|  |  |  |  |
| --- | --- | --- | --- |
| **Alumno(s):** | Johan Alejandro Ortega Díaz | **Fecha:** | 13/10/2019 |

|  |
| --- |
| OBJETIVO |
| 1) Agregar los campos faltantes en cada uno de los metodos del DAO (fecha\_nacimiento).  2) Generar el metodo delete() en DAO  3) Generar metodo para buscar contactos a través del campo nombre.  4) Generar una UI (interfaz de usuario) para probar todos los metodos del DAO (CRUD). |

|  |
| --- |
| MARCO TEÓRICO |
| Guardar datos utilizando SQLite  Guardar datos en una base de datos es ideal para datos repetidos o estructurados, como información de contacto. Esta página asume que está familiarizado con las bases de datos SQL en general y lo ayuda a comenzar con las bases de datos SQLite en Android. Las API que necesitará para usar una base de datos en Android están disponibles en el android.database.sqlitepaquete.  Crear una base de datos utilizando un ayudante de SQL  Una vez que haya definido el aspecto de su base de datos, debe implementar métodos que creen y mantengan la base de datos y las tablas. Aquí hay algunas declaraciones típicas que crean y eliminan una tabla:  private const val SQL \_ CREATE \_ ENTRIES = "CREATE TABLE $ {FeedEntry.TABLE\_NAME} (" + "$ {BaseColumns.\_ID} INTEGER PRIMARY KEY," + "$ {FeedEntry.COLLUMN\_NAME\_TITLE} TEXT," + "$ FeedIntry. COLUMN\_NAME\_SUBTITLE} TEXT) " private const val SQL \_ DELETE \_ ENTRIES = " DROP TABLE SI EXIS $ $ {FeedEntry.TABLE\_NAME} "  Al igual que los archivos que guarda en el almacenamiento interno del dispositivo , Android almacena su base de datos en la carpeta privada de su aplicación. Sus datos están seguros, ya que, de forma predeterminada, esta área no es accesible a otras aplicaciones o al usuario.  La SQLiteOpenHelperclase contiene un conjunto útil de API para administrar su base de datos. Cuando utiliza esta clase para obtener referencias a su base de datos, el sistema realiza las operaciones potencialmente prolongadas de creación y actualización de la base de datos solo cuando es necesario y no durante el inicio de la aplicación . Todo lo que necesitas hacer es llamar getWritableDatabase()o getReadableDatabase().  Conexión de base de datos persistente  Debido a que la base de datos está cerrada, getWritableDatabase() y getReadableDatabase()son costosos de llamar, debe dejar la conexión de la base de datos abierta todo el tiempo que sea posible para acceder a ella. Por lo general, es óptimo cerrar la base de datos en la onDestroy()actividad de llamada.  Leer información de una base de datos.  Para leer desde una base de datos, use el query()método, pasándole los criterios de selección y las columnas deseadas. El método combina elementos de insert() y update(), excepto que la lista de columnas define los datos que desea obtener (la "proyección"), en lugar de los datos que se deben insertar. Los resultados de la consulta se le devuelven en un Cursorobjeto. |

|  |
| --- |
| DESARROLLO |
| 1. Como primer paso vamos a crear nuestra interfaz principal en donde se encontraran todos los botones (acciones) de nuestro programa la cual es la siguiente:      1. Vamos a crear las siguientes 4 clases que serán las que usaremos para ahí meter todo nuestro código.      1. Vamos a empezar a trabajar con la clase de la Conexión la cual tiene el siguiente código:      1. Lo que se hace primero es crear los campos que tendrá nuestra tabla la cual se llama “COLUMNS\_USUARIOS” en donde tiene un id, nombre, teléfono, email y fecha de nacimiento. Después se crea el nombre de la tabla en donde se van a guardar.      1. Luego vamos a crear la tabla con nuestros campos anteriores y en donde estaremos definiendo que nuestro id sea autoincremental.      1. Ya solo se agrega un contexto y unos métodos onCreate( ) que se agregan por defecto      1. En nuestra clase usuario tendremos nuestros constructores y nuestros métodos getters y setters      1. Para el método add creamos un ContentValues el cual se usa para almacenar un conjunto de valores. Después en el put le pasamos la conexión y el valor de la columna donde esta ubicado el campo correspondiente.      1. Se crea también un método getAll donde tendremos una lista de usuarios y en él mientras estamos llenado la lista con los valores que tenemos creados.      1. Los siguientes dos métodos nos ayudaran al momento de buscar un registro donde lo estaremos buscando por el campo de nombre.      1. Por ultimo en esta clase tendremos el método que nos ayudara a borrar donde el campo que vamos a borrar es necesario borrar el registro por un nombre.      1. En nuestro método onCreate usamos el findViewById para cada uno de nuestros campos que se encuentran en nuestro layout y de igual forma incorporamos nuestro listview. También creamos un Click Listener en el List View, para cargar los datos del elemento seleccionado.      1. Metodo para el botón insertar      1. Metodo que llena los campos con los datos correspondientes a un usuario en particular      1. Metodo que se encarga de crear los views para cada registro en la base de datos para posteriorment agregarlos al list view      1. Método para el botón editar NOTA: Al momento de modificar en la aplicación y después darle click al botón manda al error de que no se pudo modificar pero si cerramos la aplicación y la volvemos a abrir ahí está la modificación correctamente realizada.      1. Acciones para el botón de buscar      1. Botón eliminar |

|  |
| --- |
| RESULTADO |
|  |

|  |
| --- |
| CONCLUSIONES |
| El uso de SQLite en android presenta muchas ventajas a la hora de desarrollar aplicaciones para esta plataforma, como muchos de nosotros estaremos familiarizados con el uso de MySql donde para hacer uso de nuestras bases de datos tenemos que desarrollar un Script que genere las tablas y los datos, pero adicional a esto es necesario montar un servidor para acceder a la información alojada en la BD.  En caso de SQLite esto no es necesario ya que para cada instancia de la aplicación existe una base de datos individual que solamente podemos acceder a través de nuestro software  Sin mencionar que es bastante sencillo el manejo de datos, todo esto hace que SQLite sea una herramienta imprescindible para el desarrollo de aplicaciones móviles en la plataforma de Android. |

|  |
| --- |
| BIBLIOGRAFÍA |
| [*https://developer.android.com/training/data-storage/sqlite*](https://developer.android.com/training/data-storage/sqlite)  <https://developer.android.com/training/data-storage/room/>  <http://www.hermosaprogramacion.com/2014/10/android-sqlite-bases-de-datos/> |