



# Projecte Patata

Dana Gomez Balada

11/02/2021  
DAM2 M8

## Índex

1.	Funcionament de l'aplicació .....	2
2.	Codi rellevant .....	3
2.1.	Base de dades .....	3
2.2.	Llista de patates .....	4
2.3.	Floating Action Buttons .....	5
2.4.	Botó afegir .....	6
2.4.1.	Actualització .....	7
2.5.	Botó cerca .....	8
2.6.	Fragment de resultat de cerca .....	9
2.7.	Splash Activity .....	11
3.	Problemes trobats .....	11

\*Els últims canvis surten en aquest color.

## 1. Funcionament de l'aplicació

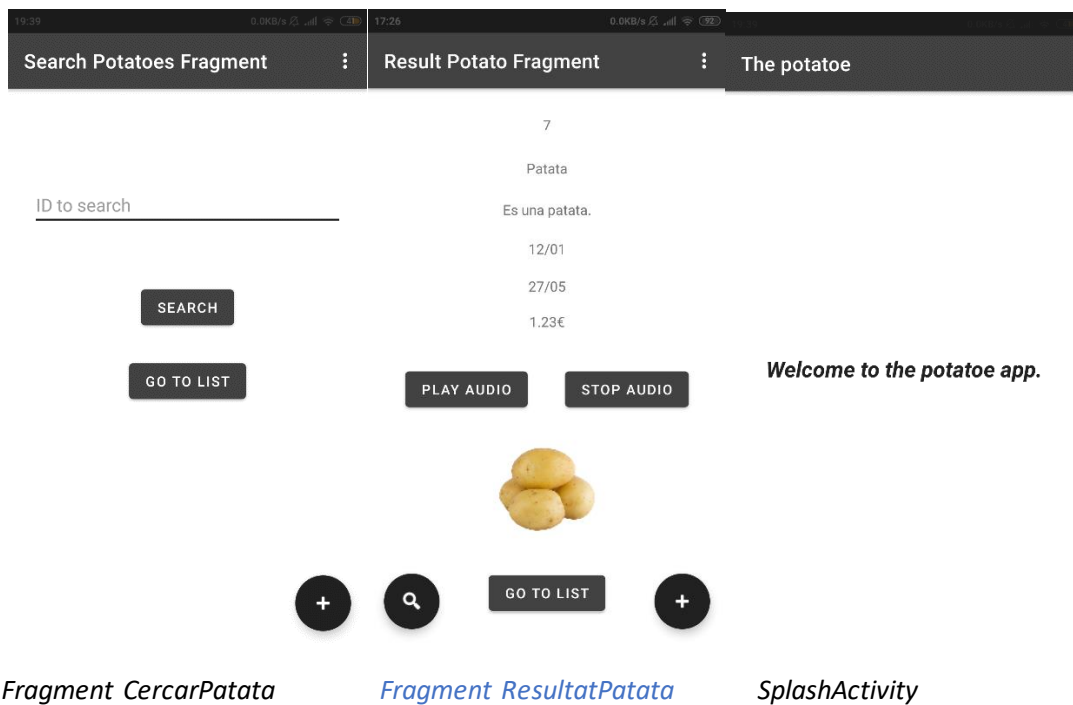
Un cop obrim l'aplicació, immediatament sortirà un missatge de benvinguda durant tres segons, i després la pantalla per fer login. Podem posar qualsevol usuari i la contrasenya ha de ser de més de 8 caràcters.

Un cop hem fet login, l'aplicació principal es carrega. Veurem una llista de totes les patates que tenim a la base de dades actualment, i podem desplaçar-nos avall per veure més. A les cantonades de la pantalla tenim dos botons, un per cercar i l'altre per afegir.

Si li donem al botó per cercar ens sortirà una nova pantalla on tindrem una caixa de text i dos botons, un per fer la cerca i l'altre per tornar. Hem d'escriure l'ID de la patata que volem buscar i, si en troba, ens portarà a una altra pantalla on mostrarà tota la informació d'aquesta patata en detall ([així com dos botons per l'àudio i un espai per la imatge](#)). Si deixem la caixa en blanc o no troba l'ID, sortirà un missatge informant de l'error.

Si li donem al botó per afegir, sortirà una nova pantalla on hi haurà múltiples caixes de text per escriure la informació rellevant, [així com dos botons per importar àudio i/o una imatge](#). També tindrem dos botons per afegir o tornar enrere. Si l'ID escrit no existeix ja, s'afegeix la nova patata i surt un missatge de confirmació. Si ja existeix o deixem algun camp en blanc, sortirà un missatge indicant l'error.

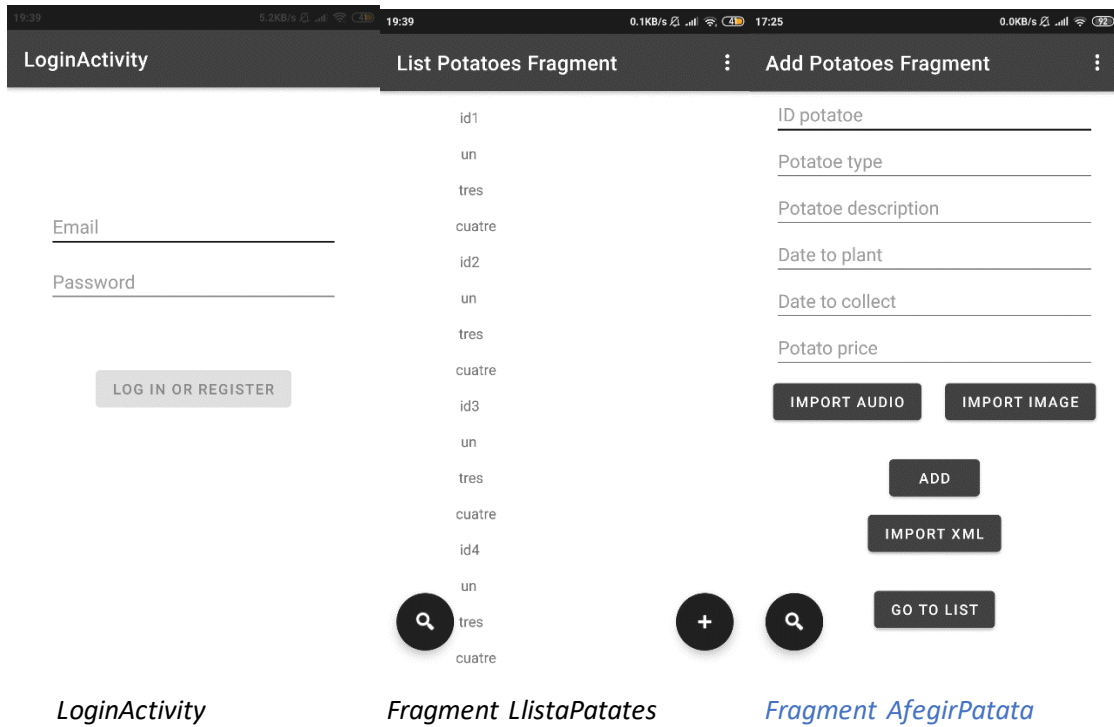
L'aplicació té una icona, nom i colors personalitzats, té una pantalla de benvinguda que dura 3 segons, cada fragment té el seu títol corresponent a la barra superior i té tres idiomes inclosos (Català, Anglès i Castellà).



Fragment CercarPatata

Fragment ResultatPatata

SplashActivity



## 2. Codi rellevant

### 2.1. Base de dades

L'aplicació conté una base de dades interna on guarda i recull tota la informació pertinent a les patates. En carregar l'aplicació, es connecta a aquesta base de dades per poder mostrar la llista inicial:

```
this.baseDades = this.openOrCreateDatabase( name: "patata", MODE_PRIVATE, factory: null);
String query = "SELECT * FROM patates";
Cursor resultat = this.baseDades.rawQuery(query, selectionArgs: null);
resultat.moveToLast();
String tipus = resultat.getString( columnIndex: 1);
```

Codi de connexió a la **BBDD**.

## 2.2. Llista de patates

Per tal de mostrar la llista de patates, fem servir un RecyclerView, que ens permet definir com volem mostrar les dades i reutilitzar aquesta organització per cada fila de la BBDD. En el fragment on tenim la llista, hem de inicialitzar el RecyclerView (a través de la classe RecyclerView que hem d'afegir, on farem ús del layout que hem creat per afegir les dades correctes).

```
// Set the adapter
if (view.findViewById(R.id.llista_patates) instanceof RecyclerView) {
    Context context = view.getContext();
    RecyclerView recyclerView = (RecyclerView) view.findViewById(R.id.llista_patates);
    if (mColumnCount <= 1) {
        recyclerView.setLayoutManager(new LinearLayoutManager(context));
    } else {
        recyclerView.setLayoutManager(new GridLayoutManager(context, mColumnCount));
    }
    this.mAdapter = new RecyclerViewPatata(this.patates, mListener);
    recyclerView.setAdapter(this.mAdapter);
}
```

Codi de creació del **RecyclerView**.

Hem de tenir en compte que, per a que es mostri la llista, hem d'afegir el RecyclerView en el layout del fragment de la llista:

```
<androidx.recyclerview.widget.RecyclerView
    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:id="@+id/llista_patates"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginLeft="16dp"
    android:layout_marginRight="16dp"
    app:layoutManager="LinearLayoutManager"
    tools:context=".LlistaPatatesFragment"
    tools:listitem="@layout/llista_patates" />
```

RecyclerView en llista\_patates\_fragment.

Per carregar les dades, ens connectem a la BBDD i creem instàncies de tipus *Patata*, que després són afegides al RecyclerView.

```
private void carregarPatates() {
    this.baseDades = getActivity().openOrCreateDatabase( name: "patata", MODE_PRIVATE, factory: null);
    String query = "SELECT * FROM patates";
    Cursor resultat = this.baseDades.rawQuery(query, selectionArgs: null);
    if (resultat != null){
        while (resultat.moveToNext()){
            String id = resultat.getString( columnIndex: 0);
            String tipus = resultat.getString( columnIndex: 1);
            String descripcio = resultat.getString( columnIndex: 2);
            String sembrar = resultat.getString( columnIndex: 3);
            String recollir = resultat.getString( columnIndex: 4);
            String preu = resultat.getString( columnIndex: 5);
            patates.add(new Patata(id,tipus,descripcio,sembrar,recollir,preu));
        }
    }
}
```

Codi creació classe *Patata*.

### 2.3. Floating Action Buttons

Els floating action buttons que tenim a les cantonades ens traslladen de fragment a fragment. Quan estem dins del fragment de cerca, el FAB de cerca desapareix, i el mateix succeeix quan estem al fragment d'afegir amb el FAB d'afegir.

```
add.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        NavHostFragment.findNavController( fragment: LlistaPatatesFragment.this)
            .navigate(R.id.action_LlistaPatates_to_AfegirPatata);
    }
});

search.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        NavHostFragment.findNavController( fragment: LlistaPatatesFragment.this)
            .navigate(R.id.action_LlistaPatates_to_CercaPatates);
    }
});
```

Codi dels **FAB**.

## 2.4. Botó afegir

El botó d'afegir (dins del fragment d'afegir) comprova que cap dels camps són buits (si ho són, surt un error), es connecta a la BBDD i comprova que l'id escrit no existeix (si existeix, surt un error). Després, fa un insert a la BBDD amb la informació dels camps de text (si hi ha algun problema, surt un error amb el missatge corresponent).

```
if (!textId.getText().toString().equals("") && !textTipus.getText().toString().equals("") && !textDesc.getText().toString().equals("") && !textSembrar.getText().toString().equals("") && !textRecollir.getText().toString().equals("") && !textPreu.getText().toString().equals("") && !rutaAudio.equals("") && !rutaImatge.equals("")) {
    baseDades = getActivity().openOrCreateDatabase("patata", MODE_PRIVATE, null);
    String query = "SELECT * FROM patates";
    Cursor resultat = baseDades.rawQuery(query, selectionArgs: null);
    Boolean esPot = true;
    String id = textId.getText().toString();
    String tipus = textTipus.getText().toString();
    String desc = textDesc.getText().toString();
    String sembrar = textSembrar.getText().toString();
    String recollir = textRecollir.getText().toString();
    String preu = textPreu.getText().toString();
    String audio = rutaAudio;
    String imatge = rutaImatge;
    if (resultat != null) {
        while (resultat.moveToNext()) {
            if (resultat.getString(columnIndex 0).equals(id)) {
                esPot = false;
            }
        }
    }
    if (esPot) {
        String sqlQuery = "INSERT INTO patates (id, tipus, descripcio, sembrar, recollir, preu, audio, imatge) " +
            "VALUES (" + id + ", " + tipus + ", " + desc + ", " + sembrar + ", " + recollir + ", " + preu + ", " + audio + ", " + imatge + ")";
        try {
            baseDades.execSQL(sqlQuery);
            Toast.makeText(getActivity().getApplicationContext(), "Afegit correctament", Toast.LENGTH_LONG).show();
        } catch (SQLException e) {
            Toast.makeText(getActivity().getApplicationContext(), e.getMessage(), Toast.LENGTH_LONG).show();
        }
    } else {
        Toast.makeText(getActivity().getApplicationContext(), "L'ID no es pot repetir!", Toast.LENGTH_LONG).show();
    }
} else {
    Toast.makeText(getActivity().getApplicationContext(), "No es poden deixar un o mes camps en blanc.", Toast.LENGTH_LONG).show();
}
```

Codi botó *Afegir*.

#### 2.4.1. Actualització

Els botons per importar àudio i imatge permeten escollir un fitxer del tipus corresponent del nostre dispositiu. Un cop escollit un fitxer, l'activity result s'encarrega d'obtenir la ruta i emmagatzemar-ho en una variable (que serà utilitzada al afegir la patata a la BBDD).

```
public void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    Uri seleccio;
    Cursor cursor;
    int indexColumna;
    switch (requestCode) {
        case 0:
            if (data!=null){
                seleccio = data.getData();
                String[] columna = {MediaStore.Audio.Media.DATA};
                cursor = getActivity().getContentResolver().query(seleccio, columna, selection: null, selectionArgs: null, sortOrder: null);
                cursor.moveToFirst();
                indexColumna = cursor.getColumnIndex(columna[0]);
                rutaAudio = cursor.getString(indexColumna);
                cursor.close();
            }
        case 1:
            if (data!=null){
                seleccio = data.getData();
                String[] columna2 = {MediaStore.Images.Media.DATA};
                cursor = getActivity().getContentResolver().query(seleccio, columna2, selection: null, selectionArgs: null, sortOrder: null);
                cursor.moveToFirst();
                indexColumna = cursor.getColumnIndex(columna2[0]);
                rutaImatge = cursor.getString(indexColumna);
                cursor.close();
            }
        default: break;
    }
}
```

Codi per obtenir la ruta del fitxer.



## 2.5. Botó cerca

El botó de cerca (dins del fragment de cerca) comprova que el camp de text no sigui buit (i mostra un missatge d'error si ho és), es connecta a la BBDD i comprova que l'ID introduït existeix (si no existeix, surt un missatge indicant l'error). Després, crea un Bundle amb l'ID i ens porta al fragment del resultat.

```
view.findViewById(R.id.botoCerca).setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        if (!textId.getText().toString().equals("")){  
            baseDades = getActivity().openOrCreateDatabase( name: "patata", MODE_PRIVATE, factory: null);  
            String query = "SELECT * FROM patates";  
            Cursor resultat = baseDades.rawQuery(query, selectionArgs: null);  
            Boolean trobat = false;  
            String id = textId.getText().toString();  
            if (resultat != null){  
                while (resultat.moveToNext() && !trobat){  
                    if (resultat.getString( columnIndex: 0).equals(id)){  
                        trobat = true;  
                    }  
                }  
            }  
            if (trobat){  
                dades = new Bundle();  
                dades.putString("ID", id);  
                NavHostFragment.findNavController( fragment: CercaPatatesFragment.this)  
                    .navigate(R.id.action_CercaPatates_to_ResultatPatata, dades);  
            }else{  
                Toast.makeText(getActivity().getApplicationContext(), getString(R.string.idNoTrobat), Toast.LENGTH_LONG).show();  
            }  
        }else{  
            Toast.makeText(getActivity().getApplicationContext(), getString(R.string.campsBlanc), Toast.LENGTH_LONG).show();  
        }  
    }  
});
```

Codi botó **Cerca**.

## 2.6. Fragment de resultat de cerca

El fragment del resultat de la cerca rep la informació del Bundle i selecciona des de la BBDD la filera corresponent amb l'ID entrat. Un cop fet això, mostra les dades per pantalla. Amb un boolean, comprova que s'hagi trobat un resultat correcte. Per mostrar la imatge, guarda la ruta en una variable i comprova si la imatge existeix (si existeix, la mostra en el espai blanc i, si no existeix o no hi ha cap imatge, mostra un missatge i deixa l'espai en blanc). Hi ha dos botons per l'àudio, un per reproduir-lo (comprova si l'àudio existeix i comprova que no s'estigui reproduint cap àudio ja) i un per parar la reproducció (si no s'està reproduint no fa res).

```
TextView id = view.findViewById(R.id.id);
TextView tipus = view.findViewById(R.id.tipus);
TextView desc = view.findViewById(R.id.desc);
TextView semb = view.findViewById(R.id.semb);
TextView reco = view.findViewById(R.id.reco);
TextView preu = view.findViewById(R.id.preu);
String getId = "";
String audio="", imatge="";

if(dades != null) {
    getId = dades.getString( key: "ID");
    baseDades = getActivity().openOrCreateDatabase( name: "patata", MODE_PRIVATE, factory: null);
    String query = "SELECT * FROM patates WHERE id='"+getId+"'";
    Cursor resultat = baseDades.rawQuery(query, selectionArgs: null);
    if (resultat != null){
        resultat.moveToNext();
        id.setText(resultat.getString( columnIndex: 0));
        tipus.setText(resultat.getString( columnIndex: 1));
        desc.setText(resultat.getString( columnIndex: 2));
        semb.setText(resultat.getString( columnIndex: 3));
        reco.setText(resultat.getString( columnIndex: 4));
        preu.setText(resultat.getString( columnIndex: 5));
        audio = resultat.getString( columnIndex: 6);
        imatge = resultat.getString( columnIndex: 7);
        trobat = true;
    }
}
```

Codi del resultat de la cerca.

```
if (trobat && imatge!=null && !imatge.equals("")){
    try{
        this.image = (ImageView) view.findViewById(R.id.imatge);
        Bitmap bitmap = BitmapFactory.decodeFile(imatge);
        bitmap = Bitmap.createScaledBitmap(bitmap, dstWidth: 100, dstHeight: 100, filter: true);
        this.image.setImageBitmap(bitmap);
    }catch (NullPointerException e){
        Toast.makeText(getActivity().getApplicationContext(), "No s'ha trobat la imatge d'aquesta patata.", Toast.LENGTH_LONG).show();
    }
}
```

Codi per mostrar la imatge.

```
String finalAudio = audio;
view.findViewById(R.id.botoPlay).setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (dades!=null){
            if (!isPlaying){
                if (trobat && finalAudio!=null && !finalAudio.equals("")){
                    Uri path = Uri.fromFile(new File(finalAudio));
                    so = MediaPlayer.create(getActivity(), path);
                    try{
                        so.start();
                        Snackbar.make(view, text: "Play", Snackbar.LENGTH_SHORT)
                            .setAction(text: "Action", listener: null).show();
                        isPlaying = true;
                    }catch(NullPointerException e){
                        Toast.makeText(getActivity().getApplicationContext(), "No hi ha cap audio en aquesta patata!", Toast.LENGTH_LONG).show();
                    }
                }else{
                    Toast.makeText(getActivity().getApplicationContext(), "No hi ha cap audio en aquesta patata!", Toast.LENGTH_LONG).show();
                }
            }else{
                Toast.makeText(getActivity().getApplicationContext(), "Ja s'esta reproduint un audio!", Toast.LENGTH_LONG).show();
            }
        }
    }
});
```

Codi del botó per reproduir l'àudio.

```
view.findViewById(R.id.botoStop).setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (dades!=null && so!=null && isPlaying){
            if (trobat && finalAudio!=null && !finalAudio.equals("")){
                Uri path = Uri.fromFile(new File(finalAudio));
                so.stop();
                so.release();
                Snackbar.make(view, text: "Stop", Snackbar.LENGTH_SHORT)
                    .setAction(text: "Action", listener: null).show();
                isPlaying = false;
            }
        }
    }
});
```

Codi del botó per parar l'àudio.

## 2.7. Splash Activity

El Splash Activity es una pantalla amb un missatge de benvinguda que es mostra quan obrim l'aplicació (per tant és l'activity MAIN). Es mostra el missatge i amb un Handler fem un comptador de temps (en aquest cas 3 segons) i amb un Intent anem a la següent Activity (en aquest cas la LoginActivity).

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_spalsh);
    scheduleSplashScreen();
}

private void scheduleSplashScreen(){
    new Handler().postDelayed(new Runnable() {

        @Override
        public void run() {
            Intent i = new Intent( packageContext: SpalshActivity.this, LoginActivity.class);
            startActivity(i);
            finish();
        }
    }, delayMillis: 3*1000);
}
```

Codi del **Splash Activity**.

## 3. Problemes trobats

En un principi va ser difícil implementar la longitud mínima de la contrasenya i els botons FAB, però aquests problemes van ser solucionats ràpidament.

No vaig poder canviar el fons blanc de l'aplicació, i la pantalla SplashActivity té una barra superior.

No vaig poder afegir la funcionalitat de XML, i no vaig tenir suficient temps per arreglar els problemes relacionats amb això (per tant, no es pot importar XML).