|  |  |  |  |
| --- | --- | --- | --- |
| **Alumno(s):** | Luis Enrique Guzmán Niño S14120070 | **Fecha:** | 23/03/2018 |

|  |
| --- |
| OBJETIVO |
| *Realizar una aplicación en Android que permita crear una lista de negra de contactos indeseables para bloquear la entrada de llamadas y mensajes.*  *La aplicación deberá tener las siguientes características y funcionalidades:*  *1) La aplicación deberá ser funcional en versiones de Android superiores a GingerBread (2.3.3).*  *2) Para poder agregar un contacto a la lista negra, éste debe ser tomado de los contactos del teléfono.*  *3) Al seleccionar un contacto, la aplicación deberá preguntar si el bloqueo será para llamadas, mensajes o ambos.*  *4) Al recibir una llamada o un mensaje de un contacto de la lista negra, estos no deberán quedar registrados en las aplicaciones nativas del dispositivo móvil (aplicación de llamadas y aplicación de mensajes).*  *5) La aplicación deberá tener un Log con las llamadas que han sido bloqueadas (Contacto, fecha y hora) y otro log con los mensajes recibidos bloqueados (Contacto, cuerpo del mensaje y fecha y hora)*  *6) Cuando se haya bloqueado una llamada o un mensaje deberá notificarse en la barra de notificaciones.* |

|  |
| --- |
| MARCO TEÓRICO |
| CALL\_PHONE –Llamar a números de teléfono directamente Servicios por los que tienes que pagar. Permite realizar llamadas sin la intervención del usuario. Nunca solicites este permiso en tus aplicaciones, muchos usuarios no instalarán tu aplicación. Si has de realizar una llamada, es mejor realizarla por medio de una intención. A diferencia de la llamada directa, no necesitas ningún permiso, dado que el usuario ha de pulsar el botón de llamada para que comience.  READ\_PHONE\_STATE –Consultar identidad estado del teléfono.. Muchas aplicaciones, como los juegos, piden este permiso para ponerse en pausa cuando recibes una llamada. Sin embargo, también permite el acceso al número de teléfono, IMEI (identificador de teléfono GSM), IMSI (identificador de tarjeta SIM) y al identificador único de 64 bits que Google asigna a cada terminal. Incluso si hay una llamada activa, podemos conocer el número al que se conecta la llamada.  READ\_CALL\_LOG y WRITE\_CALL\_LOG –Leer y modificar el registro de llamadas telefónicas. READ\_CONTACTS – Leer datos de contactos. Permite leer información sobre los contactos almacenados (nombres, correos electrónicos, números de teléfono). Algunas aplicaciones podrían utilizar esta información de forma no lícita  WRITE\_CONTACTS – Escribir datos de contactos. Permite modificar los contactos.  GET\_ACCOUNTS – Obtener Cuentas. Permiten acceder a la lista de cuentas en el Servicio de Cuentas.  Calendario:  READ\_CALENDAR – Leer datos de contactos. Permite leer información del calendario del usuario.  WRITE\_CONTACTS – Escribir datos de contactos. Permite escribir en el calendario, pero no leerlo. |

|  |
| --- |
| DESARROLLO |
| *Comenzamos con el diseño de las interfaces a utilizar para el desarrollo de esta aplicación.*    *Vamos a trabajar con fragmentos asi que creamos 2, llamadas,y para ver la lista negra de contactos.*      *Cada Pestana tendrá su correspondiente listitem asociado, asi ue creamos 2 list ítem.*    *Dichos listitem llenaran nuestras pestanas con datos sobre las llamadas bloqueadas y la lista de contactos en la lista negra.*  *Asi que damos un diseño acorde a lo antes mencionado.*      *Y por ultimo crearemos una interfaz con un listview que será donde cargaremos la lista de contactos para anadir a la lista de bloqueo.*    *Para el desarrollo de esta aplicación utilizaremos la interfaz de Android telephony. Creamos otro directorio donde estará esa interfaz.*    *Esta interfaz contara con el siguiente código que nos permitirá terminar la llamada y silenciar el tono de llamada.*    *Ahora procedemos a la creación de la base de datos que usaremos para nuestra aplicación.*  *Creando tres tablas, una para contactos en lista negra, otra para llamadas bloqueadas otra para mensajes bloqueados,*      *Y creamos los daos donde estarán los métodos para realizar acciones con cada una de las tablas.*  *Con sus correspondientes métodos, eliminar, agregar y obtener todo*      *Creamos el método dao de Contactos.*          *Creamos la actividad principal donde crearemos un adaptador para andir las pentanas de cada una de los fragmentos que usaremos, y les ponemos el titulo de nuestra elección.*    *Ahora vamos a poner el código de cada uno de los ListView que tenemos en nuestras pestanas, la primera va a cargar los datos de nuestros contactos por lo que hacemos un objeto de tipo contacto haciendo referencia a nuestro pojo y lo llenaremos con los datos que obtengamos al hacer la consulta a nuestro Dao.*    *Hacemos lo mismo con los otros ListView, solo cambiando los datos que obtendremos, por ejemplo aquí, se obtienen usando el arrayadapter de blockllamads, para ver los datos de las llamadas bloqueadas, mismas que van en la segunda pestana*    *Y ahora codificaremos la parte donde se carga la lista de contactos al hacer click sobre el botón flotante de anadir.*  *Instanciamos los elementos, y creamos un arreglo de opciones que se mostrara al hacer click sobre cualquier elemento de la lista.*  *Creamos un método confirmar donde se confirmara la acción y se llamara al pojo del contacto seleccionado y se insertara con el dao correspondiente dentro de la base de datos, en la tabla de contactos de la lista negra.*    *Para obtener el contacto seleccionado en nuestro oncreate creamos un click listener para obtener el índice de cada elemento seleccionado.*    *Y creamos el método donde vamos a consultar los contactos del teléfono e insertarlos dentro de un adaptador para llenar el listview, usando el siguiente código.*    *Con esto ya podemos tener nuestra base de datos de contactos en lista negra, pero falta la parte mas importante que es bloquear la llamada de esos contactos, para lo cual procedemos a crear otra clase.*  *Desde esta clase leeremos el estado del teléfono, y cuando reciba una llamada buscaremos el numero desde el que proviene la llamada en los que se encuentran en la lista negra de nuestra base de datos.*  *Si encontramos una coincidencia desconectamos la llamada, y crearemos un nuevo registro para esta llamada bloqueada en nuestra base de datos, donde agregaremos los datos del numero telefónico, fecha y hora*    *Y creamos la notificación donde avisaremos que se ha bloqueado la llamada*      *Con este código, desconectamos la llamada, por medio del context enviado en el método mostrado previamente.*  *Y con estos métodos obtenemos la fecha y hora del dispositivo que insertaremos en la base de datos del registro del bloqueo de la llamada.*    *Finalmente para eliminar la llamada del registro creamos una clase de tipo servicio, donde pondremos un hilo*    *Dicho hilo se ejecutara aun cuando cerremos la aplicación, y lo que hara será verificar el registro de llamadas, y cuando vea que hay una llamada de un numero que esta en la base de datos de la lista negra, lo borrara de inmediato por lo cual no lo podremos ver en el historial de llamadas, solo dentro de la base de datos de la aplcacion.* |

|  |
| --- |
| CONCLUSIONES |
| En conclusión aprendí a bloquear llamadas en Android para usar la aplicacion y estar más seguro en llamadas entrantes para asi evitar que números desconocidos entren para también para evitar factores asi aprendí a manejar más el uso de componentes para llamadas números mensajes con notificaciones detener procesos como la llamada evitar para asi saber manejar una llamada que hace Android para poder interpretar una llamada.  Hay determinadas situaciones en la vida en las cuales podríamos sentirnos acosados por unas cuantas personas, situación que lastimosamente se sale de control cuando el afectado (nosotros mismos como usuarios finales) hemos decidido en un primer momento, detener a dichas personas con una respuesta directa nuestra.  Lastimosamente si el “acosador” no entiende nuestros mensajes de rechazo, entonces la mejor opción es tratar de implementar cualquiera de las medidas que hemos mencionado en el presente artículo para bloquear llamadas en Android; si todo esto no llega a funcionarte, lo mejor es que intentes avisar a las respectivas autoridades para que las mismas estén atentas a cualquier inconveniente que pudieses estar teniendo en esos días. Además de ello, también es recomendable cambiar de número telefónico, algo que es muy fácil de hacer ya que las operadoras en diferentes partes del mundo, tienen la capacidad para modificarlo o simplemente, para entregarte otro SIM completamente distinto. |

|  |
| --- |
| BIBLIOGRAFÍA |
| [*https://blog.previewtechs.com/blocking-call-and-sms-in-android-programmatically*](https://blog.previewtechs.com/blocking-call-and-sms-in-android-programmatically)  [*https://mobiforge.com/design-development/sms-messaging-android*](https://mobiforge.com/design-development/sms-messaging-android)  [*http://www.androidbegin.com/tutorial/automatic-call-blocking-using-android-telephony-tutorial/*](http://www.androidbegin.com/tutorial/automatic-call-blocking-using-android-telephony-tutorial/)  [*https://developer.android.com/reference/android/provider/BlockedNumberContract.html*](https://developer.android.com/reference/android/provider/BlockedNumberContract.html)  [*https://source.android.com/devices/tech/connect/block-numbers*](https://source.android.com/devices/tech/connect/block-numbers)  [*https://developer.android.com/reference/android/provider/Telephony.html*](https://developer.android.com/reference/android/provider/Telephony.html)  [*https://developer.android.com/reference/android/telephony/TelephonyManager.html*](https://developer.android.com/reference/android/telephony/TelephonyManager.html) |