.

| Servicio Web | API |
|--|--|
| Todos los servicios web son API. | Todas las API no son servicios web. |
| Es compatible con XML. | Las respuestas se formatean utilizando MediaTypeFormatter de Web API en XML, JSON o cualquier otro formato dado. |
| Necesita un protocolo SOAP para enviar o recibir datos a través de la red. Por lo tanto, no tiene una arquitectura ligera. | API tiene una arquitectura ligera. |
| Puede ser utilizado por cualquier cliente que entienda XML. | Puede ser utilizado por un cliente que entienda JSON o XML. |
| El servicio web utiliza tres estilos: REST, SOAP y XML-RPC para la comunicación. | Api se puede utilizar para cualquier estilo de comunicación. |
| Proporciona soporte solo para el protocolo HTTP. | Proporciona soporte para el protocolo HTTP / s: encabezados de solicitud / respuesta de URL, etc. |

Figura 1.2.3 Diferencias entre Servicios web y APIs

Elaborado por: [16]

1.2.2.4. Metodologías de desarrollo

Una metodología de desarrollo es un conjunto de herramientas y documentos técnicamente elaborados para que los desarrolladores de software planifiquen, gestionen, implementen y evalúen sistemas de calidad. Para lograrlo deben atravesar ciertas fases que encaminan al objetivo principal ya sea en equipos grande o pequeños de programadores [18].

A lo largo de la historia de la ingeniería de software se han presentado algunos tipos de metodologías, las más conocidas, las tradicionales y las ágiles.

- **Metodologías de desarrollo tradicionales**

Catalogada por muchos como una metodología pesada a causa de su riguroso proceso o actividades en el desarrollo de software debido a que obligaba a llevar una extensa documentación, requisitos muy precisos, roles y herramientas muy detallados y hasta una planificación absoluta de todo el proyecto en la fase inicial. Todo esto con la meta

de elaborar un software de calidad, pero a la larga dando problemas dado que no se adaptaba a los cambios que necesitaba el cliente o que surgían durante las fases de desarrollo [19].

- **Metodologías de desarrollo ágiles**

Son técnicas de desarrollo de software, las cuales utilizan una serie de procesos iterativos o actividades durante el ciclo de vida del software. Tienen por objetivo reducir la sobrecarga de desarrollo haciéndolo como su nombre lo indica ágil, y que los cambios que se realicen no afecten la calidad de este [20].

- **Metodología ágil vs tradicional**

| Metodología Ágil | Metodología Tradicional |
|--|--|
| ➤ Basada en heurísticas de producción de código | ➤ Basado en estándares de desarrollo. |
| ➤ Preparado para cambios durante el desarrollo del proyecto | ➤ Resistencia a los cambios |
| ➤ Flexibilidad en los contratos debido a la respuesta de cambios | ➤ Existe un contrato prefijado |
| ➤ El cliente es parte fundamental para el equipo de desarrollo | ➤ El cliente interactúa con el equipo en reuniones. |
| ➤ Grupos pequeños de menos de 10 integrantes | ➤ Grupos grandes trabajando den distintas tareas. |
| ➤ Pocos roles | ➤ Más roles |
| ➤ Menos énfasis en la arquitectura | ➤ La arquitectura es primordial y se expresa mediante modelos. |

Tabla 1.2.5 Comparativa de aplicaciones tradicionales y ágiles

Elaborado por: [21]

1.2.2.5. Geolocalización

Es una tecnología que por medio de un dispositivo (celular, pc, tableta) obtiene datos que sirven para identificar su ubicación en un entorno físico (geoespacial) o virtual (internet). Actualmente a ganado mucha popularidad por su amplio campo de aplicaciones [22]. Esta tecnología es utilizada para dar a conocer la ubicación geográfica de una persona u objeto, es decir, muestran ubicaciones del mundo real con el propósito de ser interpretadas, analizadas y transmitidas mediante hardware, software o plataformas digitales. No obstante, también se inmiscuye en conceptos dinámicos como la posición espacial (coordenadas geográficas), referentes espaciales codificados (geo-datos) y ubicación mediante dispositivos digitales [23].

La geolocalización se puede obtener de varias formas, las más frecuentes son [22]:

- Mediante direcciones IP navegando en la web.
- Dispositivos Móviles.
- Dispositivos GPS.
- Identificación de radio frecuencia (RFID).

| Modo | Método de recolección | Tecnologías involucradas |
|--|---|---|
| Activo: Basado en el dispositivo del usuario | <ul style="list-style-type: none"> ➤ Utiliza el firmware y software del dispositivo. ➤ Utiliza el chip GPS. | <ul style="list-style-type: none"> ➤ GPS ➤ Wi-Fi ➤ 3G/4G ➤ Aplicaciones móviles |
| Pasivo: Búsqueda de datos en servidores | <ul style="list-style-type: none"> ➤ Utiliza proveedores de servicio de geolocalización. ➤ Mediante direcciones IP. | <ul style="list-style-type: none"> ➤ Ubicación IP ➤ 3G/4G ➤ Wi-Fi |

Tabla 1.2.6 Métodos de generación y recolección de datos de geolocalización

Elaborado por: [22]

- **Sistema de Posicionamiento Global (GPS)**

Es un sistema que mediante señales emitidas desde satélites estratégicamente ubicados en órbita se obtienen coordenadas de cualquier punto referencial en la superficie del planeta, permitiendo conocer la posición geográfica de cualquier ente en mundo [24].

Con la incorporación de chips GPS y otros elementos hardware como acelerómetros y giroscopios en dispositivos móviles se puede determinar fácilmente su posición. Actualmente las aplicaciones móviles pueden acceder a esos recursos aprovechando sus funcionalidades en navegación, enrutamiento y localización [23].

- **Uso de Mapas para Geolocalización**

En el mundo de las apps la navegación mediante mapas está de moda, actualmente el que domina ese mercado es Google Maps. Sin embargo, también existen otras buenas alternativas.

Google Maps

Creado por Google ha sido líder de la industria cartográfica brindando servicios como: geocodificación, enrutamiento, imágenes, entre otros [25].

Características

1. No tiene API gratuita.
2. Mapas personalizables.
3. Lo utilizan empresas como Uber, Bolt, Trip Advisor.
4. API muy fácil de utilizar.

Mapbox

Es uno de los principales competidores de Google Maps y colaborador del código abierto en bibliotecas Javascript con Leaflet, cartografía con TileMill y estilos del mapa con CartoCss. Es muy popular ya que lo utilizan empresas como Facebook, GrubHub y Airbnb [25].

Características

1. Cuenta con una API gratuita limitada.
2. Útil en la Inteligencia Empresarial
3. Se pueden agregar diferentes capas
4. Tiene mapas 3D.
5. Mapas personalizados.
6. Mapas offline.

Leaflet

Es una biblioteca de javascript de código abierto creado por Vladimir Agafonkin, quien después se uniría al proyecto de Mapbox. Actualmente Leaflet es mantenida por la comunidad de desarrolladores [26].

Características

1. Diferentes capas.
2. API gratuito.
3. Fácil de integrar y usar.
4. Compatible con todas las plataformas.
5. Grande comunidad que colaboran dando soporte.

1.2.2.6. Notificaciones push

Es un método de comunicación en el que el servidor notifica contenido nuevo o relevante a sus dispositivos suscritos, todo esto mediante alertas que se muestran en el panel de notificaciones de su dispositivo móvil [10]. Comprende un mecanismo de eventos emitidos por distintos dispositivos (celulares, tabletas, ordenadores, entre otros.) hacia servidores estableciendo comunicación y notificando dichos eventos hacia otros dispositivos clientes [27].



Figura 1.2.4 Proceso de envío de notificaciones push

Elaborado por: [45]

- **Servicios de notificaciones push populares**

Las notificaciones push tienen una variedad de proveedores de estos servicios, algunos permiten integrar este servicio gratuitamente y otros mediante pagos. Cada uno ofrecen mejores características que su competencia, a continuación, se listan los más populares [28].

- Firebase Cloud Messaging (FCM)
- Wonder Push
- OneSignal
- PushCrew

1.2.2.7. Seguridad ciudadana

- **Seguridad Social**

Busca consolidar la convivencia pacífica, el respeto de los derechos humanos y la prevención de cualquier tipo de violencia a través de la intervención de entes públicos y privados, con apoyo de la ciudadanía [8].

También implica un término amplio en los diversos campos de intervención, el concepto actual es proteger a los ciudadanos de las acciones de grupos criminales que operan por el interés público y la propiedad [29].

- **Seguridad Comunitaria**

Está enfocado a organizar y capacitar a los vecinos ante cualquier situación de delincuencia organizada, tomando medidas que garanticen su protección y calidad de vida en sus respectivas comunidades [30].

1.3.Objetivos

1.3.1. Objetivo General

Desarrollar una aplicación móvil de apoyo a la seguridad barrial para envío y localización de alertas de auxilio mediante notificaciones push en la parroquia Santa Rosa de la ciudad de Ambato.

1.3.2. Objetivos Específicos

- Identificar los entes que participan en los planes de control y seguridad barrial.
- Determinar el mejor servicio de notificaciones push.
- Definir la metodología de desarrollo de software adecuada para el proyecto.
- Implementar una aplicación móvil para envío y localización de alertas de auxilio mediante notificaciones push en la parroquia Santa Rosa de la ciudad de Ambato.

CAPITULO II: METODOLOGÍA

2.1. Materiales

2.1.1. Humanos

- Docente Tutor.
- Investigador.
- Ciudadanos de la parroquia Santa Rosa de la ciudad de Ambato.

2.1.2. Institucionales

- Facultad de Ingeniería en Sistemas, Electrónica e Industrial.
- Repositorio.
- Biblioteca Virtual.

2.1.3. Otros

El proyecto será financiado en su totalidad por el investigador.

| N.º | Detalle | Unidad | Cantidad | V. unitario | Valor total |
|-----|------------------------------------|--------|----------|--------------------------|-------------|
| 1 | Ordenador | c/u | 1 | 1300,00 | 1300,00 |
| 2 | Internet | m | 6 | 35,00 | 210,00 |
| 3 | Transporte | c/u | 20 | 0,30 | 6,00 |
| 4 | Impresiones | c/u | 300 | 0,05 | 15,00 |
| 5 | Resma de papel | c/u | 2 | 3,50 | 7,00 |
| 6 | Cuenta Google para desarrolladores | c/u | 1 | 25,00 | 25,00 |
| 7 | Dispositivo móvil | c/u | 1 | 250,00 | 250,00 |
| 8 | Dispositivos de almacenamiento | c/u | 2 | 20,00 | 40 |
| | | | | Subtotal | 1853,00 |
| | | | | Imprevistos (10%) | 185,30 |

| N.º | Detalle | Unidad | Cantidad | V. unitario | Valor total |
|-----|---------|--------|----------|--------------|-------------|
| | | | | Total | 2.038,30 |

Tabla 2.1.1 Recursos económicos

Elaborado por: Christian Chasi

2.2.Métodos

2.2.1. Modalidad de la investigación

Investigación de campo. Porque en base a encuestas y pruebas piloto que se aplicarán a los moradores de los barrios se obtendrá información que será utilizada para analizar y definir la importancia de tener una aplicación móvil tipo botón de pánico entre la comunidad barrial.

Investigación bibliográfica. Porque se utilizará fuentes de información como libros, artículos científicos, revistas para la construcción del marco teórico.

2.2.2. Población y muestra

Se ha seleccionado a los habitantes de la parroquia Santa Rosa de la ciudad de Ambato, de igual manera este caso de estudio se lo puede replicar en todos los barrios de la parroquia y de la ciudad de Ambato que cumplen las siguientes características:

1. Son organizados.
2. Son bastante participativos y colaborativos.

También se calcula la muestra de los habitantes debido a que el tamaño de la población de la parroquia Sta. Rosa según el último censo del INEC es de 22.500 personas. Para el respectivo cálculo se utilizará la siguiente fórmula:

$$n = \frac{N\sigma^2Z^2}{(N-1)e^2 + \sigma^2Z^2}$$

Donde:

n = tamaño de la muestra

N = población

Z = nivel de confianza 95% equivalente a 1,96

σ = Desviación estándar constante 0,5

e = margen de error: 0,05

$$n = \frac{22.500 * 0,5^2 * 1,96^2}{(22.500 - 1) * 0,05^2 + 0,5^2 * 1,96^2}$$

$$n = 377.73$$

| Nº | Tipo de población | Descripción | Número |
|----|--|---|--------|
| 1 | Moradores de los barrios de la parroquia Sta. Rosa | Son las personas a las que se aplicarán las encuestas, se socializará y capacitará para el uso respectivo de la aplicación. | 378 |
| | | Total | 378 |

Tabla 2.2.1 Población y muestra

Elaborado por: Christian Chasi

2.2.3. Recolección de la información

| Preguntas Básicas | Explicación |
|-----------------------------|---|
| ¿Para qué? | Para alcanzar los objetivos de la investigación. |
| ¿De qué personas u objetos? | Ciudadanos involucrados en actividades de control y seguridad barrial |
| ¿Sobre qué aspectos? | Pedidos de auxilio |
| ¿Quién, quienes? | Investigador: Christian Paúl Chasi Chango |
| ¿Cuándo? | Semestre Académico Abril 2021 – Septiembre 2021 |
| ¿Dónde? | En la parroquia Santa Rosa de la ciudad de Ambato |
| ¿Cuántas Veces? | 1 |

| | |
|-------------------------------|--|
| ¿Qué técnicas de recolección? | Encuesta |
| ¿Con qué? | Cuestionario |
| ¿En qué situaciones? | En circunstancias de delincuencia o violencia que demanden algún tipo de ayuda inmediata hacia la persona que lo necesite. |

Tabla 2.2.2 Recolección de la Información

Elaborado por: Christian Chasi

2.2.4. Procesamiento y Análisis de Datos

2.2.4.1. Procesamiento de la Información

Esta encuesta se realizó a los ciudadanos de los diferentes barrios en la parroquia Sta. Rosa de la ciudad de Ambato.

1. ¿Qué tan seguro considera que es su barrio?

| Variables | Frecuencia | Porcentaje |
|---------------------|------------|------------|
| Poco Seguro | 250 | 66.14% |
| Medianamente Seguro | 120 | 31.75% |
| Muy Seguro | 8 | 2.12% |
| Total | 378 | 100.00% |

Tabla 2.2.3 Nivel de seguridad en el barrio

Elaborado por: Christian Chasi

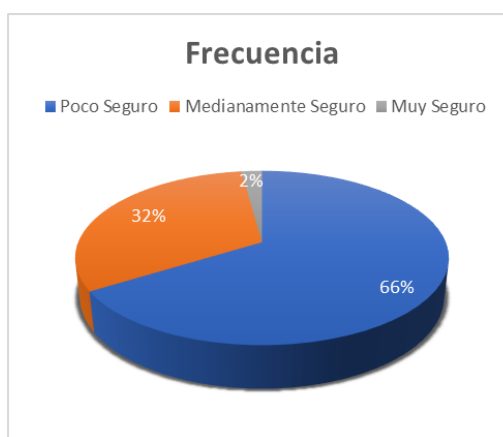


Figura 2.2.1 Resultados Nivel de Seguridad en el barrio

Elaborado por: Christian Chasi

Análisis e Interpretación:

De una muestra de 378 personas encuestadas se determinó que 370 personas que corresponden al 98% (entre poco seguro y medianamente seguro) sienten inseguridad en su barrio.

Por ende, se concluye que en los barrios de la parroquia Sta. Rosa existen problemas para lidiar con la inseguridad ciudadana.

2. ¿Ha colocado sistemas de seguridad en su hogar?

| Variables | Frecuencia | Porcentaje |
|------------------|-------------------|-------------------|
| Si | 57 | 15.08% |
| No | 321 | 84.92% |
| Total | 378 | 100.00% |

Tabla 2.2.4 Sistemas de Seguridad en el Hogar

Elaborado por: Christian Chasi

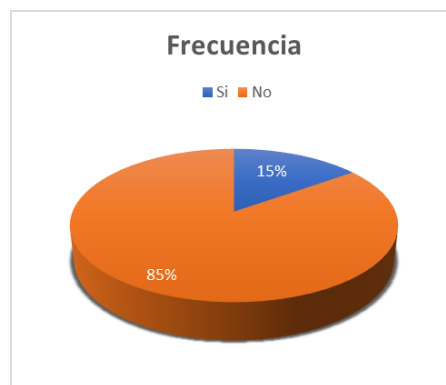


Figura 2.2.2 Resultados Sistemas Seguridad en el Hogar

Elaborado por: Christian Chasi

Análisis e Interpretación:

De una muestra de 378 personas encuestadas se determinó que el 85% no cuenta con sistemas de seguridad en su hogar.

Por ende, se concluye que la mayor parte de personas están expuestas a ser víctimas de la delincuencia al no contar con ningún sistema que les permita alertar cualquier situación de emergencia en su hogar.

3. ¿Su barrio cuenta con sistemas de seguridad o alarmas comunitarias?

| Variables | Frecuencia | Porcentaje |
|------------------|-------------------|-------------------|
| Si | 13 | 3.44% |
| No | 365 | 96.56% |
| Total | 378 | 100.00% |

Tabla 2.2.5 Sistemas de seguridad comunitaria

Elaborado por: Christian Chasi

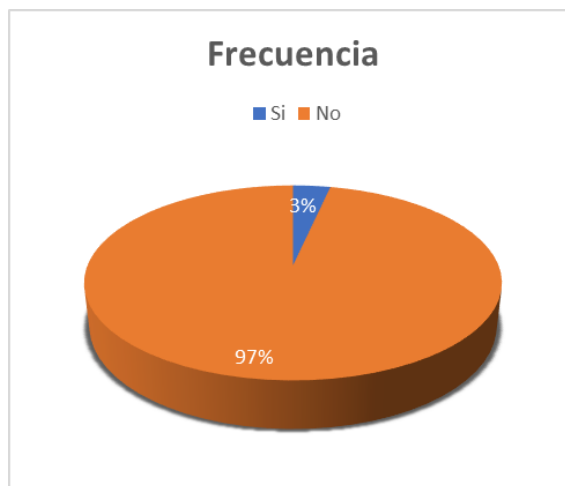


Figura 2.2.3 Resultados Sistemas de Seguridad

Elaborado por: Christian Chasi

Análisis e Interpretación:

De una muestra de 378 personas encuestadas se determinó que 365, que corresponde al 97% dan a conocer que su barrio no consta con sistemas de seguridad.

Por ende, se concluye que los morados de los barrios no cuentan con una herramienta o sistema que permita alertar a la policía o vecinos del barrio.

4. ¿Su barrio es organizado y colaborativo para brindar ayuda en situaciones de delincuencia?

| Variables | Frecuencia | Porcentaje |
|------------------|-------------------|-------------------|
| Si | 232 | 61.38% |
| No | 146 | 38.62% |
| Total | 378 | 100.00% |

Tabla 2.2.6 Su barrio es organizado y colaborativo

Elaborado por: Christian Chasi

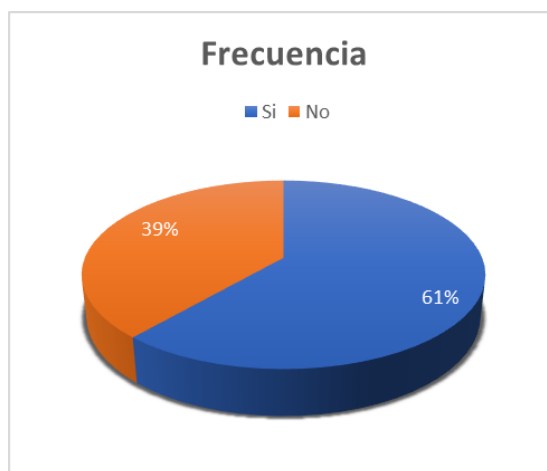


Figura 2.2.4 Resultados Barrio organizado y colaborativo

Elaborado por: Christian Chasi

Análisis e Interpretación:

De una muestra de 378 personas encuestadas, se determinó que la mayor parte son organizados y colaborativos en situaciones o actos de delincuencia.

Por ende, se concluye que los barrios actualmente se mantienen organizados para tomar medidas de seguridad ante la delincuencia.

5. ¿Está de acuerdo en que los barrios se protejan ante el aumento de la delincuencia?

| Variables | Frecuencia | Porcentaje |
|------------------|-------------------|-------------------|
| Si | 323 | 85.45% |
| No | 55 | 14.55% |

| | | |
|--------------|-----|---------|
| Total | 378 | 100.00% |
|--------------|-----|---------|

Tabla 2.2.7 De acuerdo que los barrios se protejan

Elaborado por: Christian Chasi

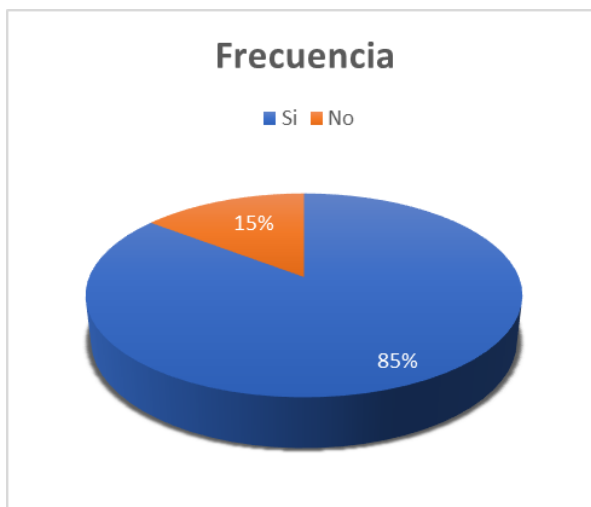


Figura 2.2.5 Resultados De acuerdo que los barrios se protejan

Elaborado por: Christian Chasi

Análisis e Interpretación:

De una muestra de 378 personas encuestadas, se determinó que, 323 que corresponden al 85% están de acuerdo en que los barrios se protejan ante el aumento de la delincuencia.

Por ende, se concluye que los barrios tienen la intención de prevenir y protegerse ante tanta inseguridad, simplemente necesitan las herramientas para poder hacerlo.

6. ¿Qué tan frecuentemente al mes se suscitan delitos en su barrio o parroquia?

| Variables | Frecuencia | Porcentaje |
|------------------|-------------------|-------------------|
| Nunca | 12 | 3.17% |
| Rara Vez | 78 | 20.63% |
| Casi Siempre | 245 | 64.81% |
| Siempre | 43 | 11.38% |
| Total | 378 | 100.00% |

Tabla 2.2.8 Frecuencia de delitos en el barrio

Elaborado por: Christian Chasi

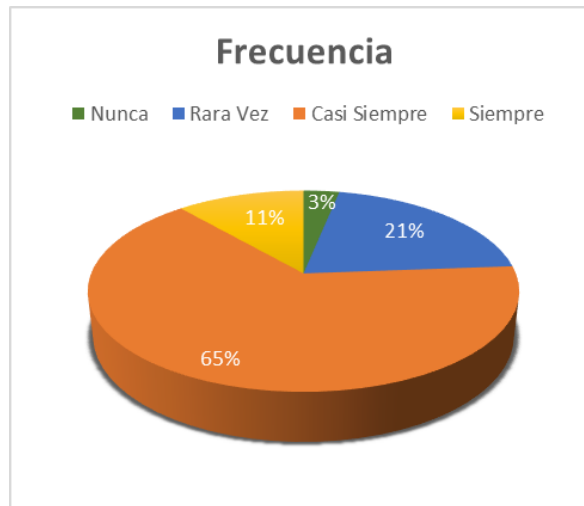


Figura 2.2.6 Resultados Frecuencia de delitos en el barrio

Elaborado por: Christian Chasi

Análisis e Interpretación:

De una muestra de 378 personas encuestadas, se determinó que 288 personas, que corresponden al 76% (entre siempre y casi siempre) manifiestan que hay gran cantidad de delitos en su barrio o parroquia.

Por ende, se concluye que los moradores de los barrios sufren constantemente delitos.

7. ¿Qué delitos son más frecuentes en su barrio o parroquia?

| Variables | Frecuencia | Porcentaje |
|-------------------|-------------------|-------------------|
| Robos a Viviendas | 197 | 52.12% |
| Robo a personas | 90 | 23.81% |
| Robo a vehículos | 29 | 7.67% |
| Asesinatos | 0 | 0.00% |
| Otros | 62 | 16.40% |
| Total | 378 | 100.00% |

Tabla 2.2.9 Delitos más frecuentes en el barrio

Elaborado por: Christian Chasi

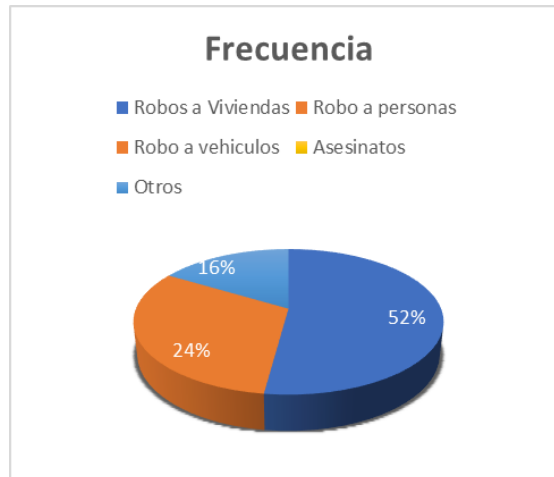


Figura 2.2.7 Resultados Delitos más frecuentes en el barrio

Elaborado por: Christian Chasi

Análisis e Interpretación:

De una muestra de 378 personas encuestadas, se determinó que la mayor cantidad de delitos en los barrios son robos.

Por ende, se concluye independientemente del tipo de delito, los moradores de los barrios están siendo víctimas de los antisociales y a causa de eso son deseables las herramientas que permitan asistir a personas que son víctimas de delitos.

8. ¿Es importante tener un botón de pánico entre la comunidad del barrio?

| Variables | Frecuencia | Porcentaje |
|------------------|-------------------|-------------------|
| Si | 370 | 97.88% |
| No | 8 | 2.12% |
| Total | 378 | 100.00% |

Tabla 2.2.10 Importancia de un botón de pánico

Elaborado por: Christian Chasi



Figura 2.2.8 Resultados Importancia de un botón de pánico

Elaborado por: Christian Chasi

Análisis e Interpretación:

De una muestra de 378 personas encuestadas, se determinó que 370 personas que corresponden al 98% consideran que es importante tener un botón de pánico entre la comunidad del barrio.

Por ende, se concluye que el barrio necesita de un botón de pánico entre vecinos para poder actuar con rapidez ante la delincuencia o alguna emergencia o acontecimiento.

9. ¿Cree que la tecnología (teléfonos inteligentes, cámaras de seguridad, sistemas, aplicaciones, entre otros) pueden ayudar a combatir la inseguridad ciudadana?

| Variables | Frecuencia | Porcentaje |
|------------------|-------------------|-------------------|
| Si | 356 | 94.18% |
| No | 22 | 5.82% |
| Total | 378 | 100.00% |

Tabla 2.2.11 Tecnología ayuda a combatir la inseguridad

Elaborado por: Christian Chasi

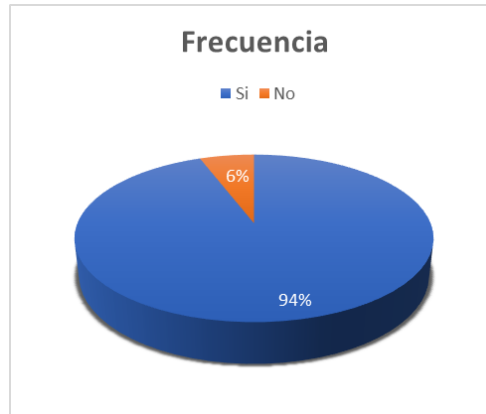


Figura 2.2.9 Resultados Tecnología ayuda a combatir la inseguridad

Elaborado por: Christian Chasi

Análisis e Interpretación:

De una muestra de 378 personas encuestadas, se determinó que, 356 que corresponden al 94% consideran que la tecnología es un medio fundamental que puede ayudar a combatir la inseguridad ciudadana.

Por ende, se concluye que el uso de la tecnología móvil aportaría considerablemente a los ciudadanos a combatir la delincuencia en los diferentes barrios de la parroquia.

10. ¿Usted estaría dispuesto a colaborar para mejorar la seguridad de su barrio?

| Variables | Frecuencia | Porcentaje |
|------------------|-------------------|-------------------|
| Si | 378 | 100.00% |
| No | 0 | 0.00% |
| Total | 30 | 100.00% |

Tabla 2.2.12 Mejorar la seguridad del barrio

Elaborado por: Christian Chasi



Figura 2.2.10 Resultados Mejorar la seguridad del barrio

Elaborado por: Christian Chasi

Análisis e Interpretación:

De una muestra de 378 personas encuestadas, se determinó que el 100% de ellas están dispuestas a colaborar para mejorar la seguridad de su barrio.

Por ende, se concluye que en los barrios hay gran entusiasmo por participar en planes o programas de prevención y protección para mejorar la seguridad barrial y parroquial.

2.2.5. Análisis e Interpretación de los resultados

Las encuestas fueron aplicadas a moradores de los diferentes barrios de la parroquia Santa Rosa. Se hicieron preguntas enfocadas a la seguridad e inseguridad que viven en sus barrios para determinar el nivel de esta, también se tomó en cuenta su criterio en entorno al auge de las tecnologías móviles y su importancia para combatir la inseguridad ciudadana.

Los resultados que se obtuvieron dan a conocer que en la mayoría de los casos carecen de las herramientas o sistemas de seguridad quizás por la falta de recursos o porque estos sistemas son muy costosos. No obstante, ellos desean participar y colaborar en actividades de seguridad ciudadana ya sea en reuniones presenciales o virtuales con el fin de estar prevenidos y capacitados para saber cómo actuar ante el aumento de los delitos en su comunidad.

CAPITULO III

RESULTADOS Y DISCUSIÓN

3.1. Análisis y discusión de los resultados.

3.1.1. Entes que participan en los planes de control y seguridad barrial

Mediante una breve entrevista con las directivas de los diferentes barrios y el GAD de la parroquia Sta. Rosa de la ciudad de Ambato se determinó que los planes de control, seguridad ciudadana y barrial están a cargo principalmente de instituciones como la policía o la Unidad de Policía Comunitaria (UPC) y del Consejo Municipal de Seguridad de la Ciudad de Ambato (COMSECA).

Estas entidades se encargan de planificar la mayor parte de actividades sociales en materia de seguridad ciudadana. En el caso de las UPC, se ubican en las zonas más estratégicas de los barrios o parroquias. Están encargadas de atender los llamados de auxilio, solicitando la ayuda de las patrullas más cercanas al lugar de los hechos. Indagando más a fondo se logró identificar que hay normas o políticas que los policías deben cumplir y las cuales los ciudadanos están en desacuerdo. Por ejemplo, si se da una alerta de auxilio y un patrullero está por el lugar del incidente, si este no está dentro de su jurisdicción difícilmente puede ayudar, ya que si llega a pasar un altercado con el delincuente el policía se involucra en serios problemas con la justicia.

Otro caso es el botón de pánico de la policía, una herramienta que durante estos años se ha dado a conocer porque en teoría la policía llega inmediatamente al lugar de los hechos con solo presionar un botón. Si embargo este no ha llegado a cumplir con las expectativas, ya que, en la mayoría de los casos, como manifestaban, si el sistema falla o tiene actualizaciones, la información de los botones de pánico se vuelve obsoleta, por ende, deben ir nuevamente con las personas para configurarlo en su dispositivo móvil. Del mismo modo las alertas de auxilio que se emiten a la policía o Ecu911 a veces no son atendidos inmediatamente ocasionando que muchos delitos queden en la impunidad.

El COMSECA, ente que se encarga de definir, planificar, ejecutar y controlar las políticas de convivencia y seguridad ciudadana, es el medio por el que se organizan

actividades como mingas, capacitaciones de bienestar y seguridad a los vecinos de los barrios con su actual programa “Barrio Limpio y Seguro”, programación de actividades de socialización con los diferentes municipios de la ciudad y en ciertos casos apoyando a proyectos que involucren temas de seguridad ciudadana. Algunas de sus políticas o normas involucran la recuperación de espacio públicos, el mejoramiento de iluminación pública, la conformación de comités de seguridad barriales, sistemas de alarmas comunitarias, cámaras de televigilancia comunal, entre otros. Con ayuda del GAD de Sta. Rosa algunas de estas políticas ya han sido implantadas en los diferentes espacios de la parroquia ayudando a reducir y controlar los delitos que se dan a diario.

En conclusión, estas instituciones tienen el fin de proteger a la ciudadanía, se reúnen con las directivas de los barrios más organizados, les capacitan y realizan actividades que les permitan estar preparados y prevenidos ante cualquier situación de emergencia. Los comités barriales saben que esa ayuda no es suficiente, y por ende deciden realizar otras actividades que complementen a las normas y políticas de seguridad barrial.

3.1.2. Metodologías de Desarrollo

Para definir la metodología de desarrollo con la cual se trabajará, se han propuesto 3 metodologías de las cuales se conocerán sus características como fases o ciclo de vida, como se aplica y al final una comparativa que permita identificar cuál se ajusta más al proyecto de desarrollo propuesto.

3.1.2.1. Extreme Programming (XP)

Esta metodología ágil se enfoca en elevar las relaciones interpersonales mediante el trabajo en equipo, la superación personal de los programadores y el entorno en el que trabajan. Buscan la retroalimentación a través de la comunicación entre clientes y el equipo de desarrollo, buscan soluciones apropiadas y que se adapten a requisitos imprecisos y muy cambiantes [21].

- **Ciclo de desarrollo de XP**



Figura 3.1.1 Ciclo de desarrollo XP

Elaborado por: Christian Chasi

El ciclo de vida de XP consiste en seis fases [31]:

a. Fase de Exploración

- Aquí interviene el cliente que es el que se encarga de tomar decisiones sobre los requisitos para luego ser trasladados a historias de usuarios.
- El equipo XP empieza a familiarizarse con las herramientas y tecnología para el desarrollo del proyecto.
- Se realiza un prototipo.

b. Fase de Planificación

- Se analizan las historias de usuario entregadas por el cliente priorizadas de mayor a menor importancia.
- El equipo realiza un cronograma y estimación del costo del proyecto.

c. Fase de Iteraciones

- Se detallan las tareas a realizar por cada historia de usuario.
- Las iteraciones van desde crear la arquitectura general del sistema hasta la fase final de producción.

- Se convoca a reuniones diarias que permitan verificar los avances o retrasos en base a cada historia de usuario.

d. Fase de Producción

- Se realizan nuevamente pruebas y evaluaciones adicionales antes y después de ser entregado al cliente.
- Continúa el desarrollo en base a las historias de usuarios faltantes.

e. Fase de Mantenimiento

- Se revisan las nuevas historias de usuarios que surgieron de la puesta en producción.
- Se asigna o incorpora personal para realizar las nuevas tareas del cliente.

f. Fase de Cierre o Muerte

- Se cierra cuando ya no existen más historias de usuario o no aportan nada al desarrollo del proyecto.

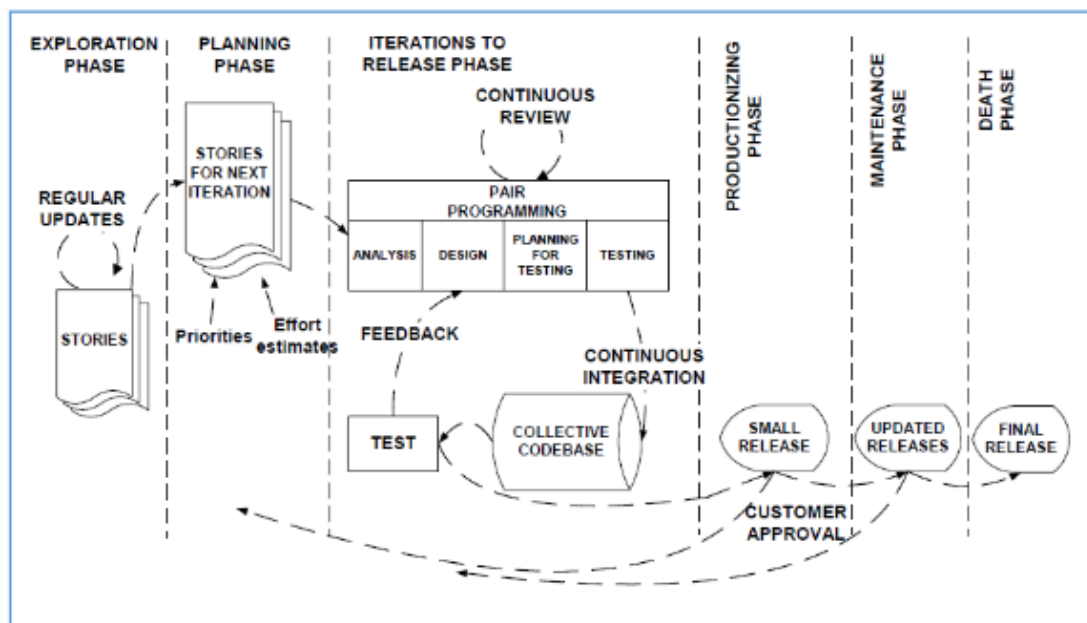


Figura 3.1.2 Fases de la metodología XP

Elaborado por: [20]

- **Roles XP**

| Rol | Actividad |
|------------------------------------|---|
| Programador | <ul style="list-style-type: none"> ➤ Codifica el ➤ Realiza pruebas unitarias |
| Cliente | <ul style="list-style-type: none"> ➤ Realiza y prioriza las historias de usuario. ➤ Diseña pruebas de aceptación. ➤ Retroalimenta al equipo de desarrollo. |
| Tester | <ul style="list-style-type: none"> ➤ Ayuda al cliente a diseñar las pruebas de aceptación. ➤ Ejecuta prueba de integración |
| Encargado de Seguimiento (tracker) | <ul style="list-style-type: none"> ➤ Realiza el seguimiento de las historias de usuario. ➤ Incorpora o elimina historias de usuario. |
| Entrenador | <ul style="list-style-type: none"> ➤ Guía y Controla que se siga con la metodología XP. |
| Consultor | <ul style="list-style-type: none"> ➤ Ayuda a resolver un problema puntual. |
| Gestor | <ul style="list-style-type: none"> ➤ El máximo responsable del proyecto ➤ Es la conexión con el cliente |

Tabla 3.1.1 Roles XP

Elaborado por: Christian Chasi

3.1.2.2.SCRUM

Es considerado una de las metodologías de desarrollo ágil más populares ya que se orienta más a los problemas de gestión del software que de las prácticas técnicas de desarrollo. Al igual que XP lleva un proceso de desarrollo iterativo denominado sprints ofreciendo mayor valor en el menor tiempo estimado en función del equipo y rol asignado [20].

Sus principales características se pueden resumir en dos:

Las iteraciones

- Las iteraciones se denominan sprints.
- Cada sprint tiene una duración de mínimo 15 días.
- Cada sprint finalizado es un presentable al cliente

Las reuniones

- Reuniones de 15 minutos para coordinación e integración del equipo.

- **Ciclo de desarrollo Scrum**

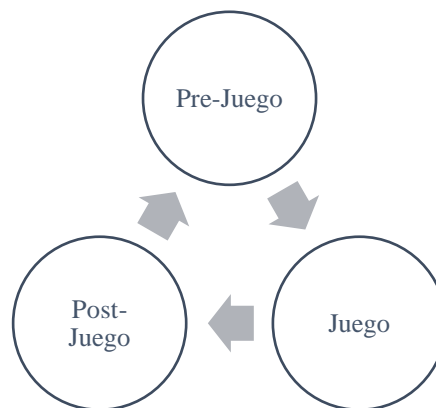


Figura 3.1.3 Ciclo de desarrollo Scrum

Elaborado por: Christian Chasi

La metodología Scrum cuenta con 3 fases principales [20]:

a. Pre-Juego

- Se elabora un backlog (lista de requisitos) del producto y se utiliza para documentar los requerimientos, historias de usuarios y características.
- Se asignan prioridades en función de los requisitos.

b. Juego

- Se ejecutan los sprints con una duración fija.
- Cada sprint atraviesa las fases de análisis, diseño y terminado.
- Se realizan reuniones diarias o daily scrum que duran 15 minutos en donde hacen retroalimentación de todas las actividades realizadas y por realizar.

c. Post-Juego

- Se verifica que se haya cumplido con los objetivos.
- En esta fase la versión del producto está listo para ser lanzado a producción con su respectiva documentación.

- **ROLES SCRUM**

| Rol | Actividad |
|---------------|--|
| Scrum Máster | <ul style="list-style-type: none"> • Facilita la comunicación entre el Product owner y el equipo • Se encarga de verificar que se siga con la filosofía Scrum. |
| Product Owner | <ul style="list-style-type: none"> • Se encarga de controlar y gestionar todo el proyecto hasta su finalización. |
| Scrum Team | <ul style="list-style-type: none"> • Son los responsables de desarrollar el proyecto basándose en los requisitos del cliente. |
| Customer | <ul style="list-style-type: none"> • Participa en la elaboración del backlog. |
| Management | <ul style="list-style-type: none"> • Tomas las decisiones y participa en la definición de metas y requerimientos. |

Tabla 3.1.2 Roles Scrum

Elaborado por: El investigador

3.1.2.3.Mobile-D

Esta metodología está orientada al desarrollo de aplicaciones móviles basándose en otras metodologías populares como XP en relación con prácticas de desarrollo, Crystal en relación con escalabilidad y Rational Unified Process (RUP) en relación al ciclo de vida [32].

Características

- Utilizado para proyectos pequeños.
- El equipo de desarrollo debe ser de menos de 10 personas.

- Se estima el tiempo de desarrollo de una aplicación móvil en menos de 10 semanas.
- **Ciclo de desarrollo Mobile-D**



Figura 3.1.4 Ciclo de desarrollo Mobile-D

Elaborado por: El investigador

La metodología Mobile-D tiene cinco fases que son:

a. Exploración

- Se establecen las partes interesadas.
- Se definen requisitos y objetivos de alcance del proyecto.
- En esta fase el equipo establece el plan y características del proyecto.

b. Inicialización

- Se definen y configuran los recursos técnicos (entorno, herramientas, tecnologías) necesarios para el desarrollo del proyecto.
- Se definen los requisitos y módulos del sistema.
- Se preparan planes para las siguientes fases.

c. Producción

- Se realizan las iteraciones para cumplir con trabajo, del mismo modo integración de código y pruebas rápidas.

d. Estabilización

- Se relaciona con la fase de producción realizando las pruebas suficientes para que el sistema se integre correctamente. No obstante, también se crea la respectiva documentación.

e. Pruebas

- Su objetivo es obtener una aplicación estable sin errores. Para ello se apoya del cliente que será el que realice pruebas para detectar fallos y corregirlos.

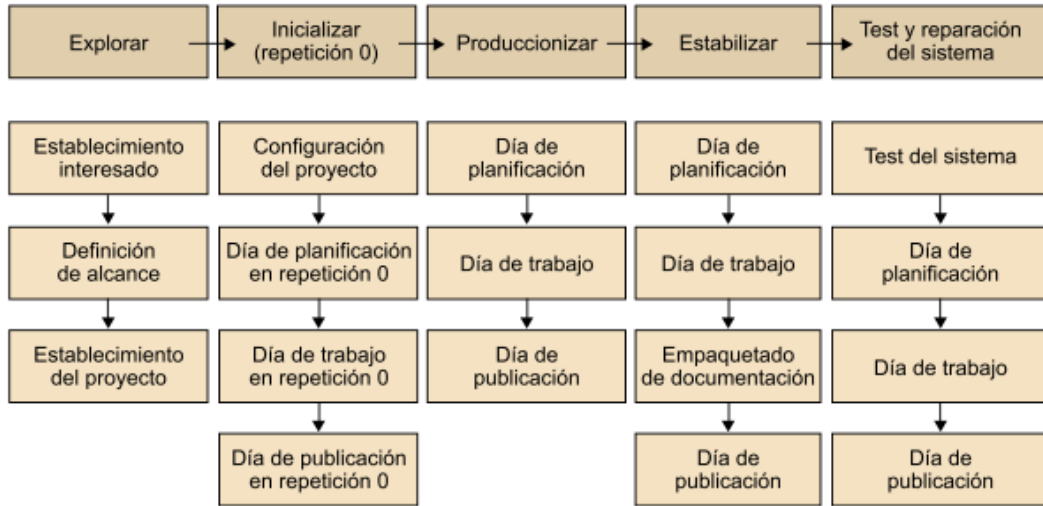


Figura 3.1.5 Ciclo de vida de la metodología Mobile-D

Elaborado por: [11]

- **Comparativa de metodologías ágiles de desarrollo**

| Criterio | Mobile-D | Scrum | XP |
|------------------------------|--|--|---|
| Año de aparición | 2005 | 1995 | 1996-1999 |
| Fases de Desarrollo | <ul style="list-style-type: none"> ➤ Exploración ➤ Inicialización ➤ Producción ➤ Estabilización ➤ Pruebas | <ul style="list-style-type: none"> ➤ Pre-Juego ➤ Juego ➤ Post-Juego | <ul style="list-style-type: none"> ➤ Exploración ➤ Planificación ➤ Iteraciones ➤ Producción ➤ Mantenimiento ➤ Fase de Cierre Muerte |
| Tipos de Aplicaciones | ➤ Aplicaciones para dispositivos móviles | ➤ Desarrollo de software en general. | ➤ Desarrollo de software en general. |

| Criterio | Mobile-D | Scrum | XP |
|------------------------------|------------------------|------------------------------------|------------------------------------|
| Tamaños de Proyectos | ➤ Pequeños | ➤ Pequeños, medianos y grandes. | ➤ Pequeños y medianos. |
| Tamaño del Equipo | ➤ Menos de 10 personas | - | ➤ Máximo 12 personas |
| Tiempos de Desarrollo | ➤ Cortos | ➤ Cortos ➤ Medianos ➤ Largos | ➤ Cortos ➤ Medianos ➤ Largos |

Tabla 3.1.3 Comparativa metodologías ágiles

Elaborado por: El investigador

• **Ventas y Desventajas**

| Metodología | Ventajas | Desventajas |
|--------------------|--|---|
| Mobile-D | <ul style="list-style-type: none"> ➤ Bajos costos de producción. ➤ Desarrollo del proyecto en el tiempo establecido. ➤ Generación de Entregables o demos | <ul style="list-style-type: none"> ➤ No es aplicable a proyectos grandes ➤ No existe una comunicación sólida. |
| SCRUM | <ul style="list-style-type: none"> ➤ El Cliente se incorpora directamente al proyecto. ➤ Se dividen las tareas para optimizarlas ➤ Aplicable a proyectos de cualquier tamaño. | <ul style="list-style-type: none"> ➤ Existe mucha dependencia de los gestores del proyecto. ➤ Compleja adaptación de los integrantes. |
| XP | <ul style="list-style-type: none"> ➤ Fomenta confianza en el equipo de desarrollo. | <ul style="list-style-type: none"> ➤ Incapaz de predecir errores durante el desarrollo. ➤ Costos un poco elevados. |

| Metodología | Ventajas | Desventajas |
|--------------------|---|--|
| | <ul style="list-style-type: none"> ➤ Eficiencia en los procesos. ➤ Flexible a cambios | <ul style="list-style-type: none"> ➤ Complejidad parecida a las metodologías tradicionales. |

Tabla 3.1.4 Ventajas y desventajas de las metodologías ágiles

Elaborado por: Christian Chasi

3.1.2.4. Metodología de desarrollo seleccionada

Después del análisis de las diferentes metodologías de desarrollo la mejor opción es Mobile-D ya que combina beneficios de metodologías como XP, Crystal y RUP y también se ajusta a las necesidades del proyecto y del investigador tomando las siguientes consideraciones:

1. La metodología Mobile-D está enfocada a aplicaciones móviles y el proyecto a desarrollar es justamente un aplicativo pequeño que no requiere de varios módulos o vistas.
2. Se adapta a cambios en el transcurso del desarrollo.
3. El estilo de desarrollo es iterativo, rápido y para grupos pequeños, lo que permitirá obtener y lanzar un producto de calidad a producción en pocas semanas.

3.1.3. Servicios de Notificaciones Push

Para determinar el mejor servicio de notificaciones push en primera instancia se ha definido tres de los mejores servicios según los sitios web populares como Capterra o Salesforce, los cuales son conocidos por sus reseñas, comentarios y valoraciones de productos o servicios de software a nivel mundial. Además, se considera la experiencia del investigador que ha trabajado con algunos de estos servicios push y las características de cada servicio tomadas de fuentes oficiales como sus páginas web.

3.1.3.1. Firebase Cloud Messaging (FCM)

Es un servicio de mensajería multiplataforma que entre varias de sus funciones cuenta con las de envío de notificaciones push desde un servidor hacia una o varias aplicaciones cliente [33].

- **Características:**

- Enviar mensajes de notificación o mensajes de datos.
- Segmentación de mensajes versátil
- Enviar mensajes desde aplicaciones cliente



Figura 3.1.6 Arquitectura Push FCM

Elaborado por: [33]

- **Flujo del ciclo de vida de FCM**

1. Desde el servidor se recibe y envía el mensaje.
2. Durante el envío del mensaje se envía una solicitud al Backend FCM.
3. Cuando el Backend recibe la solicitud se genera un ID de mensaje y otros metadatos para enviarlos a través de la capa de transporte.
4. Si está en línea el dispositivo recibe el mensaje o notificación.

- **Tipos de Mensajes**

FCM permite el envío de 2 tipos de mensajes a clientes:

1. Mensajes de notificación
2. Mensajes de datos, que son manejados por la aplicación cliente.

3.1.3.2. OneSignal

Es un servicio de mensajería que al igual que FCM ofrece notificaciones push móviles así como también notificaciones push web, mensajería en la aplicación, SMS y correo electrónico [34].

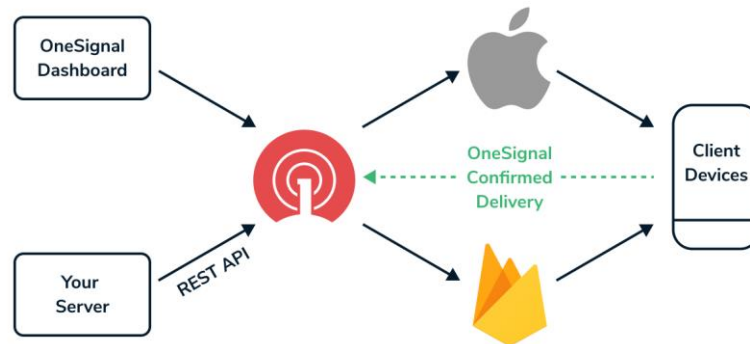


Figura 3.1.7 Arquitectura Push de OneSignal

Elaborado por: [46]

Lo mejor de utilizar este servicio es que tiene un plan gratuito para empresas de cualquier tamaño, sin embargo, tiene planes de pago adicionales que permiten utilizar más funcionalidades.

- **Servicios más relevantes de OneSignal**

1. API
2. Alertas móviles
3. Analítica en tiempo real
4. Notificaciones push web y móvil
5. Segmentaciones de todo tipo
6. Integraciones con varias plataformas

- **Compatible con múltiples plataformas como:**

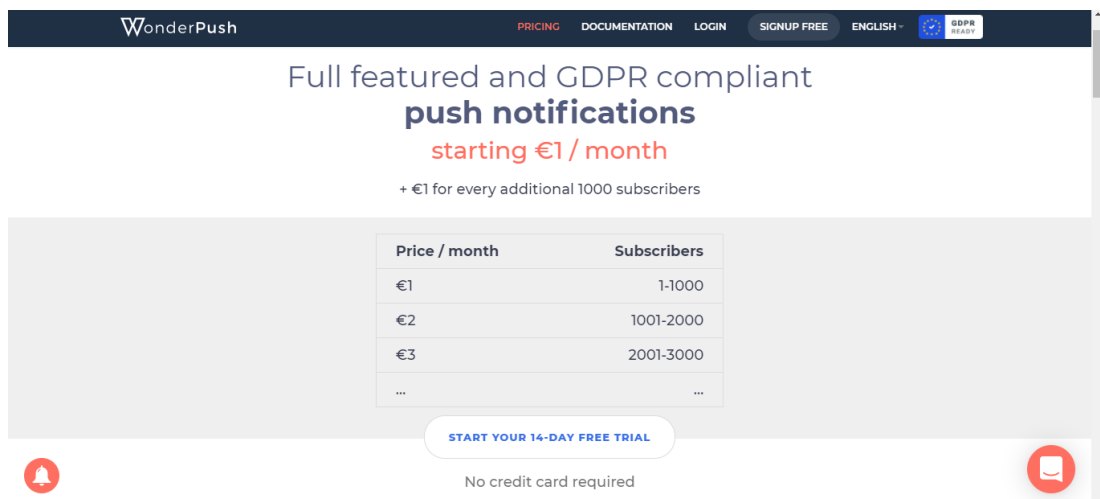
1. Unity
2. React Native
3. Cordova
4. Ionic & Ionic capacitor
5. PhoneGap
6. Flutter

7. Xamarin

3.1.3.3. Wonder Push

Es un servicio de notificaciones push para plataformas web y móviles. Ofrece servicios similares a los de OneSignal pero a diferencia de él, este no cuenta un servicio gratuito, sin embargo maneja precios muy económicos y con la facilidad de enviar más de 350 mil notificaciones por segundo haciéndolo muy eficiente para negocios de comercio electrónico o marketing [35].

Permite la personalización de sus notificaciones, así como la segmentación de usuarios para obtener mejores resultados.



The screenshot shows the WonderPush website with a dark blue header containing the logo and navigation links: PRICING, DOCUMENTATION, LOGIN, SIGNUP FREE, ENGLISH, and GDPR READY. The main content area features the text: "Full featured and GDPR compliant push notifications starting €1 / month" and "+ €1 for every additional 1000 subscribers". Below this is a table with two columns: "Price / month" and "Subscribers".

| Price / month | Subscribers |
|---------------|-------------|
| €1 | 1-1000 |
| €2 | 1001-2000 |
| €3 | 2001-3000 |
| ... | ... |

Below the table is a button that says "START YOUR 14-DAY FREE TRIAL" and the text "No credit card required". There are also two red circular icons at the bottom corners of the screenshot.

Figura 3.1.8 Página oficial Wonder Push

Elaborado por : [35]

- **Servicios más relevantes que ofrece Wonder Push**

1. Analítica en tiempo real
2. Segmentación
3. API

- **Plataformas populares compatibles**

1. Android, iOS
2. React, Ionic y Flutter
3. Wordpress, Shopify, y WooCommerce
4. Chrome y Firefox

3.1.3.4. Comparativa de servicios de notificaciones push.

| Criterio | FCM | OneSignal | Wonder Push |
|-----------------|--|---|--|
| Precio | <ul style="list-style-type: none"> ➤ Versión Gratuita ➤ Precio depende de la cantidad de uso del servicio | <ul style="list-style-type: none"> ➤ Versión Gratuita ➤ Precios desde \$9.99 mensual ➤ Cuenta con 3 planes ➤ | <ul style="list-style-type: none"> ➤ Prueba Gratuita de 14 días ➤ Precios desde \$1 por mes |
| Soporte | <ul style="list-style-type: none"> ➤ Online | <ul style="list-style-type: none"> ➤ Online ➤ 24/7 ➤ Horas de Trabajo | <ul style="list-style-type: none"> ➤ Chat Online |
| Servicios | <ul style="list-style-type: none"> ➤ Notificaciones push Web ➤ Notificaciones Push Móviles ➤ API | <ul style="list-style-type: none"> ➤ Notificaciones push Web ➤ Notificaciones Push Móviles ➤ API ➤ Marketing por Email ➤ Marketing Móvil ➤ Mensajería SMS y MMS | <ul style="list-style-type: none"> ➤ API ➤ Segmentación de audiencia. ➤ Análisis en tiempo real |
| Funciones | <ul style="list-style-type: none"> ➤ Sin capacidades de segmentación por geolocalización ➤ Pruebas A/B ➤ Integraciones limitadas y compatibilidad | <ul style="list-style-type: none"> ➤ Capacidades de segmentación por geolocalización en tiempo real ➤ Pruebas A/B ➤ Acceso ilimitado gratuito a la API ➤ Integraciones para iOS, | <ul style="list-style-type: none"> ➤ Integraciones para diferentes plataformas como Android, iOS, Chrome, Firefox, Wordpress, Woocommerce, Shopify, entre otros |

| Criterio | FCM | OneSignal | Wonder Push |
|-----------------|---|---|--|
| | con funciones de iOS ➤ Segmentación básica basada en datos demográficos y de ubicación del navegador | Android, Chrome, Firefox, entre otros. ➤ Segmentación avanzada de campañas, automatización de mensajes | ➤ Segmentación avanzada geográfica, eventos tags mediante país, idioma o zona horaria ➤ Gestión de contenido como almacenamiento de imágenes y plantillas |

Tabla 3.1.5 Comparativa de Servicios Notificaciones Push

Elaborado por : Christian Chasi y basado en [36].

• **Ventajas y Desventajas**

| Servicio Push | Ventaja | Desventaja |
|----------------------|--|---|
| OneSignal | ➤ Suscripciones móviles Ilimitadas ➤ Envío de notificaciones móviles ilimitadas. ➤ Sonidos de notificación personalizables ➤ Panel de administración muy intuitivo y usable ➤ Soporta varias plataformas para diferentes dispositivos ➤ Muy Buena Documentación | ➤ Depende de un id generado por Firebase para poder integrar sus servicios ➤ Si necesita más funcionalidades se debe pagar |

| Servicio Push | Ventaja | Desventaja |
|----------------------|--|---|
| FCM | <ul style="list-style-type: none"> ➤ Muy buena documentación ➤ Notificaciones push personalizables | <ul style="list-style-type: none"> ➤ No tiene soporte para iOS ➤ Si los datos crecen entonces deberá para más almacenamiento. |
| Wonder Push | <ul style="list-style-type: none"> ➤ Notificaciones push personalizables ➤ Integración a diversas plataformas tanto web como móvil. ➤ Buena documentación | <ul style="list-style-type: none"> ➤ Su periodo de prueba es limitado, posteriormente se debe pagar. |

Tabla 3.1.6 Ventajas y desventajas con proveedores de servicios push

Elaborado por: Christian Chasi

3.1.3.5.Servicio Push Seleccionado

Una vez analizado las principales características, ventajas y desventajas de cada uno de proveedores de servicios de notificaciones push se ha optado por escoger la opción más viable para el desarrollo de proyecto, la cuál es OneSignal, ya que al ser un proyecto de contexto social involucra grandes cantidades de dispositivos móviles y datos. Por lo tanto, su plan gratuito permite suscripciones y notificaciones personalizadas ilimitadas a dispositivos móviles, también el uso gratuito de su API y muchas otras características que se ajustan al modelo del proyecto.

3.1.4. Análisis de tecnologías Frontend

Para el desarrollo del Frontend del aplicativo se escogió una alternativa híbrida pensando en futuros desarrollos que sean compatibles principalmente para los 2 sistemas operativos móviles android e iOS. No obstante, hay que definir las tecnologías o frameworks para la construcción del aplicativo y para ellos es necesario analizarlos.

3.1.4.1.Ionic

Es un conjunto de interfaces de usuario de código abierto para el desarrollo de aplicaciones móviles y de escritorio basado en tecnologías web como HTML, CSS y JavaScript con la capacidad de integrarse con otros frameworks como Angular, React y Vue [37]. Como menciona en su página oficial Ionic ha colaborado en el desarrollo de más de 5 millones de aplicaciones móviles. Su funcionamiento se basa en una tecnología que se conoce como WebView, esta forma parte de dispositivos donde se ejecuta web, mostrando una aplicación como nativa.

Para la construcción de aplicaciones móviles en muchos de los casos es esencial el uso de los elementos nativos del dispositivo (Cámara, GPS, Sensores), Ionic proporciona acceso a estos elementos a través de plugins de Apache Cordova, estos actúan como puente entre la vista web y el sistema operativo [38].

A continuación, se muestra un gráfico que resume el funcionamiento de Ionic:

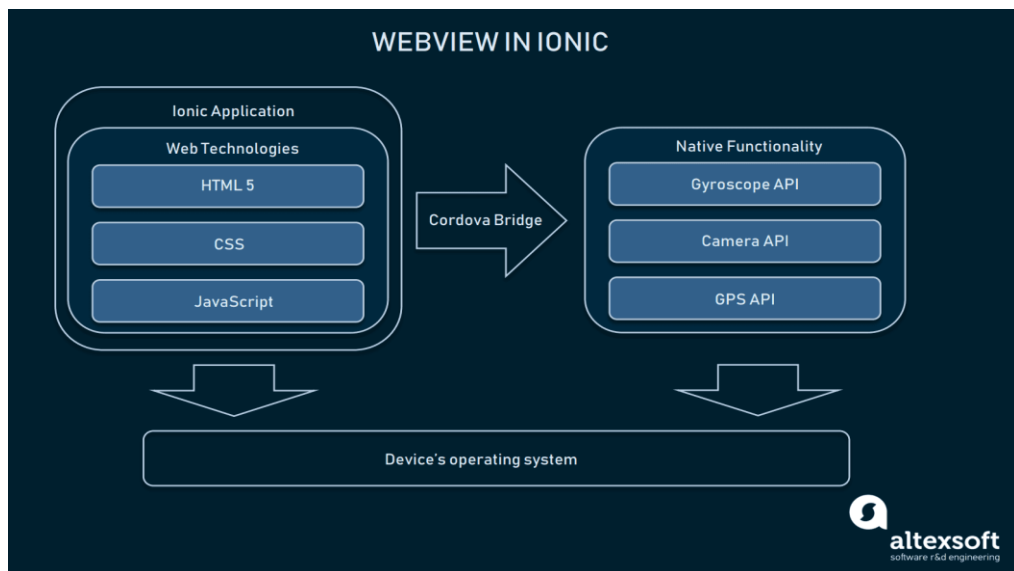


Figura 3.1.9 Diagrama que representa el funcionamiento de Ionic

Elaborado por : [38]

- **Ventajas y Desventajas de Ionic**

| Ventajas | Desventajas |
|--|--|
| Con una única base de código se puede trabajar en cualquier dispositivo o plataforma que ejecute web. | Bajo rendimiento en aplicaciones específicamente que utilicen muchos gráficos. |
| La interfaz se adapta a cualquier plataforma destino | Depende de plugins para utilizar cualquier elemento nativo. |
| Bajos costos de desarrollo y mantenimiento ya que la misma aplicación funciona para otras plataformas y no dependen de muchos programadores. | Requiere de WebView. |
| Al basarse en Javascript, uno de los lenguajes número uno del desarrollo web, no habrá dificultad de encontrar programadores. | - |
| Amplia documentación y una gran comunidad que continúa creciendo | - |

Tabla 3.1.7 Ventajas y desventajas de Ionic

Elaborado por: Christian Chasi

3.1.4.2.React Native

Es un conjunto de herramientas de código abierto creada por Facebook para revolucionar el desarrollo de aplicaciones móviles nativas multiplataforma. Se compila de forma nativa en JavaScript haciéndola compatible en dispositivos con sistema operativo iOS y Android. Está basado en ReactJs, una librería utilizada para el desarrollo web la cual ha crecido con el apoyo de su comunidad y de diferentes organizaciones lo que la ha hecho muy popular [39].

Aunque también está basado en Javascript, React a diferencia de Ionic y otras tecnologías como PhoneGap y Cordova no utiliza el WebView dado que este framework tiene acceso a los controladores móviles nativos [40].

Características

1. React utiliza las APIs nativas para renderizar los componentes, es decir para Android utiliza Java y para iOS utiliza Objective C.
2. No utiliza HTML ni CSS, las animaciones se ejecutan con APIs especiales de la librería.

Para entender mejor la diferencia entre el renderizado de otros frameworks como Ionic contra React Native a continuación se muestra la siguiente figura:

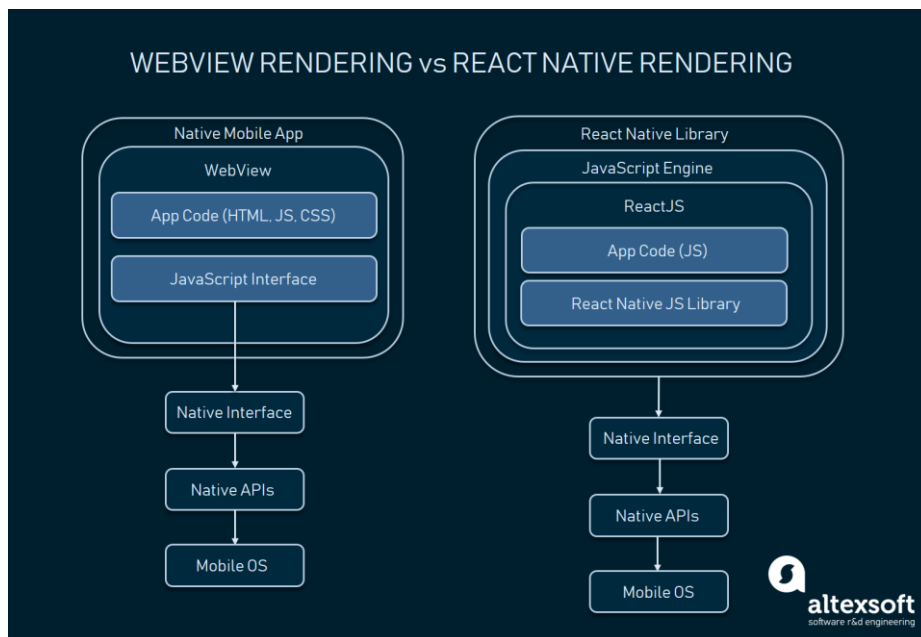


Figura 3.1.10 Renderizado WebView vs Renderizado ReactNative

Elaborado por: [40]

- **Ventajas y desventajas de React Native**

| Ventajas | Desventajas |
|---|---|
| Código reusable. | Documentación Deficiente |
| Sus componentes y módulos nativos escritos en Java y Objective C mejoran el rendimiento de la aplicación. | Mayores costos de desarrollo y mantenimiento ya que necesitas de expertos en desarrollo nativo. |

| Ventajas | Desventajas |
|--|---|
| No utiliza WebView. | Problemas cuando android o iOS actualizan su SDK. |
| Tiene una amplia comunidad de desarrolladores que aportan y continúan creciendo para mejorar el framework. | A través de la comunidad a veces reportan problemas de compatibilidad con otras librerías o durante el proceso de desarrollo. |

Tabla 3.1.8 Ventajas y desventajas de React Native

Elaborado por: Christian Chasi

3.1.4.3.Flutter

Es un SDK de código abierto creado por Google para desarrollar aplicaciones multiplataforma para iOS y Android. Utiliza el lenguaje de programación Dart y se enfoca a plataformas móviles, web y de escritorio con un único código base. Flutter trabaja en función de Widgets personalizables que permiten crear interfaces de usuario nativas [39].

Flutter al igual que React Native, cuenta con hot reload o carga en caliente lo que permite al desarrollador observar los cambios al instante durante la ejecución del emulador, siendo una herramienta útil y rápida para crear aplicaciones móviles [39].

Características

1. Los widgets sirven en toda la fase de desarrollo.
2. Flutter a diferencia de Ionic y React Native, utiliza el lenguaje de programación Dart.
3. Con la nueva versión ahora se puede crear aplicaciones web, móviles y de escritorio.

Como se muestra en la siguiente figura Flutter está diseñado para trabajar por capas, y según en la documentación cada capa puede ser opcional y reemplazable.

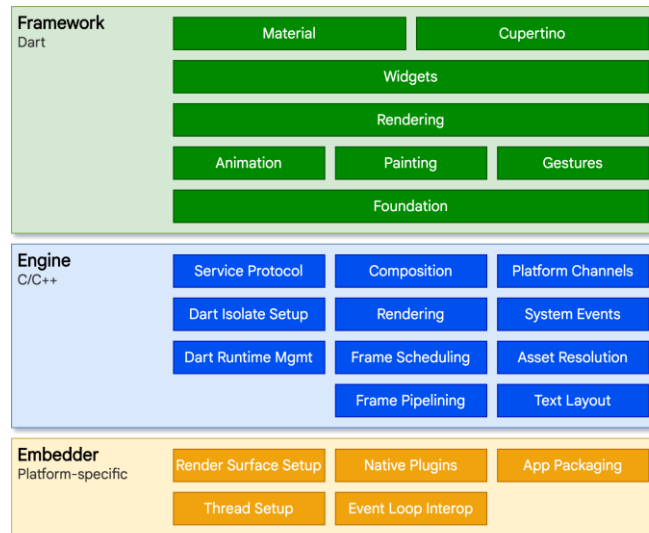


Figura 3.1.11 Arquitectura de Flutter

Elaborado por: [47]

- **Ventajas y Desventajas de Flutter**

| Ventajas | Desventajas |
|--|--|
| Los widgets permiten un rápido desarrollo de la interfaz y codificación de la aplicación. | Cuenta con un limitado número de librerías o plugins a comparación de Ionic o React Native. |
| Utiliza el lenguaje de programación Dart que es capaz de compilar para móvil, web y escritorio lo que lo hace independiente de utilizar puentes de comunicación como React Native. | No cuenta con una comunidad muy grande ya que al ser una tecnología lanzada en el año 2017 y recientemente con la versión 2 estabilizada no hay mucho soporte. |
| Al igual que Ionic y React Native cuenta con hot reload lo que permite un desarrollo rápido. | Tiene una alta curva de aprendizaje ya que usa el lenguaje Dart que no es uno de los más populares. |

| Ventajas | Desventajas |
|--|-----------------------------------|
| Sus widgets se componen de diseños como material de android y cupertino de iOS. | Las aplicaciones son más pesadas. |
| Los widgets permiten un rápido desarrollo de interfaz y codificación de la aplicación. | - |
| Es el que más se acerca al desarrollo de aplicaciones nativas por su alto rendimiento. | - |

Tabla 3.1.9 Ventajas y Desventajas de Flutter

Elaborado por: Christian Chasi

3.1.4.4. Análisis comparativo de Ionic, React Native y Flutter

| Criterio | Ionic | React Native | Flutter |
|--------------------------|---|---|---|
| Plataformas Compatibles | <ul style="list-style-type: none"> ➤ iOS ➤ Android ➤ Web | <ul style="list-style-type: none"> ➤ iOS ➤ Android ➤ Web | <ul style="list-style-type: none"> ➤ Android ➤ iOS ➤ Web ➤ Escritorio |
| Lenguaje de Programación | <ul style="list-style-type: none"> ➤ HTML ➤ CSS ➤ Javascript ➤ Typescript | <ul style="list-style-type: none"> ➤ Javascript + Swift + Objective-C + Java ➤ | <ul style="list-style-type: none"> ➤ Dart |
| Soporte de la Comunidad | <ul style="list-style-type: none"> ➤ Alto | <ul style="list-style-type: none"> ➤ Alto | <ul style="list-style-type: none"> ➤ Medio (Tecnología joven) |

| Criterio | Ionic | React Native | Flutter |
|------------------------|---|---|-------------------------------------|
| Experiencia de usuario | ➤ Medio | ➤ Medio-Alto | ➤ Alto |
| Precio | ➤ Gratuito/ Código Fuente Abierto ➤ Plan de Pago | ➤ Gratuito/ Código Fuente Abierto | ➤ Gratuito/Código Fuente Abierto |
| Código Reusable | ➤ 90% del código | ➤ 90% del código | ➤ 50% del código |

Tabla 3.1.10 Comparativa frameworks de aplicaciones móviles

Elaborado por: Christian Chasi y basado en [39]

3.1.4.5. Tecnología Frontend seleccionada

Para el desarrollo de la aplicación y basándose en el análisis de cada una de las tecnologías propuestas el que mejor se adapta al proyecto es Ionic debido a las siguientes consideraciones:

1. Tiene mayor documentación y respaldo a través de su enorme comunidad de desarrolladores que han aportado al crecimiento de este framework y al entorno de javascript.
2. Tiene componentes diseñados para las interfaces, permite la reutilización del código obteniendo código legible, fácil de mantener y reduciendo el tiempo de desarrollo.
3. Trabaja con frameworks como Angular, Vue, y React reduciendo la curva de aprendizaje en estos últimos años.
4. Se basa en el lenguaje de programación Javascript actualmente líder del mercado teniendo a disposición amplia documentación y soporte.
5. Finalmente se ajusta a las características del proyecto ya permite crear aplicaciones livianas y pequeñas que no requieran de mayores recursos del dispositivo, obteniendo un excelente rendimiento.

3.1.5. Análisis de tecnologías Backend

Generalmente las aplicaciones cliente independientemente de que plataforma sea (web, móvil o escritorio) en muchos casos utilizan la misma información o funcionalidades que permitan gestionar los datos. Por esa razón es necesario crear sistemas que atiendan a las solicitudes de cada dispositivo cliente o Frontend, ahí es donde surgen los lenguajes o frameworks para Backend.

Se realizará un breve análisis de estas las tecnologías para determinar cuál se ajusta al proyecto de investigación.

3.1.5.1.Laravel

Es un framework escrito en PHP utilizado comúnmente en el desarrollo de aplicaciones web. Según Stack Overflow y GitHub dos plataformas web top a nivel de software dan a conocer a Laravel como el framework PHP para Backend más popular [41].

Características

1. Utiliza una arquitectura MVC.
2. Cuenta con un gran gestor de dependencias dedicado.
3. Facilidad de conexión a varias bases de datos.
4. Cuenta con una comunidad activa y gran documentación.

Ventajas y desventajas de Laravel

| Ventajas | Desventajas |
|--|--|
| Curva de aprendizaje bajo, ya que utiliza PHP. | Se necesitan de programadores experimentados o expertos. |
| Es confiable ya que posee licencia Massachusetts Institute of Technology (MIT) y está alojado en GitHub. | Es fácil de aprender, pero complejo de dominar. |
| Amplio apoyo de la comunidad de desarrollares en GitHub. | Problemas con las actualizaciones y versiones. |

| Ventajas | Desventajas |
|--|-------------------------|
| Migración de datos de la Base de Datos fluida. | Tiene soporte limitado. |

Tabla 3.1.11 Ventajas y desventajas de Laravel

Elaborado por: Christian Chasi

3.1.5.2. Node.js y Express

Node.js es utilizado ampliamente en el desarrollo Backend, se basa en el motor de JavaScript V8 de Chrome, siendo el preferido para desarrollos con frameworks que se basen en javascript. Express trabaja juntamente con Node.js aprovechando su rendimiento que lo hace óptimo en tareas síncronas y asíncronas [41].

Características de Node.js y Express

1. Es el preferido por los desarrolladores en stacks javascript como MongoDB, Express, Angular y Node.js (MEAN) o MongoDB, Express, React y Node.js (MERN) o también MongoDB, Express, Vue y Node.js (MEVN).
2. Es útil para aplicaciones con múltiples llamados a servicios REST o APIs ya que es bastante rápida.
3. Empresas como Trello, Netflix, PayPal, Uber y LinkedIn utilizan Node.js.

- **Ventajas y desventajas de Node.js y Express**

| Ventajas | Desventajas |
|---|---|
| Rápido desarrollo gracias a la compilación rápida que permite ver los cambios en milisegundos. | Reduce el rendimiento en tareas de cálculos pesadas. |
| Al trabajar el lado del cliente y servidor con JavaScript el desarrollo es más flexible y rápido. | El código no es muy mantenible cuando se manejan tareas asíncronas. |
| Código reutilizable y curva de aprendizaje fácil. | - |
| Procesa solicitudes sin demora ya que no bloquea E/S de procesos, es decir es asíncronico. | - |

Tabla 3.1.12 Ventajas y desventajas de Node.js y Express

Elaborado por: Christian Chasi

3.1.5.3.Ruby on Rails

Es un framework de desarrollo web construido bajo el lenguaje de programación Ruby, es popular por su baja curva de aprendizaje y su propio entorno de paquetes personalizados, lo que le diferencia de Node.js puesto que no utiliza paquetes de terceros [41].

Características

1. Permite manipular fácilmente las Bases de Datos sin depender de paquetes de terceros.
2. Empresas como GitHub, Twiter y Aribnb utilizan este framework.

Ventajas y Desventajas de Ruby on Rails

| Ventajas | Desventajas |
|--|---|
| Utiliza el modelo MVC. | Dificultad de aplicar una nueva actualización o característica del framework. |
| Es muy rápido para programar ya que tiene sus propios módulos incorporados y no depende de terceros. | La compilación es lenta en comparación con frameworks como Node.js. |
| Amplia comunidad que respalda al framework. | Dificultad en la migración de bases de datos. |
| Es apto para aplicar técnicas de programación limpia como patrones de diseño o principios SOLID. | Al no ser muy populares se requiere de desarrolladores expertos por lo que resulta más costoso. |

Tabla 3.1.13 Ventajas y desventajas de Ruby on Rails

Elaborado por: Christian Chasi

3.1.5.4. Frameworks y lenguajes más utilizados por los desarrolladores

Como es costumbre de todos los años la popular plataforma web Stack Overflow muestra el resultado de sus estudios y encuestas aplicadas a los desarrolladores, dando a conocer en la siguiente figura el resultado de los marcos de desarrollo más utilizados del 2021, entre ellos se destaca el framework Express que está por encima de Laravel y Ruby previamente analizados.

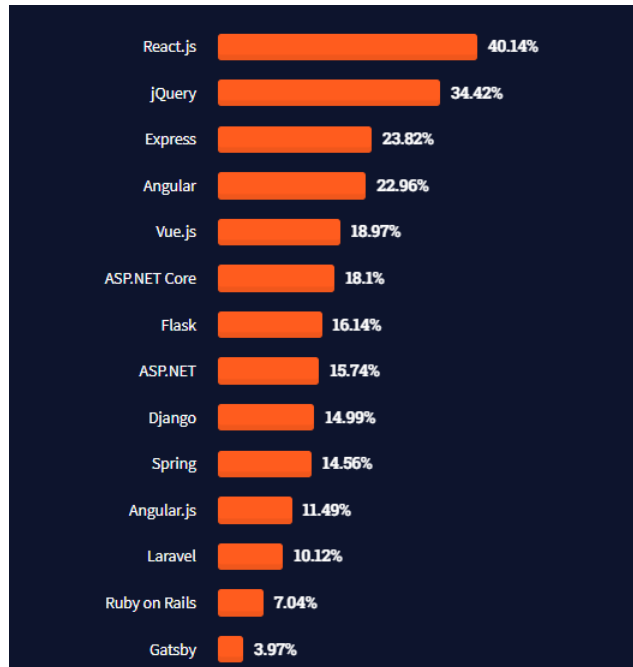


Figura 3.1.12 Frameworks más utilizados por desarrollares en el 2021

Elaborado por: [48]

De la misma manera se visualiza los lenguajes de programación, lenguajes de marcado y entornos de desarrollo más populares por la comunidad de desarrolladores. En la siguiente figura se puede evidenciar a Javascript como líder de todos los lenguajes por encima de PHP y Ruby. Además, se destaca a Node.js entre los 6 primeros lugares. La lista es más grande por lo que no se puede mostrar absolutamente todos.

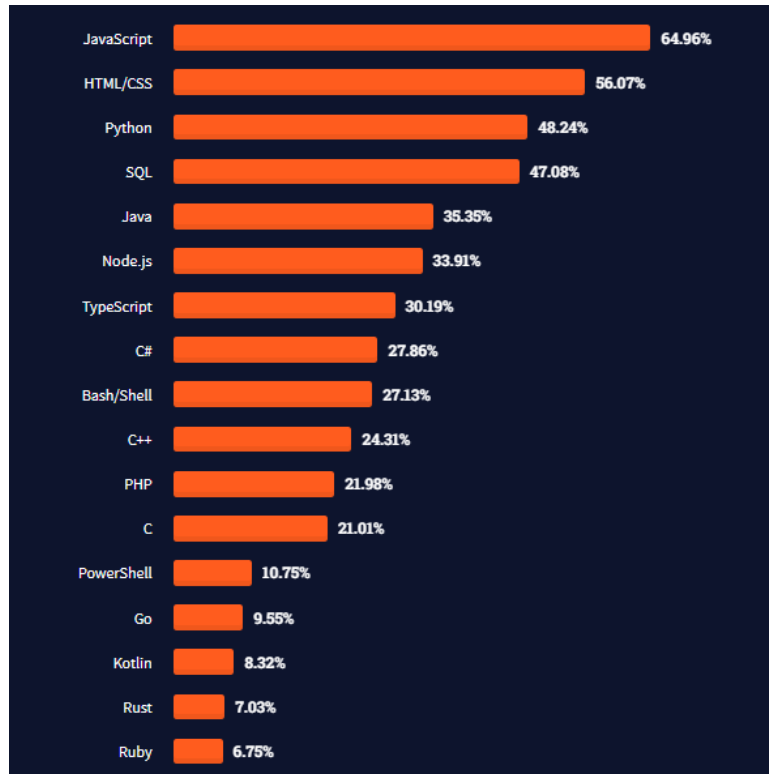


Figura 3.1.13 Lenguajes y entornos de desarrollo preferido por los desarrolladores

Elaborado por: [48]

3.1.5.5. Tecnología Backend seleccionada

Después de todo el análisis para determinar la tecnología Backend se ha optado por Node.js con Express tomando las siguientes consideraciones:

1. La aplicación cliente que involucra a varios usuarios necesariamente deberá realizar llamadas a las APIs debido a que es una aplicación de alertas de auxilio para la ciudadanía. Por lo tanto Node.js maneja eficientemente varios procesos al mismo tiempo sin tener que esperar a que terminen uno por uno, lo que lo hace la mejor alternativa.
2. La aplicación cliente será desarrollada en Ionic con el framework angular basado en Javascript lo que lo hace flexible y muy rápido tener en el cliente y servidor el mismo lenguaje.
3. La curva de aprendizaje y costos de desarrollo son considerablemente bajos.

3.1.6. Análisis de Bases de Datos

Las bases de datos son un conjunto de datos estructurados almacenados en un sistema informático también conocido como Sistema de Gestión de Base de Datos (SGBD). Tiene sus inicios en la década de los 80's en dónde se aplican los primeros modelos relacionales y manipulado de datos a través de lenguaje SQL. Posteriormente en la década del 2000 surgieron las bases de datos no relacionales que utilizaban un lenguaje de consulta diferente [42].

3.1.6.1. Bases de Datos SQL

También son conocidas como Bases de Datos Relacionales, se caracterizan porque almacenan la información mediante tablas compuestas de filas y columnas. Permiten la gestión de datos a través de un lenguaje consultas SQL.

En la siguiente tabla se muestra una comparativa de 3 Bases de Datos populares que son SQL Server de Microsoft, Oracle y MySQL.

| Producto/Característica | SQL Server | Oracle RDBMS | MYSQL |
|--------------------------------|--|--|---|
| Sistema Operativo | <ul style="list-style-type: none"> ➤ Windows Server ➤ Windows Clientes | <ul style="list-style-type: none"> ➤ Windows ➤ Unix ➤ Linux | <ul style="list-style-type: none"> ➤ Windows ➤ Unix ➤ Linux ➤ Mac |
| Controladores | <ul style="list-style-type: none"> ➤ ODBC, ➤ JDBC, ➤ ADO.NET, ➤ OLEB | - | <ul style="list-style-type: none"> ➤ ODBC, ➤ JDBC, ➤ ADO.NET |
| Licencia | <ul style="list-style-type: none"> ➤ Código Fuente Cerrado | <ul style="list-style-type: none"> ➤ Código Fuente Cerrado | <ul style="list-style-type: none"> ➤ Código Fuente Abierto ➤ GNU-GPL |

| Producto/Característica | SQL Server | Oracle RDBMS | MYSQL |
|-------------------------|------------------|-------------------|-----------------|
| Transacciones | ➤ Si | ➤ Si | ➤ Usando InnoDB |
| Lenguajes | ➤ SQL ➤ T-SQL | ➤ SQL ➤ PL-SQL | ➤ SQL |

Tabla 3.1.14 Comparativa Bases de Datos relacionales

Elaborado por: [42]

3.1.6.2. Bases de Datos No SQL

Las bases de datos No SQL se caracterizan porque a diferencia de las Bases de Datos Relacionales, estas no almacenan los datos en tablas sino en colecciones o documentos. Están diseñadas para optimizar el almacenamiento y recuperación de información utilizando su propio lenguaje de consultas diferente a SQL. Ejemplos de bases de datos No SQL son MongoDB, Cassandra y CouchDB [43].

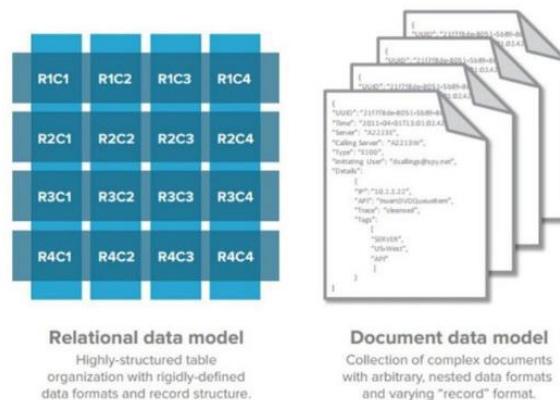


Figura 3.1.14 Almacenamiento de datos SQL y No SQL

Elaborado por: [43]

- **Diferencias entre SQL y No SQL**

| Criterio | SQL | No SQL |
|-------------------------|----------|--------------|
| Almacenamiento de datos | ➤ Tablas | ➤ Documentos |

| Criterio | SQL | No SQL |
|-----------------------|-----------------------|-------------------------------|
| Organización de datos | ➤ Esquema predefinido | ➤ Esquema dinámico |
| Escalabilidad | ➤ Vertical (Hardware) | ➤ Horizontal (Servidores) |
| Lenguaje | ➤ SQL | ➤ Lenguaje de consulta propio |

Tabla 3.1.15 Diferencias entre SQL y No SQL

Elaborado por: [42]

3.1.6.3. Análisis de rendimiento entre bases de datos SQL y No SQL

Para evaluar el rendimiento de las bases de datos SQL y No SQL en base a los experimentos del autor [42] se obtiene.

- **Rendimiento de consultas de bases de datos con 10.000 registros en milisegundos.**

| Type/Operation | Oracle | MySql | MsSql | Mongo | Redis | GraphQL | Cassandra |
|----------------|--------|-------|-------|-------|-------|---------|-----------|
| Insert | 0.076 | 0.093 | 0.093 | 0.005 | 0.009 | 0.008 | 0.011 |
| Update | 0.077 | 0.058 | 0.073 | 0.008 | 0.013 | 0.007 | 0.013 |
| Delete | 0.059 | 0.025 | 0.093 | 0.01 | 0.021 | 0.017 | 0.018 |
| Select | 0.025 | 0.093 | 0.062 | 0.009 | 0.016 | 0.010 | 0.014 |

Figura 3.1.15 Rendimiento consultas de bases de datos con 10.000 registros

Elaborado por: [42]

- **Rendimiento de consultas de bases de datos con 100.000 registros en milisegundos.**

| Type/Operation | Oracle | MySql | MsSql | Mongo | Redis | GraphQL | Cassandra |
|----------------|--------|-------|-------|-------|-------|---------|-----------|
| Insert | 0.091 | 0.038 | 0.093 | 0.005 | 0.010 | 0.008 | 0.011 |
| Update | 0.092 | 0.068 | 0.075 | 0.009 | 0.013 | 0.012 | 0.014 |
| Delete | 0.119 | 0.047 | 0.171 | 0.015 | 0.021 | 0.018 | 0.019 |
| Select | 0.062 | 0.067 | 0.060 | 0.009 | 0.015 | 0.011 | 0.014 |

Figura 3.1.16 Rendimiento consultas de bases de datos con 100.000 registros

Elaborado por: [42]

- **Resultados del experimento con relación al rendimiento de BD en milisegundos.**

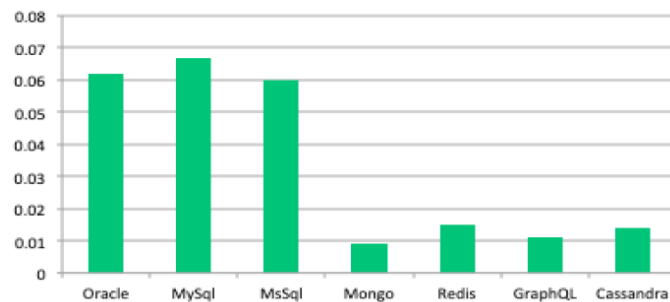


Figura 3.1.17 Resultados del rendimiento de Bases de Datos

Elaborado por: [42]

El autor concluye lo siguiente [42]:

1. Las Bases de Datos No SQL son mucho más rápidas en operaciones de lectura y escritura.
2. Las bases de datos relacionales o SQL almacenan los datos en tablas y también aplican reglas de normalización para evitar la redundancia de datos mejorando el rendimiento de las consultas.
3. Las Bases de Datos No SQL están diseñadas para trabajar con grandes volúmenes de datos.

- **Bases de datos preferidas por desarrolladores**

Según los resultados de los estudios de la plataforma web Stack Overflow las bases de datos más utilizadas son MySQL y PostgreSQL las mismas que son Bases de Datos SQL. Sin embargo, en el cuarto lugar se pueden observar a MongoDB una Base de Datos no SQL que estos últimos años se ha mantenido en crecimiento, siendo una de las preferidas por los desarrolladores que manejan pilas o stacks con Javascript.

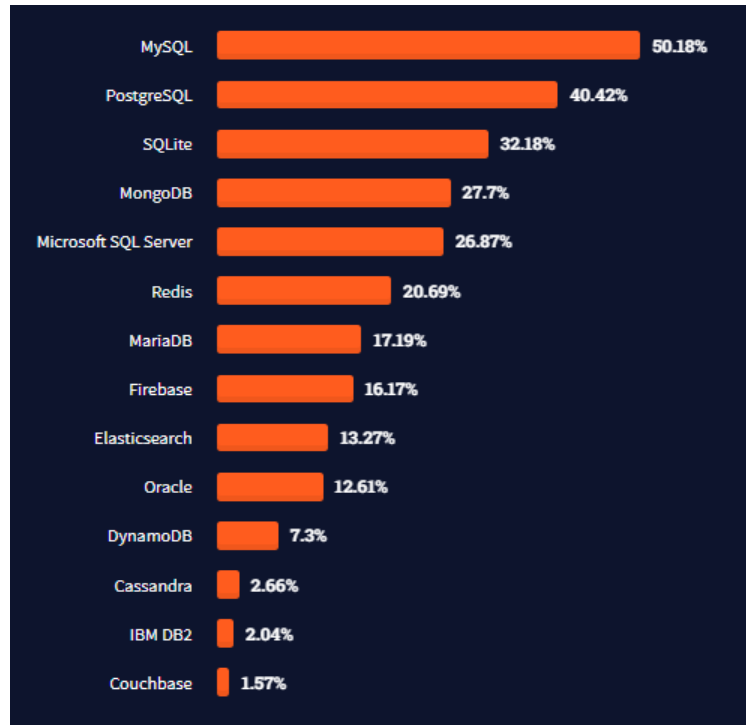


Figura 3.1.18 Bases de Datos preferidas por desarrolladores

Elaborado por: [48]

3.1.6.4. Bases de Datos con servicios en línea

Se plantea hacer uso de servidores gratuitos en internet entre ellos servidores de Bases de Datos, en ese sentido se analizarán plataformas que alojen Bases de Datos y tengan un plan gratuito.

Clever Cloud

Es una plataforma o servicios en la nube para gestionar los datos, cuentan con servicios de almacenamiento para Bases de Datos como MYSQL, POSTGRESQL, MONGODB, CASSANDRA, entre otros.

MongoDB Atlas

Es una de las más grandes y modernas plataformas de Base de Datos de MongoDB en la nube. Trabaja con Amazon Web Services (AWS), Google Cloud Platform (GCP) y Azure permitiendo automatizar y garantizar la disponibilidad y escalabilidad de los servicios en todo momento.

- **Análisis entre Clever Cloud y Mongo Atlas**

| Criterio | Clever Cloud | | MongoDB Atlas |
|----------------------------------|-------------------------------|-----------------------------------|-------------------------------|
| | MYSQL | Mongo DB | Mongo DB |
| Precios | Plan Gratuito y Desde \$5/mes | Plan Gratuito y Desde \$17,50/mes | Plan Gratuito y Desde 9\$/mes |
| Almacenamiento Máximo (Gratuito) | 10 MB | 500MB | 512MB |
| RAM | Compartida | Compartida | Compartida |
| vCPU | Compartida | Compartida | Compartida |
| Métricas | No | No | Si |

Tabla 3.1.16 Comparativa de costos y servicios de Bases de Datos en línea

Elaborado por: Christian Chasi

3.1.6.5. Bases de Datos elegida

Después de analizar las Bases de Datos SQL y No SQL, la alternativa que más se ajusta al proyecto de desarrollo es MongoDB, también se han tomado las siguientes consideraciones:

1. Es rápida para procesar grandes cantidades de información, al ser un proyecto de contexto social, en el presente y pensando a futuro involucra miles usuarios y datos.
2. Es Open Source y muy flexible para trabajar con el stack de tecnologías javascript seleccionada del lado del cliente con Ionic y Angular y del lado del servidor con Node.js y Express.js. (MEAN).
3. En cuanto a costos la mejor opción es MongoDB Atlas que proporciona un plan gratuito conveniente, además de que sus servidores están vinculados con los de Amazon, Google y Microsoft generando disponibilidad, escalabilidad y confianza.

3.2.Desarrollo de la propuesta

Como se definió previamente, se utilizará la metodología Mobile-D en la que se implementarán las siguientes fases:

- **Fase I: Exploración**

1. Definición de grupos de interés
2. Definición de alcance
3. Establecimiento del proyecto

- **Fase II: Inicialización**

1. Configuración del proyecto
2. Planificación inicial
3. Historias de Usuario
4. Planificación de Fases
5. Arquitectura de la aplicación
6. Base de Datos

- **Fase III: Producción**

1. Iteraciones (día de planificación, día de trabajo y día de liberación)

- **Fase IV: Estabilización**

1. Integración de todos los módulos

- **Fase V: Pruebas**

1. Pruebas Unitarias
2. Pruebas de Aceptación

3.2.1. Propuesta general del funcionamiento del sistema y sus actores.

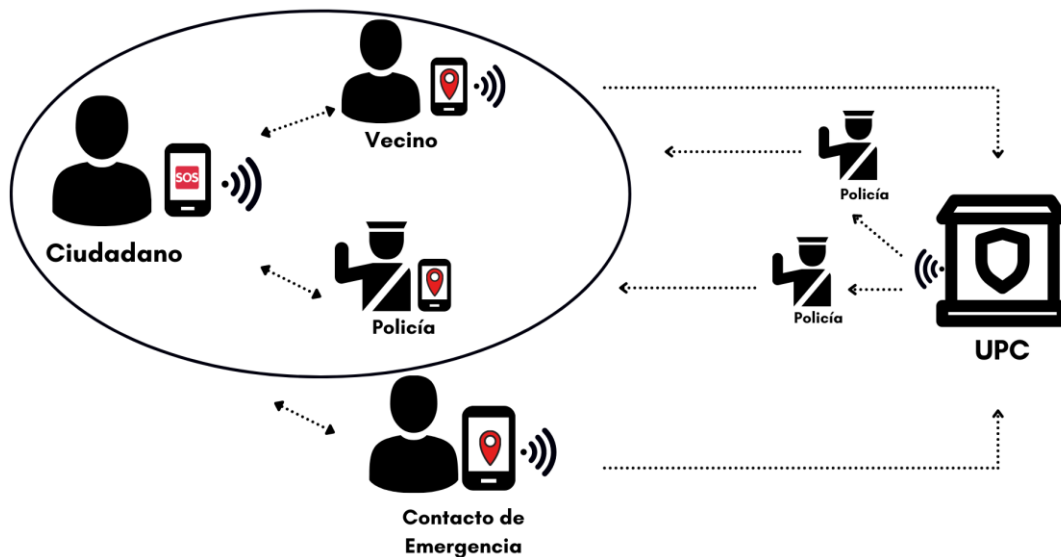


Figura 3.2.1 Diagrama general del sistema

Elaborado por: Christian Chasi

- **Descripción de la propuesta general del sistema**

| Actor | Descripción o Función |
|--------------------------------|---|
| Ciudadano | Persona o ciudadano que utiliza la aplicación móvil para enviar alertas de auxilio en base a su ubicación y un radio de distancia establecido. |
| Vecino | Persona que recibe la alerta de auxilio (siempre y cuando este en el radio de distancia) y decide si puede ayudar en conjunto con los demás vecinos o avisar a la UPC más cercana o a la policía mediante otros medios de comunicación. |
| Policía | Persona o ente que mediante otro aplicativo móvil de monitoreo de alertas recibe la alerta de auxilio del ciudadano. |
| Contactos de Emergencia | Persona que también recibe la alerta de auxilio (sin estar en el radio de distancia) y decide ayudar o avisar a la policía mediante otros medios de comunicación. |

Tabla 3.2.1 Descripción de la propuesta general del diagrama

Elaborado por: Christian Chasi

3.2.2. Definición del modelo de proceso

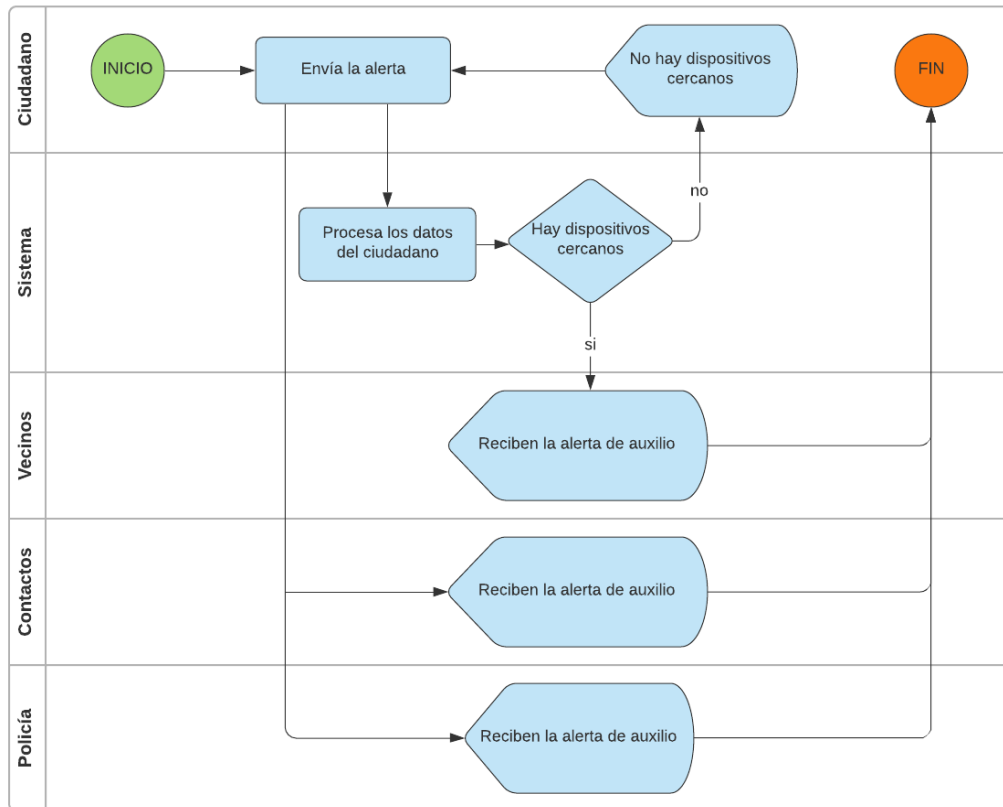


Figura 3.2.2 Diagrama de procesos del sistema para el envío de alertas de auxilio

Elaborado por: Christian Chasi

3.2.3. Fase I: Exploración

En esta fase se definen las bases del proyecto tales como grupos interesados, requerimientos, recursos y limitaciones que permitan conocer hasta dónde se llegará con la aplicación. Todo esto con el objetivo de que las siguientes fases se ejecuten exitosamente.

3.2.3.1. Definición de los grupos de interés

- **Ciudadanos de la parroquia Sta. Rosa**

Son los usuarios principales que utilizarán el aplicativo para alertar o recibir llamados de auxilio entre la comunidad del barrio.

- **GAD de la parroquia Sta. Rosa**

Es la entidad que ayudará a promover el uso del aplicativo planificando reuniones y capacitaciones. También proveerá información que ayudará al investigador a cumplir con los objetivos del proyecto.

- **Equipo de Desarrollo**

1. Programador: Christian Chasi
2. Tutor del proyecto: Ing. Julio Balarezo

3.2.3.2. Definición del alcance

- **Levantamiento de requerimientos iniciales**

En esta etapa se busca definir brevemente los requerimientos básicos de la aplicación. Los requerimientos son definidos por el investigador, y los ciudadanos que utilizarán el aplicativo, todo esto con el objetivo de tener un aplicativo seguro, intuitivo y que satisfaga las necesidades de los involucrados.

En cuestión de recursos físicos para montar servicios y la Base de Datos hay limitaciones ya que el aplicativo hará uso de servidores gratuitos en internet.

- **Requerimientos Funcionales**

Aplicación Móvil – Barrio Seguro

1. Registro de Usuarios.
2. Autenticación de usuarios.
3. Envío de alertas de auxilio.
4. Notificaciones Push.
5. Mapas y Geolocalización.
6. Rangos de distancia de las alertas de auxilio.
7. Contactos de emergencia.
8. Historial de alertas.
9. Administración del perfil de usuario.
10. Restablecimiento de contraseña.

Aplicación móvil – Monitoreo de alertas

1. Registro de Usuarios.
2. Autenticación de usuarios.
3. Perfil de Usuario.
4. Mapas y Geolocalización
5. Historial de alertas.
6. Restablecimiento de la contraseña.

- **Delimitación del Proyecto**

En base a las herramientas de desarrollo, requerimientos e instrumentos de investigación se han establecido algunas limitaciones en el uso del aplicativo.

1. El aplicativo para los ciudadanos de los barrios, por el momento solo estará disponible para dispositivos móviles con sistema operativo android a partir de la versión 5 en adelante y se podrá descargar desde la Play Store de Google.
2. El dispositivo deberá tener internet fijo o datos móviles para el envío de alertas de auxilio.
3. El aplicativo **Barrio Seguro** es para que lo utilicen personas de barrios organizados, colaborativos y participativos de la parroquia Sta. Rosa.
4. El aplicativo de **Monitoreo de Alertas** surge como propuesta por parte de los ciudadanos para que lo utilicen las autoridades o entes que tengan la obligación de prestar ayuda.
5. Se utilizarán servidores gratuitos para la Base de Datos y los servicios REST, por lo tanto, tendrá limitaciones en cuanto al almacenamiento de la información.

- **Supuestos**

1. Si la aplicación recibe financiación de las autoridades se trabajaría para eliminar las limitaciones del sistema.

3.2.3.3. Establecimiento del proyecto

En esta etapa se establecen y definen ciertas características relevantes para el desarrollo del proyecto, como tecnologías de desarrollo, IDEs, lenguajes, librerías y demás.

| Tecnología | Descripción |
|----------------------------|---|
| Sistema Operativo | ➤ Windows 10 Home |
| Lenguaje/s de Programación | ➤ Typescript ➤ Javascript |
| IDE de Desarrollo | ➤ Visual Studio Code ➤ Android Studio |
| Tecnologías Cliente | ➤ Ionic ➤ Angular |
| Tecnologías Servidor | ➤ Node.js ➤ Express ➤ MongoDB |
| Hardware | ➤ Computador Portátil MSI, procesador i7, RAM 16GB, SSD 256GB, HDD 1TB ➤ Xiaomi Redmi Note 8, android 10 |

Tabla 3.2.2 Tecnologías para el desarrollo del proyecto – Fase I

Elaborado por: Christian Chasi

3.2.4. Fase II: Inicialización

En esta fase se preparan todos los recursos físicos, tecnológicos, entorno del proyecto. También se recaba toda la información y se planifican las tareas a realizar en las fases posteriores; de esta fase depende el éxito de las siguientes.

3.2.4.1. Configuración del proyecto

- **Preparación del ambiente**

Para empezar con el desarrollo y codificación del aplicativo se utiliza un computador portátil Core i7 de 9na generación, en un sistema operativo Windows 10 HOME. En el, se instalaron tecnologías de desarrollo como Java, Ionic, Angular, Node.js, Express.js, Postman, MongoDB, Git y Git Hub, Visual Studio Code, Android Studio,

entre otros. También se configuraron variables de entorno ya que ciertas herramientas trabajan mediante consola y es necesario configurarlas.

Para mantener un control de versiones y respaldos se hace uso la herramienta de Git y Git Hub. La tecnología de Node.js permitirá montar el servidor local, en el que se crearan y probaran los servicios REST. Al utilizar APIs de terceros como OneSignal, Leaflet y Mapbox es necesario crear cuentas en sus plataformas para poder utilizar sus servicios. Para las pruebas de la aplicación se utilizarán emuladores de Android Studio y también un dispositivo móvil Redmi Note 8 con android 10. Finalmente, para lanzar la aplicación a producción se utilizarán servidores gratuitos en plataformas como HEROKU (Servicios REST), MongoDB Atlas (Base de Datos) y plataformas de paga como Google Play Store (App Cliente).

A continuación, se muestra paso a paso la creación y configuración de OneSignal (API de notificaciones Push).

1. Como primer paso se crea un nuevo proyecto en OneSignal escogiendo la opción de Google Android (FCM).

New App / Website

Add your app or website. Need help? [Read our getting started docs.](#)

Name of your app or website
Alarmas

What organization should it belong to?
Unknown

What platform do you wish to use for this app? You can set up more later.

Apple iOS (APNs) Google Android (FCM) Web Email SMS

> More Options

Next: Configure Your Platform

Figura 3.2.3 Proyecto en OneSignal

Elaborado por: Christian Chasi y [49]

2. Luego se crea un proyecto en Firebase (FCM). Este paso es necesario ya que Firebase es propiedad de Google y obliga a crear un proyecto en su plataforma para poder utilizar notificaciones push en dispositivos android.

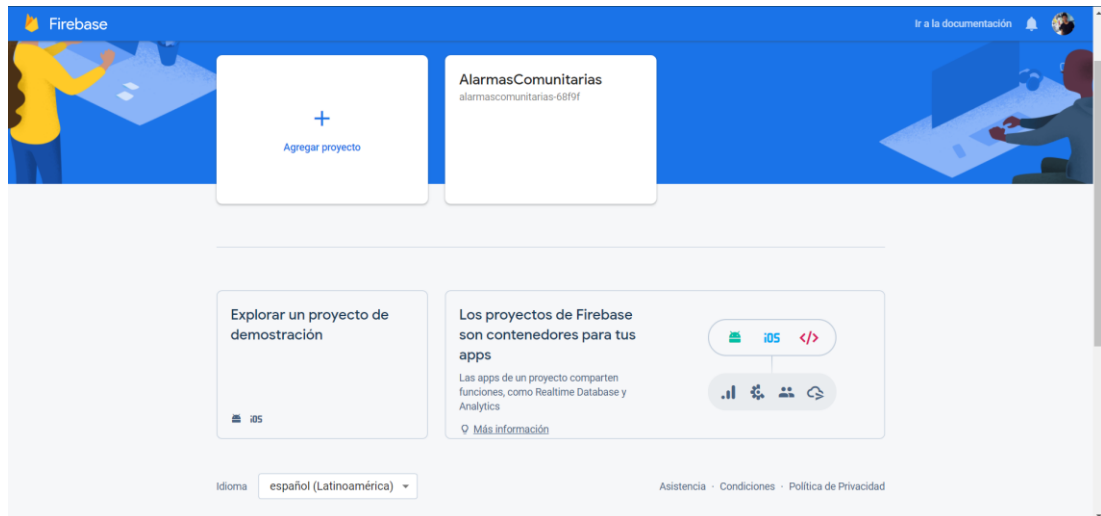


Figura 3.2.5 Proyecto en Firebase

Elaborado por: Christian Chasi y [50]

3. Luego se configura los id del proyecto de Firebase en OneSignal.

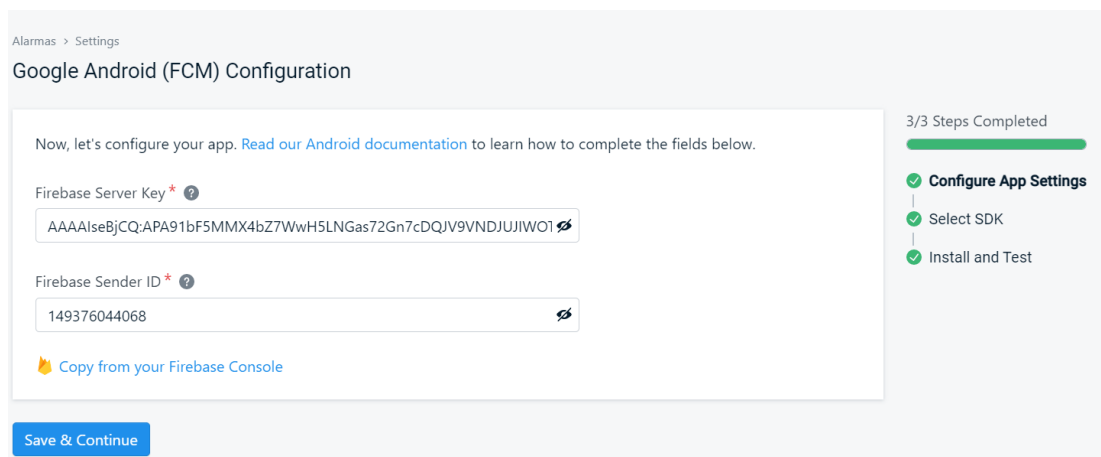


Figura 3.2.4 Configuración final de OneSignal

Elaborado por: Christian Chasi y [49]

- **Estructura del proyecto en Ionic**

Como se analizó en los puntos anteriores el Framework Ionic trabaja conjuntamente con Angular por lo tanto ya tienen su estructura de trabajo predefinida.

En la siguiente figura se muestra la estructura del proyecto para el desarrollo de las 2 aplicaciones. Algunos de los directorios fueron creados automáticamente por el asistente de Angular y Ionic y otros creadas por el programador con el fin de mantener un código bien organizado y limpio.

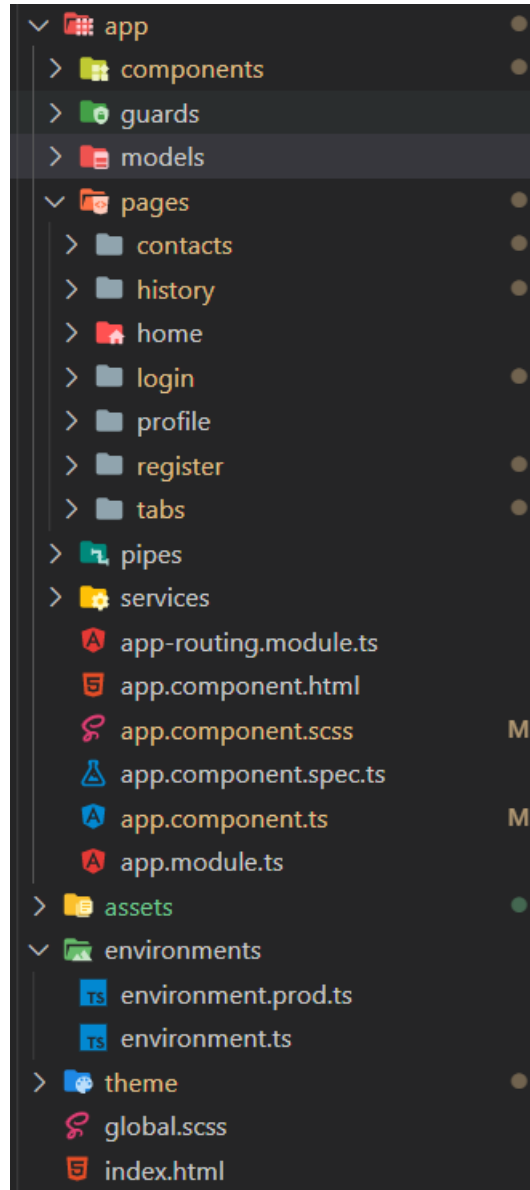


Figura 3.2.6 Estructura de directorios de la aplicación Barrio Seguro y Monitoreo de Alertas

Elaborado por: Christian Chasi

Descripción de los directorios más importantes:

app: Este directorio es la raíz de todo el proyecto creado por angular.

components: Directorio creado por el programador que contiene los componentes que se van a reutilizar durante el desarrollo de la aplicación.

guards: Directorio creado por el programador que contiene la lógica para proteger las rutas de la aplicación.

models: Directorio creador por el programador que contiene los modelos o entidades que se utilizan para mapear los datos.

pages: Directorio creado por el programador que contiene los módulos principales del aplicativo.

services: Directorio creado por el programador que contiene toda la lógica para las llamadas a la API de servicios REST.

assets: Directorio creado por angular que contiene todos los recursos (imágenes, iconos, fuentes, etc.) que se utilizan en la aplicación.

environments: Directorio creado por angular que contiene los enlaces o url con los que se conectará la aplicación ya sea en modo de pruebas o en producción.

theme: Directorio creado por angular en el que se maneja los estilos scss generales de la aplicación.

- **Capacitación del personal de desarrollo**

Para el manejo de algunas tecnologías que no se conocen a profundidad se utiliza la documentación oficial de las herramientas, frameworks y librerías.

- **Comunicación con los interesados**

Se mantienen pequeñas reuniones con el personal a cargo del GAD de la parroquia Sta. Rosa y los vecinos de los barrios como se muestra en los gráficos del **Anexo 1**, con el fin de socializar acerca de requerimientos, funcionalidades del aplicativo, ventajas, capacitación, pruebas y corrección de errores.

3.2.4.2. Planificación inicial

En esta etapa se consolida y analiza todo lo que se ha venido recabando, es decir se detalla más a fondo los requerimientos e historias de usuario.

- **Requerimientos Funcionales**

Como se definió anteriormente en los diagramas generales y de proceso se implementarán 2 aplicaciones móviles. La primera es la de ciudadanos la que les permitirá emitir, recibir y geolocalizar las alertas de auxilio. La segunda aplicación surge como propuesta por parte de los mismos ciudadanos para que pueda ser utilizada por la policía o las autoridades que estén en la obligación de brindar ayuda, en esta a diferencia de la app de ciudadanos, solo se geolocalizarán y se mantendrá un registro de las alertas.

Aplicación Móvil – Barrio Seguro

| Id | Requerimiento | Descripción |
|------------|------------------------------------|---|
| RF1 | Registro de Usuarios | El ciudadano crea una cuenta con sus datos personales para posteriormente iniciar sesión. |
| RF2 | Autenticación de Usuario | El usuario inicia sesión posterior a su registro, esta acción solo la realizará una vez. |
| RF3 | Envío de alertas de auxilio | Se envía la alerta de auxilio (con sus datos personales y ubicación) utilizando tecnología push hacia otros dispositivos. |
| RF4 | Ubicación mediante GPS | La opción de obtener ubicación debe estar disponible en el registro de usuario, en el envío de alertas de auxilio y en el perfil de usuario. |
| RF5 | Notificaciones Push | Estas notificaciones contendrán la información de la persona que solicita la ayuda y al dar clic en ellas se visualizará la ubicación e información en un mapa. |
| RF6 | Geolocalización de alertas | En función de la información que llega de la persona que solicita ayuda, se debe mostrar en un mapa su ubicación y datos personales. |

| Id | Requerimiento | Descripción |
|-------------|----------------------------------|--|
| RF7 | Radio de alcance | Para el envío de alertas puede haber distintas distancias de alcance con el fin de llegar a personas que estén más alejadas del lugar. |
| RF8 | Contactos de emergencia | Los ciudadanos pueden agregar contactos (familiares o amigos) de emergencia. |
| RF9 | Historial de Alertas | Cada vez que llegue una alerta, debe almacenarse en el dispositivo manteniendo un registro de esta, que posteriormente, puede ser visualizada o eliminada. |
| RF10 | Perfil de usuario | El usuario puede actualizar sus datos personales y ubicación. |
| RF11 | Restablecer la contraseña | El usuario restablece la contraseña en caso de que la olvide. |

Tabla 3.2.3 Requerimientos funcionales aplicación móvil – Barrio Seguro

Elaborado por: Christian Chasi

Aplicación Móvil – Monitoreo de alertas

| Id | Requerimiento | Descripción |
|-------------|-----------------------------------|--|
| RF12 | Registro de Usuarios | El usuario puede crear una cuenta con sus datos personales. |
| RF13 | Autenticación de Usuario | Una vez que el usuario ha creado la cuenta deberá iniciar sesión. |
| RF14 | Geolocalización de alertas | El usuario podrá visualizar en el mapa la ubicación y los datos de la persona que solicita ayuda. |
| RF15 | Notificaciones Push | El usuario recibirá en su dispositivo las notificaciones de alertas que contendrán información del ciudadano que solicita ayuda. |

| Id | Requerimiento | Descripción |
|-------------|------------------------------------|---|
| RF16 | Historial de Notificaciones | Cada notificación de auxilio se almacenará en el dispositivo para mantener un registro de esta. |
| RF17 | Perfil de usuario | El usuario puede actualizar sus datos personales y ubicación. |
| RF18 | Restablecer la contraseña | El usuario restablece la contraseña en caso de que la olvide. |

Figura 3.2.7 Requerimientos funcionales aplicación móvil – Monitoreo de alertas

Elaborado por: Christian Chasi

- **Requerimientos no Funcionales**

| Id | Requerimiento | Descripción | Prioridad |
|-------------|----------------------------|--|------------------|
| RNF1 | Seguridad | Las aplicaciones deben emplear seguridad para la protección de datos o utilización de sus funciones. Para ello se utilizará JSON Web Tokens (JWT). | ALTA |
| RNF2 | Usabilidad | Las interfaces deben ser intuitivas para el usuario. | ALTA |
| RNF3 | Escalabilidad | El sistema debe ser escalable y adaptarse al aumento del número de usuarios. | ALTA |
| RNF4 | Tiempo de Respuesta | El sistema debe tener tiempos de respuesta rápidos a los llamados de las REST APIs. | ALTA |

Tabla 3.2.4 Requerimientos no funcionales

Elaborado por: Christian Chasi

- **Otros Requerimientos**

| Id | Requerimiento | Descripción | Prioridad |
|-------------|-----------------------------------|---|------------------|
| ORF1 | Herramientas de desarrollo | Por el lado del cliente utilizará el framework Ionic y Angular, por el lado del servidor Node.js y Express. | ALTA |
| ORF2 | Plataformas | El framework Ionic permite desarrollar tanto para android e iOS, pero este aplicativo estará disponible por el momento solo para android por cuestiones de costos. | ALTA |
| ORF3 | Base de Datos | Utilizará una Base de Datos MongoDB debido a la velocidad de lectura y escritura de datos y la flexibilidad de comunicación entre las tecnologías de desarrollo en el cliente y servidor. | ALTA |

Tabla 3.2.5 Otros Requerimientos

Elaborado por: Christian Chasi

- **Análisis de Requerimientos**

En función de los requerimientos se han establecido los siguientes módulos:

Aplicación Móvil – Barrio Seguro

| Módulos | Descripción | Código Requerimiento | Estimado en horas | Prioridad |
|--|---|-----------------------------|--------------------------|------------------|
| Módulo de registro de Usuarios | Los usuarios se registran en la aplicación. | RF1 | 70 | ALTA |
| Módulo de Autenticación de Usuarios | Los usuarios inician sesión en la aplicación. | RF2 | 40 | ALTA |
| | Restablecer la contraseña del usuario. | RF11 | 20 | BAJA |

| Módulos | Descripción | Código Requerimiento | Estimado en horas | Prioridad |
|--|--|-----------------------------|--------------------------|------------------|
| Módulo de alertas de auxilio | Envío de alertas de auxilio. | RF3 | 20 | ALTA |
| | Ubicación actual mediante GPS | RF4 | 10 | ALTA |
| | Notificaciones push. | RF5 | 20 | ALTA |
| | Rango de distancia de alertas de auxilio. | RF7 | 10 | ALTA |
| | Mapas y Geolocalización | RF6 | 40 | ALTA |
| Módulo de Perfil de Usuario | Los usuarios gestionan su perfil. | RF10 | 40 | MEDIA |
| Módulo de Historial de Alertas | Mantener un registro de las alertas de auxilio | RF9 | 30 | MEDIA |
| | Mapas y Geolocalización | RF6 | 40 | ALTA |
| Módulo de Contactos de emergencia | Gestión de los contactos del usuario. | RF8 | 40 | BAJA |

Tabla 3.2.6 Módulos de la aplicación móvil – Barrio Seguro

Elaborado por: Christian Chasi

Aplicación Móvil – Monitoreo de alertas

| Módulos | Descripción | Código Requerimiento | Estimado en horas | Prioridad |
|--|---|---------------------------------|------------------------------|------------------|
| Módulo de registro de Usuarios | Los usuarios se registran en la aplicación. | RF11 | 20 | ALTA |
| Módulo de Autenticación de Usuarios | Los usuarios inician sesión en la aplicación. | RF12 | 10 | ALTA |
| | Restablecer la contraseña del usuario. | RF18 | 20 | MEDIA |
| Módulo de Perfil de Usuario | Los usuarios gestionan su perfil. | RF17 | 10 | ALTA |
| Módulo de alertas de auxilio | Notificaciones push. | RF15 | 10 | ALTA |
| | Historial de las alertas de auxilio | RF16 | 10 | ALTA |
| | Mapas y Geolocalización | RF13 | 40 | ALTA |

Tabla 3.2.7 Módulos aplicación móvil – Monitoreo de alertas

Elaborado por: Christian Chasi

3.2.4.3. Historias de Usuario

Las historias de usuario son una descripción más precisa de lo que el cliente (en este caso los ciudadanos y el investigador) requiere para el sistema, al ser descrito por el cliente no se deben utilizar tecnicismos, al contrario, debe utilizar un lenguaje sencillo que sea entendible para el equipo de desarrollo. A continuación, se muestra un formato de una matriz de historia de usuario básico y sus características.

| Historia De Usuario | |
|----------------------------|---------------------|
| Numero: | |
| Nombre de Historia: | |
| Usuario: | Programador: |
| Prioridad: | Riesgo: |
| Descripción: | |
| Observaciones: | |

Tabla 3.2.8 Plantilla de una historia de usuario

Elaborado por: Christian Chasi

Cada elemento se refiere a:

- **Número:** Identificador de la historia de usuario.
- **Nombre de la historia:** El nombre que corresponde al módulo o requisito del sistema para identificarlo en las tareas de las siguientes fases.
- **Usuario:** Es la persona o grupo quien va a interactuar con el aplicativo.
- **Programador:** Es la persona que se encarga de la implementación de la historia de usuario.
- **Prioridad:** Es la importancia que se le asigna a la historia de usuario, esta puede ser: ALTA, MEDIA O BAJA.
- **Riesgo:** Es el nivel de riesgo que puede presentarse durante la implementación de la historia de usuario, este puede ser: ALTO, MEDIO, BAJO.
- **Descripción:** El cliente redacta lo que requiere para el sistema.
- **Observación:** Se agregan detalles a tomar en cuenta en la historia de usuario.

A continuación, se implementan las historias de usuario correspondientes al proyecto de investigación:

Aplicación Móvil – Barrio Seguro

| Historia De Usuario | |
|---|-------------------------------------|
| Numero: 001 | |
| Nombre de Historia: Registro de ciudadanos | |
| Usuario: Ciudadanos | Programador: Christian Chasi |
| Prioridad: Alta | Riesgo: Bajo |

| |
|---|
| <p>Descripción:</p> <ul style="list-style-type: none"> • El usuario con rol ciudadano debe crear una cuenta con sus datos personales (nombre, apellido, celular, dirección, cédula, email) y su ubicación actual (obtenida mediante el GPS del dispositivo). • Se debe validar que sea una cédula real. • En caso de que la cédula, correo o celular ya existan se debe mostrar el mensaje al usuario. • Antes del registro todos los datos deben ser validados. |
| <p>Observaciones:</p> <ul style="list-style-type: none"> • El usuario debe dar permisos para utilizar el GPS del dispositivo. |

Tabla 3.2.9 Historia de usuario – Registro de usuarios

Elaborado por: Christian Chasi

| | |
|---|-------------------------------------|
| Historia De Usuario | |
| Numero: 002 | |
| Nombre de Historia: Autenticación de Usuario | |
| Usuario: Ciudadanos | Programador: Christian Chasi |
| Prioridad: Alta | Riesgo: Bajo |
| <p>Descripción:</p> <ul style="list-style-type: none"> • El usuario debe iniciar sesión con el correo y contraseña registrados previamente. • En la pantalla principal también debe haber las opciones para restablecer la contraseña y registrarse si el usuario no tiene una cuenta. | |
| <p>Observaciones:</p> <ul style="list-style-type: none"> • El usuario debe tener un correo válido donde recibirá la nueva contraseña de acceso. • Por seguridad en el inicio de sesión se requiere implementar autenticación basada en tokens. | |

Tabla 3.2.10 Historia de usuario – Autenticación de usuario

Elaborado por: Christian Chasi

| |
|----------------------------|
| Historia De Usuario |
|----------------------------|

| | |
|--|-------------------------------------|
| Numero: 003 | |
| Nombre de Historia: Alertas de auxilio | |
| Usuario: Ciudadanos | Programador: Christian Chasi |
| Prioridad: Alta | Riesgo: Alto |
| Descripción: <ul style="list-style-type: none"> • Si el usuario ingresa a la aplicación y está autenticado puede enviar alertas de auxilio (presionando el botón de pánico). • En la alerta de auxilio se envía la ubicación (actual o con la que se registró) • El usuario puede escoger el radio de distancia de alcance de envío de la alerta o utilizar una distancia ya definida para que la acción del pedido de auxilio sea rápida. • Los dispositivos objetivo (vecinos) recibirán la alerta con la información mediante notificaciones push (con un sonido predeterminado) que al dar clic se abrirá un mapa con información de la persona. • El mapa se debe mostrar los marcadores con la ubicación e información de: <ol style="list-style-type: none"> 1. El ciudadano que envía la alerta de auxilio. 2. El ciudadano que recibe la alerta. | |
| Observaciones: <ul style="list-style-type: none"> • Cada vez que se envíe una alerta deberá esperar un lapso corto de tiempo para volver a enviar otra. • Cuando llegue la notificación o alerta se debe emitir un sonido que identifique que es un pedido de auxilio. • Para que se escuche la alerta es importante tener activado el sonido del dispositivo y de las notificaciones. | |

Tabla 3.2.11 Historia de usuario –Alertas de auxilio

Elaborado por: Christian Chasi

| Historia De Usuario | |
|--|-------------------------------------|
| Numero: 004 | |
| Nombre de Historia: Configuración y Perfil de Usuario | |
| Usuario: Ciudadanos | Programador: Christian Chasi |
| Prioridad: Media | Riesgo: Bajo |

| |
|--|
| <p>Descripción:</p> <ul style="list-style-type: none"> • El usuario puede gestionar su perfil, es decir actualizar sus datos personales (nombre, apellido, dirección, etc.), ubicación, correo y la contraseña. • También podrá silenciar las alertas si así lo requiere. |
| <p>Observaciones:</p> <ul style="list-style-type: none"> • El usuario no puede actualizar el número de cédula. |

Tabla 3.2.12 Historia de usuario – Autenticación de usuario con rol ciudadano

Elaborado por: Christian Chasi

| Historia De Usuario | |
|--|-------------------------------------|
| Numero: 005 | |
| Nombre de Historia: Historial de alertas | |
| Usuario: Ciudadanos | Programador: Christian Chasi |
| Prioridad: Media | Riesgo: Bajo |
| <p>Descripción:</p> <ul style="list-style-type: none"> • Las notificaciones se almacenan en el dispositivo con el fin de mantener un registro o listado de alertas. • El módulo debe tener una opción para eliminar todo el historial de alertas. | |
| Observaciones: Ninguna | |

Tabla 3.2.13 Historia de usuario – Historial de alertas

Elaborado por: Christian Chasi

| Historia De Usuario | |
|---|-------------------------------------|
| Numero: 006 | |
| Nombre de Historia: Contactos de emergencia. | |
| Usuario: Ciudadanos | Programador: Christian Chasi |
| Prioridad: Baja | Riesgo: Bajo |
| <p>Descripción:</p> <ul style="list-style-type: none"> • El usuario puede agregar y eliminar contactos de emergencia para que reciban las alertas de auxilio. | |
| <p>Observación:</p> <ul style="list-style-type: none"> • Para agregar un contacto, este deberá estar registrado en la aplicación. | |

Tabla 3.2.14 Historia de usuario – Contactos de emergencia

Elaborado por: Christian Chasi

Aplicación Móvil – Monitoreo de Alertas

| Historia De Usuario | |
|--|-------------------------------------|
| Numero: 007 | |
| Nombre de Historia: Registro de usuarios | |
| Usuario: Policía | Programador: Christian Chasi |
| Prioridad: Alta | Riesgo: Medio |
| Descripción: <ul style="list-style-type: none">• El usuario con rol policía puede crear una cuenta con sus datos personales (nombre, apellido, celular, dirección, cédula, email).• Antes del registro todos los datos ingresados deben ser validados. | |
| Observación: Ninguna | |

Tabla 3.2.15 Historia de usuario – Registro de usuarios con rol policía

Elaborado por: Christian Chasi

| Historia De Usuario | |
|---|-------------------------------------|
| Numero: 008 | |
| Nombre de Historia: Autenticación de Usuario | |
| Usuario: Policía | Programador: Christian Chasi |
| Prioridad: Alta | Riesgo: Bajo |
| Descripción: <ul style="list-style-type: none">• El usuario debe iniciar sesión con el correo y contraseña registrados previamente.• En la pantalla principal también debe haber las opciones para restablecer la contraseña y registrarse si el usuario no tiene una cuenta. | |
| Observación: Ninguna | |

Tabla 3.2.16 Historia de usuario – Autenticación de usuarios

Elaborado por: Christian Chasi

| Historia De Usuario | |
|----------------------------|--|
| Numero: 009 | |

| | |
|---|-------------------------------------|
| Nombre de Historia: Alertas de auxilio | |
| Usuario: Policía | Programador: Christian Chasi |
| Prioridad: Alta | Riesgo: Alto |
| Descripción: | |
| <ul style="list-style-type: none"> Las notificaciones se almacenan en el orden en que llegan en el dispositivo con el fin de mantener un registro o listado de alertas y poder visualizarlas posteriormente. | |
| Observación: Cuando llegue la notificación o alerta se debe emitir un sonido que identifique que es un pedido de auxilio. | |

Tabla 3.2.17 Historia de usuario – Gestión de alertas de auxilio.

Elaborado por: Christian Chasi

| Historia De Usuario | |
|---|-------------------------------------|
| Numero: 010 | |
| Nombre de Historia: Configuración y Perfil de Usuario | |
| Usuario: Policía | Programador: Christian Chasi |
| Prioridad: Baja | Riesgo: Bajo |
| Descripción: | |
| <ul style="list-style-type: none"> El usuario con rol policía puede configurar gestionar su perfil, es decir podrá actualizar sus datos personales, correo y contraseña de inicio de sesión, también eliminar e historial de notificaciones y cerrar sesión. | |
| Observaciones: | |
| <ul style="list-style-type: none"> Ninguna | |

Tabla 3.2.18 Historia de usuario – Autenticación de usuario con rol policía

Elaborado por: Christian Chasi

3.2.4.4. Planificación de Fases

Una vez que se han analizado y definidos a fondo los requisitos se describe la planificación de las siguientes fases y sus respectivas iteraciones.

| Fase | Iteración | Descripción |
|----------------|------------------|---|
| Inicialización | Iteración 0 | Configuración del proyecto, análisis de |

| Fase | Iteración | Descripción |
|------------|--|---|
| | | requerimientos y planificación de fases. |
| Producción | Aplicación Móvil – Barrio Seguro | |
| | Iteración 1: Módulo de Registro de Ciudadanos | Implementación del módulo de registro de usuarios, story cards, código fuente e interfaces de usuario. |
| | Iteración 2: Módulo de Autenticación de usuarios | Implementación del módulo de inicio de sesión de usuarios, story cards, código fuente e interfaces de usuario. |
| | Iteración 3: Módulo de Alertas de Auxilio | Implementación del módulo para envío de alertas de auxilio, story cards, código fuente e interfaces de usuario. |
| | Iteración 4: Módulo de Historial de alertas. | Implementación del módulo de historial de alertas de auxilio, story cards, código fuente e interfaces de usuario. |
| | Iteración 5: Módulo de Contactos | Implementación del módulo del módulo de contactos, story cards, código fuente e interfaces de usuario. |
| | Iteración 6: Módulo de Configuración y Perfil de Usuario | Implementación del módulo del perfil del usuario, story cards, |

| Fase | Iteración | Descripción |
|----------------|---|---|
| | | código fuente e interfaces de usuario. |
| | Aplicación Móvil – Monitoreo de Alertas | |
| | Iteración 7: Módulo de Registro de Usuarios | Implementación del módulo de registro de usuarios, story cards, código fuente e interfaces de usuario. |
| | Iteración 8: Módulo de Autenticación de Usuarios | Implementación del módulo de inicio de sesión de usuarios, story cards, código fuente e interfaces de usuario. |
| | Iteración 9: Módulo de alertas de auxilio | Implementación del módulo para monitoreo de alertas de auxilio, story cards, código fuente e interfaces de usuario. |
| | Iteración 10: Módulo de Configuración y Perfil de Usuario | Implementación del módulo del perfil del usuario, story cards, código fuente e interfaces de usuario. |
| Estabilización | Iteración de Estabilización | Integración de todos los módulos (Backend y Frontend) de las 2 aplicaciones. |
| Pruebas | Iteración pruebas del sistema | Se realizarán las respectivas pruebas unitarias y de aceptación de las 2 aplicaciones. |

Tabla 3.2.19 Planificación de Fases – Fase II

3.2.4.5.Arquitectura de la aplicación

La arquitectura de la aplicación para ciudadanos de los barrios es de 3 capas Presentación, Servicios y Datos. La aplicación móvil cliente envía la petición hacia el servidor (que contiene los servicios REST), este realiza la respectiva consulta a la Base de Datos (MongoDB) y los resultados lo retorna al servidor para ser procesados y enviados mediante una llamada a la API de OneSignal, este también los procesa y envía hacia los dispositivos clientes (vecinos, contactos de emergencia y app de monitoreo de alertas).

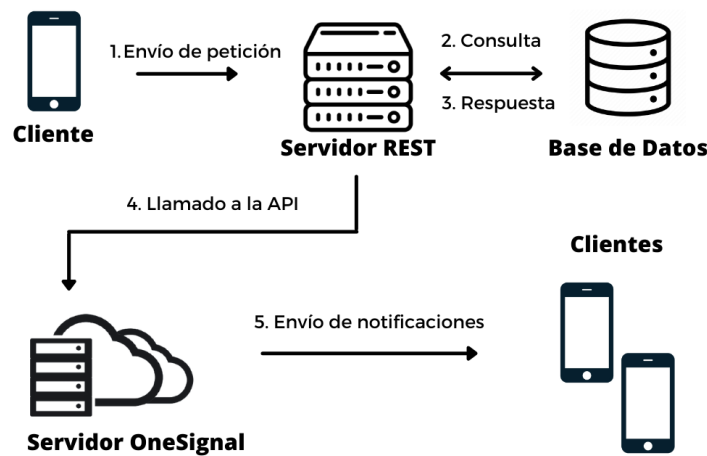


Figura 3.2.8 Arquitectura de la aplicación

3.2.4.6.Base de Datos

MongoDB es una base de datos No SQL por lo tanto no existen relaciones como tal, en lugar de tablas utiliza colecciones cuyos datos se estructuran en formato JSON por lo tanto no ocupan mucho espacio de almacenamiento. También se hace hincapié a que mongo crea automáticamente el id de tipo ObjectId.

➤ **Colección ciudadanos**

```
ciudadano: {
  keys :{
    _id:ObjectId,
    ubicacion: Point,
    telefonos: {},
    activo: Boolean,
    suscrito: Boolean,
    nombre: String,
    apellido: String,
    cedula: String,
    direccion: String,
    email: String,
    celular: String,
    genero: String,
    rol: ObjectId,
    password: String,
    id_dispositivo: String
  }
}
```

Figura 3.2.9 Base de Datos - Colección ciudadanos

Elaborado por: Christian Chasi

➤ **Colección roles**

```
roles:{
  keys:{
    _id:ObjectId,
    rol:String
  }
}
```

Figura 3.2.10 Base de datos - Colección roles

Elaborado por: Christian Chasi

➤ **Colección policías**

```
policia: {
  keys :{
    _id:ObjectId,
    activo: Boolean,
    suscrito: Boolean,
    nombre: String,
    apellido: String,
    email: String,
    celular: String,
    rol: ObjectId,
    password: String,
    id_dispositivo: String
  }
}
```

Figura 3.2.11 Base de Datos - Colección policías

Elaborado por: Christian Chasi

3.2.5. Fase III: Producción

En esta fase se implementa todo el aplicativo en función de los requisitos o historias de usuario. También es importante hacer hincapié a que en esta metodología el ciclo de desarrollo es iterativo, lo que significa que en cada iteración se implementa en este caso los módulos predefinidos previamente, para posteriormente pasar a la fase de estabilización y pruebas.

Aplicación Móvil – Barrio Seguro

3.2.5.1. Iteración 1: Registro de ciudadanos

- **Día de planificación**

| Número | Tipo | Dificultad | | Esfuerzo | | Prioridad |
|--|---------|---------------|----------|----------|-------------------|-----------|
| | | Antes | Después | Estimado | Gastado | |
| 1 | Nuevo | Fácil | Fácil | 70h | 75h | Alta |
| | Arreglo | Moderada | Moderada | | | |
| | Mejora | Difícil | Difícil | | | |
| Descripción: Los usuarios con rol ciudadano se registran mediante la aplicación proporcionando sus datos personales y ubicación obtenida mediante el GPS del dispositivo. Si el registro es exitoso se redirige automáticamente a la pantalla de autenticación caso contrario aparecen un mensaje de aviso al usuario para que corrija el campo invalido. | | | | | | |
| Fecha | | Estado | | | Comentario | |
| 03/05/2021 | | Definido | | | | |
| 12/05/2021 | | Realizado | | | | |

Tabla 3.2.20 Story Card – Registros de usuarios con rol ciudadano

Elaborado por: Christian Chasi

- **Día de trabajo**

A continuación, se presentan fragmentos del código implementado para el registro de usuarios (Frontend). Se visualiza lo siguiente:

1. Se valida el formulario.
2. Se obtienen los datos del formulario
3. Se envía a través del método que llama a los servicios REST

```
1 async register() {
2   this.formularioEnviado = true;
3   if (this.formulario.invalid || this.genero == "") {
4     this.formulario.markAllAsTouched();
5     this.srvAlertSmS.presentAlert("Hay campos vacíos o incorrectos revíselos", "Campos Erróneos", "", true)
6     this.srvAlertSmS.closeModalAlert(3000)
7     return;
8   }
9
10  const { nombre, apellido, cedula, celular, ubicacion, direccion, email, password } = this.formulario.value;
11  this.ciudadano = new Ciudadano(null, nombre, apellido, cedula, null, null, direccion, email, celular);
12  this.ciudadano.ubicacion = ubicacion;
13  this.ciudadano.setPassword = password;
14  this.ciudadano.setGenero = this.genero;
15  await this.srvAlertSmS.presentLoading("Por favor espere...");
16  try {
17    this.srvAuth.registrar(this.ciudadano).then((res: any) => {
18      this.srvAlertSmS.loading.dismiss();
19      if (res != false) {
20        if (res.data.status == "ok") {
21          this.srvAlertSmS.presentAlert(res.data.msg, "Registro", "", true);
22          this.navCtrl.navigateRoot('login', { animated: true });
23        }
24      }
25      else {
26        this.srvAlertSmS.presentAlert('No se encontro el ID del dispositivo. Cierre la aplicación e intente
nuevamente', "Error", "", true);
27      }
28      this.srvAlertSmS.closeModalAlert(3000);
29    }, (err) => {
30      console.log('Error Registro', err);
31      this.srvAlertSmS.loading.dismiss();
32      if (err.status == 0) {
33        this.srvAlertSmS.presentAlert("No se pudo conectar con el servidor", "Servidor", "", true)
34      }
35      else if (err.status == 500) {
36        this.srvAlertSmS.presentAlert("Error en el Servidor", "Servidor", "", true)
37      }
38      else if (err.error) {
39        this.srvAlertSmS.presentAlert(err.error.msg, "Error al Registrar", "", true);
40      }
41      else {
42        this.srvAlertSmS.presentAlert(err.error.msg, "Registro", "", true);
43      }
44      this.srvAlertSmS.closeModalAlert(3000);
45    }
46  );
47  } catch (error) {
48  }
49  }
50  }
51  }
52  }
53  }
```

Figura 3.2.12 Iteración 1 – Codificación de registro de usuario (Frontend)

Elaborado por: Christian Chasi

Para la parte de los servicios (Backend) se muestra el fragmento de código más importante para ejecutar el proceso de registro de usuarios, en este se visualiza:

1. Recepción de los datos enviados desde el cliente.
2. Manejo de los datos y encriptación de la contraseña.
3. Registro de la información en la Base de Datos

```
usuario.js

const registrarCiudadanos = async (req = request, res = response) => {
  try {
    const {
      _id,
      password,
      latitud,
      longitud,
      rol,
      tokenDispositivo,
      ...ciudadano
    } = req.body;
    const role = await Rol.findOne({ rol }); // Obtengo el ID del rol
    const ubicacion = { type: "Point", coordinates: [longitud, latitud] };
    const usuario = new Ciudadano(ciudadano);
    usuario.rol = role;
    usuario.ubicacion = ubicacion;
    //Encripto contraseña
    const salt = bcryptjs.genSaltSync();
    const passwordCrypt = bcryptjs.hashSync(password, salt);
    usuario.password = passwordCrypt;
    usuario.id_dispositivo = tokenDispositivo;
    await usuario.save();
    res.json({ data: { status: "ok", msg: "Registro Exitoso" } });
  } catch (errores) {
    res.status(500).json({ msg: "Error en el servidor", errores });
  }
};
```

Figura 3.2.13 Iteración 1 – Codificación de registro de usuario (Backend)

Elaborado por: Christian Chasi

- **Día de liberación**

En esta etapa se integra la parte del Frontend y Backend para el módulo de registro de usuarios, se realizan pruebas de funcionamiento y se libera.

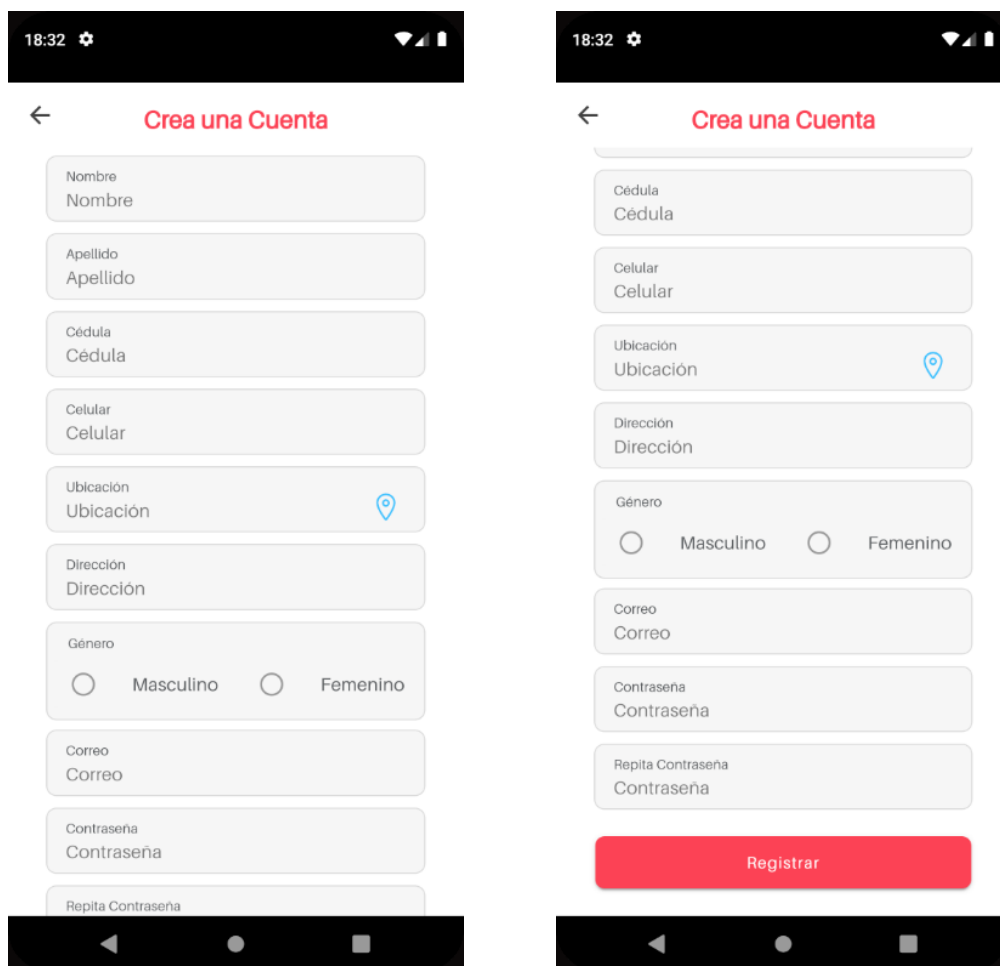


Figura 3.2.14 Iteración 1 – Interfaz registro de usuario

Elaborado por: Christian Chasi

3.2.5.2. Iteración 2: Autenticación de usuarios

- **Día de planificación**

| Número | Tipo | Dificultad | | Esfuerzo | | Prioridad |
|--------|---------|------------|----------|----------|---------|-----------|
| | | Antes | Después | Estimado | Gastado | |
| 2 | Nuevo | Fácil | Fácil | 40h | 40h | Alta |
| | Arreglo | Moderada | Moderada | | | |
| | Mejora | Difícil | Difícil | | | |

Descripción: Los usuarios con rol ciudadano pueden iniciar sesión (posteriormente de haberse registrado) con su correo y contraseña. También podrán restablecer su contraseña mediante su correo electrónico registrado en caso de que el usuario lo

| requiera. El aplicativo hace uso de autenticación mediante JSON Web Tokens (JWT) permitiendo un inicio de sesión seguro para el usuario. | | |
|--|-----------|------------|
| Fecha | Estado | Comentario |
| 13/05/2021 | Definido | |
| 20/05/2021 | Realizado | |

Tabla 3.2.21 Story Card – Autenticación de usuarios con rol ciudadano

Elaborado por: Christian Chasi

- **Día de Trabajo**

A continuación, se presentan fragmentos del código más importante implementado para la autenticación de usuarios. Se visualiza lo siguiente:

1. Se valida el formulario.
2. Se obtienen los datos del formulario
3. Se envía a través del método que llama a los servicios REST

```

login.page.ts

async login() {
  if (this.formLogin.invalid) {
    this.formLogin.markAllAsTouched();
    return
  }
  await this.alertSms.presentLoading("Validando Datos...");
  try {
    this.ciudadano = this.formLogin.value;

    var valido = await this.srvAuth.loguearse(this.ciudadano);
    if (valido) {
      this.alertSms.loading.dismiss();
      await this.srvCitizen.getCiudadanoInfo();
      this.navCtrl.navigateRoot('tabs/home', { animated: true });
      this.srvContactos.obtenerContactos();
    }
  } catch (error) {
    console.log('Error Login:', error);
    this.alertSms.loading.dismiss();
    if (error.error.msg) {
      this.alertSms.presentAlert(error.error.msg, "Login", '', true)
    }
    else {
      this.alertSms.presentAlert("Ha ocurrido un error al iniciar sesión", "Login", '', true)
    }
    this.alertSms.closeModalAlert(2000)
  }
}

```

Figura 3.2.15 Iteración 2 – Codificación de autenticación de usuario (Frontend)

Elaborado por: Christian Chasi

Para la parte de los servicios (Backend) se muestra el fragmento de código más importante para ejecutar el proceso de autenticación de usuarios, en este se visualiza:

1. Validación y verificación del correo y contraseña.
2. Validación del dispositivo con el que se autentica.

```
autenticacion.js

const loginCiudadano = async (req = request, res = response) => {
  const { email, password, tokenDispositivo } = req.body;

  try {
    const usuario = await Ciudadano.findOne({ email });

    if (!usuario) {
      return res.status(400).json({ msg: "Email/Contraseña no son correctos" });
    }

    const validaPassword = bcryptjs.compareSync(password, usuario.password);

    if (!validaPassword) {
      return res.status(401).json({ msg: "Email/Contraseña no son correctos" });
    }

    const token = await generarJWT(usuario._id);
    //Activo al usuario para recibir notificaciones
    usuario.suscrito = true;
    usuario.save();

    if (tokenDispositivo == usuario.id_dispositivo) {
      return res.json({ data: { access_token: token } });
    }

    const existeCiudadanoDispositivo = await Ciudadano.findOne({
      id_dispositivo: tokenDispositivo,
    });

    if (!existeCiudadanoDispositivo) {
      usuario.id_dispositivo = tokenDispositivo;
      await usuario.save();
      return res.json({ data: { access_token: token } });
    }

    await Ciudadano.findOneAndUpdate(
      { _id: existeCiudadanoDispositivo._id },
      { id_dispositivo: null }
    );
    usuario.id_dispositivo = tokenDispositivo;
    await usuario.save();
    return res.json({ data: { access_token: token } });
  } catch (error) {
    res.status(500).json({ error: "Contactese con el administrador" });
  }
};
```

Figura 3.2.16 Iteración 2 – Codificación de autenticación de usuario (Backend)

Elaborado por: Christian Chasi

- **Día de Liberación**

En esta etapa se integra la parte del Frontend y Backend para la autenticación de usuarios, se realizan pruebas de funcionamiento y se libera.

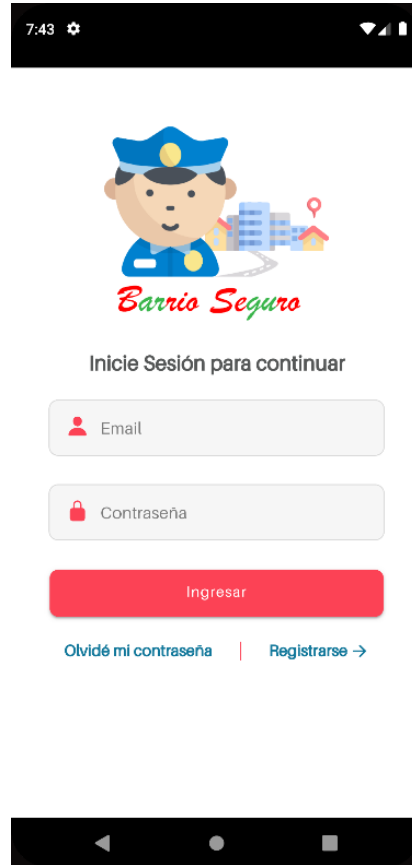


Figura 3.2.17 Iteración 2 – Interfaz autenticación de usuario

Elaborado por: Christian Chasi

3.2.5.3. Iteración 3: Alertas de auxilio

- **Día de planificación**

| Número | Tipo | Dificultad | | Esfuerzo | | Prioridad |
|--------|---------|------------|----------|----------|---------|-----------|
| | | Antes | Después | Estimado | Gastado | |
| 3 | Nuevo | Fácil | Fácil | 100h | 150h | Alta |
| | Arreglo | Moderada | Moderada | | | |
| | Mejora | Difícil | Difícil | | | |

Descripción: Si es el usuario ya está autenticado, al iniciar la aplicación se abre el módulo de alertas de auxilio, para enviar una alerta de auxilio se presiona el botón de pánico y se envían los pedidos de auxilio hacia los dispositivos más cercanos en

un radio de 500 metros (defecto). También existe la opción de cambiar la distancia de alcance de las alertas y la opción de obtener la ubicación actual. Los dispositivos cercanos reciben la alerta mediante notificaciones push que se visualizan en el panel de notificaciones o top bar de su dispositivo android.

Cuando el usuario de clic en una notificación se abre el mapa con su información y la de la persona que solicita ayuda.

Como medida de seguridad el sistema siempre valida al usuario que la envía alerta, y en cada llamada (POST, GET, PUT, DELETE) a los servicios REST API se envía el token (JWT) de autenticación.

| Fecha | Estado | Comentario |
|--------------|---------------|-------------------|
| 21/05/2021 | Definido | - |
| 28/05/2021 | Realizado | - |

Tabla 3.2.22 Story Card – Envío de alertas de auxilio

Elaborado por: Christian Chasi

- **Día de Trabajo**

A continuación, se presentan fragmentos del código más importante implementado para el envío de alertas de auxilio (Frontend). Se visualiza lo siguiente:

1. Se obtiene la ubicación actual (latitud, longitud) y el radio.
2. Se envían los datos a través del método que llama al servicio REST.
3. Se desactiva y activa el botón de pánico mientras se envía la alerta de auxilio.

```

home.page.ts

async enviarAlerta() {
  try {
    const ubi = this.ubicacion.split(',');
    this.helpSignal = { lat: ubi[0], long: ubi[1], radio: this.radio };
    this.renderer.addClass(this.imgPanic.nativeElement, "btnPanic");// Desactivando botón panico
    await this.srvAlertSms.presentAlert("Enviando Señal de auxilio...", "Aviso");
    const respuesta: any = await this.srvCitizen.enviarNotificacionAyuda(this.ciudadano.getId, this.helpSignal)
    this.srvAlertSms.alert.dismiss();
    if (respuesta.ok == true) {
      await this.srvAlertSms.presentAlert(`Notificación Enviada a ${respuesta.data.recipients} dispositivo/s cercanos`, "Aviso", "Espere 10 segundos...");
      this.renderer.addClass(this.imgPanic.nativeElement, "btnPanic");
      //Si se envia la notificacion espero 10s
      this.srvAlertSms.closeModalAlert(10000);
    }
    else {
      await this.srvAlertSms.presentAlert(`${respuesta.sms}`, "Aviso");
      this.srvAlertSms.closeModalAlert(2000);
    }
    this.renderer.removeClass(this.imgPanic.nativeElement, "btnPanic");
  } catch (error) {
    this.srvAlertSms.alert.dismiss();
    if (error.statusText == "Unauthorized") {
      this.srvAlertSms.presentToast('Usuario Deshabilitado, contacte al administrador');
      this.srvCitizen.clearStorage();
      this.navCtrl.navigateRoot('login', { animated: true });
      this.srvOneSignal.obtenerAppIdOneSignal();
    }
    else {
      await this.srvAlertSms.presentAlert(error.error.msg, "Error");
    }
  }
  this.srvAlertSms.closeModalAlert(3000);
  this.renderer.removeClass(this.imgPanic.nativeElement, "btnPanic");
}
}

```

Figura 3.2.19 Iteración 3 – Codificación de envío de alertas de auxilio (Frontend)

Elaborado por: Christian Chasi

También se muestra la configuración de OneSignal para recibir notificaciones push.

```

onesignal.service.ts

async configuracionPush() {
  this.oneSignal.startInit(this._appId, this._googleProjectNumber);
  this.oneSignal.inFocusDisplaying(this.oneSignal.OSInFocusDisplayOption.Notification);
  this.oneSignal.enableSound(true);
  this.oneSignal.handleNotificationReceived().subscribe((sms) => {
    this.notificacionRecibida(sms);
  });
  this.oneSignal.handleNotificationOpened().subscribe(async (sms) => {
    await this.notificacionRecibida(sms.notification);
    await this.presentModal();
  });
  this.oneSignal.getIds().
  then((info) => {
    this._tokenDispositivo = info.userId;
    this.guardarTokenEnDispositivo(this._tokenDispositivo);
  })
  this.oneSignal.endInit();
}
}

```

Figura 3.2.18 Iteración 3 – Codificación OneSignal en el proyecto

Elaborado por: Christian Chasi

Para la parte de los servicios (Backend) se muestra el fragmento de código más importante para ejecutar el proceso de envío de alertas de auxilio, en este se visualiza:

1. Obtención de la información (latitud, longitud y radio) enviado desde el cliente.
2. Búsqueda de dispositivos cercanos en base al radio.
3. Búsqueda para obtener los dispositivos que pertenecen a los contactos del usuario.
4. Construcción del cuerpo de la petición de OneSignal.
5. Llamado a la API de OneSignal para enviar las notificaciones.

```

const postnotificacion = async (req = request, res = response) => {
  try {
    const { id } = req.params;
    const { radio, lat, long } = req.body;
    //Declaro variables
    const ciudadano = req.usuario; // obtengo los datos del ciudadano
    const idUsuarioPost = new mongoose.Types.ObjectId(id);

    // Obtengo los telefonos de los familiares
    const setTelefonos = await Ciudadano.findById(id)
      .select("telefonos")
      .map((tel) => tel.telefonos.map((element) => element.telefono));
    // Obtener los id's en función de los telefonos
    const dispositivosFamiliares = await Ciudadano.find({
      celular: setTelefonos,
      suscrito: true,
    }).distinct("id_dispositivo");

    const setDispositivosPolicia =
      (await Policia.aggregate([
        { $group: { _id: "$id_dispositivo" } },
        { $limit: 25 },
      ])) || [];

    const dispositivosPolicia =
      setDispositivosPolicia.map((item) => item._id) || [];
    const setDispositivosCercanos = await Ciudadano.aggregate([
      {
        $geoNear: {
          near: { type: "Point", coordinates: [+long, +lat] },
          key: "ubicacion",
          distanceField: "distancia",
          maxDistance: radio,
          query: { _id: { $ne: idUsuarioPost }, activo: true, suscrito: true },
        },
      },
      { $limit: 100 },
      {
        $project: {
          _id: 0,
          id_dispositivo: 1, // quiero obtener el campo token
        },
      },
    ]);

    const dispositivosCercanos =
      setDispositivosCercanos.map((item) => item.id_dispositivo) || []; //Obtengo los tokens

    //Combino los 3 arrays con id's de dispositivos
    let allDispositivos = [
      ...dispositivosPolicia,
      ...dispositivosCercanos,
      ...dispositivosFamiliares,
    ];

    if (allDispositivos.length === 0) {
      return res.json({ ok: false, sms: "No hay dispositivos cercanos" });
    }
    //Armos el cuerpo de la peticion a la API de OneSignal
    const content = {
      app_id: "35151193-9ae6-44e6-91d3-774247a8d343", //Id de OneSignal
      data: {
        nombre: ciudadano.nombre,
        apellido: ciudadano.apellido,
        direccion: ciudadano.direccion,
        celular: ciudadano.celular,
        latitud: lat,
        longitud: long,
      },
      contents: {
        en: `🚨 ${ciudadano.nombre} ${ciudadano.apellido} needs your help`,
        es: `🚨 ${ciudadano.nombre} ${ciudadano.apellido} necesita tu ayuda`,
      },
      headings: { en: "Help alert", es: "Alerta de Auxilio" },
      android_channel_id: "d3374278-abe0-4653-91db-6249e04d2515",
      include_player_ids: allDispositivos,
    };

    const response = await sendNotificacion(content);
    if (response.id.length > 0) {
      return res.json({ ok: true, data: response });
    }
    return res.json({
      ok: false,
      sms: "No se han encontrado dispositivos activos",
      response,
    });
  } catch (error) {
    console.log("Error:", error);
    res.status(500).json({ error: "Error 500 en el servidor", error });
  }
};

```

Figura 3.2.20 Iteración 3 – Codificación de envío de alertas (Backend)

Elaborado por: Christian Chasi

- **Día de Liberación**

En esta etapa se integra la parte del Frontend y Backend para el envío de alertas de auxilio y el aviso mediante notificaciones push, se realizan pruebas de funcionamiento y se libera.

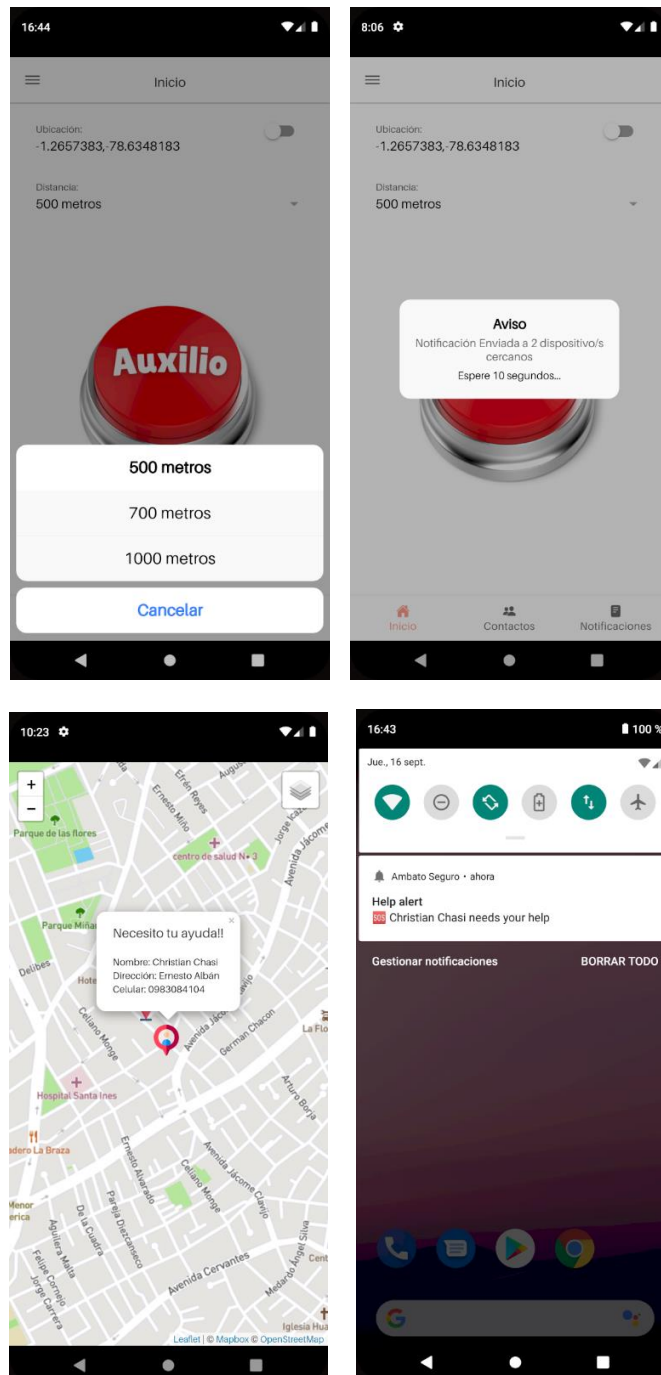


Figura 3.2.21 Iteración 3 – Interfaz de envío de alertas de auxilio

Elaborado por: Christian Chasi

Iteración 4: Contactos de emergencia

- **Día de planificación**

| Número | Tipo | Dificultad | | Esfuerzo | | Prioridad |
|--|---------|------------|----------|----------|------------|-----------|
| | | Antes | Después | Estimado | Gastado | |
| 4 | Nuevo | Fácil | Fácil | 40h | 40h | Baja |
| | Arreglo | Moderada | Moderada | | | |
| | Mejora | Difícil | Difícil | | | |
| Descripción: Los usuarios con rol ciudadano pueden agregar hasta 5 contactos de emergencia que también deben estar registrados. Las alertas de auxilio les llegarán a esos contactos aún sin estar en el rango de distancia especificado. | | | | | | |
| Fecha | | Estado | | | Comentario | |
| 14/06/2021 | | Definido | | | | |
| 19/06/2021 | | Realizado | | | | |

Tabla 3.2.23 Story Card – Contactos de emergencia

Elaborado por: Christian Chasi

- **Día de Trabajo**

A continuación, se presenta fragmento del código más importante implementado para los contactos de emergencia. Se visualiza lo siguiente:

1. Métodos para obtener, agregar y eliminar contactos a través de los servicios REST.


```
contacts.page.ts

async obtenerContactos(event:any){
  try {
    this.contactos = await this.srvContact.obtenerContactos();
  } catch (error) {
    this.srvAlertSms.presentToast("Error al cargar contactos", "danger")
  }
  event.target.complete();
}

closeSlide() {
  if(this.ionSlidin){
    this.ionSlidin.closeOpened();
  }
}

async agregarContacto() {
  try {
    const contacto = {
      nombre: this.miformulario.controls["nombre"].value,
      telefono: this.miformulario.controls["telefono"].value,
    }
    this.llamandoAPI=true;
    const contactos = await this.srvContacts.guardarContacto(this.usuario.getId(), contacto);
    this.llamandoAPI=false;

    this.srvAlertSms.presentToast("Contacto Guardado", "success");
    return true;
  } catch (err) {
    this.llamandoAPI=false;
    this.srvAlertSms.presentToast(err.error.msg, "danger")
    return false;
  }
}

async eliminar(telefono: string) {
  try {
    this.contactos = await this.srvContact.eliminarContacto(this.ciudadano.getId(), telefono);
    this.srvAlertSms.presentToast("Contacto eliminado", "success");
    this.srvContact.obtenerContactos();
  } catch (err) {
    this.srvAlertSms.presentToast(err.error.msg, "danger")
  }
}
```

Figura 3.2.22 Iteración 4 – Codificación contactos de emergencia (Frontend)

Elaborado por: Christian Chasi

Para la parte de los servicios (Backend) se muestra el fragmento de código más importante para obtener, guardar y eliminar de contactos de emergencia, en este se visualiza:

1. Obtención de los datos que provienen del cliente.
2. Validación de teléfonos

```
usuarios.js

const postTelefonos = async (req = request, res = response) => {
  try {
    const { id } = req.params;
    const { telefono,nombre } = req.body;
    const condicionFiltrar = { _id: id };
    const condicionAgregar = { $push: { telefonos:{nombre:nombre, telefono} } };
    const { telefonos } = await Usuario.findOneAndUpdate(
      condicionFiltrar,
      condicionAgregar,
      { new: true }
    );
    if (telefonos) {
      return res.json({ ok: true, msg: "Telefono Guardado",telefonos });
    }
    return res.status(400).json({ ok: false, msg: "Ha ocurrido un error a guardar el telefono" });
  } catch (error) {
    res.status(500).json({ error: "Contactese con el administrador" });
  }
};

const deleteTelefonos = async (req = request, res = response) => {
  try {
    const { id } = req.params;
    const { telefono } = req.query;
    const condicionFiltrar = { _id: id };
    const condicionAgregar = { $pull: { telefonos: {telefono} } };
    await Usuario.findOneAndUpdate(
      condicionFiltrar,
      condicionAgregar,
      { new: true }
    )
    res.json({ ok:true, msg:"Telefono Eliminado" });
  } catch (error) {
    res.status(500).json({ error: "Contactese con el administrador" });
  }
};

const getTelefonos = async (req = request, res = response) => {
  try {
    const { id } = req.params;
    const { telefonos } = await Usuario.findOne({ _id: id });
    res.json({ telefonos: telefonos });
  } catch (error) {
    res.status(500).json({ error: "Contactese con el administrador" });
  }
};
```

Figura 3.2.23 Iteración 4 – Codificación contactos de emergencia (Backend)

Elaborado por: Christian Chasi

- **Día de Liberación**

En esta etapa de integra el Frontend y Backend implementado para el módulo de contactos, se realizan pruebas de funcionamiento y se libera.

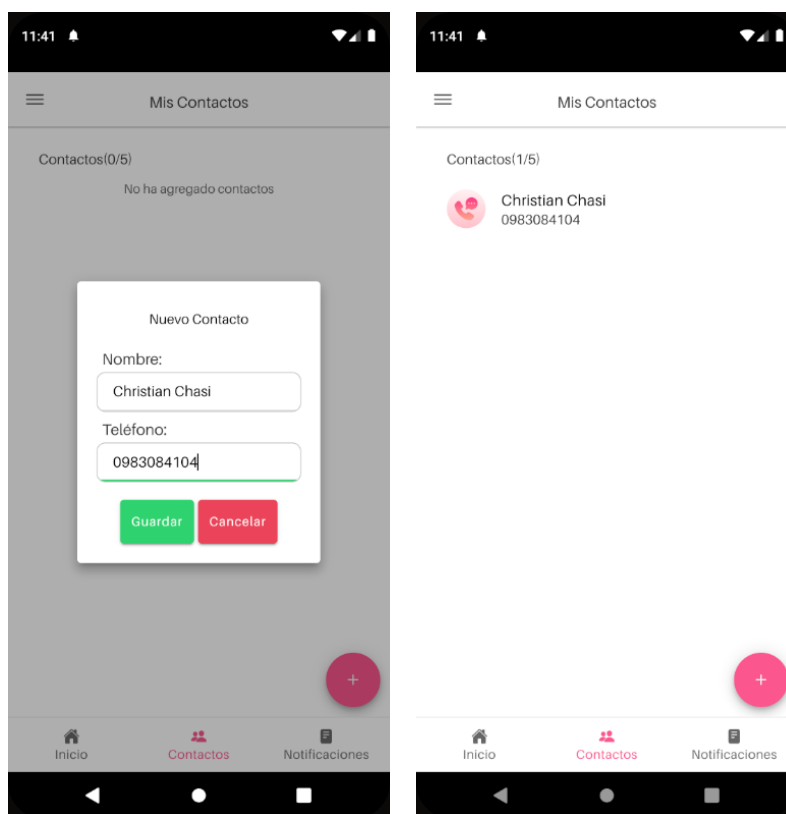


Figura 3.2.24 Iteración 4 – Interfaz contactos de emergencia

Elaborado por: Christian Chasi

3.2.5.4. Iteración 5: Historial de alertas

- **Día de planificación**

| Número | Tipo | Dificultad | | Esfuerzo | | Prioridad |
|--|---------|---------------|----------|----------|-------------------|-----------|
| | | Antes | Después | Estimado | Gastado | |
| 5 | Nuevo | Fácil | Fácil | 70h | 60h | Baja |
| | Arreglo | Moderada | Moderada | | | |
| | Mejora | Difícil | Difícil | | | |
| Descripción: Los usuarios con rol ciudadano recibirán alertas de auxilio que se almacenarán automáticamente en su dispositivo manteniendo un registro de estas. | | | | | | |
| Fecha | | Estado | | | Comentario | |
| 21/06/2021 | | Definido | | | | |
| 26/06/2021 | | Realizado | | | | |

Tabla 3.2.24 Story Card – Historial de alertas

Elaborado por: Christian Chasi

- **Día de Trabajo**

A continuación, se presenta fragmento del código más importante implementado para el módulo del historial de alertas. Se visualiza lo siguiente:

1. Se cargan todas las notificaciones del que se almacenaron en el dispositivo.
2. Se muestra el componente con el mapa y la información del usuario.

```
history.page.ts

notificaciones: AdditionalData[] = []
ciudadano: Ciudadano;

ngOnInit() {
  this.notificaciones = this.srvOneSignal.arrayMensajes;
  this.srvOneSignal.emitirNotificacionHistorial.subscribe((noty) => {
    this.notificaciones.unshift(noty);
    this.appref.tick();
  })
}

historialNotificaciones(index: number) {
  const noty = this.notificaciones[index];
  const { latitud, longitud, nombre, apellido, celular, direccion } = noty.payload.additionalData;
  this.ciudadano = new Ciudadano(null, nombre, apellido, null, latitud, longitud, direccion, null, celular);
  this.showModal();
}

async showModal() {
  const modal = await this.modalController.create({
    component: ViewAlertComponent,
    componentProps: {
      ciudadano_auxilio: this.ciudadano
    }
  });
  return await modal.present();
}

async eliminarNotificaciones() {
  this.notificaciones = [];
  this.srvOneSignal.eliminarNotificaciones();
}
}
```

Figura 3.2.25 Iteración 5 – Codificación historial de alertas

Elaborado por: Christian Chasi

- **Día de Liberación**

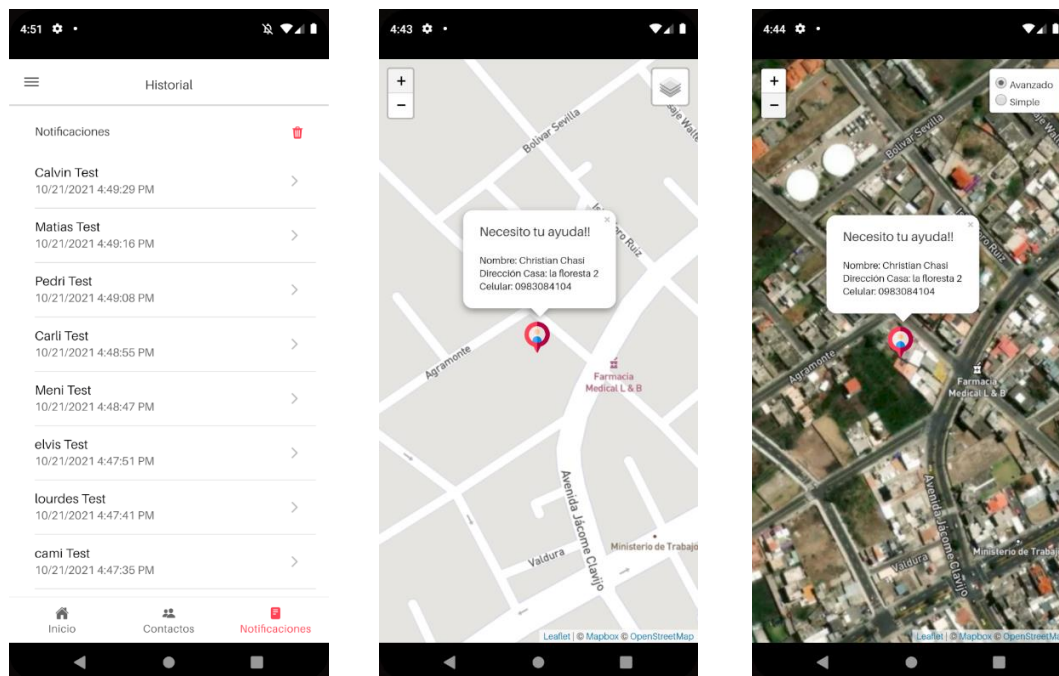


Figura 3.2.26 Iteración 5 – Interfaz de historial de notificaciones

Elaborado por: Christian Chasi

3.2.5.5. Iteración 6: Configuración y Perfil de Usuario

- **Día de planificación**

| Número | Tipo | Dificultad | | Esfuerzo | | Prioridad |
|---|---------|---------------|----------|----------|-------------------|-----------|
| | | Antes | Después | Estimado | Gastado | |
| 6 | Nuevo | Fácil | Fácil | 40h | 50h | Baja |
| | Arreglo | Moderada | Moderada | | | |
| | Mejora | Difícil | Difícil | | | |
| Descripción: Mediante un menú lateral se tendrá acceso a la opción de configuración que permitirá silenciar las notificaciones y acceder al perfil de usuario para modificar sus datos personales o de inicio de sesión. | | | | | | |
| Fecha | | Estado | | | Comentario | |
| 27/06/2021 | | Definido | | | | |
| 01/07/2021 | | Realizado | | | | |

Tabla 3.2.25 Story Card – Perfil de Usuario

Elaborado por: Christian Chasi

- **Día de Trabajo**

A continuación, se presenta fragmento del código más importante implementado para modificar el perfil de usuario. Se visualiza lo siguiente:

1. La función que permite al usuario actualizar sus datos personales y de autenticación.
2. Se valida los datos modificados.
3. Se llama a los métodos que se conectan con los servicios REST.

```
profile.page.ts

async actualizarDatos() {
  if (this.formulario.invalid || !this.formulario.dirty) { return }
  this.isActualizandoDatos = true;
  const { password, nombre, apellido, direccion, ubicacion, email, celular } = this.formulario.controls;
  const ciudadanoInfo = new Ciudadano(this.ciudadano.getId());
  if (nombre.dirty) ciudadanoInfo.setNombre = nombre.value;
  if (password.dirty) ciudadanoInfo.setPassword = password.value;
  if (apellido.dirty) ciudadanoInfo.setApellido = apellido.value;
  if (direccion.dirty) ciudadanoInfo.setDireccion = direccion.value;
  if (email.dirty) ciudadanoInfo.setEmail = email.value;
  if (celular.dirty) ciudadanoInfo.setCelular = celular.value;
  if (ubicacion.dirty) ciudadanoInfo.ubicacion = ubicacion.value;
  try {
    const response = await this.srvCitizen.actualizarUsuario(ciudadanoInfo);
    this.isActualizandoDatos = false;
    this.ciudadano = this.srvCitizen.instanciarCiudadano(response);
    this.srvCitizen.guardarCiudadanoLS(this.ciudadano);
    this.srvAlertSms.presentToast('Datos actualizados', 'success');
    if (email.dirty || password.dirty) {
      this.srvAlertSms.presentToast('Inicie sesión nuevamente', 'success');
      timer(2500).subscribe(() => this.exit());
    }
    this.formulario.reset(this.ciudadano);
  } catch (error) {
    if (error.statusText == "Unauthorized") {
      this.srvAlertSms.presentToast('Usuario Deshabilitado, contacte al administrador');
      this.srvCitizen.clearStorage();
      this.navCtrl.navigateRoot('login', { animated: true });
      this.srvOneSignal.obtenerAppIdOneSignal();
    }
    else {
      this.srvAlertSms.presentToast(error.error.msg, "danger");
    }
    this.formulario.reset(this.ciudadano);
    this.isActualizandoDatos = false;
  }
}
```

Figura 3.2.27 Iteración 6 – Codificación del perfil de usuario (Frontend)

Elaborado por Christian Chasi

Para la parte de los servicios (Backend) se muestra el fragmento de código más importante que le permite al usuario actualizar ciertos datos personales. Se visualiza lo siguiente

1. Recepción de los datos que provienen del cliente.
2. Actualización de los datos.

```
usuario.js

const updateCiudadano = async (req = request, res = response) => {
  try {
    const { id } = req.params;
    const { rol, password, latitud, longitud, ...ciudadano } = req.body;

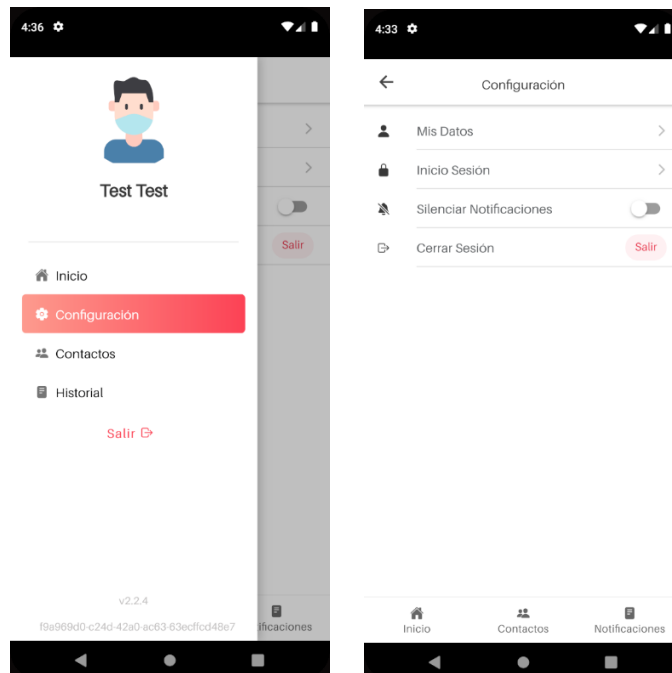
    if (latitud && longitud) {
      const ubicacion = { type: "Point", coordinates: [longitud, latitud] };
      ciudadano.ubicacion = ubicacion;
    }
    if (password) {
      const salt = bcryptjs.genSaltSync();
      ciudadano.password = bcryptjs.hashSync(password, salt);
    }

    const ciudadanoActualizado = await Ciudadano.findByIdAndUpdate(
      id,
      ciudadano,
      { new: true }
    );
    return res.json(ciudadanoActualizado);
  } catch (error) {
    res.status(500).json({ error: "Contactese con el administrador" });
  }
};
```

Figura 3.2.28 Iteración 6 – Codificación perfil de usuario (Backend)

Elaborado por: Christian Chasi

- **Día de Liberación**



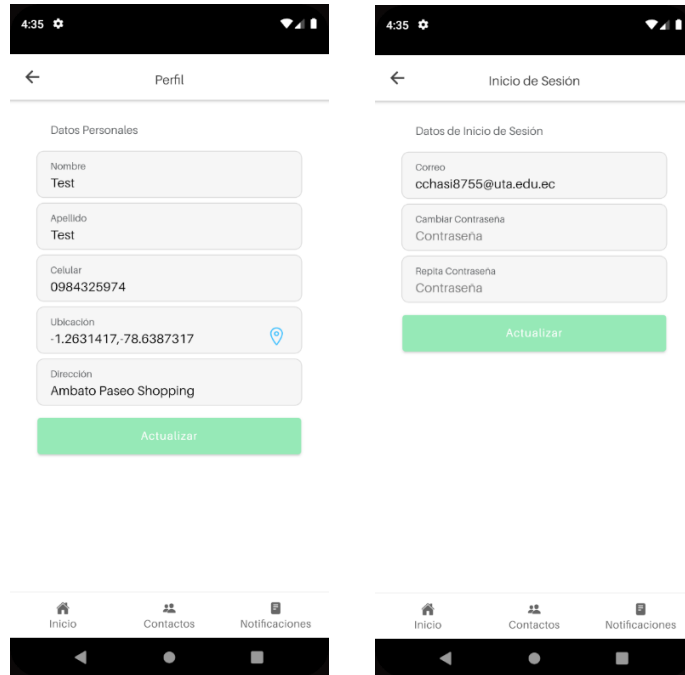


Figura 3.2.29 Iteración 6 – Interfaz perfil de usuario

Elaborado por: Christian Chasi

Aplicación Móvil – Monitoreo de Alertas

3.2.5.6. Iteración 7: Registro de usuarios

- Día de planificación

| Número | Tipo | Dificultad | | Esfuerzo | | Prioridad |
|---|---------|---------------|----------|----------|-------------------|-----------|
| | | Antes | Después | Estimado | Gastado | |
| 7 | Nuevo | Fácil | Fácil | 20h | 20h | Alta |
| | Arreglo | Moderada | Moderada | | | |
| | Mejora | Difícil | Difícil | | | |
| Descripción: Los usuarios con rol policía se registran mediante la aplicación proporcionando sus datos personales. | | | | | | |
| Fecha | | Estado | | | Comentario | |
| 12/07/2021 | | Definido | | | | |
| 15/07/2021 | | Realizado | | | | |

Tabla 3.2.26 Story Card – Registro de usuarios con rol policía

Elaborado por: Christian Chasi

- **Día de trabajo**

A continuación, se presentan fragmentos del código implementado para el registro de usuarios (Frontend). Se visualiza lo siguiente:

1. Se valida el formulario.
2. Se obtienen los datos del formulario
3. Se envían los datos a través del método que llama a los servicios REST

```
async register() {
  if (this.formulario.invalid) {
    this.formulario.markAllAsTouched();
    this.srvAlertSmS.presentAlert("Hay campos vacíos o incorrectos revíselos", "Campos Erróneos", "", true)
    this.srvAlertSmS.closeModalAlert(3000)
    return;
  }

  const { nombre, apellido, celular, email, password } = this.formulario.value;
  this.policia = new Policia(null, nombre, apellido, email, celular);
  this.policia.setPassword = password;
  await this.srvAlertSmS.presentLoading("Por favor espere...");
  this.srvAuth.registrar(this.policia).then((res: any) => {
    this.srvAlertSmS.loading.dismiss();
    if (res != false) {
      if (res.data.status == "ok") {
        this.srvAlertSmS.presentAlert(res.data.msg, "Registro", "", true);
        this.navCtrl.navigateRoot('login', { animated: true });
      }
    }
  })
  else {
    this.srvAlertSmS.presentAlert('No se encontro el ID del dispositivo. Cierre la aplicación e intente nuevamente', "Error", "", true);
  }
  this.srvAlertSmS.closeModalAlert(3000);

}, (err) => {
  this.srvAlertSmS.loading.dismiss();
  if (err.status == 0) {
    this.srvAlertSmS.presentAlert("No se pudo conectar con el servidor", "Servidor", "", true)
  }
  else if (err.status == 500) {
    this.srvAlertSmS.presentAlert("Error en el Servidor", "Servidor", "", true)
  }
  else if (err.error) {
    this.srvAlertSmS.presentAlert(err.error.msg, "Error al Registrar", "", true);
  }
  else {
    this.srvAlertSmS.presentAlert(err.error.msg, "Registro", "", true);
  }
  this.srvAlertSmS.closeModalAlert(3000);
}
);
}
```

Figura 3.2.30 Iteración 7 – Codificación de registro de usuario (Frontend)

Elaborado por: Christian Chasi

Para la parte de los servicios (Backend) se muestra el fragmento de código más importante para ejecutar el proceso de registro de usuarios, en este se visualiza:

1. Recepción de los datos que provienen del cliente.
2. Manejo de los datos y encriptación de la contraseña.

3. Guardado de la información en la Base de Datos

```
usuario.js

const registrarPolicias = async (req = request, res = response) => {
  try {
    const { _id, password, tokenDispositivo, rol, ...policia } = req.body;
    const role = await Rol.findOne({ rol });
    const usuario = new Policia(policia);
    usuario.rol = role;
    //Encripto contraseñas
    const salt = bcryptjs.genSaltSync();
    const passwordCrypt = bcryptjs.hashSync(password, salt);
    usuario.password = passwordCrypt;
    usuario.id_dispositivo = tokenDispositivo ;
    //Guardo el ciudadano y el dispositivo
    await usuario.save();
    res.json({ data: { status: "ok", msg: "Registro Exitoso" } });
  } catch (errores) {
    console.log("Errores", errores);
    res.status(500).json({ msg: "Error en el servidor", errores });
  }
};
```

Figura 3.2.31 Iteración 7 – Codificación de registro de usuario (Backend)

Elaborado por: Christian Chasi

- **Día de liberación**

En esta etapa se integra la parte del Frontend y Backend para el módulo de registro de usuarios, se realizan pruebas de funcionamiento y se libera.

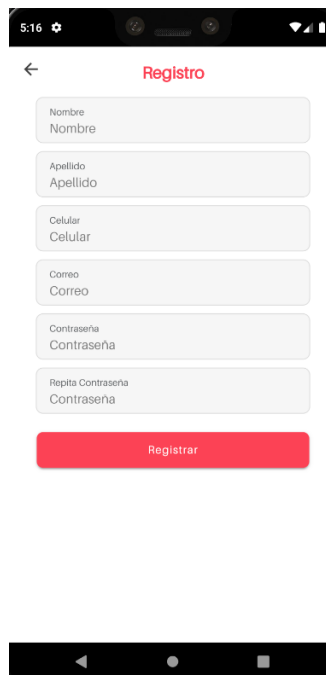


Figura 3.2.32 Iteración 7 – Interfaz de registro de usuarios

Elaborado por Christian Chasi

3.2.5.7. Iteración 8: Autenticación de usuarios

- **Día de planificación**

| Número | Tipo | Dificultad | | Esfuerzo | | Prioridad |
|--|---------|------------|----------|----------|------------|-----------|
| | | Antes | Después | Estimado | Gastado | |
| 8 | Nuevo | Fácil | Fácil | 30h | 30h | Alta |
| | Arreglo | Moderada | Moderada | | | |
| | Mejora | Difícil | Difícil | | | |
| Descripción: Los usuarios con rol policía pueden iniciar sesión (posteriormente de haberse registrado) con su correo y contraseña. También podrán restablecer su contraseña mediante su correo electrónico en caso de que lo requieran. | | | | | | |
| Fecha | | Estado | | | Comentario | |
| 15/07/2021 | | Definido | | | | |
| 17/07/2021 | | Realizado | | | | |

Tabla 3.2.27 Story Card – Autenticación de usuarios con rol policía

Elaborado por: Christian Chasi

- **Día de Trabajo**

A continuación, se presentan fragmentos del código más importante implementado para la autenticación de usuarios. Se visualiza lo siguiente:

1. Se valida el formulario.
2. Se obtienen los datos del formulario
3. Se envía a través del método que llama a los servicios REST

```
login.page.ts

async login() {
  if (this.formLogin.invalid) {
    this.formLogin.markAllAsTouched();
    return
  }
  await this.alertSms.presentLoading("Validando Datos...");
  try {
    this.policia = this.formLogin.value;

    var valido = await this.srvAuth.loguearse(this.policia);
    if (valido) {
      this.alertSms.loading.dismiss();
      await this.srvCitizen.getCiudadanoInfo();
      this.navCtrl.navigateRoot('tabs/history', { animated: true });
    }
  } catch (error) {
    this.alertSms.loading.dismiss();
    if (error.error.msg) {
      this.alertSms.presentAlert(error.error.msg, "Login", '', true)
    }
    else {
      this.alertSms.presentAlert("Ha ocurrido un error al iniciar sesión", "Login", '', true)
    }
  }
  this.alertSms.closeModalAlert(2000);
}
}
```

Figura 3.2.33 Iteración 8 – Codificación de autenticación de usuario (Frontend)

Elaborado por: Christian Chasi

Para la parte de los servicios (Backend) se muestra el fragmento de código más importante para ejecutar el proceso de autenticación de usuarios, en este se visualiza:

1. Validación y verificación del correo y contraseña.
2. Validación del dispositivo con el que se autentica.

```
autenticacion.js

const loginPolicia = async (req = request, res = response) => {
  const { email, password, tokenDispositivo } = req.body;
  try {
    const usuario = await Policia.findOne({ email });

    if (!usuario || !usuario.activo) {
      return res.status(400).json({ msg: "Email/Contraseña no son correctos" });
    }

    const validaPassword = bcryptjs.compareSync(password, usuario.password);

    if (!validaPassword) {
      return res.status(401).json({ msg: "Email/Contraseña no son correctos" });
    }
    const token = await generarJWT(usuario._id);

    usuario.suscrito = true; //Activo al usuario para recibir notificaciones

    if (tokenDispositivo == usuario.id_dispositivo) {
      usuario.save();
      return res.json({ data: { access_token: token } });
    }

    const existePoliciaDispositivo = await Policia.findOne({
      id_dispositivo: tokenDispositivo,
    });

    if (existePoliciaDispositivo) {
      existePoliciaDispositivo.id_dispositivo = null;
      await existePoliciaDispositivo.save();
    }
    usuario.id_dispositivo = tokenDispositivo;
    await usuario.save();
    return res.json({ data: { access_token: token } });
  } catch (error) {
    console.log("error:", error);
    res.status(500).json({ error: "Contactese con el administrador" });
  }
};
```

Figura 3.2.34 Iteración 8 – Codificación de autenticación de usuario (Backend)

Elaborado por: Christian Chasi

- **Día de Liberación**

En esta etapa se integra la parte del Frontend y Backend para la autenticación de usuarios, se realizan pruebas de funcionamiento y se libera.



Figura 3.2.35 Iteración 8 – Interfaz autenticación de usuario

Elaborado por: Christian Chasi

3.2.5.8. Iteración 9: Alertas de auxilio

- **Día de planificación**

| Número | Tipo | Dificultad | | Esfuerzo | | Prioridad |
|--------|---------|------------|----------|----------|---------|-----------|
| | | Antes | Después | Estimado | Gastado | |
| 9 | Nuevo | Fácil | Fácil | 12h | 12h | Alta |
| | Arreglo | Moderada | Moderada | | | |
| | Mejora | Difícil | Difícil | | | |

| Descripción: En este módulo el usuario o policía puede visualizar en una tabla o grid todas las alertas emitidas por los ciudadanos de los barrios. Cada alerta tiene la opción “Ver”, esta despliega el mapa con la información del ciudadano. | | |
|--|-----------|------------|
| Fecha | Estado | Comentario |
| 17/07/2021 | Definido | |
| 20/07/2021 | Realizado | |

Tabla 3.2.28 Story Card – Alertas de auxilio

Elaborado por: Christian Chasi

- **Día de Trabajo**

A continuación, se presenta fragmento del código más importante implementado para el módulo de Alertas de auxilio. Se visualiza lo siguiente:

1. Se obtienen los datos almacenados en el dispositivo.
2. Se muestra el componente del mapa con la información del usuario.

```

history.page.ts

historialNotificaciones(index: number) {
  const noty = this.notificaciones[index];
  const { latitud, longitud, nombre, apellido, celular, direccion } = noty.payload.additionalData;
  this.ciudadano = new Ciudadano(null, nombre, apellido, null, latitud, longitud, direccion, null, celular);
  this.showModal();
}

async showModal() {
  const modal = await this.modalController.create({
    component: ViewAlertComponent,
    componentProps: {
      ciudadano_auxilio: this.ciudadano
    }
  });
  return await modal.present();
}

```

Figura 3.2.36 Iteración 9 – Codificación para mostrar las alertas de auxilio

Elaborado por. Christian Chasi

- **Día de liberación**

En esta etapa se integra la parte del Frontend y Backend para mostrar las alertas de auxilio, se realizan pruebas de funcionamiento y se libera.

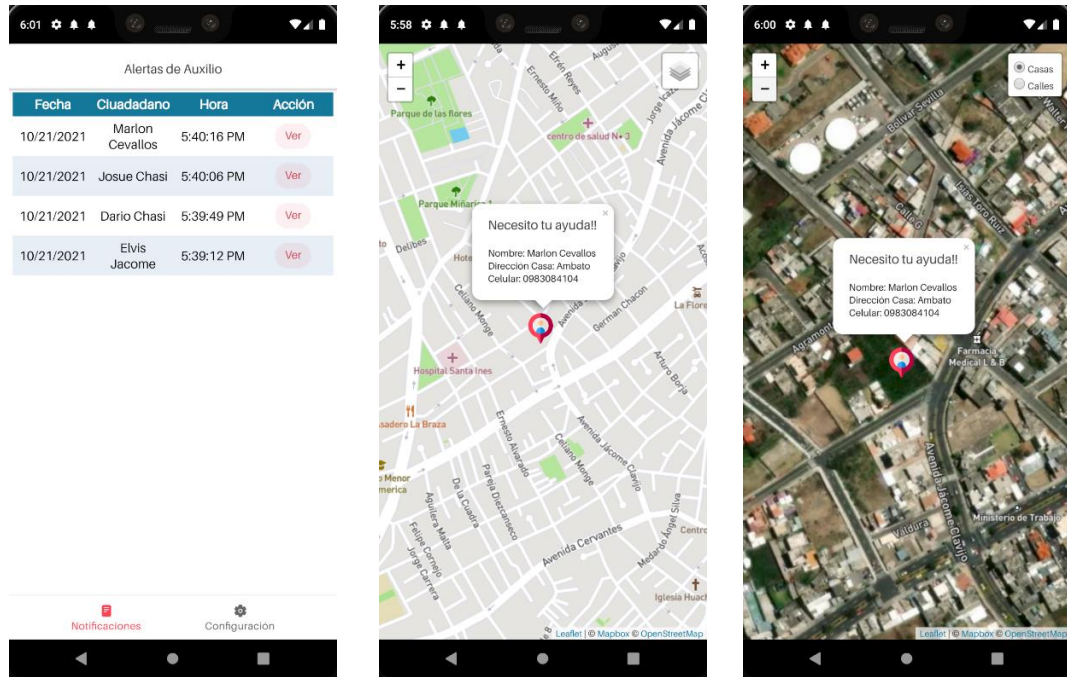


Figura 3.2.37 Iteración 9 – Interfaz alertas de auxilio

Elaborado por: Christian Chasi

3.2.5.9. Iteración 10: Configuración y Perfil de Usuario

- Día de planificación

| Número | Tipo | Dificultad | | Esfuerzo | | Prioridad |
|--|---------|---------------|----------|----------|-------------------|-----------|
| | | Antes | Después | Estimado | Gastado | |
| 10 | Nuevo | Fácil | Fácil | 24h | 24h | Alta |
| | Arreglo | Moderada | Moderada | | | |
| | Mejora | Difícil | Difícil | | | |
| <p>Descripción: Los usuarios con rol policía pueden actualizar sus datos personales cuando este lo requiera necesario, también podrán borrar el historial de notificaciones o desactivar las notificaciones de alertas de auxilio. Esta opción se encuentra como “Configuración” en la parte inferior derecha.</p> | | | | | | |
| Fecha | | Estado | | | Comentario | |
| 23/07/2021 | | Definido | | | | |
| 27/07/2021 | | Realizado | | | | |

Tabla 3.2.29 Story Card – Configuración y perfil de usuario

Elaborado por: Christian Chasi

- **Día de Trabajo**

A continuación, se presenta fragmento del código más importante implementado para el módulo de configuración y perfil de usuario. Se visualiza lo siguiente:

1. La función que permite al usuario actualizar sus datos personales y de autenticación.
2. Validación y obtención de los datos del formulario que fueron modificados.
3. Llamadas a los métodos que se conectan con el servicio REST.

```
mis-datos.component.ts

async actualizarDatos() {

  if (this.formulario.invalid || !this.formulario.dirty) { return }
  this.isActualizandoDatos = true;
  const { nombre, apellido, celular } = this.formulario.controls;
  const policiaInfo = new Policia(this.policia.getId);
  if (nombre.dirty) policiaInfo.setNombre = nombre.value;
  if (apellido.dirty) policiaInfo.setApellido = apellido.value;
  if (celular.dirty) policiaInfo.setCelular = celular.value;
  try {
    const response = await this.srvCitizen.actualizarUsuario(policiaInfo);
    this.policia = this.srvCitizen.instanciarPolicia(response);
    this.srvCitizen.guardarPoliciaLS(this.policia);
    this.srvAlertSms.presentToast('Datos actualizados', 'success');
  } catch (error) {
    if (error.statusText == "Unauthorized") {
      this.srvAlertSms.presentToast('Usuario Deshabilitado, contacte al administrador');
      this.srvCitizen.clearStorage();
      this.navCtrl.navigateRoot('login', { animated: true });
      this.srvOneSignal.obtenerAppIdOneSignal();
    }
    else if (error.error.msg) {
      this.srvAlertSms.presentToast(error.error.msg, "danger");
    }
    else{
      this.srvAlertSms.presentToast("No se ha podido actualizar los datos", "danger");
    }
  }
  this.formulario.reset(this.policia);
  this.isActualizandoDatos = false;
}
}
```

Figura 3.2.38 Iteración 10 – Codificación perfil de usuario (Frontend)

Elaborado por Christian Chasi

Para la parte de los servicios (Backend) se muestra el fragmento de código más importante que le permite al usuario actualizar sus datos personales.


```
usuario.js

const updatePolicia = async (req = request, res = response) => {
  try {
    const { id } = req.params;
    const { rol, password, ...policia } = req.body;

    if (password) {
      const salt = bcryptjs.genSaltSync();
      policia.password = bcryptjs.hashSync(password, salt);
    }
    const ciudadanoActualizado = await Policia.findByIdAndUpdate(id, policia, {
      new: true,
    });

    return res.json(ciudadanoActualizado);
  } catch (error) {
    res.status(500).json({ error: "Contactese con el administrador" }, error);
  }
};
```

Figura 3.2.39 Iteración 10 – Codificación perfil de usuario (Backend)

Elaborado por: Christian Chasi

• **Día de Liberación**

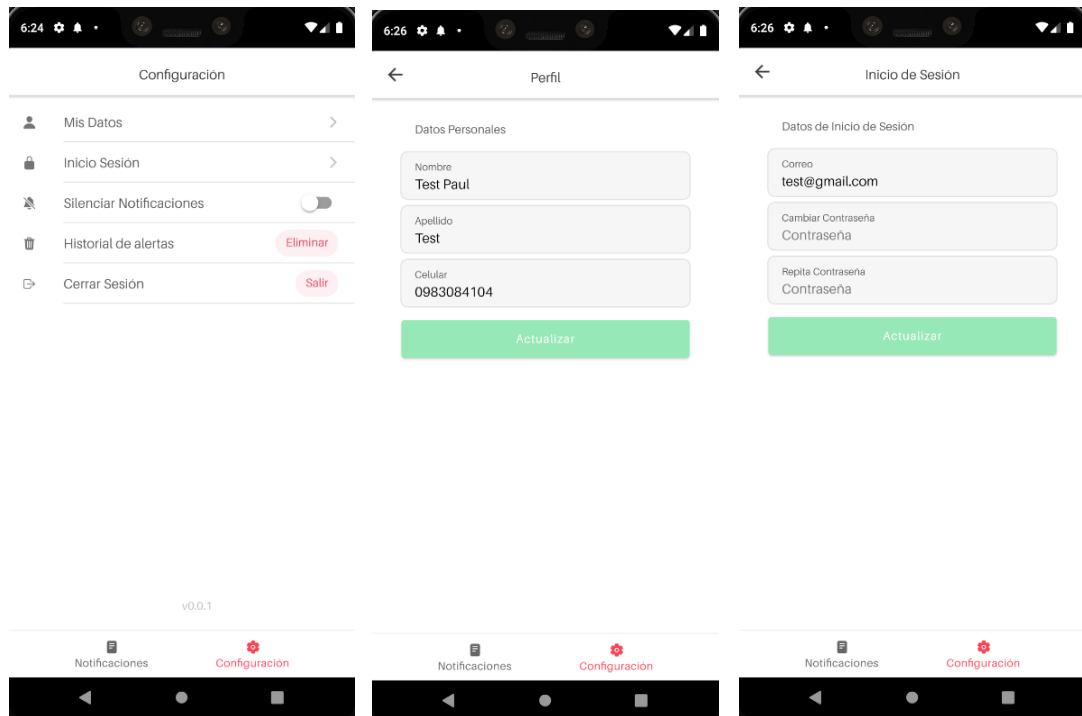


Figura 3.2.40 Iteración 10 – Interfaces de configuración y perfil de usuario

Elaborado por: Christian Chasi

3.2.6. Fase IV: Estabilización

En esta fase se consigue una aplicación completamente funcional con todos sus módulos integrados a nivel de Frontend y Backend.

Es decir, se integró todos los módulos (registro, autenticación, alertas, contactos, historial, configuración y perfil de usuario) de las dos aplicaciones con los servicios REST y Base de Datos.

Como resultados las dos aplicaciones trabajan de acuerdo de los requerimientos y funcionalidades especificadas consiguiendo un aplicativo estable. También se realiza el respectivo manual de usuario.

3.2.7. Fase V: Pruebas

En esta fase o etapa de desarrollo se verifican las funcionalidades del sistema en función de los requisitos, se realizan pruebas que garantizarán el correcto funcionamiento de las aplicaciones con el objetivo de lanzarlo a producción.

3.2.7.1. Pruebas Unitarias

Permiten estimar el funcionamiento correcto de una sección o unidad de código para detectar errores de codificación o lógica durante el desarrollo del sistema.

La aplicación de **Monitoreo de Alertas** trabaja en base al mismo código fuente de la aplicación **Barrio Seguro**, por lo tanto, las siguientes pruebas son válidas para comprobar el funcionamiento de ambos aplicativos.

| | |
|-------------------------|--|
| N° | PU001 |
| Nombre: | Validación de ingreso de datos en los formularios. |
| Inicialización: | En los formularios de autenticación, registro, perfil de usuario y contactos. |
| Salida Esperada: | Que el formulario valide y muestre los respectivos mensajes de validación durante o después del ingreso de datos. |
| Procedimiento: | Se ingresan datos diferentes al que corresponde cada campo del formulario (texto, números, caracteres especiales) para verificar que estos se validen. |


| | |
|--------------------------------|---|
| <p>Salida Obtenida:</p> | <p>Cuando ingreso datos incorrectos o que no corresponden al campo, el formulario realiza la validación y muestra el respectivo mensaje al usuario cuando son:</p> <ul style="list-style-type: none"> • Solo letras o números. • La longitud mínima y máxima de caracteres. • Si es un correo válido o no. |
| <p>Capturas:</p> |  |

Tabla 3.2.30 Prueba Unitaria – Validación de formularios

Elaborado por: Christian Chasi

| | |
|--------------------------------|---|
| <p>N°</p> | <p>PU002</p> |
| <p>Nombre:</p> | <p>Mensaje de aviso en el envío de alertas de auxilio.</p> |
| <p>Inicialización:</p> | <p>En el módulo de alertas de auxilio.</p> |
| <p>Salida Esperada:</p> | <p>Mostrar un mensaje en la pantalla que confirme el envío de la alerta y el total de dispositivos.</p> |
| <p>Procedimiento:</p> | <p>Presionar el botón de pánico.</p> |
| <p>Salida Obtenida:</p> | <p>Al presionar el botón de pánico se envía la alerta y después aparece un mensaje de confirmación con el número de dispositivos al que llegó la alerta de auxilio.</p> |


| | |
|------------------|--|
| Capturas: |  |
|------------------|--|

Tabla 3.2.31 Prueba Unitaria – Mensaje de aviso de alerta de auxilio

Elaborado por: Christian Chasi

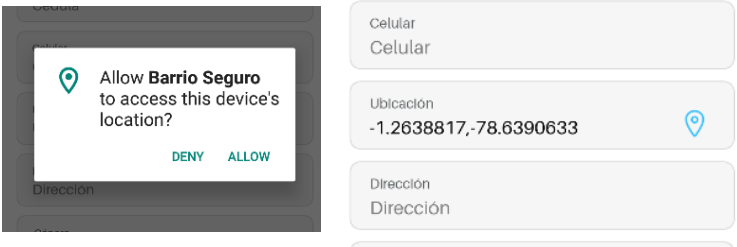
| | |
|-------------------------|---|
| N° | PU003 |
| Nombre: | Obtener la ubicación mediante el GPS |
| Inicialización: | En el módulo de registro, en el módulo de alertas de auxilio y en el módulo de actualizar el perfil. |
| Salida Esperada: | Al presionar el icono de obtener ubicación debe cargar la ubicación actual (latitud y longitud) del usuario. |
| Procedimiento: | Dirigirse al módulo de registro, alertas de auxilio o perfil de usuario y presionar el ícono de obtener ubicación. (Es importante dar permisos de Ubicación). |
| Salida Obtenida: | Cuando presiona el botón para obtener la ubicación se carga la ubicación actual (latitud y longitud) del usuario. |
| Capturas: |  |

Tabla 3.2.32 Prueba Unitaria – Ubicación mediante GPS

Elaborado por: Christian Chasi

| | |
|----------------|-------------------------------------|
| N° | PU004 |
| Nombre: | Sonidos de las notificaciones push. |

| | |
|-------------------------|--|
| Inicialización: | Cuando llega una notificación push. |
| Salida Esperada: | Emitir un sonido de alerta de sirena de policía. |
| Procedimiento: | Enviar una notificación de auxilio presionando el botón de pánico. Esperar la notificación push en otro dispositivo |
| Salida Obtenida: | Cuando llega la notificación push emite el sonido de alerta de sirena de policía que permite identificar que se trata de un pedido de auxilio. |

Tabla 3.2.33 Prueba Unitaria – Sonidos de las notificaciones push

Elaborado por: Christian Chasi

| | |
|-------------------------|---|
| N° | PU005 |
| Nombre: | Lectura y escritura de información en el almacenamiento del dispositivo. |
| Inicialización: | Al autenticarse, en el módulo de historial de alertas y en el módulo de contactos. |
| Salida Esperada: | Que la información se almacene en dispositivo, pueda ser leída y reescrita. |
| Procedimiento: | Mediante las herramientas de desarrollador se monitorea el almacenamiento del dispositivo verificando que la información del usuario y los tokens se almacenen y se muestren. Otra manera en que se comprueba es cuando se cierra la aplicación y se abre nuevamente, si el usuario ya está autenticado y tiene el token en el almacenamiento, abre directamente al módulo de las alertas de auxilio de lo contrario va directo al módulo de autenticación. |
| Salida Obtenida: | Con las herramientas de desarrollador se visualiza como se almacena la información en el dispositivo cuando el usuario se autentica, cuando llega una alerta de auxilio y cuando se agrega un contacto. También al cerrar y abrir la aplicación se dirige al módulo de alertas de auxilio lo que significa que la información se lee correctamente del almacenamiento del dispositivo. |

| | |
|------------------|--|
| Capturas: | |
|------------------|--|

Tabla 3.2.34 Prueba Unitaria – Escritura y lectura de información en el dispositivo

Elaborado por: Christian Chasi

| | |
|-------------------------|--|
| N° | PU006 |
| Nombre: | Restablecimiento de contraseña. |
| Inicialización: | En el módulo de autenticación. |
| Salida Esperada: | Se envía un correo electrónico con la nueva contraseña de inicio de sesión. Posteriormente se ingresa la nueva contraseña y se autentica. |
| Procedimiento: | En el módulo de autenticación se escoge la opción de Olvidé mi Contraseña , luego en el campo que dice Email se ingresa un correo, el formulario lo valida y se envía. |
| Salida Obtenida: | Si el correo es válido, llegará un mensaje en la bandeja de entrada que contiene la nueva contraseña de inicio de sesión. Cuando se ingresa el correo y la nueva contraseña el usuario se autentica correctamente. |
| Capturas: | |

Tabla 3.2.35 Prueba Unitaria – Restablecer contraseña

Elaborado por: Christian Chasi

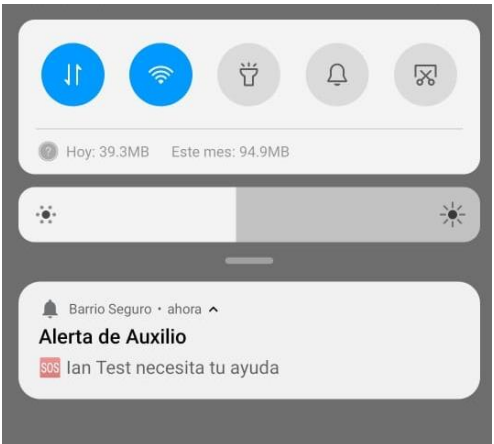
| | |
|-------------------------|--|
| N° | PU007 |
| Nombre: | Recepción de la notificación push o alerta de auxilio. |
| Inicialización: | En el status bar del dispositivo celular. |
| Salida Esperada: | Que aparezca la notificación push de una alerta de auxilio en el status bar del dispositivo. |
| Procedimiento: | Se envía una alerta de auxilio desde un dispositivo y se visualiza la llegada de esta en otro dispositivo. |
| Salida Obtenida: | Cuando se emite una alerta de auxilio desde un dispositivo, esta aparece en el estatus bar del dispositivo que recibió la alerta. |
| Capturas: |  <p>The image shows a close-up of the top portion of an iPhone screen. At the very top is the status bar with icons for cellular signal, Wi-Fi, battery, and a notification bell. Below the status bar is a notification card. The notification card has a bell icon on the left, followed by the text 'Barrio Seguro · ahora ^'. Below that, in bold, is 'Alerta de Auxilio'. At the bottom of the notification card, there is a red 'SOS' icon followed by the text 'Ian Test necesita tu ayuda'.</p> |

Tabla 3.2.36 Prueba Unitaria – Recepción de la notificación push

Elaborado por: Christian Chasi

| N° | Prueba | Id Prueba | Cumple |
|-----------|--|------------------|---------------|
| 1 | Validación de ingreso de datos en los formularios. | PU001 | Si |
| 2 | Mensaje de aviso en el envío de alertas de auxilio. | PU002 | Si |
| 3 | Obtener la ubicación mediante el GPS. | PU003 | Si |
| 4 | Sonidos de las notificaciones push. | PU004 | Si |
| 5 | Lectura y escritura de información en el almacenamiento del dispositivo. | PU005 | Si |
| 6 | Restablecimiento de contraseña. | PU006 | Si |

| | | | |
|---|--|-------|----|
| 7 | Recepción de la notificación push o alerta de auxilio. | PU007 | Si |
|---|--|-------|----|

Tabla 3.2.37 Resumen de Pruebas Unitarias

Elaborado por: Christian Chasi

3.2.7.2. Pruebas de Aceptación

Estas pruebas se realizan en función de las historias de usuarios y se evalúa el funcionamiento descrito en ellas.

| Prueba de Aceptación | |
|--|-----------------------------|
| Número: 001 | Evaluación: Aprobado |
| Historia de usuario: 001 | |
| Nombre de caso de prueba: Registro de ciudadanos | |
| Descripción: <ul style="list-style-type: none"> • El usuario con rol ciudadano debe crear una cuenta con sus datos personales (nombre, apellido, celular, dirección, cédula, email) y su ubicación actual (obtenida mediante el GPS del dispositivo). • Se debe validar que sea una cédula real. • En caso de que la cédula, correo o celular ya existan se debe mostrar el mensaje al usuario. • Antes del registro todos los datos deben ser validados. | |
| Resultado: <ul style="list-style-type: none"> • Se ingresan los datos en el formulario y se obtiene la ubicación pulsando el botón (debe dar permisos de ubicación). • Si la cédula es falsa, aparece un mensaje que ingrese una cédula válida. • Si la cédula, correo o celular ya existen entonces se muestra un mensaje al usuario que describe el campo inválido. • Cuando todos los datos están correctos el registro es exitoso. | |
| Conclusión: Con esto se cumplen los prerequisites de registro de usuarios y validación de los datos ingresados. | |

Tabla 3.2.38 Prueba de aceptación – Registro de Usuarios con rol ciudadano

Elaborado por: Christian Chasi

| Prueba de Aceptación | |
|---|-----------------------------|
| Número: 002 | Evaluación: Aprobado |
| Historia de usuario: 002 | |
| Nombre de caso de prueba: Autenticación de Usuarios | |
| Descripción: | |
| <ul style="list-style-type: none"> • El usuario debe iniciar sesión con el correo y contraseña registrados previamente. • En la pantalla principal también debe haber las opciones para restablecer la contraseña y registrarse si el usuario no tiene una cuenta. | |
| Resultado: | |
| <ul style="list-style-type: none"> • Para iniciar sesión el usuario ingresa los datos correo y contraseña, estos se validan tanto en el Frontend y Backend, si son correctos ingresa a la pantalla principal o de lo contrario aparece un aviso de datos incorrectos. • Si el usuario quiere restablecer la contraseña, debe ir a la opción de “Olvidé mi contraseña” ubicada en la pantalla principal del login, posteriormente ingresar su correo con el que se registró se validan que los datos sean correctos y se envía la nueva contraseña al correo electrónico del usuario. | |
| Conclusión: Con esto se cumplen los prerrequisitos de autenticación y restablecimiento de la contraseña. | |

Tabla 3.2.39 Prueba de aceptación – Autenticación de usuarios con rol ciudadano

Elaborado por: Christian Chasi

| Prueba de Aceptación | |
|---|-----------------------------|
| Número: 003 | Evaluación: Aprobado |
| Historia de usuario: 003 | |
| Nombre de caso de prueba: Alertas de auxilio. | |
| Descripción: | |
| <ul style="list-style-type: none"> • Si el usuario ingresa a la aplicación y está autenticado puede enviar alertas de auxilio (presionando el botón de pánico). • En la alerta de auxilio se envía la ubicación (actual o con la que se registró) | |

| |
|---|
| <ul style="list-style-type: none"> • El usuario puede escoger el radio de distancia de alcance de envío de la alerta o utilizar una distancia ya definida para que la acción del pedido de auxilio sea rápida. • Los dispositivos objetivo recibirán la alerta con la información mediante notificaciones push (con un sonido predeterminado) que al dar clic se abrirá un mapa con información de la persona. • El mapa se debe mostrar los marcadores con la ubicación e información de: <ol style="list-style-type: none"> 3. El ciudadano que envía la alerta de auxilio. 4. El ciudadano que recibe la alerta. |
| <p>Resultado:</p> <ul style="list-style-type: none"> • Al presionar el botón de pánico se envía una alerta a los dispositivos cercanos en un radio de 500 metros. • Se puede modificar la ubicación y la distancia de alcance de la alerta. • Si escojo otras distancias mayores a 500 metros, la alerta de auxilio llega a más dispositivos o personas. • Cuando la persona recibe la notificación en su dispositivo, al dar clic en esta, se abre una vista con un mapa que contiene dos marcadores de posición con modales que muestran la información de los ciudadanos. |
| <p>Conclusión: Con esto se cumplen los prerequisites de rangos de distancia, ubicación actual, alertas sonoras de las notificaciones push, mapas y marcadores.</p> |

Tabla 3.2.40 Prueba de aceptación – Alertas de auxilio

Elaborado por: Christian Chasi

| Prueba de Aceptación | |
|--|-----------------------------|
| Número: 004 | Evaluación: Aprobado |
| Historia de usuario: 004 | |
| Nombre de caso de prueba: Configuración y Perfil de usuario. | |
| <p>Descripción:</p> <ul style="list-style-type: none"> • El usuario puede gestionar su perfil, es decir actualizar sus datos personales (nombre, apellido, dirección, etc.), ubicación, correo y la contraseña. • También podrá silenciar las alertas si así lo requiere. | |
| Resultado: | |

| |
|--|
| <ul style="list-style-type: none"> • Para acceder al perfil, el usuario debe pulsar el botón del menú o deslizar su dedo hacia la derecha y aparecerán entre varias opciones una que dice “Configuración”. Al dar clic en esta aparecen otros submenús en los que el usuario puede modificar sus datos personales o de inicio de sesión. Cuando se modifica los datos de algún campo, estos se validan y si todo esta correcto se activa el botón para poder actualizar sus datos. • También está la opción para evitar recibir notificaciones o alertas de auxilio. |
| <p>Conclusión: Con esto se cumplen los prerequisites de actualizar datos personales, actualizar la ubicación actual y silenciar las notificaciones.</p> |

Tabla 3.2.41 Prueba de aceptación – Perfil de usuario

Elaborado por: Christian Chasi

| Prueba de Aceptación | |
|--|-----------------------------|
| Número: 005 | Evaluación: Aprobado |
| Historia de usuario: 005 | |
| Nombre de caso de prueba: Historial de alertas. | |
| Descripción: | |
| <ul style="list-style-type: none"> • Las notificaciones se almacenan en el dispositivo con el fin de mantener un registro o listado de alertas. • El módulo debe tener una opción para eliminar todo el historial de alertas. | |
| Resultado: | |
| <ul style="list-style-type: none"> • Cuando llega una notificación, esta se almacena en el dispositivo, se puede acceder al listado a través del módulo de historial de alertas, una vez allí, se previsualiza del nombre y la hora de la alerta de auxilio, al pulsar el registro se muestra un mapa con la información de la persona. • El historial de alertas se elimina pulsando el ícono del basurero. | |
| Conclusión: Con esto se cumple el prerequisite de historial de alertas. | |

Tabla 3.2.42 Prueba de aceptación – Historial de alertas

Elaborado por: Christian Chasi

| Prueba de Aceptación | |
|-----------------------------|-----------------------------|
| Número: 006 | Evaluación: Aprobado |

| |
|--|
| Historia de usuario: 006 |
| Nombre de caso de prueba: Contactos de emergencia. |
| Descripción: <ul style="list-style-type: none"> • El usuario puede agregar y eliminar contactos de emergencia para que reciban las alertas de auxilio. |
| Resultado: <ul style="list-style-type: none"> • El usuario pulsa un botón que abre una ventana modal en el que se debe ingresar el nombre y número de celular, cuando ya se ingresa los datos, el formulario los valida y se activa el botón que guarda la información. • Si el número de celular ingresado no corresponde a ningún usuario registrado se muestra el respectivo mensaje de error. Caso contrario el contacto se guarda y se muestra en la lista. • Para eliminar un contacto se pulsa sobre él y se mostrara el icono para borrarlo. |
| Conclusión: Con esto se cumplen los prerequisites de agregar y eliminar contactos de emergencia. |

Tabla 3.2.43 Prueba de aceptación – Contactos de emergencia

Elaborado por: Christian Chasi

Aplicación Móvil – Monitoreo de alertas

| Prueba de Aceptación | |
|--|-----------------------------|
| Número: 007 | Evaluación: Aprobado |
| Historia de usuario: 007 | |
| Nombre de caso de prueba: Registro de usuarios | |
| Descripción: <ul style="list-style-type: none"> • El usuario con rol policía debe crear una cuenta con sus datos personales (nombre, apellido, celular, correo y contraseña). • El usuario se registra siempre y cuando todos los datos del formulario son válidos. | |
| Resultado: <ul style="list-style-type: none"> • Si el correo o celular ya existen entonces se muestra un mensaje al usuario que describe el campo inválido. | |

| |
|--|
| <ul style="list-style-type: none"> • Cuando todos los datos están correctos se crea la cuenta del usuario. |
| <p>Conclusión: Con esto se cumplen los requisitos de registro de usuarios y validación de los datos ingresados.</p> |

Tabla 3.2.44 Prueba de aceptación – Registro de Usuarios con rol policía

Elaborado por: Christian Chasi

| Prueba de Aceptación | |
|---|-----------------------------|
| Número: 008 | Evaluación: Aprobado |
| Historia de usuario: 008 | |
| Nombre de caso de prueba: Autenticación de usuarios | |
| <p>Descripción:</p> <ul style="list-style-type: none"> • El usuario debe iniciar sesión con el correo y contraseña registrados previamente. • En la pantalla principal también debe haber las opciones para restablecer la contraseña y registrarse si el usuario no tiene una cuenta. | |
| <p>Resultado:</p> <ul style="list-style-type: none"> • Al ingresar los datos (correo y contraseña) se validan tanto en el Frontend y Backend, si son correctos ingresa al módulo principal caso contrario notifica al usuario que el correo y contraseña no son correctos. | |
| <p>Conclusión: Con esto se cumplen los requisitos de autenticación y restablecimiento de la contraseña.</p> | |

Tabla 3.2.45 Prueba de aceptación – Autenticación de usuarios con rol policía

Elaborado por: Christian Chasi

| Prueba de Aceptación | |
|--|-----------------------------|
| Número: 009 | Evaluación: Aprobado |
| Historia de usuario: 009 | |
| Nombre de caso de prueba: Alertas de auxilio. | |
| <p>Descripción:</p> <ul style="list-style-type: none"> • Las notificaciones se almacenan en el orden en que llegan en el dispositivo con el fin de mantener un registro o listado de alertas y poder visualizarlas posteriormente. | |

| |
|---|
| <p>Resultado:</p> <ul style="list-style-type: none"> • Cuando llega una notificación, esta se almacena en el dispositivo, se puede acceder al listado a través del módulo de historial de alertas, una vez allí, se previsualiza la fecha, nombre y la hora de la alerta de auxilio, al pulsar en el botón que dice “VER” en un elemento del registro se muestra un mapa con la ubicación e información de la persona que solicitó ayuda. |
| <p>Conclusión: Con esto se cumplen los requisitos de almacenar y visualizar alertas de auxilio.</p> |

Tabla 3.2.46 Prueba de aceptación – Alertas de auxilio

Elaborado por: Christian Chasi

| Prueba de Aceptación | |
|---|-----------------------------|
| Número: 010 | Evaluación: Aprobado |
| Historia de usuario: 010 | |
| Nombre de caso de prueba: Configuración y Perfil de usuario. | |
| <p>Descripción:</p> <ul style="list-style-type: none"> • El usuario con rol policía puede configurar gestionar su perfil, es decir podrá actualizar sus datos personales, correo y contraseña de inicio de sesión, también eliminar el historial de notificaciones o alertas y cerrar sesión. | |
| <p>Resultado:</p> <ul style="list-style-type: none"> • Si el usuario quiere actualizar su información debe dirigirse a la opción ubicada en la parte inferior derecha que dice Perfil. Una vez ahí edita los campos que desea y se activará el botón que dice Actualizar. Si todos los datos están correctos se actualiza exitosamente. | |
| <p>Conclusión: Con esto se cumplen los prerequisites de actualizar datos personales, de inicio de sesión y eliminar registro de alertas.</p> | |

Tabla 3.2.47 Prueba de aceptación – Perfil de usuario

Elaborado por: Christian Chasi

3.2.7.3. Conclusión de pruebas

En función de todas las pruebas realizadas se concluye que se han cumplido con todos los requisitos definidos previamente para el desarrollo de las 2 aplicaciones para dispositivos móviles. Las pruebas han sido satisfactorias dando como resultado

aplicaciones completamente funcionales y listas para ser lanzadas a producción para ser utilizadas por los ciudadanos de los barrios descargándolas desde la Play Store de Google.

3.2.8. Publicación en la Tienda de aplicaciones

Las aplicaciones se publica en la Google Play y están disponibles para todos los dispositivos android.

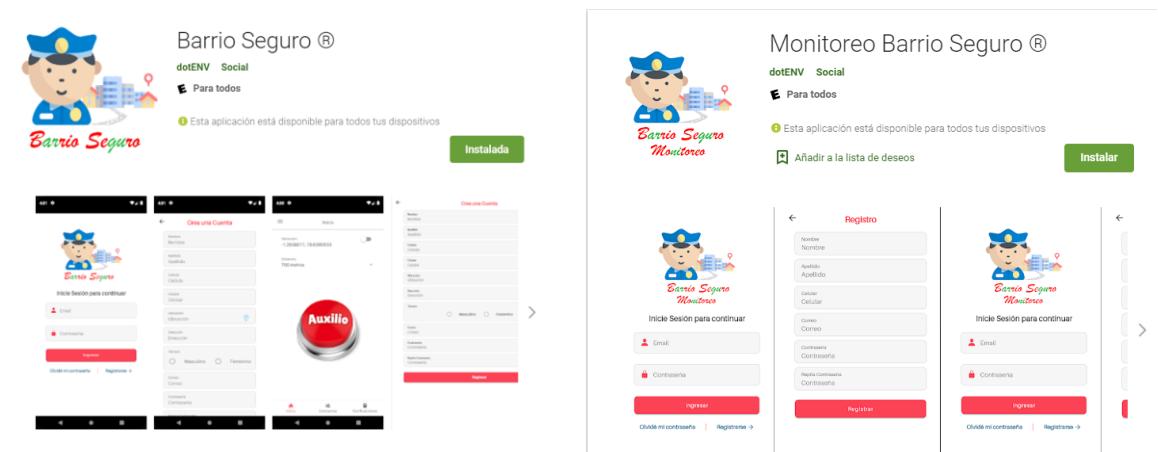


Figura 3.2.41 Aplicaciones publicadas en la Google Play Store

Elaborado: Christian Chasi

CAPITULO IV

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

- La seguridad de los barrios está a cargo de entes como policía, el COMSECA y la colaboración de los ciudadanos de barrios organizados contra la delincuencia.
- La tecnología push es una herramienta muy interesante que permite notificar eventos en tiempo real, en ese sentido el que mejor se adaptó al sistema fue OneSignal por su rapidez, fácil configuración e integración al proyecto.
- Ionic y Angular permitieron desarrollar el aplicativo en un corto tiempo, utilizando buenas y modernas prácticas de programación para poder mantener un código legible.
- La aplicación es rápida y notifica los eventos a todos los ciudadanos que tengan el aplicativo y estén en el rango de distancia definido o seleccionado.
- La alerta de auxilio llega aún si el teléfono está bloqueado o la aplicación está cerrada.
- La metodología Mobile-D se adecuó correctamente al proyecto y no influyó en costos extra, además permitió el desarrollo rápido de las aplicaciones cliente y servidor.
- Utilizar servidores gratuitos en internet no interfieren en el rendimiento de la aplicación, la principal desventaja serían las limitaciones del plan gratuito cuando la cantidad de usuarios crece.

Recomendaciones

- Se recomienda que los usuarios siempre utilicen un correo al que tengan acceso en caso de que olviden su contraseña y puedan restablecerla fácilmente.
- Se recomienda evitar lugares muy cerrados ya que el GPS puede dar ubicaciones menos precisas.
- Se recomienda que los usuarios en lo posible mantengan siempre activadas las notificaciones del aplicativo y el celular.

- Se recomienda utilizar servidores propios en caso de que el número de usuarios crezca.
- Se recomienda que los ciudadanos sean más participativos y asistan a reuniones de socialización o capacitación para el uso del aplicativo.
- En un futuro se recomienda trabajar también con tecnología SMS.
- Si existe apoyo económico se podría analizar el lanzamiento de la aplicación para dispositivos iOS.

BIBLIOGRAFÍA

- [1] S. Vizcarra y D. Bonilla, «Tema central Inseguridad y mecanismos barriales de», *Rev. Latinoam. Estud. Secur.*, vol. 19, 2016.
- [2] H. V. Fonseca, «El desarrollo tecnológico en materia policial: una receta de éxito para la prevención del delito», *Rev. Relac. Int. Estrateg. y Secur.*, vol. 15, n.º 1, pp. 79-97, 2020, doi: 10.18359/ries.4243.
- [3] Instituto Nacional de Estadísticas y Censos, «Tecnologías de la Información y Comunicación, 2020», 2021. Accedido: jun. 22, 2021. [En línea]. Disponible en: https://www.ecuadorencifras.gob.ec/documentos/web-inec/Estadisticas_Sociales/TIC/2020/202012_Principales_resultados_Multiproposito_TIC.pdf.
- [4] Fiscalía General del Estado, «Fiscalía General del Estado | Cifras de robos», 2021. <https://www.fiscalia.gob.ec/estadisticas-de-robos/>.
- [5] Instituto Nacional de Estadísticas y Censos (INEC), «Encuesta de Victimización y Percepción de Inseguridad 2011». https://www.ecuadorencifras.gob.ec/camas/web-inec/Victimizacion/FRESULTADOS_INDICADORES_ENVIPI_2011.xlsx&usg=AOvVaw0HIHBr-pbN.
- [6] La Hora, «En Tungurahua incrementan llamadas por violencia intrafamiliar pero las denuncias disminuyen : Noticias Tungurahua : La Hora Noticias de Ecuador, sus provincias y el mundo», 2018. <https://lahora.com.ec/tungurahua/noticia/1102156596/en-tungurahua-incrementan-llamadas-por-violencia-intrafamiliar-pero-las-denuncias-disminuyen>.
- [7] La Hora, «Robos, un problema que se vive a diario en Santa Rosa | Diario La Hora», 2021. <https://www.lahora.com.ec/tungurahua/robos-un-problema-que-se-vive-a-diario-en-santa-rosa/> (accedido ene. 17, 2022).
- [8] Ministerio del Interior (MDI), «PLAN ESPECÍFICO DE SEGURIDAD PÚBLICA Y CIUDADANA», 2019. [En línea]. Disponible en:

<https://www.defensa.gob.ec/wp-content/uploads/downloads/2019/07/plan-nacional-min-interior-web.pdf>.

- [9] S. G. P. BORBOR, «DESARROLLO DE APLICACIÓN MÓVIL UTILIZANDO TECNOLOGÍA PUSH PARA NOTIFICACIONES DE CALIFICACIONES EN LA UNIDAD EDUCATIVA SANTA TERESITA», 2016.
- [10] M. C. B. ELENA, «ANÁLISIS DE LA TECNOLOGÍA PUSH PARA EL DESARROLLO DE UNA APLICACIÓN MÓVIL ANDROID DE NOTIFICACIÓN DE EVENTOS Y MENSAJERÍA GRUPAL EN TIEMPO REAL PARA LA CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES», 2018.
- [11] R. Ramírez Vique, «Métodos para el desarrollo de aplicaciones móviles».
- [12] B. Fling, «Mobile Design and Development Practical Concepts and Techniques for Creating Mobile Sites and Web Apps (Animal Guide) by Brian Fling (z-lib.org)», p. 334, 2009.
- [13] L. Delía, N. Galdamez, P. Thomas, y P. Pesado, «Un Análisis Experimental de Tipo de Aplicaciones para Dispositivos Móviles».
- [14] Bambumobile, «Tecnologías clave para Desarrollo de Aplicaciones - Bambú Mobile». <https://www.bambu-mobile.com/tecnologias-clave-para-el-desarrollo-de-apps/> (accedido ago. 28, 2021).
- [15] Kellton Tech, «Top Mobile app development frameworks 2021». <https://www.kelltontech.com/kellton-tech-blog/top-7-mobile-app-development-frameworks-in-2021> (accedido ago. 30, 2021).
- [16] guru99, «API vs Web Service: What's the Difference?» <https://www.guru99.com/api-vs-web-service-difference.html> (accedido nov. 30, 2021).
- [17] RedHat, «¿Qué es una API?» <https://www.redhat.com/en/topics/api/what-are-application-programming-interfaces> (accedido nov. 30, 2021).
- [18] Y. Daniel y A. Balaguera, «Metodologías ágiles en el desarrollo de aplicaciones

para dispositivos móviles. Estado actual Agile methodologies in the development of applications for mobile devices. present state».

- [19] M. E. P. J. Gabriel, «Metodologías de desarrollo de software».
- [20] S. Alsaqqa, S. Sawalha, y H. Abdel-Nabi, «Agile Software Development: Methodologies and Trends», *Int. J. Interact. Mob. Technol.*, vol. 14, n.º 11, pp. 246-270, jul. 2020, Accedido: ago. 19, 2021. [En línea]. Disponible en: <https://online-journals.org/index.php/i-jim/article/view/13269>.
- [21] A. O. Duarte y M. Rojas, «Las Metodologías de Desarrollo Ágil como una Oportunidad para la Ingeniería del Software Educativo», *Rev. Av. en Sist. e Informática*, vol. 5, n.º 2, 2008.
- [22] Evaluando Software, «Qué es la geolocalización y cómo funciona», 2021. <https://www.evaluandosoftware.com/la-geolocalizacion-funciona/>.
- [23] A. Leszczynski, «Geolocation», en *International Encyclopedia of Human Geography*, Elsevier, 2020, pp. 101-106.
- [24] A. Pozo-Ruz, A. Ribeiro, M. C. García-Alegre, L. García, D. Guinea, y F. Sandoval, «SISTEMA DE POSICIONAMIENTO GLOBAL (GPS): DESCRIPCIÓN, ANÁLISIS DE ERRORES, APLICACIONES Y FUTURO».
- [25] Fulcrum, «Mapbox vs. Google Maps: Choose Your Perfect Map API». <https://fulcrum.rocks/blog/mapbox-vs-google-maps/> (accedido sep. 01, 2021).
- [26] Mapsvg, «Mapbox Vs Leaflet: What Makes These Two So Different | MapSVG Blog». <https://mapsvg.com/blog/mapbox-vs-leaflet> (accedido sep. 01, 2021).
- [27] I. Warren, A. Meads, S. Srirama, T. Weerasinghe, y C. Paniagua, «Push notification mechanisms for pervasive smartphone applications», *IEEE Pervasive Comput.*, vol. 13, n.º 2, pp. 61-71, abr. 2014, doi: 10.1109/MPRV.2014.34.
- [28] Capterra, «Best Push Notifications Software 2021 | Reviews of the Most Popular Tools & Systems». <https://www.capterra.com/push-notifications-software/> (accedido ago. 27, 2021).
- [29] P. O. E. Garcés, «La influencia del tráfico de drogas en la inseguridad ciudadana

- en el sector de La Mariscal en la ciudad de Quito periodo 2010 - 2014», 2015.
- [30] J. S. Moreira, «Incidencia de la seguridad comunitaria y el capital social en barrios urbanos de San José, Costa Rica», *URVIO - Rev. Latinoam. Estud. Secur.*, n.º 19, p. 90, 2016, doi: 10.17141/urvio.19.2016.2409.
- [31] J. Fernández González, «Introducción a las metodologías ágiles Otras formas de analizar y desarrollar».
- [32] H. K. Flora y S. V Chande, «A Systematic Study on Agile Software Development Methodologies and Practices», Accedido: ago. 19, 2021. [En línea]. Disponible en: www.ijcsit.com.
- [33] Develou, «Android Push Notifications Con Firebase Cloud Messaging». <https://www.develou.com/android-push-notifications-firebase-cloud-messaging/> (accedido ago. 12, 2021).
- [34] Capterra, «OneSignal Precios, Alternativas & Más 2021 - Capterra». <https://www.capterra.com/p/163452/OneSignal/> (accedido ago. 12, 2021).
- [35] WonderPush, «Pricing - Push notifications - WonderPush». <https://www.wonderpush.com/pricing/> (accedido ago. 27, 2021).
- [36] OneSignal, «Comparación de OneSignal y Firebase Cloud Messaging - OneSignal». <https://onesignal.com/onesignal-vs-firebase-cloud-messaging> (accedido ago. 26, 2021).
- [37] Ionic Framework, «Open-Source UI Toolkit to Create Your Own Mobile or Desktop Apps». <https://ionicframework.com/docs>.
- [38] AltexSoft, «Pros and Cons of Ionic Mobile App Development | AltexSoft», 2020. <https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-ionic-mobile-development/> (accedido ago. 30, 2021).
- [39] A. E. Fentaw y Cross, «Cross platform mobile application development: a comparison study of React Native Vs Flutter», 2020.
- [40] AltexSoft, «ReactJS and React Native: Overview & Pros and Cons | AltexSoft». <https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-reactjs-and-react-native/> (accedido ago. 31, 2021).

- [41] Statistics and Data, «Most Popular Backend Frameworks – 2012/2021 - New Update - Statistics and Data». <https://statisticsanddata.org/data/most-popular-backend-frameworks-2012-2021/> (accedido ago. 31, 2021).
- [42] R. Čerešňák y M. Kvet, «Comparison of query performance in relational a non-relation databases», en *Transportation Research Procedia*, 2019, vol. 40, pp. 170-177, doi: 10.1016/j.trpro.2019.07.027.
- [43] D. Kunda y H. Phiri, «A Comparative Study of NoSQL and Relational Database».
- [44] Stack Overflow, «Format XML and JSON». <https://stackoverflow.com/questions/55930613/how-to-convert-a-string-that-is-json-or-xml-like-to-a-useable-python-object> (accedido nov. 30, 2021).
- [45] M. A. Alvarez, «Integrantes del servicio de notificaciones push desde la web», 2017. <https://desarrolloweb.com/articulos/notificaciones-push-web-firebase.html>.
- [46] Blog.expo.dev, «Push Notifications and In-App Messaging with OneSignal and Expo | by George Deglin | Exposition». <https://blog.expo.dev/push-notifications-and-in-app-messaging-with-onesignal-and-expo-3abf77b7f288> (accedido ago. 12, 2021).
- [47] Flutter, «Flutter | Flutter architectural overview». <https://flutter.dev/docs/resources/architectural-overview> (accedido ago. 31, 2021).
- [48] Stack Overflow, «Stack Overflow Developer Survey 2021». <https://insights.stackoverflow.com/survey/2021> (accedido ago. 31, 2021).
- [49] «All Applications | OneSignal». <https://app.onesignal.com/> (accedido sep. 11, 2021).
- [50] «Firebase console». <https://console.firebase.google.com/> (accedido sep. 11, 2021).

ANEXOS

3.3. Anexo 1

- Reuniones de socialización y capacitación con los ciudadanos de los barrios de la parroquia Sta. Rosa



Figura 3.3.1 Exposición del proyecto a los moradores del barrio las Lajas

Elaborado por: Christian Chasi



Figura 3.3.4 Socialización de la aplicación a los moradores del barrio las Lajas

Elaborado por: Christian Chasi



Figura 3.3.3 Aplicación de encuestas a los moradores del barrio Las Lajas

Elaborado por: Christian Chasi



Figura 3.3.2 Socialización de la aplicación con los moradores de La Ciudadela 12 de Noviembre

Elaborado por: Christian Chasi



Figura 3.3.6 Reuniones a la que asiste la policía

Elaborado por: Christian Chasi



Figura 3.3.5 Exposición del aplicativo (con el Ing. Balarezo, Tutor del Proyecto) en la policía

Elaborado por: Christian Chasi

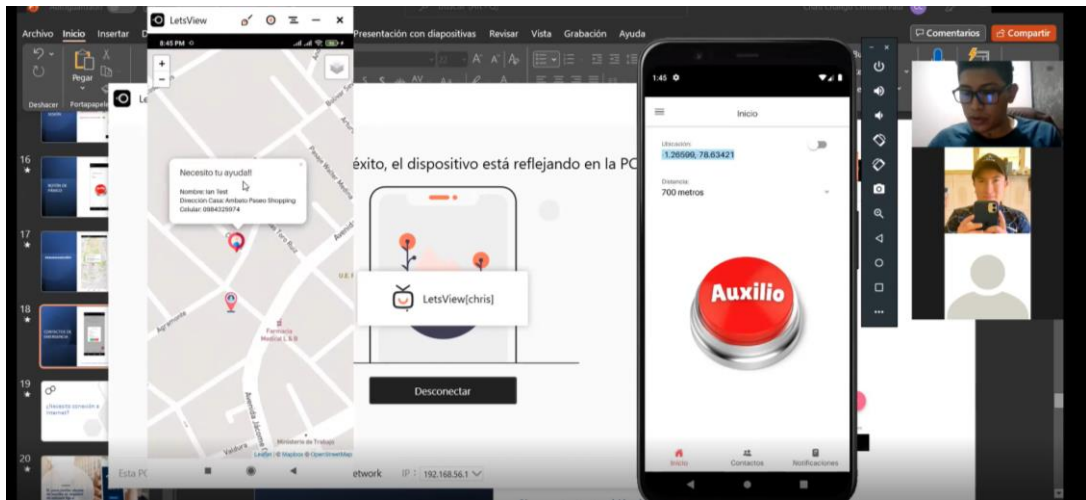


Figura 3.3.7 Reunión Virtual con el barrio Guayaquil – Parroquia Sta. Rosa

Elaborado por: Christian Chasi

3.4. Anexo 2

MANUAL DE USUARIO

A continuación, se muestra el respectivo manual de usuario que describe los pasos para utilizar los 2 aplicativos móviles. Estos manuales fueron compartidos al GAD y a los presidentes de los barrios de la parroquia Sta. Rosa.





Manual Aplicación Barrio Seguro



1. Buscar la aplicación como **"Barrio Seguro"** en la Google Play.

2. Descargamos e instalamos.

3. Abrimos la app y continuamos con los siguientes pasos:



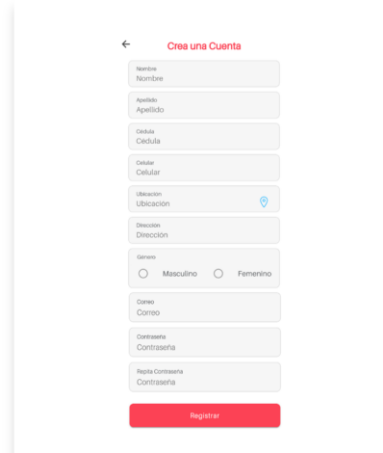
INICIO DE SESIÓN

- Al abrir por primera vez la aplicación se mostrará la siguiente interfaz.
- Nos pedirá que ingresemos el correo y contraseña
- Si no tenemos esos datos procedemos a crear una cuenta mediante la opción **"Registrarse"**.



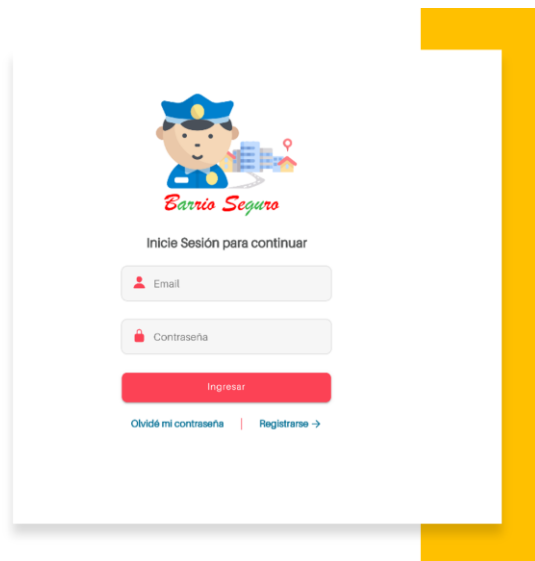
REGISTRO DE USUARIO

- ❑ En cada campo se debe ingresar los datos que solicita.
- ❑ En el campo de **Ubicación** se debe presionar el ícono de ubicación color azul para llenar ese campo automáticamente.
- ❑ Una vez que todos los campos están llenos se presiona **Registrar** y si aparece el mensaje de registro exitoso está listo para autenticarse.



PROCESO DE AUTENTICACIÓN

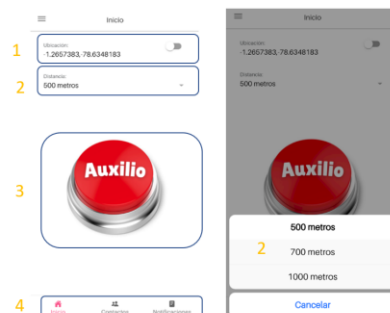
- ❑ Con una cuenta ya creada se procede a ingresar el correo y contraseña para iniciar sesión



MÓDULO DE ALERTAS DE AUXILIO

Una vez autenticado, como muestran las figuras tendremos las siguientes opciones:

1. Latitud, longitud del ciudadano y botón para obtener su ubicación actual.
2. Opciones de rangos de distancia.
3. Botón de pánico.
4. Accesos rápidos a otros módulos.



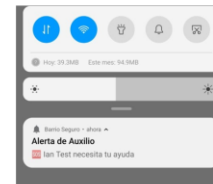
MÓDULO DE ALERTAS DE AUXILIO

- ❑ En la **Figura 1** se muestra el respectivo mensaje que indica el numero de dispositivos al que fue enviado la alerta de auxilio.
- ❑ En la **Figura 2** se muestra la notificación push o alerta de auxilio cuando llega al dispositivo del vecino.

FIGURA 1



FIGURA 2



GEOLOCALIZACIÓN

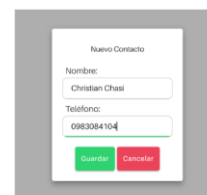


- ❑ Al abrir una notificación o alerta de auxilio se mostrarán los mapas con los que se puede visualizar la ubicación del ciudadano que solicita ayuda.
- ❑ También existe la opción para cambiar el tipo de mapa de simple a avanzado.

CONTACTOS DE EMERGENCIA

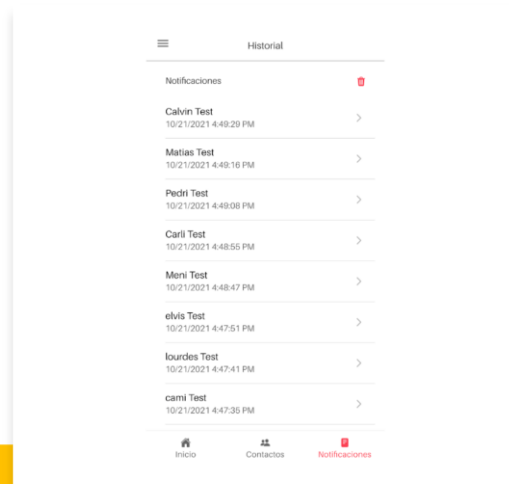
Para agregar un contactos debemos pulsar en el botón con el símbolo de + ubicado en la parte inferior derecha.

- ❑ En la sección de contactos de emergencia podemos agregar hasta 5 contactos que recibirán la alerta de auxilio sin estar en el rango de distancia.
- ❑ Para agregar un contacto de emergencia, este también debe tener instalada la aplicación y tener una cuenta.



Historial de alertas de auxilio.

- ❑ Cuando llega una notificación push o alerta de auxilio, si el usuario la abre, esta se almacena automáticamente en el dispositivo manteniendo un registro.
- ❑ Cada registro guardado tiene la información del ciudadano y también puede ser visto.
- ❑ Para eliminar el historial de alertas se debe presionar en el icono con el basurero.

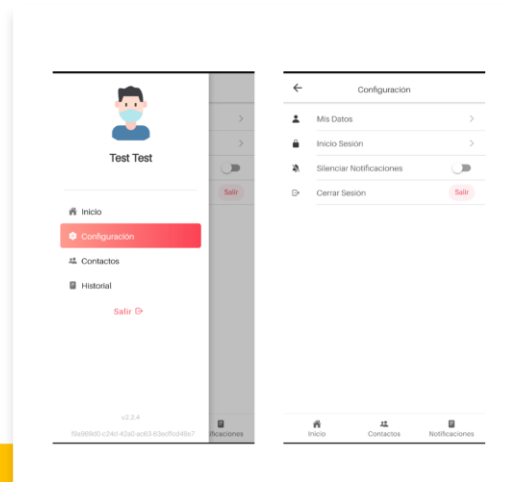


Configuración y Perfil de usuario

Para acceder a esta opción se desliza el dedo hacia la derecha o se presiona el icono de menú.

Existen las siguientes opciones:

1. **MIS DATOS:** Modificar nuestra información personal.
2. **INICIO DE SESIÓN:** Modificar el correo o la contraseña.
3. **SILENCIAR NOTIFICACIONES:** Desactivar las alertas de auxilio.
4. **CERRAR SESIÓN:** Salir de nuestra cuenta y dejar de recibir alertas.



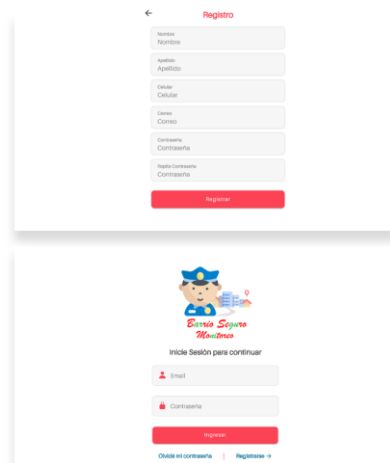


Manual de usuario aplicación Monitoreo de Alertas



INICIO DE SESIÓN Y REGISTRO DE USUARIO

- Al igual que la aplicación de **Barrio Seguro** la aplicación de Monitoreo de Alertas sigue los mismos pasos a excepción que tiene menos campos en su formulario de **Registro**.

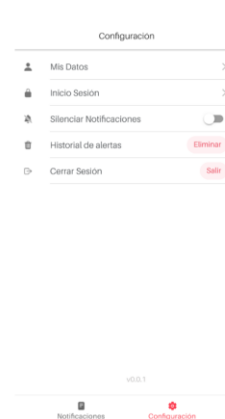


Configuración

Para acceder a esta opción se dirige al menú que esta en la parte inferior.

Existen las siguientes opciones:

1. **MIS DATOS:** Modificar nuestra información personal.
2. **INICIO DE SESIÓN:** Modificar el correo o la contraseña.
3. **SILENCIAR NOTIFICACIONES:** Desactivar las alertas de auxilio.
4. **HISTORIAL DE ALERTAS:** Eliminar todo el historial de alertas
5. **CERRAR SESIÓN:** Salir de nuestra cuenta y dejar de recibir alertas.

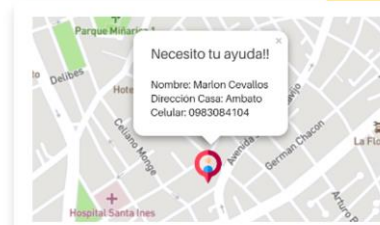


Monitoreo de Alertas

- ❑ Cada alerta de auxilio que llega al dispositivo se almacena cuando es abierta.
- ❑ Al pulsar en la opción **VER** se mostrará el mapa con la información y ubicación de la persona.

Alertas de Auxilio

| Fecha | Ciudadano | Hora | Acción |
|------------|-----------------|------------|--------|
| 10/21/2021 | Marlon Cevallos | 5:40:16 PM | Ver |
| 10/21/2021 | Josue Chasi | 5:40:06 PM | Ver |
| 10/21/2021 | Dario Chasi | 5:39:49 PM | Ver |



3.5. Anexo 3

A continuación, se muestra las preguntas guía para la mini entrevista aplicada a la directiva de los diferentes barrios y GAD de la parroquia Santa Rosa de la Ciudad de Ambato.

CUESTIONARIO DE PREGUNTAS GUIA PARA LA ENTREVISTA A LAS DIRECTIVAS DE LOS BARRIOS Y GAD DE STA. ROSA

- ¿Podría mencionar a los entes o instituciones que están a cargo de la seguridad de los diferentes barrios de la parroquia?
- ¿Cuál es la función de los entes mencionados anteriormente?
- ¿Cuáles son algunas políticas o actividades de control y seguridad barrial?
- ¿Estas instituciones mantienen comunicación constante con los diferentes GADs municipales?
- ¿Son suficientes las políticas implementadas por los diferentes organismos encargados de la seguridad ciudadana?
- ¿Los entes encargados del control y seguridad barrial pueden financiar algún proyecto que aporte a la seguridad de los barrios?

3.6. Anexo 4

Cuestionario de preguntas aplicadas a los ciudadanos de los diferentes barrios de la parroquia Sta. Rosa de la ciudad de Ambato.

**ENCUESTA PARA EVALUAR LA INSEGURIDAD CIUDADANA EN LA
PARROQUIA SANTA ROSA DE LA CIUDAD DE AMBATO**

I. CUESTIONARIO

1. ¿Que tan seguro considera que es su barrio?

- Poco Seguro
- Medianamente Seguro
- Muy Seguro

2. ¿Ha colocado sistemas de seguridad en su hogar?

Si No

3. Su barrio cuenta con sistemas de seguridad o alarmas comunitarias?

Si No

4. ¿Su barrio es organizado y colaborativo para brindar ayuda en situaciones de delincuencia?

Si No

5. ¿Está de acuerdo en que los barrios se protejan ante el aumento de la delincuencia?

Si No

6. Que tan frecuentemente al mes se suscitan delitos en su barrio o parroquia?

- Nunca

- Rara vez
- Casi siempre
- Siempre

7. Que delitos son más frecuentes en su barrio o parroquia (Escoga 1 o más respuestas)

- Robos a Viviendas
- Robo a personas
- Robos a vehículos
- Asesinatos
- Otros

8. ¿Es importante tener un botón de pánico entre la comunidad del barrio?

Si No

9. ¿Cree que la tecnología (teléfonos inteligentes, cámaras de seguridad, sistemas, aplicaciones, entre otros) pueden ayudar a combatir la inseguridad ciudadana?

Si No

10. Usted estaría dispuesto a colaborar para mejorar la seguridad de su barrio?

Si No