

Manual de Usuario

Vamos a desarrollar una aplicación para el mantenimiento de una tabla de medicamentos que son fabricados por determinados laboratorios. El esquema de la base de datos es el siguiente:



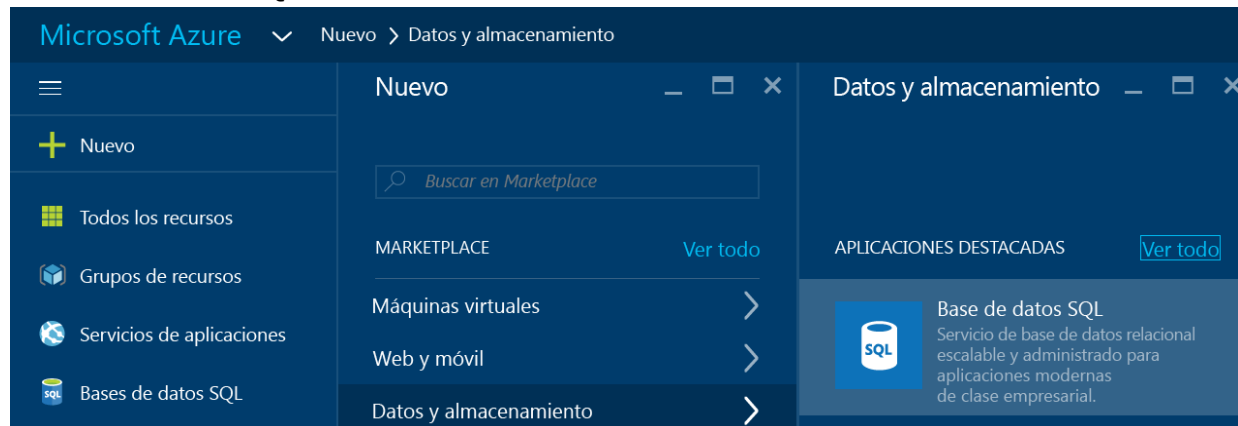
La aplicación contará con las siguientes características:

- Los Usuarios, Roles, Permisos y Laboratorios deberán estar cargados en la base de datos. Al menos deberán haber los siguientes roles: "Administrador", "Usuario" e "Invitado" y al menos un usuario con cada rol. De los cuales, están cargados los siguientes usuarios:
 - Con Rol de Administrador:
 - Usuario: admin1, password: admin1.
 - Usuario: admin2, password: admin2.
 - Con Rol de Usuario:
 - Usuario: user1, password: user1.
 - Usuario: user2, password: user2.
 - Con Rol de Invitado:
 - Usuario: invitado1, password: invitado1.
 - Usuario: invitado2, password: invitado2.
- La aplicación contiene:
 - Una pantalla de Login
 - Una Pantalla para mostrar todos los datos de los medicamentos.
 - Aquellos medicamentos cuya cantidad disponible sea menor que 10 deberá aparecer la cantidad en rojo en la rejilla (el resto en negro).
 - Si el usuario no tiene Rol de "Invitado" Al seleccionar un medicamento se cargarán sus valores en las cajas de texto y la lista de laboratorios correspondientes.
 - Al pinchar sobre Salir se cerrará la aplicación.
 - Al pinchar sobre Limpiar se vaciará cualquier contenido de las cajas de texto y se seleccionará cualquier posible laboratorio de la lista de laboratorios.
 - Si el usuario tiene Rol de "Administrador", al pinchar sobre Insertar se insertará el medicamento cuyos valores estén en las cajas de texto y la lista de laboratorios correspondientes.
 - Si el usuario tiene Rol de "Administrador", al pinchar sobre Borrar se borrará el medicamento cuyo Identificador esté en la cajas de texto correspondiente.
 - Si el usuario tiene Rol de "Administrador", al pinchar sobre Modificar se modificarán los valores del medicamento el medicamento cuyos valores almacenados en la base de datos difieran de los que estén en las cajas de texto y la lista de laboratorios correspondientes.

1. Creación de la Base de Datos SQLServer en Azure

Para crear la Base de Datos hemos utilizado Azure (servicio en la Nube que ofrece Microsoft). Hemos utilizado una cuenta como estudiantes de la Universidad de Málaga, para poder utilizarlo de forma gratuita. Hemos seguidos los siguientes pasos:

1. Clicamos en **Nuevo, Datos y almacenamiento** y por último en Base de datos SQL.



2. Ahora, se nos desplegará un formulario para rellenar con la información de la Base de Datos.

The screenshot shows the 'Base de datos SQL' configuration form. It has a dark blue header with the title 'Base de datos SQL' and window controls. The form contains several fields with red asterisks indicating required information: 'Nombre de base de datos' (Database name) with a placeholder 'Escriba el nombre de la base de datos'; 'Suscripción' (Subscription) with a dropdown menu showing 'Contenido de SQL DB'; 'Grupo de recursos' (Resource group) with radio buttons for 'Crear nuevo' (selected) and 'Usar existente', followed by an empty text field; 'Seleccionar origen' (Select origin) with a dropdown menu showing 'Base de datos en blanco'; 'Servidor' (Server) with the text 'Configurar los valores obligatorios' and a red exclamation mark icon; 'Plan de tarifa' (Pricing tier) with a dropdown menu showing 'S2 estándar'; and 'Intercalación' (Collation) with a text field showing 'SQL_Latin1_General_CP1_CI_AS'. Each section has a right-pointing arrow to proceed.

En nuestro caso, el nombre de la Base de Datos es "GI" y el grupo de recursos "eCorp". El plan de tarifa hemos dejado el que venia por defecto que era el gratuito.. Todos los demás aspectos los hemos dejado por defecto.

Ahora, tendremos que pulsar sobre "Configurar los valores obligatorios" para crear y configurar un servidor que sustente a nuestra Base de Datos. Una vez hayamos hecho click sobre esta opción, le damos a "Crear un nuevo servidor". Y si nos desplegará el siguiente formulario:

Nuevo servidor

* Nombre de servidor

Introduzca un nombre de servidor

.database.windows.net

* Inicio de sesión del administrador del servidor

Escribir nombre de usuario

* Contraseña

* Confirmar contraseña

* Ubicación

Este de Australia

Crear servidor V12 (última actualización)

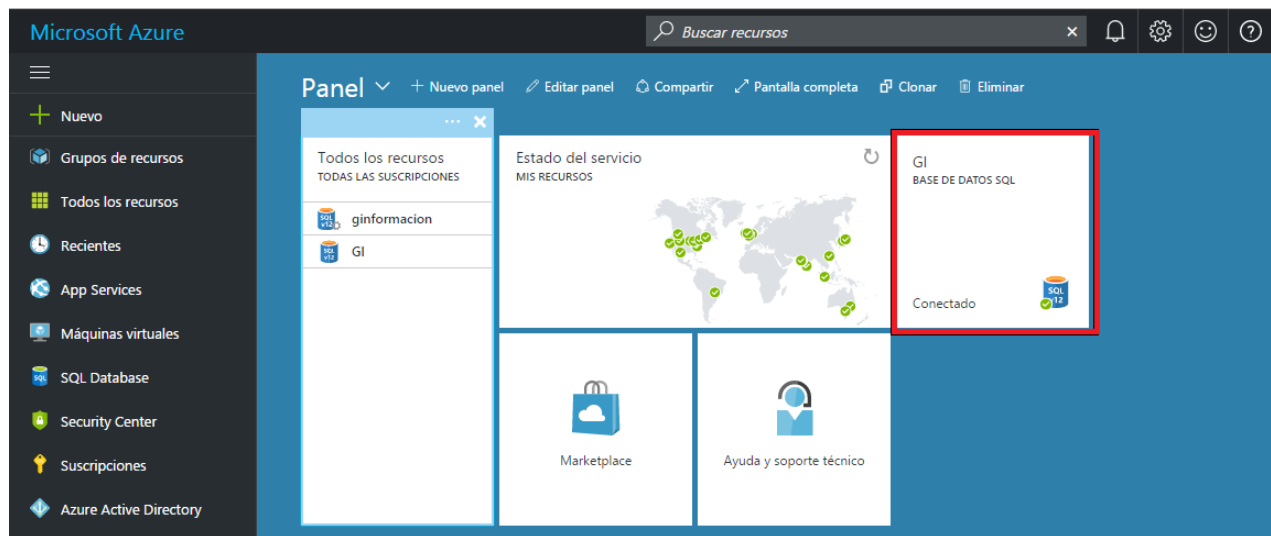
Sí No

☒ Permitir que los servicios de Azure accedan al servidor ⓘ

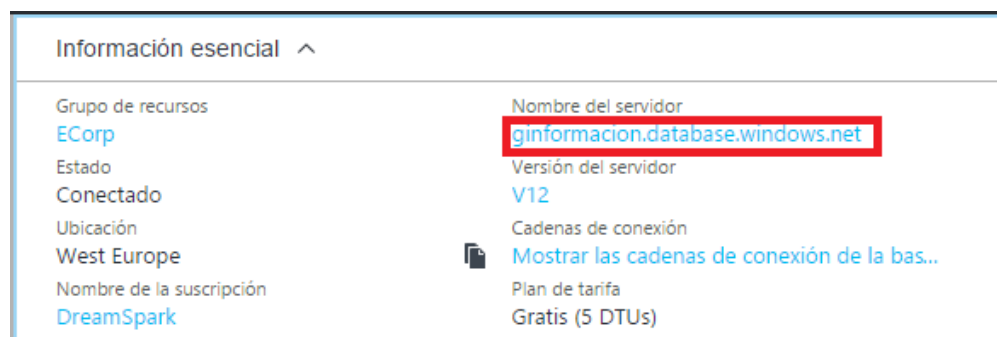
En nuestro caso, el nombre del servidor es "ginformacion.database.windows.net". También tendremos que crear una cuenta para la administración del servidor y gestión del mismo. En nuestro caso la cuenta es "admingestion".

Una vez relleno el formulario, aceptamos y volveremos al formulario anterior de la Base de Datos. Volvemos a aceptar éste y estará todo creado.

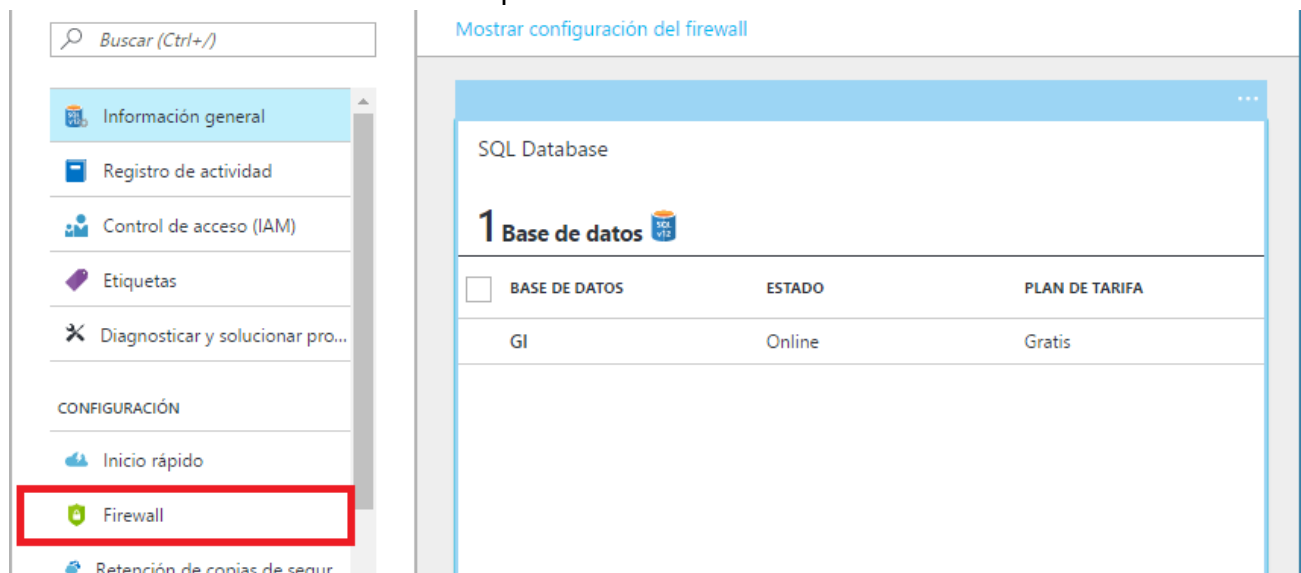
Una vez esté todo creado, deberemos indicarle a nuestro servidor qué rango de direcciones IP admite. Para ello, estando en la página principal de Azure, hacemos click en nuestra Base de Datos.



Una vez estamos dentro de la información de la base de datos, clicamos sobre el nombre del servidor:



Buscamos en el menú lateral la opción "Firewall":

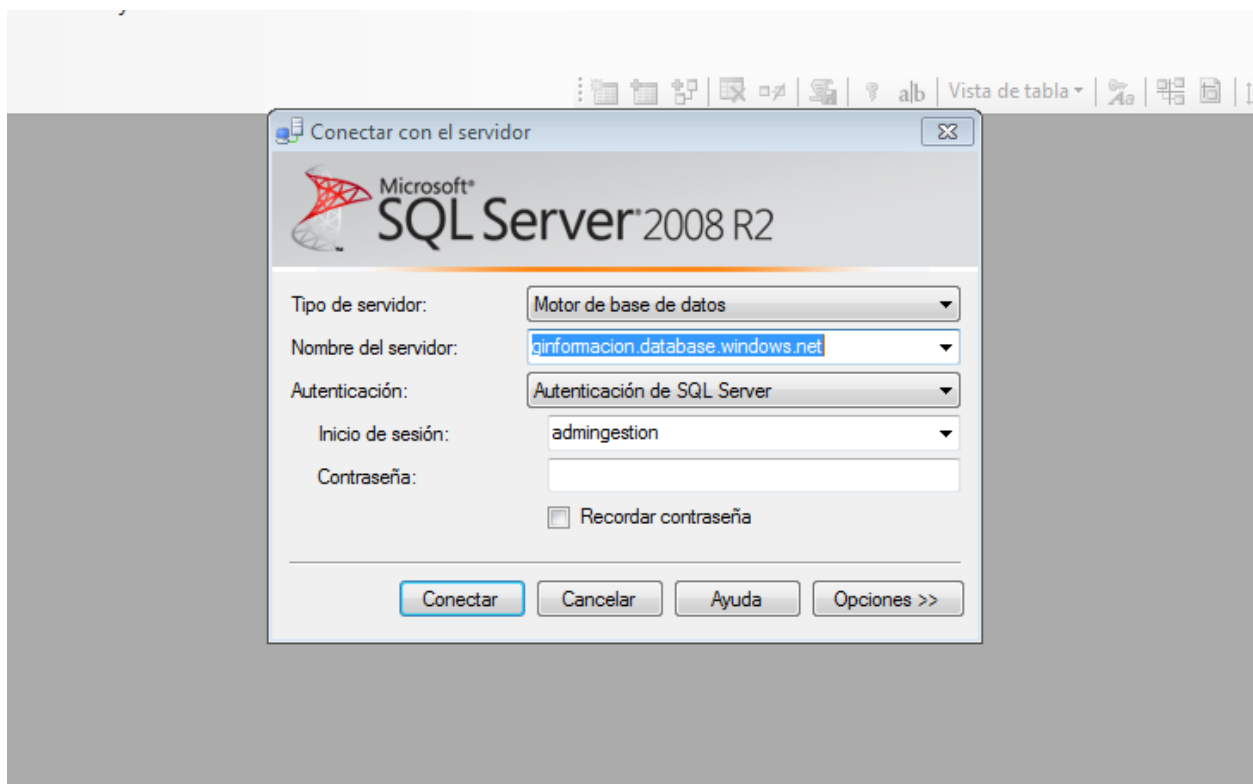


Dentro de esta opción podremos configurar los rangos de IP que queramos que nuestro servidor acepte. En nuestro caso, lo hemos configurado para que acepte todas las IPs (0.0.0.0 - 255.255.255.255) para simplificar la práctica.

NOTA: si nos conectamos a través del Wifi de la UMA no podremos conectarnos por el proxy.

2. ¿Cómo conectarnos a la Base de Datos con Microsoft SQL Management Studio 2008?

Desde la propia máquina virtual de la asignatura, abrimos Microsoft SQL Management e iniciamos sesión con los datos de nuestro servidor, tal y como aparece en la siguiente imagen.



Introducimos la contraseña pertinente y tendremos acceso al servidor.

Una vez dentro del servidor, tal y como hemos hecho en las prácticas anteriores, creamos las tablas, las relacionamos e introducimos los datos necesarios. Aquí se adjunta una imagen con las tablas creadas, relacionadas y ejecutando una consulta para mostrar el contenido de una tabla:

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Explorador de objetos' (Object Explorer) displays the database structure for 'ginformacion.database.windows.net'. The 'Bases de datos' (Databases) folder is expanded, showing 'Bases de datos del sistema' (System Databases) and 'GI'. Under 'GI', the 'Tablas' (Tables) folder is expanded, showing 'Tablas del sistema' (System Tables) and user tables: 'dbo.tLaboratorio', 'dbo.tMedicamento', 'dbo.tPantalla', 'dbo.tPermiso', 'dbo.tRol', and 'dbo.tUsuario'. The 'dbo.tMedicamento' table is selected. The main pane shows the 'SQLQuery1.sql - gi...mingestion (109)' query editor with the query: `SELECT * FROM tMedicamento`. The 'Resultados' (Results) tab displays the query output as a table with 17 rows and 4 columns: 'ID_MEDICAMENTO', 'NOMBRE_MEDICAMENTO', 'CANTIDAD_DISPONIBLE', and 'LABORATORIO'. The status bar at the bottom indicates 'Consulta ejecutada correctamente.' (Query executed successfully).

ID_MEDICAMENTO	NOMBRE_MEDICAMENTO	CANTIDAD_DISPONIBLE	LABORATORIO	
1	2	ADRENOL	2	11
2	3	ADVANTAN	3	9
3	4	ADVANTAN	4	12
4	5	ADVANTAN	5	10
5	6	ADVANTAN	6	11
6	7	AERIUS	7	18
7	8	AERIUS	8	18
8	9	AEROCROM	9	15
9	10	AERODIOL	10	24
10	11	AEROGAL N4	11	13
11	12	AEROTINA	12	15
12	13	AEROTINA	13	15
13	14	AEROTINA	14	11
14	15	AEROTINA D	15	9
15	16	AEROTINA D	16	11
16	17	AEROVAC G	17	10
17	18	AEROXINA	18	19

3. ¿Cómo conectarnos a la Base de Datos desde la Aplicación Android?

Para conectarse a la Base de Datos hemos decidido usar el driver jTDS.

El driver jTDS es un driver JDBC 3.0 open source para SQL Server (6.5,7,2000,2005,2008, 2012) y Sybase. (<http://jtds.sourceforge.net/>)

Gradle:

compile group: 'net.sourceforge.jtds', name: 'jtds', version: '1.3.1'

Código usado para establecer conexión:

```
Class.forName("net.sourceforge.jtds.jdbc.Driver");
ConnectionURL = "jdbc:jtds:sqlserver://ginformacion.database.windows.net:1433/GI;";
connection = DriverManager.getConnection(ConnectionURL, "admingestion", "Informacion1");
```

4. ¿Cómo iniciar sesión?

Para iniciar sesión, la aplicación se conecta a la Base de Datos definido por la URL:

```
jdbc:jtds:sqlserver://ginformacion.database.windows.net:1433/GI;
```

La aplicación permitirá realizar ciertas operaciones dependiendo de los permisos que dicho usuario tiene en la Base de datos. Los permisos que tiene cada usuario se obtendrá al iniciar sesión. Para iniciar sesión escribiremos nuestro usuario en el campo Username y nuestra contraseña en el campo Password; después hacemos click en el botón de abajo LOGIN y accederemos a la aplicación con los permisos correspondientes a nuestro tipo de usuario.

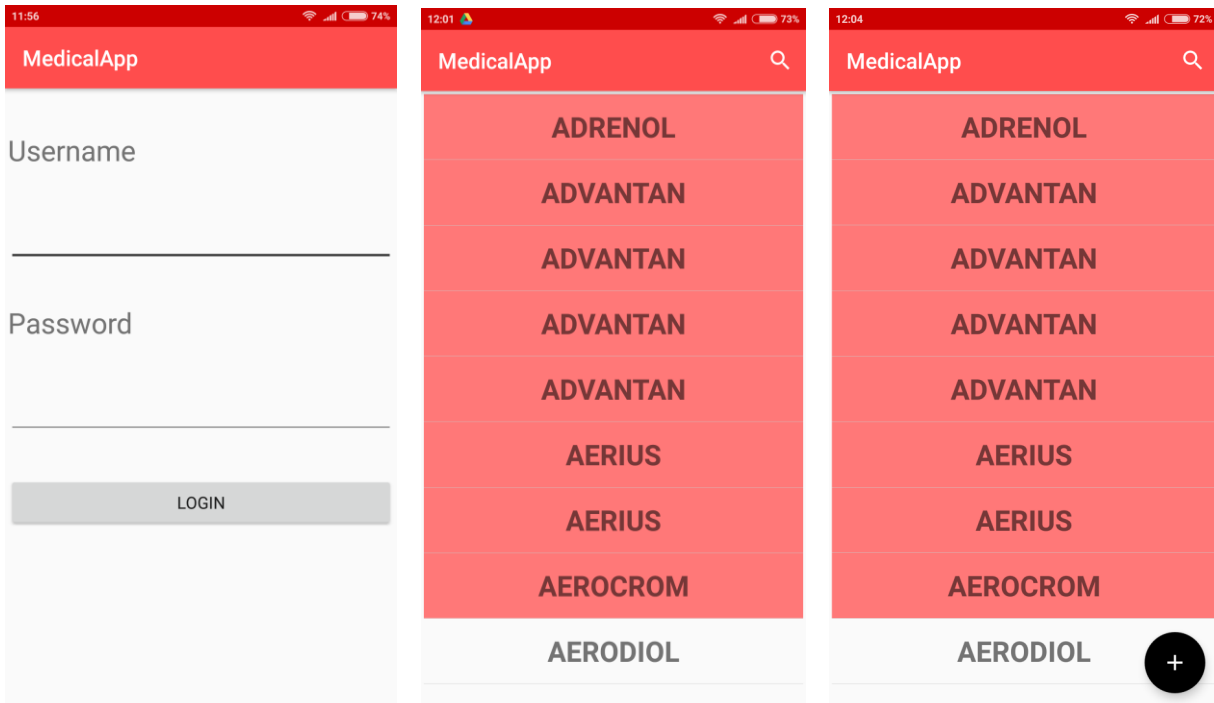
- **Operaciones que tiene cada usuario**

- Administrador: visualizar la lista y los datos, editar, borrar y añadir.
- Usuario: visualizar la lista y los datos.
- Invitado: visualizar la lista.

Las operaciones que tiene cada usuario determinará los elementos del interfaz gráfico que se presentarán en la aplicación.

Podemos observar en el siguiente código cómo se comprueban las credenciales del usuario y cómo se obtienen el rol que tiene cada usuario. La variable RolName pasa al resto de las clases para determinar las vistas que tiene permitido cada usuario y las operaciones que puede realizar.

```
String query = "SELECT * FROM tUsuario WHERE nombre='" + mUsername + "' " +  
    "AND password COLLATE Latin1_General_CS_AS ='" + mPassword + "'";  
Statement stmt = con.createStatement();  
ResultSet rs = stmt.executeQuery(query);  
if(rs.next())  
{  
    z = "Login Successful";  
    rolName = rs.getString("rolName");  
    isSuccess=true;  
    con.close();  
}  
else  
{  
    z = "Invalid Credentials!";  
    isSuccess = false;  
}
```

En la interfaz, si el usuario es administrador, aparecerá además de la lista de medicamentos un botón flotante para añadir un nuevo medicamento.

5. ¿Cómo visualizar los datos de un fármaco?

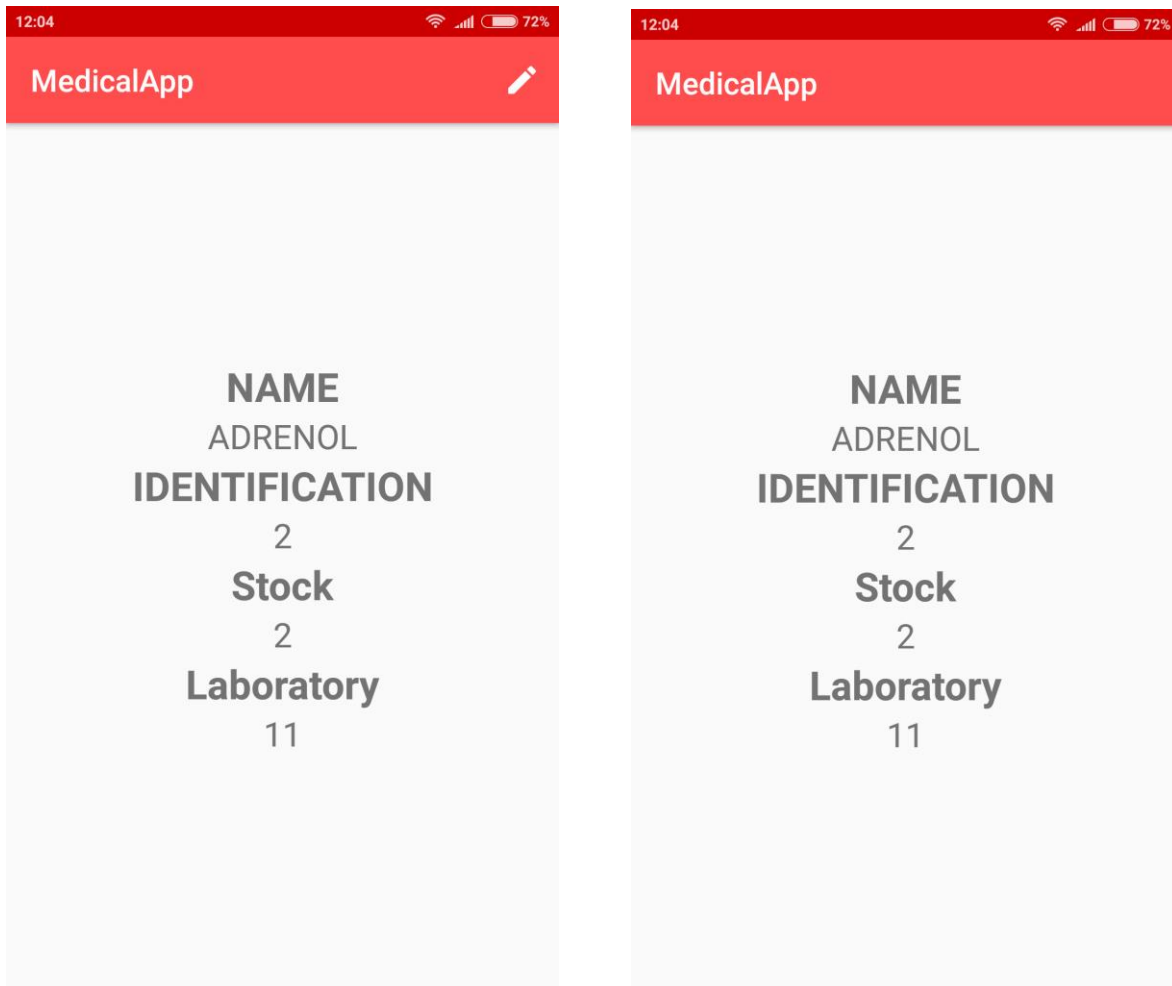
Tras el inicio de sesión, aparecerá una lista de todos los fármacos de la Base de Datos. Los elementos de la lista aparecerán en color rojo, si la cantidad disponible es inferior a 10.

Para poder visualizar la información que tiene cada fármaco basta con pulsar el elemento de la lista correspondiente.

El código que tenemos a continuación se corresponde con la obtención de datos:

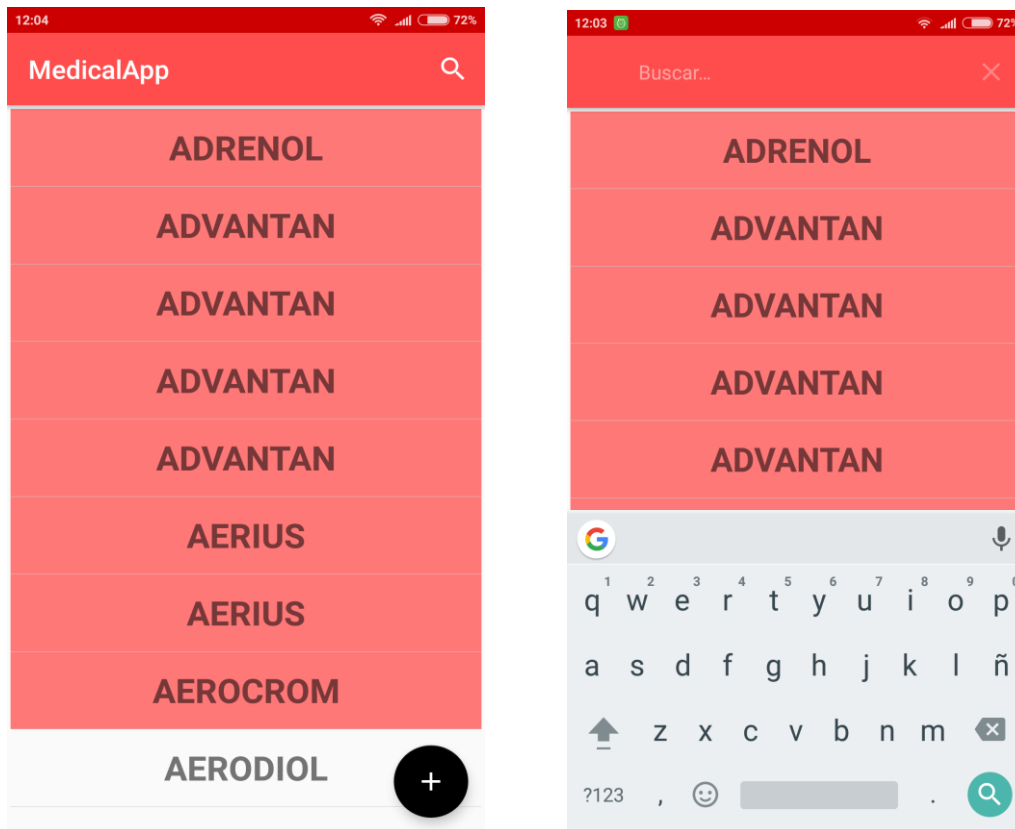
```
String query = "SELECT * FROM tMedicamento;";
Statement stmt = con.createStatement();
ResultSet rs = stmt.executeQuery(query);
while(rs.next()){
    DataProvider dataprovider = new DataProvider(rs.getString("NOMBRE_MEDICAMENTO"),
        rs.getInt("ID_MEDICAMENTO"),rs.getInt("CANTIDAD_DISPONIBLE"),rs.getInt("LABORATORIO"));
    medicineList.add(dataprovider);
}
```

En la interfaz, tendremos los campos de la base de datos correspondiente a la tupla de un medicamento y además, si el usuario es administrador, aparecerá el botón de editar medicamento (Lápiz) en la esquina superior derecha.



6. ¿Cómo buscar un fármaco?

Para realizar la búsqueda de un fármaco tenemos que pulsar el icono de la lupa que aparece en el menú (parte superior derecha de la aplicación) y nos abrirá una ventana con un cuadro de texto en la barra de menú para escribir el nombre del medicamento a buscar.



La búsqueda de un fármaco se basa en el componente `SearchView` de Android. Para definir el funcionamiento del `SearchView` tenemos que implementar el siguiente método:

```
//This method is implemented to let the user filter an element of the recycler view
@Override
public boolean onQueryTextChange(String newText) {
    //the newText is the text that the user inputs in the TextView of the SearchView
    newText = newText.toLowerCase();
    ArrayList<DataProvider> filteredMedicineList = new ArrayList<>();
    for(DataProvider dataProvider: medicineList){
        String name = dataProvider.getMedicineName().toLowerCase();
        if(name.startsWith(newText)){
            filteredMedicineList.add(dataProvider);
        }
    }
    //We shall change the elements shown in the recycler view with elements that the filter shows us
    adapter.setFilter(filteredMedicineList);
    return true;
}
```

El código anterior realiza una búsqueda en la lista de fármacos en la lista que hemos obtenido de la Base de Datos, y todos los fármacos que coinciden con el filtro se muestran por pantalla.

7. ¿Cómo eliminar un fármaco?

Para la eliminación de un elemento de la lista nos hemos basado en la propuesta de Material Design de google y hemos decidido aplicar la técnica de Contextual Action Mode.

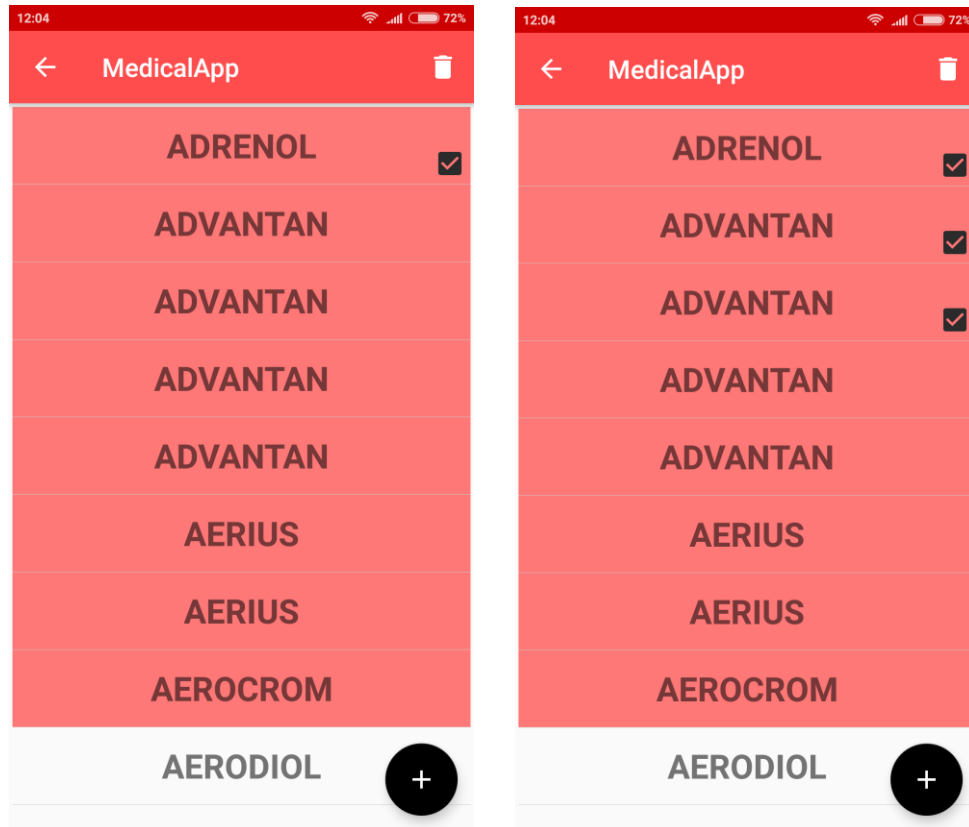
La eliminación de los fármacos se realiza mediante el código que tenemos a continuación:

```
//Changes the data when we use the delete menu
public void updateAdapter(ArrayList<DataProvider> listToDelete, ArrayList<DataProvider>list ) {
    medicineList = list;
    for (DataProvider dataProvider : listToDelete) {
        medicineList.remove(dataProvider);
    }
    try {
        DeleteData deleteData = new DeleteData(listToDelete);
        deleteData.execute().get();
    } catch (InterruptedException e) {
        e.printStackTrace();
    } catch (ExecutionException e) {
        e.printStackTrace();
    }
    notifyDataSetChanged();
}
```

Cómo podemos observar el código anterior borra los fármacos en la lista y llama a la clase DeleteData, que es una clase que ejecuta la sentencia SQL necesario para la eliminación de los datos del fármaco en la base de datos.

```
// Change below query according to your own database.
Statement stmt = con.createStatement();
for(DataProvider dataProvider: listToDelete){
    String query = "DELETE FROM tMedicamento WHERE ID_MEDICAMENTO="+dataProvider.getIdentification()+"";
    stmt.executeUpdate(query);
}
```

En la interfaz hemos optado por usar el método del click prolongado, que al hacerlo sobre un elemento de la lista, cambiamos de vista en la barra de menú y mostramos la opción de borrar (icono de papelera) y podemos o borrar el elemento seleccionado o seguir seleccionando elementos de la lista que quedarán marcados con un CheckBox para su posterior eliminación o si son vueltos a pulsar se deselectan.



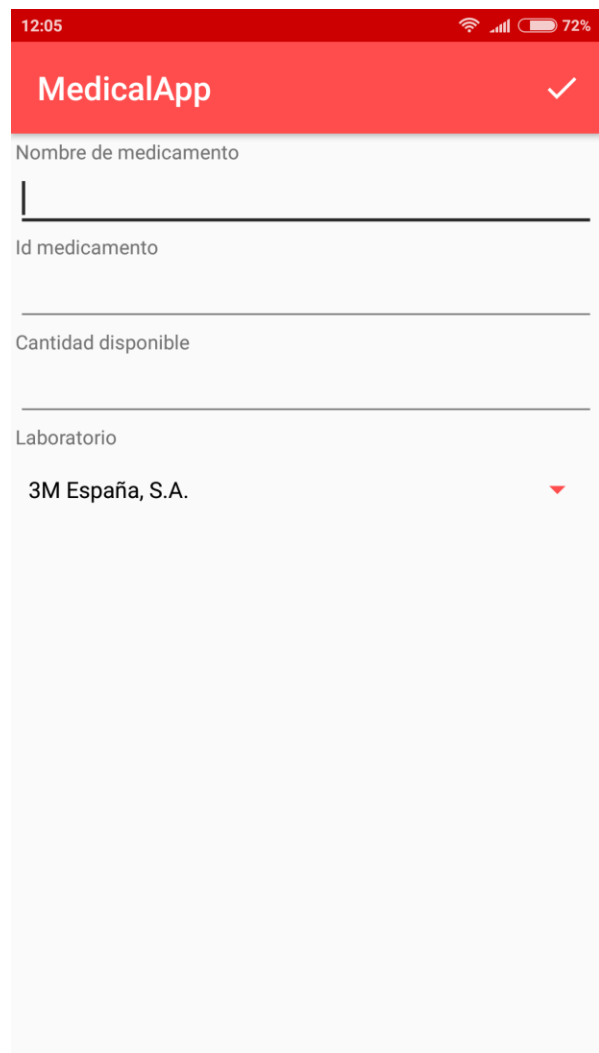
8. ¿Cómo crear un fármaco?

Para crear un fármaco tenemos que pulsar el botón de la parte inferior derecha. Tras pulsar dicho botón aparecerá un formulario a rellenar sobre la información que se desea insertar en la base de datos.

La creación de un fármaco también se basa en la propuesta del Material Design y consiste en utilizar un FloatingActionButton.

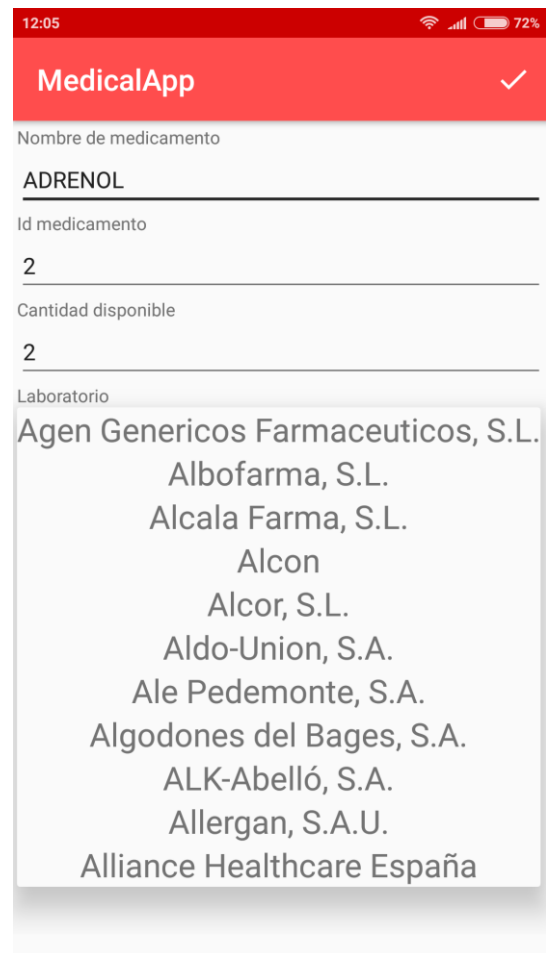
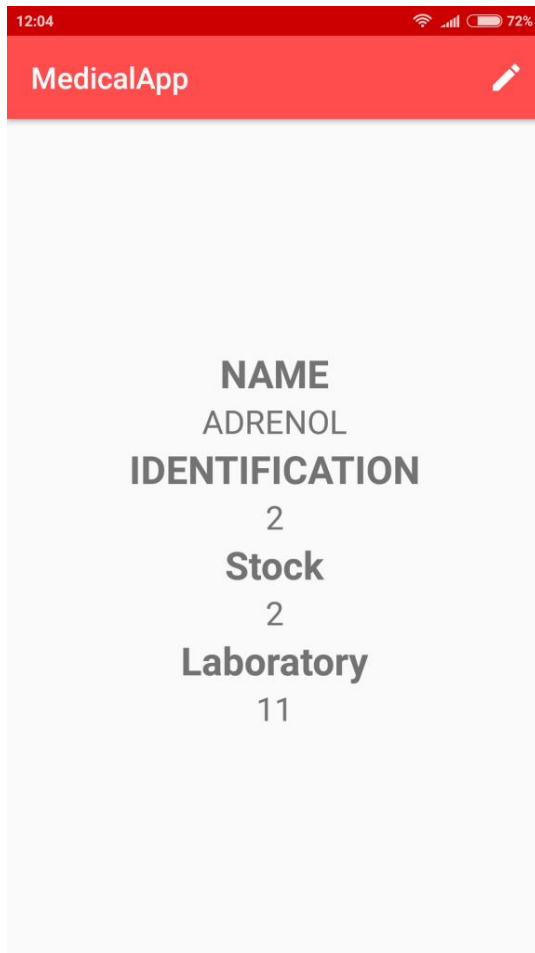
Para finalizar la inserción tenemos que pulsar el botón del "tick" (parte superior derecha) en el GUI del formulario. Este menú ejecutará el siguiente código:

```
String query = "INSERT INTO tMedicamento VALUES(" + id + "," + name + "," + amount + "," + laboratorio + ")";
try {
    Statement stmt = con.createStatement();
    stmt.executeUpdate(query);
    Intent intent = new Intent(InsertActivity.this, MainActivity.class);
    intent.putExtra("rolName", rolName);
    startActivity(intent);
} catch (java.sql.SQLException e) {
    z = "Este identificador ya existe";
}
```



9. ¿Cómo editar los datos de un fármaco?

La modificación de los datos de un fármaco se realiza pulsando elemento deseado de la lista, entrando en la pantalla de información del fármaco y seleccionando el menú de editar (el icono del lápiz). Una vez editado el fármaco, para guardar los cambios realizados hacemos click en el icono de "tick" en la parte superior derecha.



Para implementar la modificación de los datos de un fármaco hemos tenido que distinguir dos casos:

1. Si se cambia el ID, tenemos que borrar el fármaco y volver a insertarlo en la bases de datos. Esto se debe a que el ID del fármaco es el Primary Key de la Tabla.
2. Si no se cambia el ID, aprovechamos la operación UPDATE.

```

if (idOriginal == id) {
    // Change below query according to your own database.
    try {
        String query = "UPDATE tMedicamento SET NOMBRE_MEDICAMENTO='" + name + "' " +
            ",CANTIDAD_DISPONIBLE=" + amount + " ,LABORATORIO=" + laboratorio + " WHERE ID_MEDICAMENTO=" +
            idOriginal + ";";
        Statement stmt = con.createStatement();
        stmt.executeUpdate(query);
    } catch (java.sql.SQLException ex) {
        z = "El id ya existe";
    }
    Intent intent = new Intent(EditActivity.this, MainActivity.class);
    intent.putExtra("rolName", rolName);
    startActivity(intent);
} else {
    // Change below query according to your own database.
    try {
        String query = "INSERT INTO tMedicamento VALUES(" + id + "," + name + "," + amount + "," + laboratorio + ")";
        String query2 = "DELETE FROM tMedicamento WHERE ID_MEDICAMENTO=" + idOriginal + ";";
        Statement stmt = con.createStatement();
        stmt.executeUpdate(query);
        stmt.executeUpdate(query2);
        Intent intent = new Intent(EditActivity.this, MainActivity.class);
        intent.putExtra("rolName", rolName);
        startActivity(intent);
    } catch (java.sql.SQLException ex) {
        z = "El id ya existe";
    }
}
}

```

Integrantes:

- Adrián Cárdenas Jiménez
- Adrián García López
- Daniel María López Torres
- David Luque Fernández
- John Carlo Purihin Enríquez