

# Documentación

---

## Proyecto agenda

Iván Gallego

### Índice de contenidos

1. [Leer datos al inicio](#)
2. [Añadir datos](#)
3. [Eliminar datos](#)
4. [Modificar datos](#)

### Leer datos al inicio

#### cargarDatos

El método `cargarDatos` hace una consulta a la base de datos y utiliza el método `leerContactos` para extraer los contactos existentes de la base de datos

```
private void cargarDatos() {
    contactos = new ArrayList<>();
    if (dbContactos != null) {
        contactosDatabase = dbContactos.getReadableDatabase();
        Cursor cursor = contactosDatabase.rawQuery(
            "SELECT * FROM contactos", null);
        leerContactos(cursor);
        contactosDatabase.close();
    }
}
```

#### leerContactos

El método `leerContactos` recorre con el cursor todos los resultados de la consulta hecha por el método `cargarDatos` y construye un objeto `Contactos` a partir de la información obtenida

```
private void leerContactos(Cursor cursor) {
    cursor.moveToFirst();
    Contacto c;
    while (!cursor.isAfterLast()) {
        c = new Contacto();
        c.setNombre(cursor.getString(0));
        c.setApellido(cursor.getString(1));
        c.setTelefono(cursor.getString(2));
        c.setCorreo(cursor.getString(3));
        Bitmap imagen = convertirBytesBitmap(cursor.getBlob(4));
    }
}
```

```
        c.setImagen(imagen);
        c.setAmigo(cursor.getInt(5) == 1);
        c.setFamilia(cursor.getInt(6) == 1);
        c.setTrabajo(cursor.getInt(7) == 1);
        contactos.add(c);
        cursor.moveToNext();
    }
}
```

## Añadir datos

### guardarContacto

El método `guardarContacto` comprueba que no exista un contacto con el nombre igual, a continuación, lo introduce en la base de datos utilizando `ContentValues`

```
private void guardarContacto(Contacto c) {
    contactosDatabase = dbContactos.getWritableDatabase();
    SQLiteDatabase readableDatabase = dbContactos.getReadableDatabase();
    long count = DatabaseUtils.queryNumEntries(
        readableDatabase, "contactos", "nombre = ?",
        new String[]{c.getNombre()});
    if (count == 0){
        ContentValues values = new ContentValues();
        values.put("nombre", c.getNombre());
        values.put("apellido", c.getApellido());
        values.put("telefono", c.getTelefono());
        values.put("correo", c.getCorreo());
        Bitmap b = c.getImagen();
        byte[] bytesImagen = null;
        if (b != null) {
            bytesImagen = convertirImagenBytes(c.getImagen());
        }
        values.put("imagen", bytesImagen);
        values.put("amigo", c.isAmigo());
        values.put("trabajo", c.isTrabajo());
        values.put("familia", c.isFamilia());
        contactosDatabase.insert("contactos", null, values);
    }
    contactosDatabase.close();
    readableDatabase.close();
}
```

## Eliminar datos

### eliminarContacto

El método `eliminarContacto` elimina un contacto de la base de datos utilizando su nombre como referencia

```
private void eliminarContacto(Contacto contacto) {
    if (dbContactos != null) {
        contactosDatabase = dbContactos.getWritableDatabase();
        contactosDatabase.delete(
            "contactos", "nombre = ?", new String[] {contacto.getNombre()});
        contactosDatabase.close();
    }
}
```

## Modificar datos

### editarContacto

El método `editarContacto` comprueba que no exista un contacto con el nombre nuevo a editar, a continuación, edita el contacto anterior por el nuevo en la base de datos

```
private void editarContacto(Contacto editado, Contacto nuevo) {
    contactosDatabase = dbContactos.getWritableDatabase();
    SQLiteDatabase readableDatabase = dbContactos.getReadableDatabase();
    long count = DatabaseUtils.queryNumEntries(
        readableDatabase, "contactos", "nombre = ?",
        new String[]{nuevo.getNombre()});
    if (count == 0){
        ContentValues values = new ContentValues();
        values.put("nombre", nuevo.getNombre());
        values.put("apellido", nuevo.getApellido());
        values.put("telefono", nuevo.getTelefono());
        values.put("correo", nuevo.getCorreo());
        Bitmap b = nuevo.getImagen();
        byte[] bytesImagen = null;
        if (b != null) {
            bytesImagen = convertirImagenBytes(nuevo.getImagen());
        }
        values.put("imagen", bytesImagen);
        values.put("amigo", nuevo.isAmigo());
        values.put("trabajo", nuevo.isTrabajo());
        values.put("familia", nuevo.isFamilia());
        contactosDatabase.update(
            "contactos", values, "nombre = ?", new String[]{editado.getNombre()});
    }
    contactosDatabase.close();
    readableDatabase.close();
}
```

Para poder realizar las operaciones anteriores, debe haber las siguientes variables de clase

```
private BDContactos dbContactos;  
private SQLiteDatabase contactosDatabase;  
public ArrayList<Contacto> contactos;  
private Contacto contactoTemp;
```

Siendo:

1. **dbContactos** - La base de datos
2. **contactosDatabase** - La base de datos sobre la que se puede escribir o leer dependiendo del case
3. **contactos** - La lista de contactos
4. **contactoTemp** - El contacto que se utiliza de manera auxiliar para saber cuál es el contacto editado, para posteriormente editarlo en la base de datos