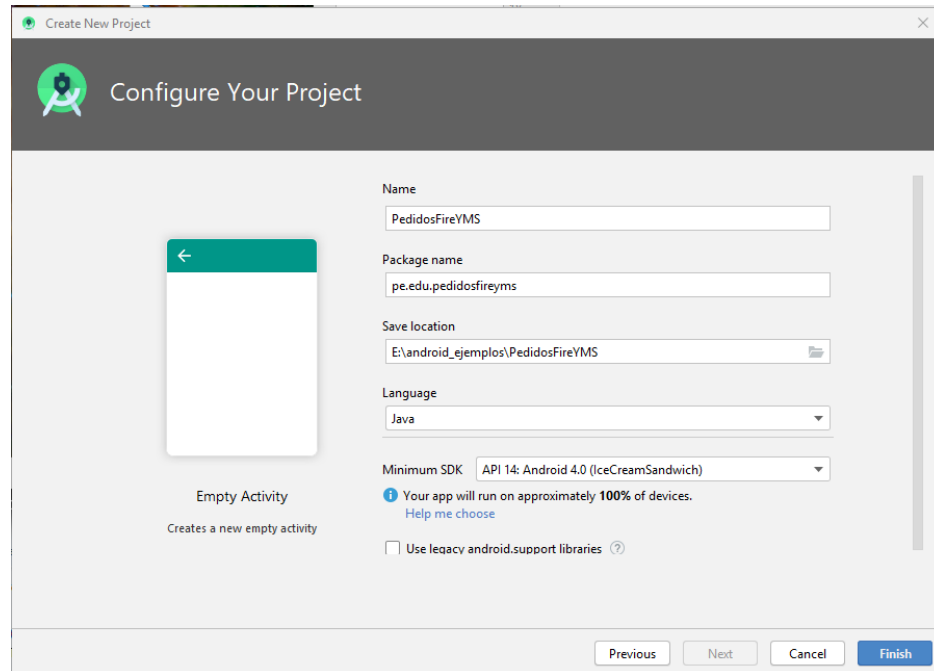


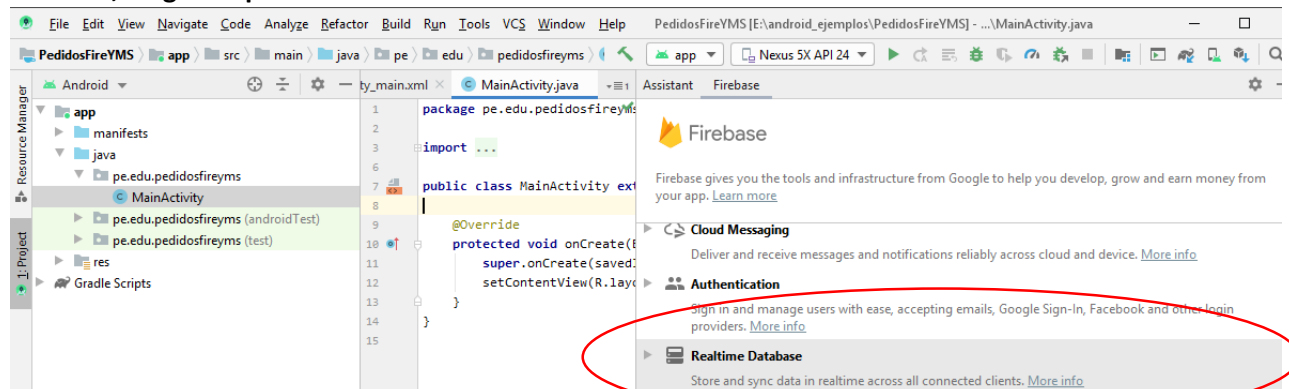
SEMANA 6 – PRACTICA 1

Instrucciones:

1. Iniciamos creando un proyecto en blanco al cual denominaremos PedidosFire



2. Dirigirse al menú Tools, elegir la opción Firebase, se mostrará el lateral derecho como se muestra, elegir la opción RealTime eDataBase.



Elegir la opción **Save and retrieve data**.



Realtime Database

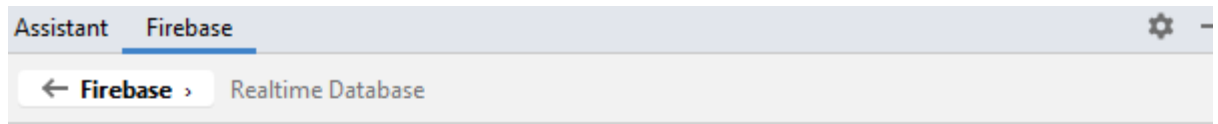
Store and sync data in realtime across all connected clients. [More info](#)



[Save and retrieve data](#)

Pedro Yuri Marquez Solis

3. Ahora se muestra la secuencia del proceso: Iniciar con el primer paso con el botón Connect to FireBase.



Save and retrieve data

Our cloud database stays synced to all connected clients in realtime and remains available when your app goes offline. Data is stored in a JSON tree structure rather than a table, eliminating the need for complex SQL queries.

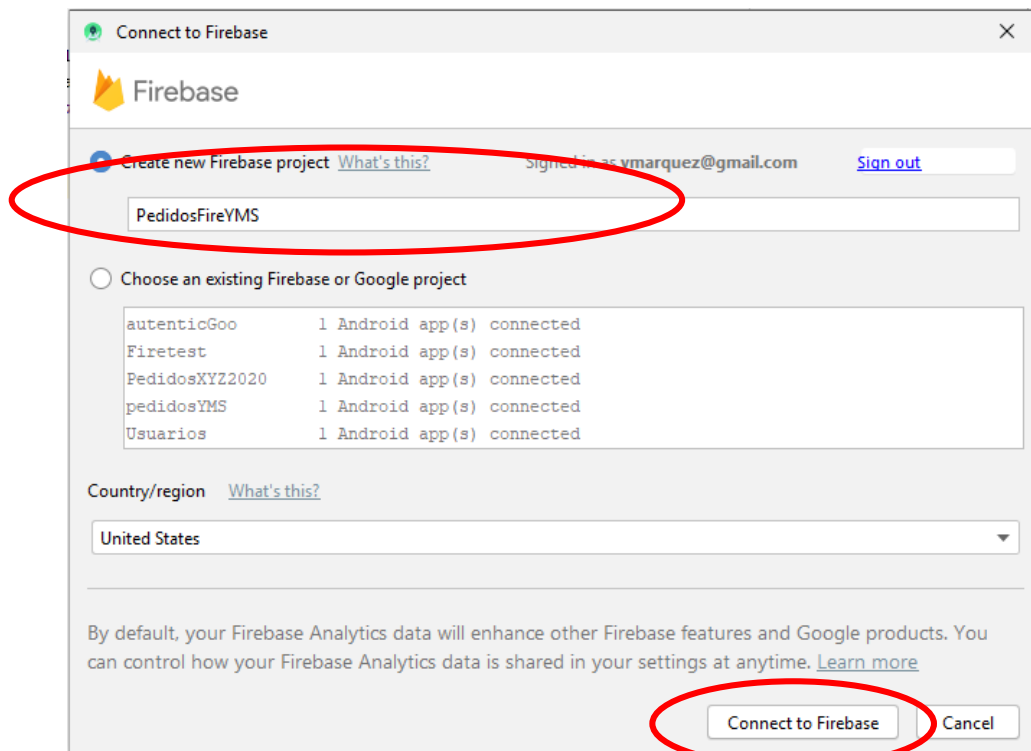
[Launch in browser](#)

1 Connect your app to Firebase

Connect to Firebase

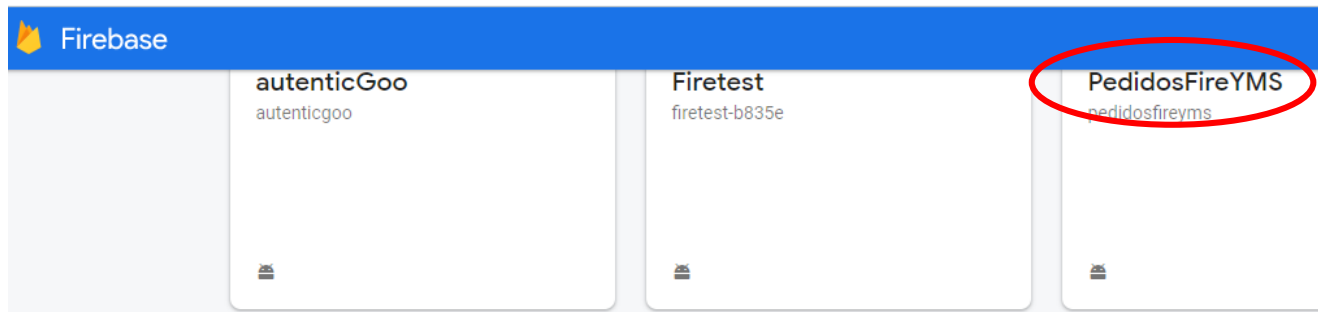
El proceso que se realizará activará (Connect to Firebase) creará un proyecto en el FireBase del desarrollador, el cual debe tener una cuenta de google obligatoriamente.

4. Se mostrará una ventana que permite elegir el nombre del proyecto, darle click a **Connect to Firebase**

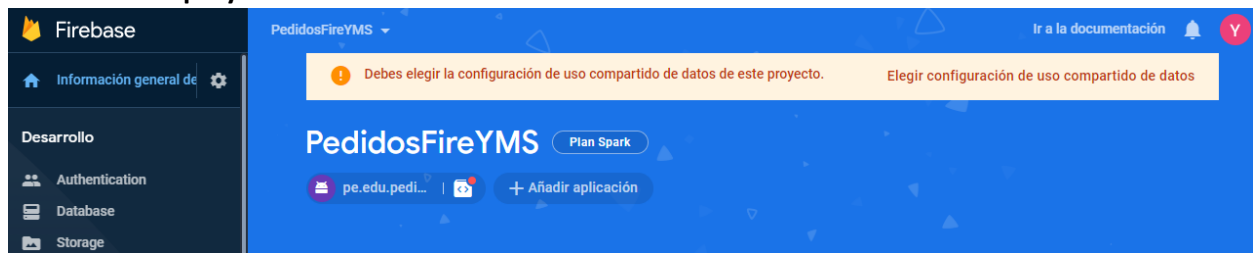
The screenshot shows the 'Connect to Firebase' dialog box. The 'Create new Firebase project' option is selected, and the project name 'PedidosFireYMS' is entered in the text field. The 'Choose an existing Firebase or Google project' option is also visible, with a list of existing projects: 'autenticGoo', 'Firetest', 'PedidosXYZ2020', 'pedidosYMS', and 'Usuarios'. The 'Country/region' dropdown is set to 'United States'. At the bottom, the 'Connect to Firebase' button is highlighted with a red circle, and the 'Cancel' button is also visible.

Este proceso tomará un tiempo, sólo debemos esperar unos minutos.

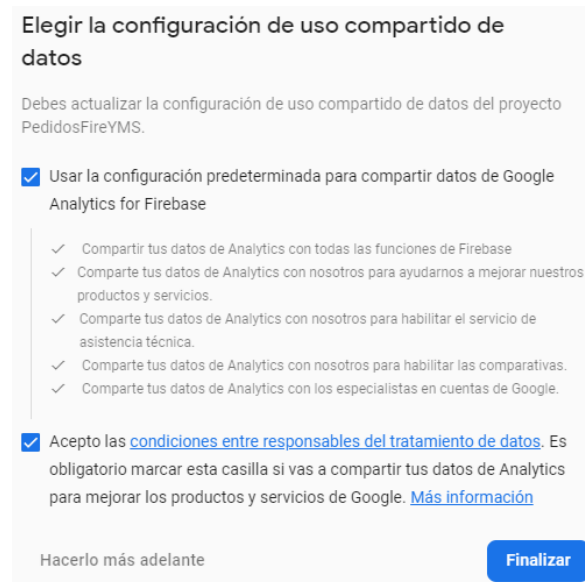
5. Dirigirse a la consola de Firebase (en el buscador escribir firebase console) y visualizar la lista de proyectos: Se mostrará la consola con el proyecto creado.



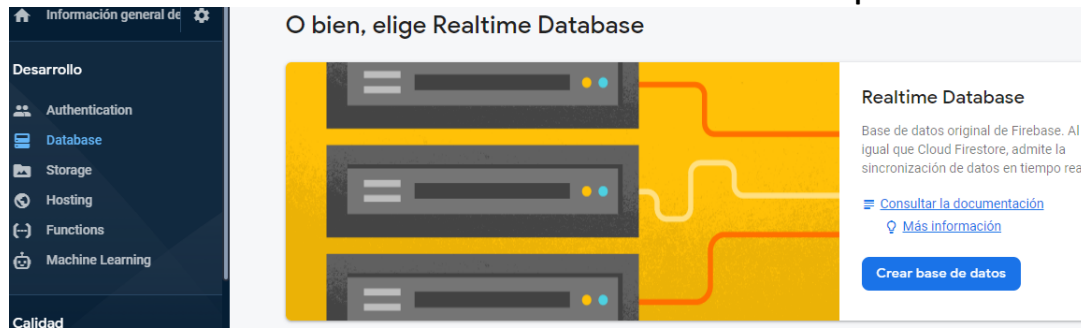
6. Seleccionar el proyecto :



Seleccionar la opción de uso compartido de datos:



7. Ahora elegir en el menú lateral la opción dataBase y RealTime Database:



En la ventana que se muestra ahora elegir: Empezar con el modo de prueba.



Se mostrará el nodo raíz de la base de datos creada:

Modificar las reglas a:

```
{
  "rules": {
    ".read": "true",
    ".write": "true",
  }
}
```



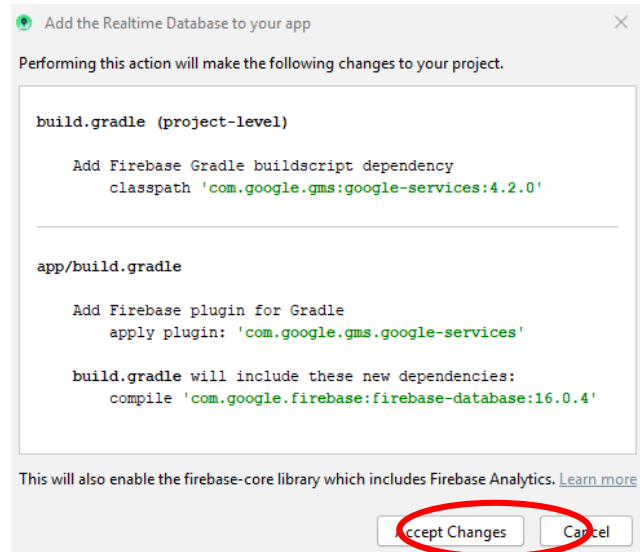
8. Agregar la base de datos a la aplicación

Retornamos al asistente y agregamos la base de datos a la aplicación.

② Add the Realtime Database to your app

Add the Realtime Database to your app

Se mostrará un cuadro de diálogo, darle clic a Accept Changes.

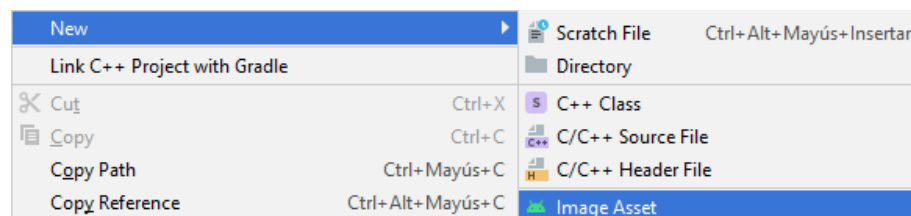


El asistente mostrará una respuesta como esta indicativa que ya se tiene la conexión con la base de datos del proyecto creado.

② Add the Realtime Database to your app

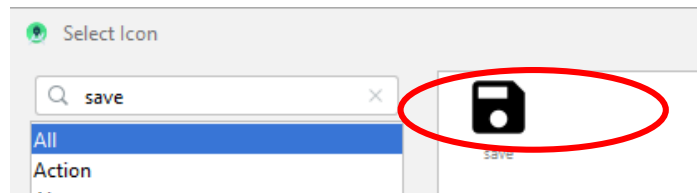
✓ Dependencies set up correctly

9. Obtenemos los elementos gráficos del menú:

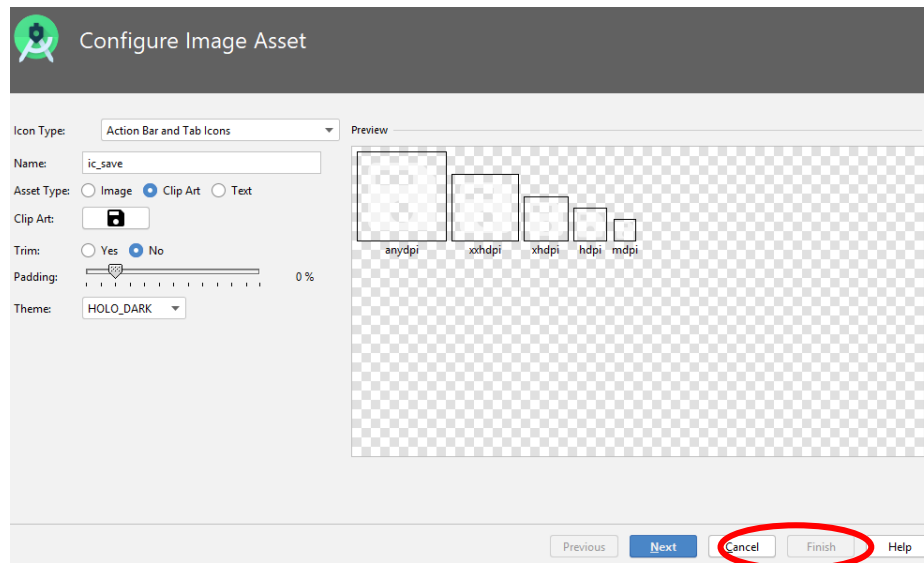


En la ventana que se muestra elegir: en Icon Type: Action Bar and tab Icons, y en el name ingresar por ejemplo ic_save. Se sugieren : ic_add, ic_update, ic_list para el resto de recursos icon.

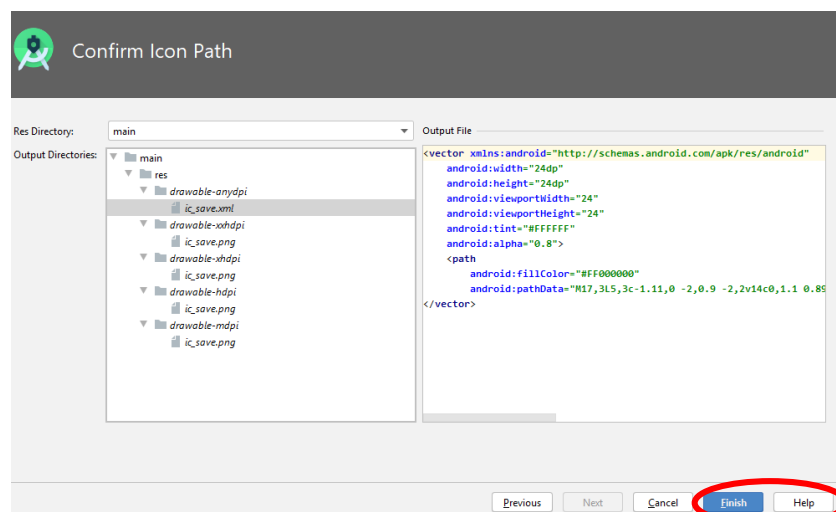
Para asignar el ícono efectuar clic al botón ClipArt y efectuar la búsqueda, elegir el icon a asociar.



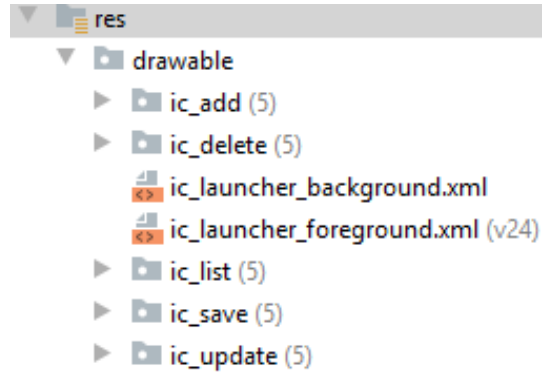
Se mostrarán las versiones del icon elegido, darle clic a next:



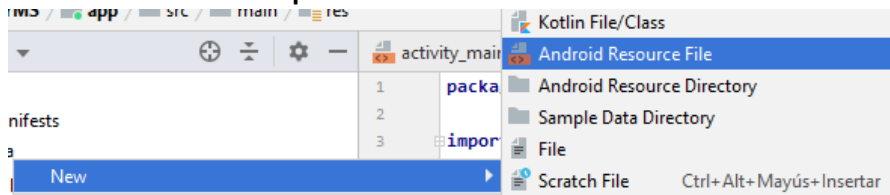
Finalmente confirmar la creación del recurso xml:



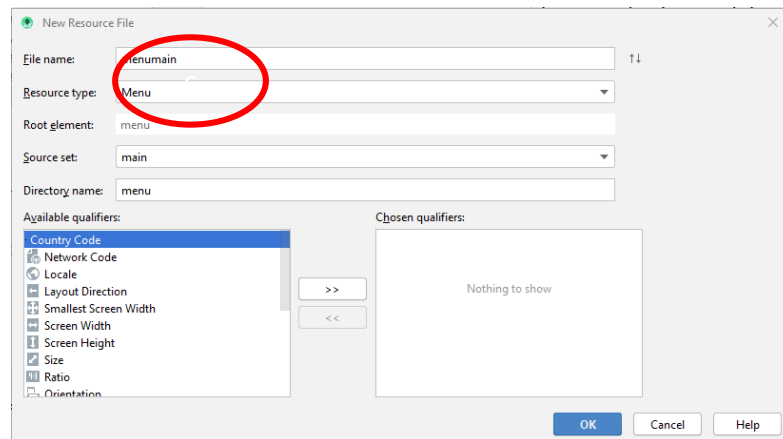
10. Repetir el proceso hasta obtener en res los siguientes recursos:



11. Ahora vamos a crear el menú: para ello crea un nuevo Android Resource File



Ingresar los siguientes datos:



Click en Ok.

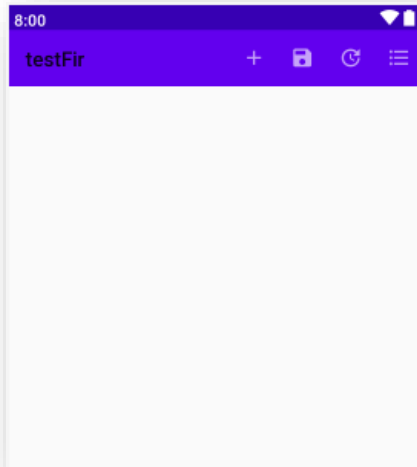
12. Crear el menú:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto"
      xmlns:android="http://schemas.android.com/apk/res-auto">
    <item android:id="@+id/icon_add"
          android:icon="@drawable/ic_add"
          android:title="Agregar"
          app:showAsAction="always">
    </item>
    <item android:id="@+id/icon_save"
          android:icon="@drawable/ic_save"
          android:title="Guardar"
          app:showAsAction="always">
```



```
</item>
<item android:id="@+id/icon_update"
      android:icon="@drawable/ic_update"
      android:title="Actualizar"
      app:showAsAction="always">
</item>
<item android:id="@+id/icon_list"
      android:icon="@drawable/ic_list"
      android:title="Listar"
      app:showAsAction="always">
</item>
</menu>
```

El menu creado deberá apreciarse como se muestra:



13. Insertar el menú en el MainActivity:

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu1, menu);
    return super.onCreateOptionsMenu(menu);
}

@Override
public boolean onOptionsItemSelected(@NonNull MenuItem item) {
    switch (item.getItemId())
    {
        case R.id.icon_add:{
            Toast.makeText(this, "agregar", Toast.LENGTH_SHORT).show();
            break;
        }
        case R.id.icon_save:{
            Toast.makeText(this, "guardar", Toast.LENGTH_SHORT).show();
            break;
        }
    }
    return super.onOptionsItemSelected(item);
}
```

14. Implementación de la clase Pedido:

```
public class Pedidos {

    private String idpedido;
```




```
private String tipoped;
private String distrito;
private String descripcion;
private String entregar;
private String direccion;

public Pedidos() {
}

public Pedidos(String idpedido, String tipoped, String distrito, String descripcion, String
entregar, String direccion) {
    this.idpedido = idpedido;
    this.tipoped = tipoped;
    this.distrito = distrito;
    this.descripcion = descripcion;
    this.entregar = entregar;
    this.direccion = direccion;
}

public String getIdpedido() {
    return idpedido;
}

public void setIdpedido(String idpedido) {
    this.idpedido = idpedido;
}

public String getTipoped() {
    return tipoped;
}

public void setTipoped(String tipoped) {
    this.tipoped = tipoped;
}

public String getDistrito() {
    return distrito;
}

public void setDistrito(String distrito) {
    this.distrito = distrito;
}

public String getDescripcion() {
    return descripcion;
}

public void setDescripcion(String descripcion) {
    this.descripcion = descripcion;
}

public String getEntregar() {
    return entregar;
}

public void setEntregar(String entregar) {
    this.entregar = entregar;
}

public String getDireccion() {
    return direccion;
}

public void setDireccion(String direccion) {
    this.direccion = direccion;
}
}
```

15. Implementar la clase MyFirebaseApp

Con esta clase lograremos la sincronización de la aplicación cuando estemos operando en modo desconectado

```
import com.google.firebase.database.FirebaseDatabase;

public class MyFirebaseApp extends android.app.Application{
    @Override
    public void onCreate() {

        super.onCreate();
        FirebaseDatabase.getInstance().setPersistenceEnabled(true);
    }
}
```

16. Agregar los datos a mostrar en las listas:

```
<resources>
    <string name="app_name">PedidosXYZ2020</string>
    <string-array name="tiposped">
        <item>"Carga simple"</item>
        <item>"Carga pesada"</item>
        <item>"Sobre"</item>
    </string-array>
    <string-array name="distritos">
        <item>"Chilca"</item>
        <item>"El tambo"</item>
        <item>"Huancayo"</item>
        <item>"Chupaca"</item>
    </string-array>
    <string name="mensa_tipoPed">
        "Indique el tipo de pedido"
    </string>
    <string name="mensa_distrito">
        "Elija el distrito"
    </string>
</resources>
```

17. Definir el Layout del MainActivity:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:orientation="vertical">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="Pedidos a atender"
        android:textSize="28dp"></TextView>

    <Spinner
        android:id="@+id/spTipoPed"
        android:layout_width="366dp"
        android:layout_height="38dp"
        android:entries="@array/tiposped"
        android:prompt="@string/mensa_tipoPed">
```



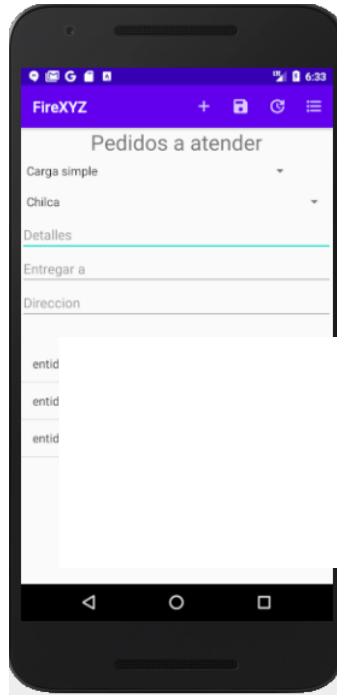
```
</Spinner>

<Spinner
    android:id="@+id/spdistrito"
    android:layout_width="match_parent"
    android:layout_height="43dp"
    android:entries="@array/distritos"
    android:prompt="@string/mensa_distrito"></Spinner>

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="textMultiline"
    android:hint="Detalles"
    android:id="@+id/edtdetalles"
></EditText>
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Entregar a"
    android:id="@+id/edtentregar"
>
</EditText>
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Direccion "
    android:id="@+id/edtdireccion"
>
</EditText>

<Space
    android:layout_width="match_parent"
    android:layout_height="34dp" />

<ListView
    android:id="@+id/lstpedidos"
    android:layout_width="match_parent"
    android:layout_height="match_parent"></ListView>
</LinearLayout>
```



18. Implementación del MainActivity:

```
public class MainActivity extends AppCompatActivity {
    Spinner sptipo, spdistrito;
    EditText edtdetalle, edtdireccion, edtentregar;

    ListView listViewPedidos;
    private List<Pedido> listPedidos = new ArrayList<Pedido>();
    ArrayAdapter<Pedido> arrayAdapterPedidos;

    private DatabaseReference mDataBase;
    private FirebaseDatabase firebaseDatabase;
    Pedido ped_sel;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        sptipo=findViewById(R.id.spTipoPed);
        spdistrito=findViewById(R.id.spdistrito);
        edtdetalle=findViewById(R.id.edtdetalles);
        edtdireccion=findViewById(R.id.edtdireccion);
        edtentregar=findViewById(R.id.edtentregar);

        listViewPedidos=findViewById(R.id.lstpedidos);
        mDataBase= FirebaseDatabase.getInstance().getReference();

        inicializarFirebase();
        listarPedidos();
        listViewPedidos.setOnItemClickListener(new AdapterView.OnItemClickListener()
        {
            @Override
            public void onItemClick(AdapterView<?> adapterView, View view, int i,
long l) {
```



```
        ped_sel=(Pedido) adapterView.getItemAtPosition(i);
        edtdetalle.setText(ped_sel.getDescripcion());
        edtentregar.setText(ped_sel.getEntregar());
        edtdireccion.setText(ped_sel.getDireccion());
        //buscar el indice del texto a mostrar
        String qtipopedido=ped_sel.getTipoped();
        String qDistrito=ped_sel.getDistrito();
        sptipo.setSelection(obtenerPosicionItem(sptipo,qtipopedido));
        spdistrito.setSelection(obtenerPosicionItem(spdistrito, qDistrito));
    }
    });
}

public static int obtenerPosicionItem(Spinner spinner, String texto){
    int posicion=0;
    for (int i= 0;i<spinner.getCount();i++){
        if (spinner.getItemAtPosition(i).toString().equalsIgnoreCase(texto))
posicion=i;
    }
    return posicion;
}

private void listarPedidos() {
    mDataBase.child("Pedido").addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            listPedidos.clear();
            int n=0;
            for (DataSnapshot objSnapShot : dataSnapshot.getChildren()){
                Pedido p=objSnapShot.getValue(Pedido.class);
                n++;
                listPedidos.add(p);
                arrayAdapterPedidos = new ArrayAdapter<Pedido>(MainActivity.this,
android.R.layout.simple_list_item_1,listPedidos);
                listViewPedidos.setAdapter(arrayAdapterPedidos);
            }
        }

        @Override
        public void onCancelled(@NonNull DatabaseError databaseError) {

        }
    });
}

private void inicializarFirebase(){
    FirebaseApp.initializeApp(this);
    firebaseDatabase=FirebaseDatabase.getInstance();
    mDataBase=firebaseDatabase.getReference();
}

public void registrarpedido(){
    String tipo= sptipo.getSelectedItem().toString();
    String distrito = spdistrito.getSelectedItem().toString();
    String descripcion= edtdetalle.getText().toString();
    String entrega = edtentregar.getText().toString();
    String direccion = edtdireccion.getText().toString();

    if (!TextUtils.isEmpty(descripcion) ){
```



```
String id = mDataBase.push().getKey();
Pedido pedido = new
Pedido(id,tipo,distrito,descripcion,entrega,direccion);
mDataBase.child("Pedido").child(id).setValue(pedido);
Toast.makeText(this, "pedido registrado", Toast.LENGTH_SHORT).show();
    }
    else Toast.makeText(this, "debe agregar una descripción",
Toast.LENGTH_SHORT).show();
    }

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu1, menu);
    return super.onCreateOptionsMenu(menu);
}

@Override
public boolean onOptionsItemSelected(@NonNull MenuItem item) {
    switch (item.getItemId()){
        case R.id.icon_save:{
            registrarpedido();
            break;
        }
        case R.id.icon_update:{
            actualizarPedido();
            break;
        }
        case R.id.icon_add:{
            //Limpiar controles
            Toast.makeText(this, "Limpiar controles", Toast.LENGTH_SHORT).show();
            break;
        }
        case R.id.icon_list:{
            Toast.makeText(this, "listado ", Toast.LENGTH_SHORT).show();
            break;
        }
    }
    return true;
}

public void actualizarPedido(){
    Pedido p = new Pedido();
    p.setIdpedido(ped_sel.getIdpedido());
    p.setDescripcion(edtdetalle.getText().toString());
    p.setDireccion(edtdireccion.getText().toString());
    p.setEntregar(edtentregar.getText().toString());
    p.setDistrito(spdistrito.getSelectedItem().toString());
    p.setTipoped(sptipo.getSelectedItem().toString());
    mDataBase.child("Pedido").child(p.getIdpedido()).setValue(p);
    Toast.makeText(this, "actualizado", Toast.LENGTH_SHORT).show();
}
}
```