Splash Screen

activity_splash.xml:

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:orientation="vertical"
   android:gravity="center"
   android:background="@color/black">

   <ProgressBar
       android:id="@+id/progressBar"
       style="?android:attr/progressBarStyle"
       android:layout_marginTop="20dp"
       android:layout_width="match_parent"
       android:layout_height="wrap_content" />

</LinearLayout>
```

SplashActivity:

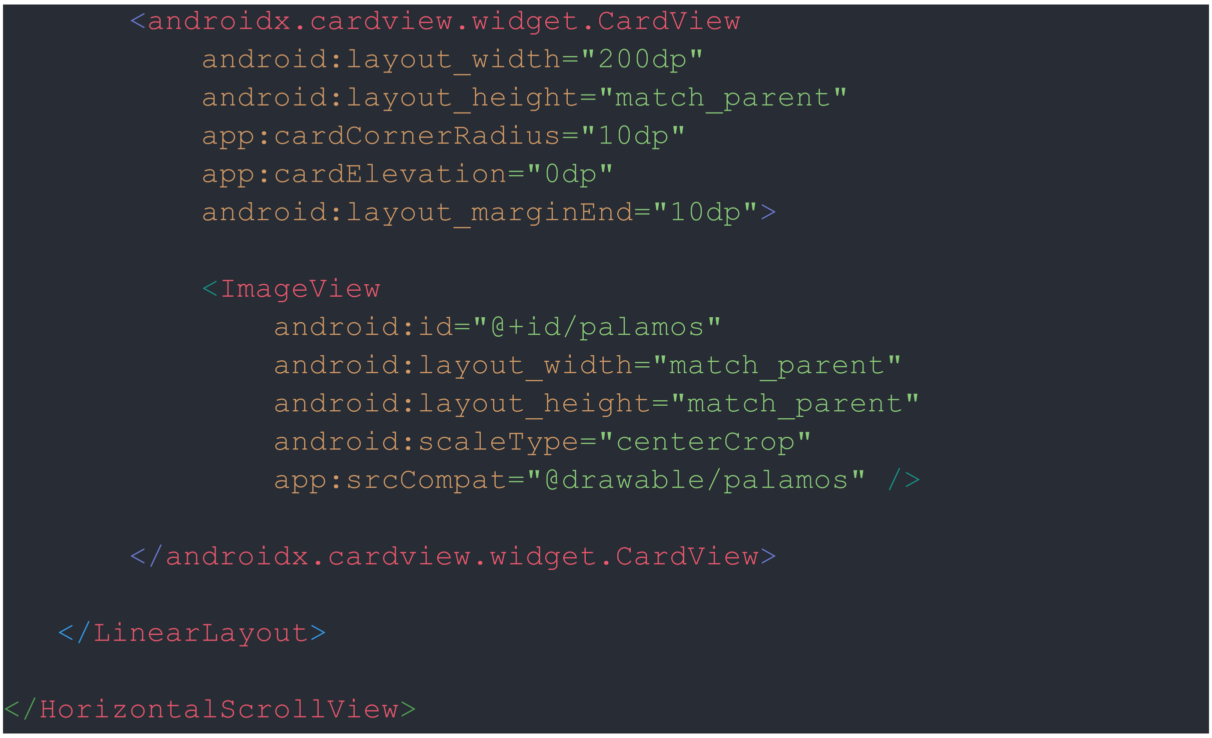
```
@SuppressLint("CustomSplashScreen")
class SplashActivity : AppCompatActivity() {

   override fun onCreate(savedInstanceState: Bundle?) {
       super.onCreate(savedInstanceState)
       setContentView(R.layout.activity_splash)

       Handler(Looper.getMainLooper()).postDelayed({
           startActivity(Intent(this, MainActivity::class.java))
           finish()
       }, 3000)
   }
}
```

AndroidManifest:

```
<activity
   android:name=".SplashActivity"
   android:exported="true">
   <intent-filter>
       <action android:name="android.intent.action.MAIN" />

       <category android:name="android.intent.category.LAUNCHER"
/>
   </intent-filter>
</activity>
```

Horizontal Scroll View

```
<HorizontalScrollView
   android:id="@+id/horizontal_scroll_view"
   android:layout_width="match_parent"
   android:layout_height="120dp"
   android:layout_marginTop="30dp"
   android:layout_marginHorizontal="20dp"
   app:layout_constraintStart_toStartOf="parent">

   <LinearLayout
       android:layout_width="wrap_content"
       android:layout_height="match_parent"
       android:orientation="horizontal">

       <androidx.cardview.widget.CardView
           android:layout_width="200dp"
           android:layout_height="match_parent"
           app:cardCornerRadius="10dp"
           app:cardElevation="0dp"
           android:layout_marginEnd="10dp">

           <ImageView
               android:id="@+id/estartit"
               android:layout_width="match_parent"
               android:layout_height="match_parent"
               android:scaleType="centerCrop"
               app:srcCompat="@drawable/estartit" />

       </androidx.cardview.widget.CardView>

       <androidx.cardview.widget.CardView
           android:layout_width="200dp"
           android:layout_height="match_parent"
           app:cardCornerRadius="10dp"
           app:cardElevation="0dp"
           android:layout_marginEnd="10dp">

           <ImageView
               android:id="@+id/pals"
               android:layout_width="match_parent"
               android:layout_height="match_parent"
               android:scaleType="centerCrop"
               app:srcCompat="@drawable/pals" />

       </androidx.cardview.widget.CardView>
```



```
        <androidx.cardview.widget.CardView
            android:layout_width="200dp"
            android:layout_height="match_parent"
            app:cardCornerRadius="10dp"
            app:cardElevation="0dp"
            android:layout_marginEnd="10dp">

            <ImageView
                android:id="@+id/palamos"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:scaleType="centerCrop"
                app:srcCompat="@drawable/palamos" />

        </androidx.cardview.widget.CardView>

    </LinearLayout>

</HorizontalScrollView>
```

View Pager 2

activity_main.xml

```
<androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="270dp"
    android:layout_marginTop="30dp"
    android:layout_marginHorizontal="20dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:cardCornerRadius="10sp"
    app:cardElevation="0dp">

    <androidx.viewpager2.widget.ViewPager2
        android:id="@+id/slider"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</androidx.cardview.widget.CardView>
```

slider.xml

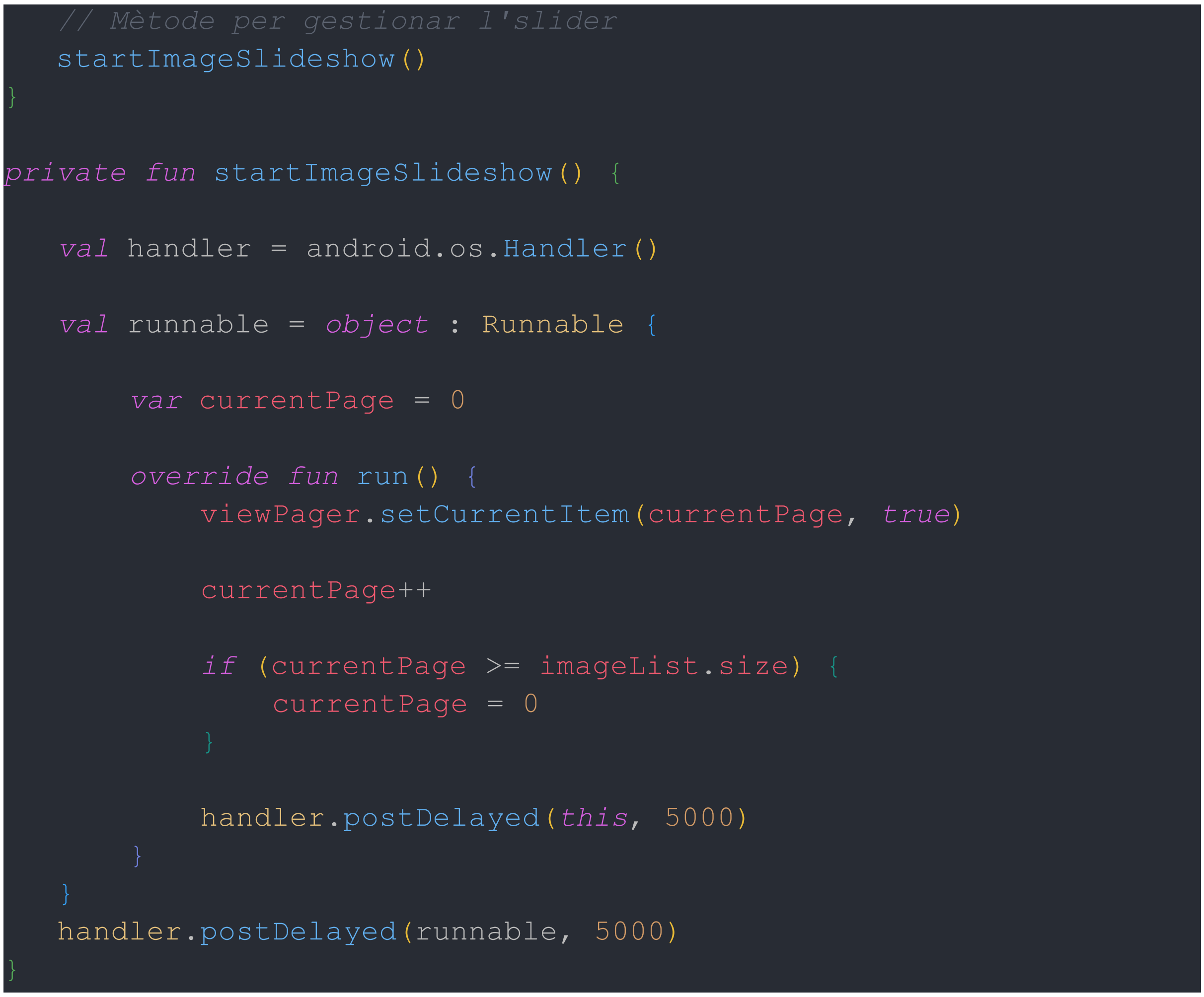
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent">


    <ImageView
        android:id="@+id/imageView"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:contentDescription="sliderImage"
        android:scaleType="fitXY"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"/>

</androidx.constraintlayout.widget.ConstraintLayout>
```

SliderAdapter.kt

```
class SliderAdapter(private val context: Context, private var
imageList: List<Int>) :
RecyclerView.Adapter<SliderAdapter.SliderViewHolder>() {


   class SliderViewHolder(itemView: View) :
RecyclerView.ViewHolder(itemView) {
       val imageView: ImageView =
itemView.findViewById(R.id.imageView)
   }

   override fun getItemCount(): Int = imageList.size

   override fun onCreateViewHolder(parent: ViewGroup, viewType:
Int): SliderViewHolder {
       val view =
LayoutInflater.from(context).inflate(R.layout.slider, parent,
false)
       return SliderViewHolder(view)
   }

   override fun onBindViewHolder(holder: SliderViewHolder,
position: Int) {
       holder.imageView.setImageResource(imageList[position])
   }
}
```

MainActivity

```
lateinit var viewPager: ViewPager2
lateinit var sliderAdapter: SliderAdapter

val imageList = listOf(
   R.drawable.estartit2,
   R.drawable.estartit3,
   R.drawable.estartit4,
   R.drawable.estartit5
)

override fun onCreate(savedInstanceState: Bundle?) {
   super.onCreate(savedInstanceState)
   setContentView(R.layout.activity_main)

   // Inicialitzar slider
   viewPager = findViewById(R.id.slider)
   sliderAdapter = SliderAdapter(this, imageList)
   viewPager.adapter = sliderAdapter
```



```kotlin
    // Mètode per gestionar l'slider
    startImageSlideshow()
}

private fun startImageSlideshow() {

    val handler = android.os.Handler()

    val runnable = object : Runnable {

        var currentPage = 0

        override fun run() {
            viewPager.setCurrentItem(currentPage, true)

            currentPage++

            if (currentPage >= imageList.size) {
                currentPage = 0
            }

            handler.postDelayed(this, 5000)
        }
    }
    handler.postDelayed(runnable, 5000)
}
```

Recycle View

activity_recycle.xml

```
<androidx.recyclerview.widget.RecyclerView
   android:id="@+id/recyclerView"
   android:layout_width="match_parent"
   android:layout_height="400sp"
   app:layout_constraintBottom_toBottomOf="parent"
   app:layout_constraintEnd_toEndOf="parent"
   app:layout_constraintStart_toStartOf="parent"
   app:layout_constraintTop_toTopOf="parent" />
```

items_user.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:layout_marginTop="10sp"
   android:orientation="horizontal"
   android:gravity="center_vertical">

   <ImageView
       android:id="@+id/item_image"
       android:layout_width="60sp"
       android:layout_height="60sp" />

   <TextView
       android:id="@+id/item_name"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:text="User Name"
       android:textSize="18sp"
       android:layout_marginStart="25dp" />

</LinearLayout>
```

UserListClass

```
data class UserListClass(val name: String, val imageResourceId:
Int)
```

UserListAdapter

```
class UserListAdapter(private val userList: List<UserListClass>) :
RecyclerView.Adapter<UserListAdapter.UserViewHolder>() {

    override fun onCreateViewHolder(parent: ViewGroup, viewType:
Int): UserViewHolder {
        val view =
LayoutInflater.from(parent.context).inflate(R.layout.item_user,
parent, false)
        return UserViewHolder(view)
    }

    override fun onBindViewHolder(holder: UserViewHolder, position:
Int) {
        val currentUser = userList[position]
        holder.nameTextView.text = currentUser.name

holder.imageView.setImageResource(currentUser.imageResourceId)
    }

    override fun getItemCount(): Int = userList.size

    class UserViewHolder(itemView: View) :
RecyclerView.ViewHolder(itemView) {
        val imageView: ImageView =
itemView.findViewById(R.id.item_image)
        val nameTextView: TextView =
itemView.findViewById(R.id.item_name)
    }
}
```

RecycleActivity

```
class SecondActivity : AppCompatActivity() {

    private lateinit var recyclerView: RecyclerView
    private lateinit var userAdapter: UserListAdapter
    private lateinit var userList: MutableList<UserListClass>

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_second)

        recyclerView = findViewById(R.id.recyclerView)
        recyclerView.layoutManager = LinearLayoutManager(this)

        userList = mutableListOf(
            UserListClass("John Doe", R.drawable.user),
            UserListClass("Jane Smith", R.drawable.building)
        )

        userAdapter = UserListAdapter(userList)
        recyclerView.adapter = userAdapter
    }
}
```

Bottom Bar

activity_main.xml

```
<com.google.android.material.bottomappbar.BottomAppBar
   android:id="@+id/bottomAppBar"
   android:layout_width="match_parent"
   android:layout_height="70dp"
   app:fabAlignmentMode="center"
   app:fabCradleMargin="10dp"
   app:fabCradleRoundedCornerRadius="10dp"
   app:fabCradleVerticalOffset="10dp"
   app:layout_constraintBottom_toBottomOf="parent"
   app:layout_constraintEnd_toEndOf="parent"
   app:layout_constraintStart_toStartOf="parent"
   tools:ignore="BottomAppBar">

   <LinearLayout
       android:layout_width="match_parent"
       android:layout_height="wrap_content"
       android:gravity="center"
       android:orientation="horizontal"
       android:weightSum="2">

       <ImageView
           android:id="@+id/add"
           android:layout_width="0dp"
           android:layout_height="35dp"
           android:layout_gravity="center"
           android:layout_weight="1"
           app:srcCompat="@drawable/plus" />

       <ImageView
           android:id="@+id/home"
           android:layout_width="0dp"
           android:layout_height="50dp"
           android:layout_gravity="center"
           android:layout_weight="1"
           app:srcCompat="@drawable/home" />

   </LinearLayout>

</com.google.android.material.bottomappbar.BottomAppBar>
```

Map

Project Build Gradle:

```
id("com.google.android.libraries.mapsplatform.secrets-gradle-plugi
n") version "2.0.1" apply false
```

Module Build Gradle:

```
id("com.google.android.libraries.mapsplatform.secrets-gradle-plugi
n")
```

```
implementation("com.google.android.gms:play-services-maps:19.0.0")
```

AndroidManifest.xml

```
<uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.INTERNET" />
```

```
<meta-data
   android:name="com.google.android.geo.API_KEY"
   android:value="AIzaSyBB0KizOiZqjB90xyrIC2tiJTsvjYQIVx4" />
```

activity_main.xml

```
<com.google.android.gms.maps.MapView
   android:id="@+id/mapView"
   android:layout_width="match_parent"
   android:layout_height="500dp"
   android:layout_marginTop="30dp"
   app:layout_constraintBottom_toBottomOf="parent"
   app:layout_constraintEnd_toEndOf="parent"
   app:layout_constraintStart_toStartOf="parent"
   app:layout_constraintTop_toTopOf="parent" />
```

MainActivity.kt

```
class MainActivity : AppCompatActivity(), OnMapReadyCallback {

    private lateinit var mMap: GoogleMap
    private lateinit var mapView: MapView

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        mapView = findViewById(R.id.mapView)

        mapView.onCreate(savedInstanceState)
        mapView.getMapAsync(this)
    }

    override fun onMapReady(googleMap: GoogleMap) {
        mMap = googleMap

val estartit = LatLng(42.0512, 3.1895)
mMap.addMarker(MarkerOptions().position(estartit).title("Marker in
Estartit"))
mMap.moveCamera(CameraUpdateFactory.newLatLng(estartit))
    }
}
```

Action Bar Menu

- Crear menú en el layout
- menu.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/spinner"
        android:title="Spinner"/>

    <item
        android:id="@+id/music"
        android:title="Music"/>

</menu>
```

MainActivity.kt

```kotlin
override fun onCreateOptionsMenu(menu: Menu?): Boolean {
    menuInflater.inflate(R.menu.main_menu, menu)
    return true
}

override fun onOptionsItemSelected(item: MenuItem): Boolean {
    return when (item.itemId) {
        R.id.spinner -> {
            val intent = Intent(this, MainActivity::class.java)
            startActivity(intent)
            true
        }
        R.id.music -> {
            val intent = Intent(this, ActivityMusic::class.java)
            startActivity(intent)
            true
        }
        else -> super.onOptionsItemSelected(item)
    }
```

Intent

Enviar dades

```
val intent = Intent(this, SecondActivity::class.java)

intent.putExtra("STRING_KEY", "Hola, aquesta és una cadena!")
intent.putExtra("NUMBER_KEY", 12345)
startActivity(intent)
```

Rebre dades

```
val stringData = intent.getStringExtra("STRING_KEY")
val numberData = intent.getIntExtra("NUMBER_KEY", 0)

val textView = findViewById<TextView>(R.id.textViewResult)
textView.text = "Rebut: $stringData i Número: $numberData"
numberText.text = numberValue.toString()
```

Shared Preferences

Guardar dades a les shared

```kotlin
val sharedPreferences = getSharedPreferences("sharedPref", Context.MODE_PRIVATE)
val editor = sharedPreferences.edit()

editor.putString("STRING_KEY", "Hola SharedPreferences!")
editor.putInt("NUMBER_KEY", 123)
editor.putBoolean("BOOLEAN_KEY", true)

editor.apply()

val intent = Intent(this, SecondActivity::class.java)
startActivity(intent)
```

Recuperar

```kotlin
val sharedPreferences = getSharedPreferences("sharedPref", Context.MODE_PRIVATE)

// Recuperar dades
val savedString = sharedPreferences.getString("STRING_KEY", "Cap text trobat")
val savedNumber = sharedPreferences.getInt("NUMBER_KEY", 0)
val savedBoolean = sharedPreferences.getBoolean("BOOLEAN_KEY", false)

// Mostrar dades al layout
val textView = findViewById<TextView>(R.id.textViewResult)
textView.text = "String: $savedString\nNúmero: $savedNumber\nBooleà: $savedBoolean"
```

Array JSON a shredPrefs i recuperar per RecycleView

Enviar

```
val house = House(
   poblation = poblation.selectedItem.toString(),
   address = address.text.toString(),
   price = price.text.toString().toFloat(),
   lat = lat.text.toString().toFloat(),
   long = long.text.toString().toFloat(),
   imageName = imageNames.random()
)

val sharedPreferences = getSharedPreferences("houses",
Context.MODE_PRIVATE)
val editor = sharedPreferences.edit()

val gson = Gson()

// Obtenir la llista actual de cases des de SharedPreferences
val currentCasesJson = sharedPreferences.getString("houses_list",
"[]")
val caseListType = object : TypeToken<MutableList<House>>() {}.type
val caseList = gson.fromJson<MutableList<House>>(currentCasesJson,
caseListType)

caseList.add(house)

// Convertir a JSON
val newCasesJson = gson.toJson(caseList)

try {
   editor.putString("houses_list", newCasesJson)
   editor.apply()

   Toast.makeText(this, "Casa/apartament afegida correctament!!",
Toast.LENGTH_SHORT).show()

   val intent = Intent(this, RecycleActivity::class.java)
   startActivity(intent)

} catch (e: Exception) {
   Toast.makeText(this, "Error al afegir la casa/apartament",
Toast.LENGTH_SHORT).show()
}
```

Rebre

```
val sharedPreferences = getSharedPreferences("houses",
Context.MODE_PRIVATE)
val gson = Gson()
val currentHousesJson = sharedPreferences.getString("houses_list",
"[]")

// Definir el tipus de la llista
val houseListType = object : TypeToken<MutableList<House>>()
{}.type

// JSON a llista
val houseList =
gson.fromJson<MutableList<House>>(currentHousesJson,
houseListType)

// Convertir perque el adapter ho pugui llegir
val housesList = houseList.map { house ->
    HousesList(
        house.address,
        house.price,
        resources.getIdentifier(house.imageName, "drawable",
packageName),
        house.lat,
        house.long
    )
}.toMutableList()

// Configura el RecyclerView
recyclerView = findViewById(R.id.recyclerView)
recyclerView.layoutManager = LinearLayoutManager(this)
```