

# Red\_Bog

**Julian David Rodriguez Fernandez, Diego David Valbuena Muñoz, Esteban Durán Jiménez, Nicolas Fernando Davila Peñuela, Edwar Mauricio Cortes Bernal, Valentina Ladeus Machado**

## No. de equipo de trabajo: 1

### I. INTRODUCCIÓN

**R**edBog es una aplicación para Android que busca dar información sobre los actos delictivos que suceden en la ciudad de Bogotá D.C. La descripción de la implementación de estructuras de datos, los requerimientos funcionales del software, la etapa preliminar para la interfaz de usuario, las pruebas del prototipo y análisis comparativo, entre otros, son algunas de las características que serán abordadas a mayor profundidad en el presente documento en función de otorgar al usuario bogotano una herramienta para combatir la delincuencia común.

### II. DESCRIPCIÓN DEL PROBLEMA A RESOLVER

La ciudad de Bogotá presenta índices de criminalidad muy elevados. En lo corrido del año 2022 solo en la localidad de Kennedy se han robado cerca de 4521 celulares, 700 motocicletas y 1072 bicicletas. Se han cometido 9425 atracos, se han cometido 563 ataques sexuales, entre otros delitos. La percepción de seguridad en la ciudad es nula, los recursos para acceder a información detallada no son inmediatos y la desconfianza en las autoridades cada día es mayor. Sin embargo, la información clara de los eventos delictivos y actuar ciudadano han demostrado combatir de manera efectiva este flagelo social que se vive actualmente. Es por todo esto que el objetivo de RedBog es otorgar una herramienta a los usuarios para informar sobre actos delictivos y acceder a reportes previos.

### III. USUARIOS DEL PRODUCTO DE SOFTWARE

El público objetivo de RedBog son todas aquellas personas que vivan o visiten la ciudad de Bogotá y deseen registrar o informarse sobre actividad delictiva o violenta en lugares específicos de la ciudad.

### IV. REQUERIMIENTOS FUNCIONALES DEL SOFTWARE

#### Datos a ser ingresados en el sistema:

- **Nombre del usuario:** Cadena de caracteres para identificar al creador de diferentes reportes.

- **Número de teléfono :** Número tipo entero con el que se identificarán los reportes creados.
- **Contraseña:** Cadena de caracteres que se valida al inicio del programa para dar entrada al sistema a los usuarios.
- **Localidad:** Cadena de caracteres que permite identificar el lugar donde sucedieron cada uno de los hechos ilícitos
- **Tipo de acto Ilícito :** Cadena de caracteres que permite identificar el acto ilícito
- **Comentario:** Cadena de caracteres que sirve para que los usuarios puedan detallar el acto ilícito del que fueron víctimas.

#### Operaciones a ser realizadas en cada pantalla que se presenta:

- **Login:** En la primera pantalla de la app los usuarios pueden realizar la operación de registro. Esta operación requiere que ingresen su número de teléfono móvil y una contraseña previamente registrada en el sistema para ingresar a la app, en esta pantalla también se encuentra un botón que permite ingresar a la pantalla de registro.
- **Registro:** En la pantalla de registro los usuarios pueden guardar su información en el sistema, para esto deben ingresar su nombre, teléfono, correo electrónico y contraseña (esta última se debe confirmar para garantizar que el usuario no haya cometido errores de digitación), luego al dar click sobre el botón “registrarme” ya pueden entrar a la app con el Login.
- **Reportes generales:** En la pantalla de reportes generales se pueden revisar todos los reportes generados por usuarios de la app, estos están organizados de los más recientes a los menos recientes.
- **Mis reportes:** En la pantalla de mis reportes se pueden visualizar todos los reportes generados por el usuario en cuestión, estos se organizan del más antiguo al más reciente, adicionalmente es posible desde esta pantalla eliminar reportes generados dando click encima del icono de la basura e ingresar a la pantalla de edición de reportes dando click en texto “editar comentario”.
- **Crear :** En esta pantalla se pueden crear reportes, para esto se debe ingresar la localidad donde sucedió el acto ilícito, el tipo de acto ilícito y un breve comentario de lo sucedido, luego al dar click en crear reporte, este se podrá visualizar tanto en mis reportes como en reportes generales.

- **Edición de reportes :** En esta pantalla el usuario puede editar los reportes que haya realizado, el usuario ingresa a esta pantalla al dar clic en el texto de editar reportes que aparece en la parte inferior de cada reporte en la sección de mis reportes. Para editar un reporte, el usuario debe ingresar la información que se desee actualizar en los mismos campos que se tienen cuando se quiere crear un reporte.

### Los reportes del sistema y otras salidas:

En el sistema las salidas principales que se tienen son mensajes que permiten confirmar si la función en mención se realizó correctamente o no. Estos mensajes se tienen para las funciones de : Registro de usuario, Login, Creación de reporte, Eliminación de reporte y Edición de reporte.

### Usuarios que pueden ingresar datos al sistema:

Los usuarios que pueden ingresar datos al sistema son los que tengan una cuenta registrada en el sistema, esto para que se pueda identificar quién hizo cada uno de los reportes y también para que estos reportes puedan ser editados por los usuarios que así lo requieran.

### Funcionalidades de la app:

#### 1. Registrar usuario:

*Descripción:* Esta funcionalidad sirve para que los usuarios puedan registrarse en la app, para esto deben ingresar: nombre, correo electrónico, número de teléfono móvil y una contraseña.

*Acciones iniciadas y comportamiento esperado:* Las acciones iniciadoras de esta actividad son la de dar clic sobre el botón de registrar en la pantalla de login y la de ingresar los datos de localidad, tipo de acto ilícito y breve comentario de lo sucedido. Se espera en esta funcionalidad que se guarden los datos ingresados por el usuario en una base de datos para que de esta manera puedan ser utilizados en el resto de funciones de la app.

*Comportamiento anormal:* Cuando un usuario previamente registrado ingresa su información, el sistema arrojará un error indicando que el usuario ya existe.

#### Requerimientos funcionales:

- **Ingreso nombre:** Debe mostrar una viñeta donde se pueda ingresar el nombre del usuario, para esto en el código se utilizó un “Textinput layout” (un objeto creado específicamente por Android Studio el entorno donde se programó) este toma los datos ingresados por el usuario para que posteriormente se puedan almacenar permanentemente en la base de datos.

- **Ingresar correo electrónico :** Se debe mostrar una viñeta para ingresar el correo electrónico, está funciona igual que la viñeta de ingresar nombre.
- **Ingresar número de teléfono móvil :** Se debe mostrar una viñeta para ingresar el número de teléfono móvil. Esta funciona igual que la viñeta de ingresar nombre. Sin embargo, si el usuario intenta ingresar cualquier entrada diferente de un número entero, se genera un error pidiéndole que rectifique si su entrada es un número entero.
- **Ingresar contraseña:** Se deben mostrar dos viñetas, en la primera se añade la contraseña y en la segunda se confirma que sea la misma cadena de caracteres. Ambas viñetas funcionan de la misma manera que lo hace la viñeta de “*ingreso, nombre*”
- **Guardar datos:** Permite tomar todos los datos ingresados y guardarlos en la base de datos de usuarios.

#### 2. Login usuarios:

*Descripción:* Esta funcionalidad permite a los usuarios ingresar a la app si se encuentran registrados en la base de datos de la misma.

*Acciones iniciadas y comportamiento esperado:* Las acciones iniciadas son el ingreso de los datos requeridos para validar el usuario, estos son el celular y la contraseña del usuario, también el dar clic en el botón de login, se espera que se verifiquen los datos ingresados con los anteriormente almacenados y se determine si el usuario puede ingresar.

#### Requerimientos funcionales:

- **Ingreso número de celular:** debe mostrar una viñeta donde se pueda ingresar el número de celular del usuario, si la entrada es diferente de un número entero, sale un
- **Ingrese contraseña:** debe mostrar una viñeta donde se pueda ingresar el celular del usuario.
- **Verificación de usuario:** debe tomar la información ingresada por el usuario en las viñetas y buscar en la base de datos si esta información si concuerda. Si esto es así, permite al usuario ingresar a la app. En caso contrario aparece en la pantalla un mensaje informando al usuario que la información ingresada está errada.

#### 3. Crear reporte:

*Descripción:* Esta característica del programa permite a los usuarios generar y publicar los diferentes actos

ilícitos a los que se han visto expuestos, detallando así la localidad y los hechos sucedidos.

*Acciones iniciadas y comportamiento esperado:* Esta funcionalidad inicia cuando, en la pantalla de inicio, el usuario da clic al botón de “crear reporte”. Consecuentemente, se espera que aparezca un formulario en el cual se piden los detalles del acto ilícito a reportar. Al llenar los campos requeridos para generar la publicación, el usuario deberá confirmar y compartir su reporte al dar clic sobre el botón “Crear reporte”. Una vez completada esta acción, se espera que el reporte generado ya pueda ser visualizado tanto por el usuario que lo creó como por otros usuarios en la sección de reportes generales.

*Requerimientos funcionales:*

- *Ingreso tipo de acto ilícito:* Dropdown menú que debe mostrar al usuario las categorías de actos ilícitos que se pueden presentar y que puede reportar.
- *Ingreso tipo de localidad:* Dropdown menú que debe mostrar las localidades de Bogotá con el fin de que el usuario seleccione el lugar en donde desea generar el reporte.
- *Sección de comentarios:* Es un cuadro de texto en donde el usuario podrá redactar los hechos sucedidos con detalle e informar a los demás usuarios de la plataforma que las situaciones que se presentan en las diferentes localidades.

#### 4. Borrar reporte:

*Descripción:* Esta funcionalidad permite a los usuarios borrar los registros creados por ellos.

*Acciones iniciadoras y comportamiento esperado:* Se tiene como acción iniciadora el dar clic sobre el icono de la basura ubicado en cada reporte dentro de la pantalla de “Mis reportes”, posteriormente se despliega una ventana emergente que permite al usuario confirmar si desea realmente eliminar el reporte, se espera que siempre se despliega este menú para evitar que se eliminen reportes por error y también luego de que confirme la eliminación inmediatamente aparezca la lista de reportes actualizada.

*Requerimientos funcionales:*

- *Icono inicial de eliminación:* Se debe contratar con un icono inicial de eliminación que al ser seleccionado cree una ventana emergente para que el usuario pueda confirmar la eliminación.
- *Ventana emergente de eliminación:* Se debe tener una ventana emergente que permita al usuario confirmar dando clic en un botón si desea eliminar el reporte en cuestión.

- *Botón eliminador :* debe existir un botón eliminador que permita a los usuarios finalmente eliminar el reporte y adicionalmente actualice la lista automáticamente y la despliegue en la pantalla de “Mis reportes”, se espera que se elimine el mismo reporte que se seleccionó y que tanto en la pantalla de “Mis reportes” como en la de la de “Reportes generales” este reporte quede eliminado de manera definitiva.

#### 5. Editar reporte:

*Descripción:* Permite al usuario editar en los reportes que ellos crearon cualquiera de los datos ingresados e inmediatamente actualiza las pantallas.

*Acciones iniciadas y comportamiento esperado:* Esta funcionalidad inicia cuando se selecciona la opción de editar comentario que se puede ver en la parte inferior de cada reporte de la pantalla de mis reportes, luego dirige al usuario a la pantalla de editar reporte y le permite actualizar los datos de, localidad, tipo de acto delictivo y comentario sobre la experiencia vivida.

*Requerimientos funcionales:*

- *Ingreso tipo de acto ilícito:* Dropdown menú que debe mostrar al usuario las categorías de actos ilícitos que se pueden presentar y que puede reportar.
- *Ingreso tipo de localidad:* Dropdown menú que debe mostrar las localidades de Bogotá con el fin de que el usuario seleccione el nuevo dato a actualizar.
- *Sección de comentarios:* Es un cuadro de texto en donde el usuario podrá editar su comentario sobre los hechos sucedidos con detalle e informar a los demás usuarios de la plataforma sobre las situaciones que se presentan en las diferentes localidades.

### V. DESCRIPCIÓN DE LA INTERFAZ DE USUARIO PRELIMINAR

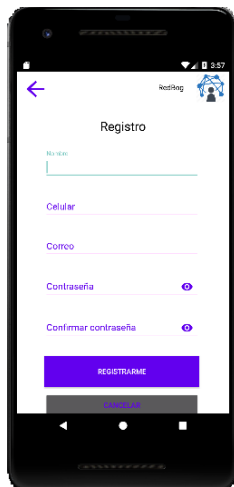
#### Pantalla inició:

Se realizó una interfaz sencilla en la cual el usuario al iniciar la aplicación solo encontrará dos cuadros de texto para poder ingresar su usuario y contraseña que debió registrar previamente, además encontrará dos botones, uno para ingresar a la app y el otro para realizar el proceso de registro



### Pantalla de Registro:

El usuario para poder registrarse necesitará llenar los cuatro cuadros de texto que corresponden a su nombre, celular, correo y contraseña; una vez hecho esto también encontrará el botón de registrar.



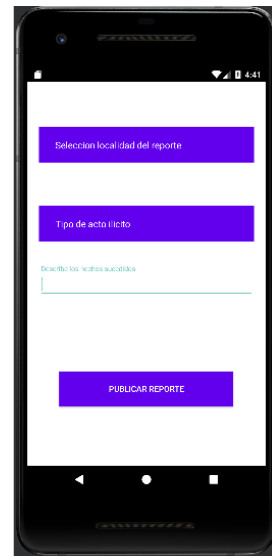
### Pantalla Principal:

Al iniciar en la app, el usuario podrá encontrar tres opciones, las cuales serían ver reportes, ver reportes creados por el usuario y crear usuarios;



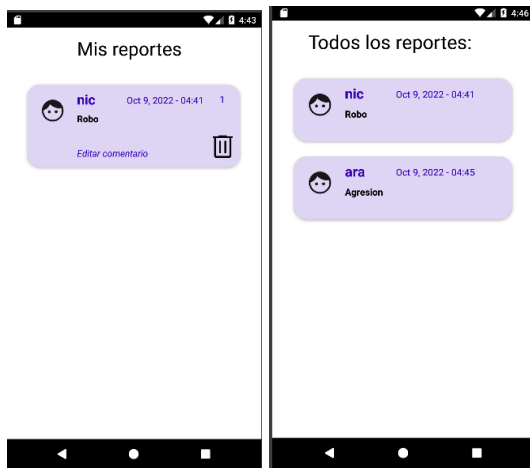
### Pantalla Crear reporte:

Aquí el usuario podrá elegir el tipo de delito entre las tres opciones seleccionadas, asalto, agresión y robo; también tendrá que elegir la localidad donde presenció el delito; al final hay un cuadro de texto para que pueda dejar un comentario



### Pantalla de Reportes:

La pantalla mis reportes y reportes son muy similares en la manera en que el usuario podrá ver los cada uno de los reportes registrados, solo que en la de mis reportes estará implementado las funciones faltantes del CRUD es decir, podemos leer, borrar o actualizar cualquier reporte, siempre y cuando sea creado por el usuario de la cuenta.



## VI. ENTORNOS DE DESARROLLO Y DE OPERACIÓN

El entorno de desarrollo que se utilizó fue Android Studio eligiendo como lenguaje de programación Java para el manejo de datos y la lógica del aplicativo y el sub lenguaje XML para la interfaz de usuario. Se decidió que el aplicativo final funcionará en el sistema operativo Android en dispositivos móviles.

Para el almacenamiento de datos se usó SQLite con el fin de manejar la base de datos tanto de los usuarios como de los reportes y así poder mantener la relación entre estos.

## VII. PROTOTIPO DE SOFTWARE INICIAL

- <https://github.com/JulianDavidRF/RedBog>

## VIII. DISEÑO, IMPLEMENTACIÓN Y APLICACIÓN DE LAS ESTRUCTURAS DE DATOS

En el proyecto realizado se implementaron los tres tipos básicos de estructuras de datos vistos, estos son, arreglos, colas, pilas y listas enlazadas, a continuación se da una explicación clara de cómo se implementan cada una de estas estructuras:

1. **Pilas:** Esta estructura de datos se utilizó en el proyecto para almacenar temporalmente los diferentes reportes generados por los usuarios y generar un ordenamiento básico de los mismos, ya que como esta estructura cumple con un comportamiento LIFO, al ingresar los diferentes reportes se tenían que los más recientes quedaban más cerca del TOP de la pila y los menos recientes más alejados, por lo tanto, al realizar la función de push para tomar cada reporte y visualizarlo en la app, estos quedaban organizados del más reciente al más antiguo.
2. **Colas:** Esta estructura de datos se utilizó en el proyecto para almacenar temporalmente los datos de los reportes específicos de cada usuario, ya que la funcionalidad de Búsqueda parcial de los datos tiene

como propósito principal permitir a los usuarios visualizar sus reportes, borrarlos y editarlos, se aprovecha que la estructura tiene un comportamiento FIFO, esto es que el último elemento en ingresar es el primero que salir, de esta manera cuando se realizaba la función de dequeue para cada elemento y estos se visualizaban en la aplicación, se tenía que estos quedaban organizados de más antigua a más reciente, mostrando así un historial de los reportes realizados por el usuario.

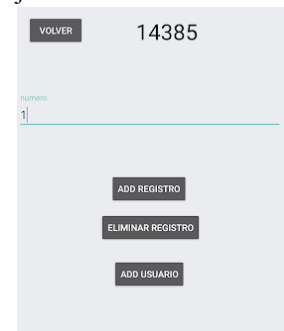
3. **Listas enlazadas:** Esta estructura de datos se utilizó para tomar los reportes previamente ordenados en las colas o pilas respectivamente y mediante una clase específica de Android Studio (*el IDE específico para el desarrollo de apps móviles*) se presentan ordenadamente en cuadros independientes para que el usuario pudiera entender la información y posteriormente realizar las modificaciones o eliminaciones pertinentes.

## IX. PRUEBAS DEL PROTOTIPO Y ANÁLISIS COMPARATIVO.

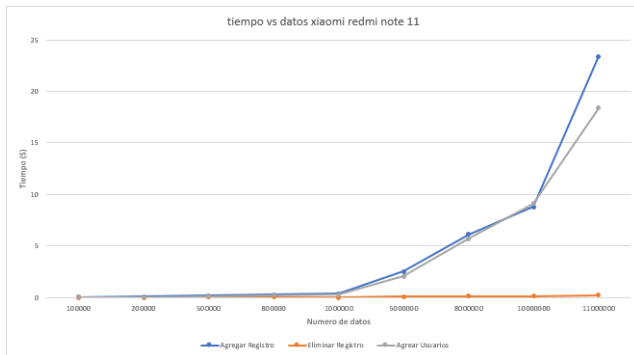
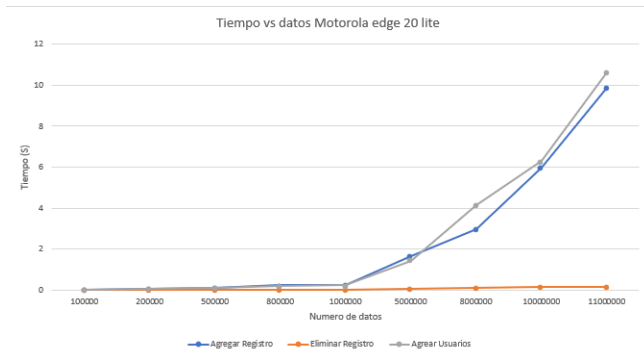
Para determinar que tan optimizado está el funcionamiento de los algoritmos usados en el prototipo, se escogieron 3 funcionalidades que representan el mayor costo computacional, siendo estas la acción de crear un usuario, la opción de crear un reporte y la opción de eliminar un reporte. Adicional para organizar los datos en estas funcionalidades se hizo uso de las estructuras de cola para los usuarios y listas para los reportes.

La manera como se testean estas funcionalidades es mediante la toma de tiempo de ejecución para cada una con diferentes tamaños de datos que van desde el 1 hasta 11 millones, que fue lo máximo permitido por el hardware en los que se ejecutó el prototipo (celulares).

Para la prueba se creó un botón adicional en la aplicación que lleva a una nueva pantalla donde se localizan una casilla donde se introduce el número de datos a testear y tres botones que simulaban la ejecución de cada una de las funciones con unos casos de prueba predispuestos, de manera que arroje en la parte superior el tiempo en nanosegundos que se demoró ejecutando los datos.



El resultado de los tiempos de ejecución para cada funcionalidad y número de datos se ven expresados en la siguiente gráfica, cabe aclarar que los tiempos están expresados en segundos para una mayor comodidad



Se realizó el mismo proceso para dos celulares de diferente referencia, para comparar cómo se desempeñaba la aplicación en ambos, de esto podemos decir que en el celular Note 11 hubo un mayor tiempo de ejecución de los datos, pero en ambos casos la aplicación fallaba al tratar de ejecutar más de 12 millones de datos.

Del comportamiento de la gráfica se puede apreciar que en un principio el programa se comporta de forma muy eficaz, tendiendo a ser de complejidad  $O(\log n)$  en los tres casos, pero al llegar al procesamiento de más de 1 millón de datos se incrementa exponencialmente ( $O(n^2)$ ) en los casos de agregar un reporte o usuario, y en el caso de la eliminación sigue siendo logarítmica.

#### X. INFORMACIÓN DE ACCESO AL VIDEO DEMOSTRATIVO DEL PROTOTIPO DE SOFTWARE

El video demostrativo lo puede consultar en:  
<https://youtu.be/2yC30dyXTRQ>

#### XI. ROLES Y ACTIVIDADES

INTEGRANTE	ROL(ES)	ACTIVIDADES REALIZADAS (Listado)
------------	---------	----------------------------------

<b>Julian David Rodriguez Fernandez,</b>	Experto Líder	-Desarrollo de registro y login. -Desarrollo e investigación, pruebas prototipo.
<b>Diego David Valbuena Muñoz,</b>	Secretario Animador Co Líder	-Colaboración / investigación prueba de prototipo. -Revisión y depuración código. -Motivación.
<b>Esteban Durán Jiménez</b>	Coordinador Co Líder	-Coordinación de reuniones de trabajo -Colaboración con CRUD mis registros - Edición y Comprobación del código - Apoyo con el Reporte técnico
<b>Nicolas Fernando Davila Peñuel</b>	Técnico	-Apoyo en el desarrollo del documento y presentación. -Apoyo en el desarrollo de estructuras
<b>Edward Mauricio Cortes Bernal</b>	Investigador Observador	-Revisión general de documentación y reporte técnico
<b>Valentina Ladeus Machado</b>	Animadora Líder Técnica	-Dirección del desarrollo del prototipo. -Apoyo en documentación

#### XII. DIFICULTADES Y LECCIONES APRENDIDAS

- La primera dificultad que experimentamos fue aprender sobre la programación en el entorno de Android Studio.
- En esta primera entrega, faltó organización y sincronía como grupo, pues no supimos organizarnos de una manera efectiva y que permitiese un trabajo más provechoso. Debido a estos problemas iniciales, aprendimos que como grupo debemos mejorar nuestra comunicación y la forma en que transmitimos nuestras ideas para que estas queden claras a los demás integrantes. Adicionalmente, aprendimos que debemos trabajar en nuestra organización y delegación de actividades de cada uno de los integrantes para que el trabajo sea equitativo.

#### XIII. REFERENCIAS BIBLIOGRÁFICAS

- [1] James T.Streib y otros: *Guide to data Structures, A Concise Introduction Using Java*, Springer, 2017.
- [2] Estadísticas de criminalidad en Bogotá D.C consultado el 8 de octubre del 2022 <https://analitica.scj.gov.co/analytics/saw.dll?Portal>