



---

# DOCUMENTACIÓN DE CÓDIGO DE APLICACIÓN MEMORAMA PARA 2DO PARCIAL

---

Gerardo Yael De León Medina



11 DE MAYO DEL 2023  
UNIVERSIDAD AUTÓNOMA DE AGUASCALIENTES  
Diseño de Móviles



Primero para la pantalla inicial, se usan 2 imágenes: Una para el fondo de inicio, y otra para el logo de la universidad. También se usan 2 TextView para el nombre del alumno y el título de la aplicación.

Después, se maneja un Button para acceder a la siguiente vista, que sería el juego como tal.

```

<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:gravity="center"
    android:orientation="vertical"
    android:background="@drawable/background"> <!-- Agregar una imagen de fondo -->

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="90dp"
        android:text="Gerardo Yael de León Medina \n                276698"
        android:textColor="@color/white"
        android:textSize="26sp"
        android:textStyle="bold|italic" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="32dp"
        android:text="Memorams"
        android:textColor="@color/white"
        android:textSize="36sp"
        android:textStyle="bold|italic" />

    <Button
        android:id="@+id/startButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="onClick"

```

Este es el XML que se usó para esta actividad, que sería el Main.

```

<ImageView
    android:id="@+id/imageView"
    android:layout_width="195dp"
    android:layout_height="103dp"
    app:srcCompat="@drawable/uua" />

<Space
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />

</LinearLayout>

```

Y de aquí se puede ver cómo tenemos el background y el logo de la uaa desde la carpeta de Drawable.

Después, en el código principal, tenemos el siguiente código que nos va a servir para cambiar de actividades.

```

package com.example.memorama

import ...

class MainActivity : AppCompatActivity() {

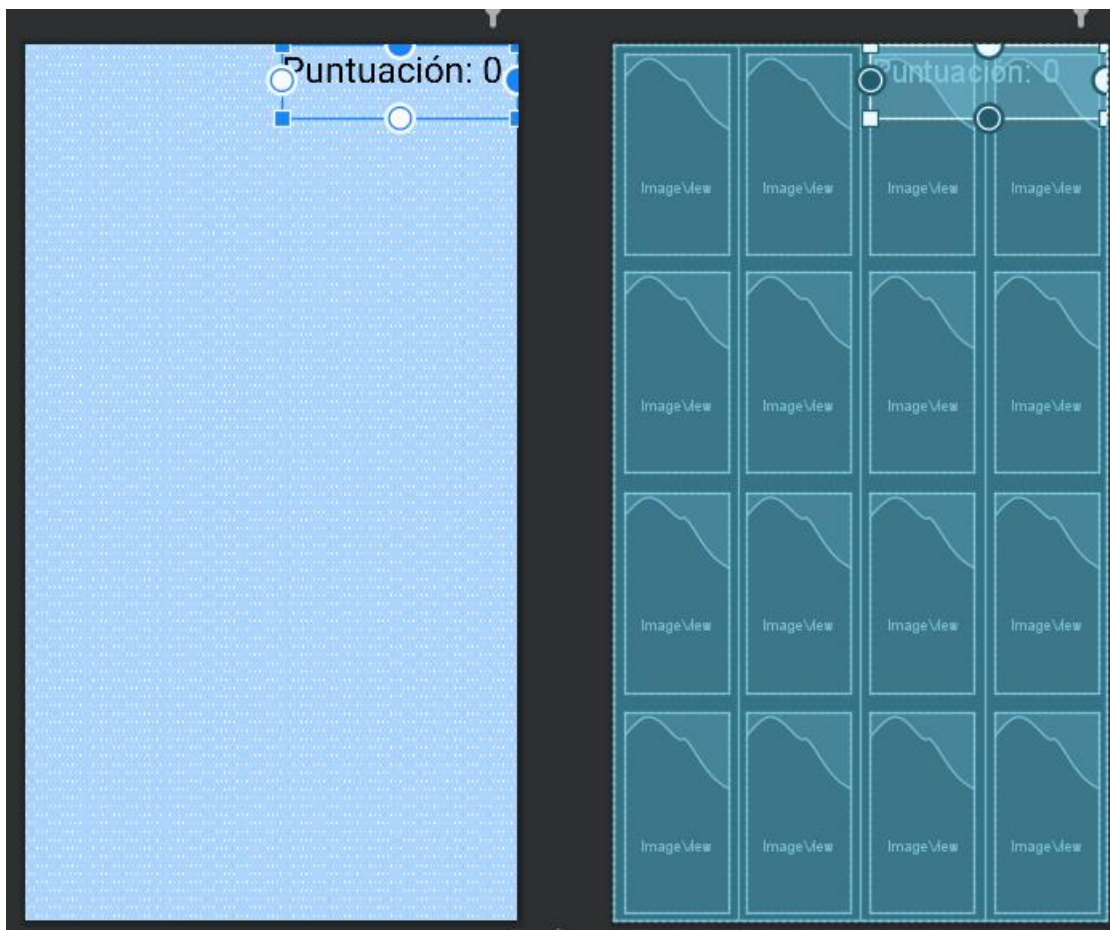
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val startButton = findViewById<Button>(R.id.startButton)
        startButton.setOnClickListener { it: View!
            val intent = Intent( packageContext: this, GameActivity::class.java)
            startActivity(intent)
        }
    }
}

```

Primero asignamos una variable al botón para poder acceder al método “intent” y al “startActivity”. Esto con el fin de que se coloque la actividad como pantalla principal.

Después, en la siguiente actividad, en el XML, tenemos la siguiente vista:



Para las imágenes, tenemos un “GridLayout” que nos va a servir para acomodar las imágenes en 4 filas y 4 columnas (ya que usamos 8 cartas, que serían 16 por adivinar). Y a parte a eso, tenemos un “TextView” con un parent hacia la derecha y arriba, con el fin de que cargue a parte y se acomode de forma estratégica.

```

<?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"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/background4"
    android:gravity="center"
    android:orientation="vertical">

    <TextView
        android:id="@+id/conteo"
        android:layout_width="197dp"
        android:layout_height="62dp"
        android:text="Puntuación: 0"
        android:textColor="#000000"
        android:textSize="30sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <GridLayout
        android:id="@+id/gridLayout"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:columnCount="4"
        android:rowCount="4">

        <ImageView
            android:layout_width="10dp"
            android:layout_height="15dp"
            android:layout_rowWeight="1"

```

Así se ven las etiquetas de la actividad (El imageView se repite 16 veces para las 16 imágenes)

```

<ImageView
    android:layout_width="10dp"
    android:layout_height="15dp"
    android:layout_rowWeight="1"
    android:layout_columnWeight="1"
    android:layout_margin="8dp"
    android:scaleType="centerInside" />

```

Luego, lo que se hace es cargar las 8 imágenes que se tienen en el Drawable, con el fin de usarlas cada que se seleccione una. Luego el arreglo se va a desordenar para que no queden acomodadas.

Luego se van a acomodar las imágenes en base al GridLayout por medio de un for, contando la cantidad de imágenes puestas dentro de este Layout. Y para cada carta acomodada, se va a asignar una imagen trasera en caso de que no se hayan volteado. Luego, cada que se haga click en alguna imagen, lo que va a pasar es que se va a llamar una función llamada “voltearCarta”, esto por medio de un “setOnClickListener”.

```

class GameActivity: AppCompatActivity() {
    private lateinit var cards: MutableList<Int>
    private lateinit var gridLayout: GridLayout
    private var conteo: Int = 0;
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_game)

        // Cargar el arreglo de las cartas con las imágenes
        cards = mutableListOf(
            R.drawable.imagen1,
            R.drawable.imagen2,
            R.drawable.imagen3,
            R.drawable.imagen4,
            R.drawable.imagen5,
            R.drawable.imagen6,
            R.drawable.imagen7,
            R.drawable.imagen8
        )
        cards.addAll(cards)

        // Shuffle the cards
        cards.shuffle()

        // Set up the card grid
        gridLayout = findViewById<GridLayout>(R.id.gridLayout)
        for (i in 0 until gridLayout.childCount) {
            val imageView = gridLayout.getChildAt(i) as ImageView
            imageView.setImageResource(R.drawable.card_back)
            imageView.setOnClickListener { voltearCarta(imageView, cards[i]) }
        }
    }
}

```

Después, dentro del método “voltearCarta” vamos a meter la lógica en donde digamos si es que la primera carta es igual a la segunda que se clickee.

Primero, se comprueba que la primera carta esté volteada (que lo hacemos con un null de una variable creada en este método), y lo mismo con la segunda. Después comprobamos con un if para ver si están volteadas, y en caso de que sean correctas, la primera carta y la segunda carta toman el valor de la primera carta volteada, y las variables vuelven a null.

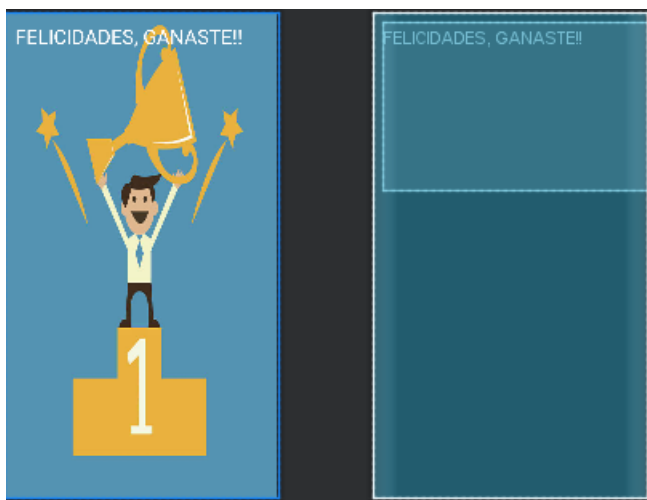
```
private fun voltearCarta(imageView: ImageView, card: Int) {
    if (firstCard == null) {
        firstCard = imageView
        imageView.setImageResource(card)
    } else if (secondCard == null && imageView != firstCard) {
        secondCard = imageView
        imageView.setImageResource(card)

        // Para checar si la primera carta es igual a la segunda seleccionada
        if (firstCard!!.drawable.constantState == secondCard!!.drawable.constantState) {
            firstCard = null
            secondCard = null
            this.conteo++
            val conteoTexto = findViewById<TextView>(R.id.conteo);
            conteoTexto.setText("Puntuación: " + this.conteo.toString());
        } else {
            Handler(Looper.getMainLooper()).postDelayed({
                firstCard!!.setImageResource(R.drawable.card_back)
                secondCard!!.setImageResource(R.drawable.card_back)
                firstCard = null
                secondCard = null
            }, delayMillis: 1000)
        }
    }
}
```

Y antes de esto, se define una variable global para el conteo de cartas adivinadas. Luego sólo se le asigna ese conteo al TextView por medio del id “conteo”.

Después, en caso de que estén incorrectas, por medio de las variables, los ImageViews tienen la imagen “card\_back” (la que tienen al cargar la actividad).

Luego, para comprobar si todas las cartas están volteadas, se comprueba en base de que no tengan la imagen “card\_back”, si ninguna la tiene, entonces nos lleva a un XML por medio del “Inflater” pasando el texto de Victoria y un pequeño ícono.



El tercer XML tiene lo siguiente:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/custom_toast_layout"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:background="@drawable/ganador"
    android:orientation="horizontal"
    android:padding="16dp">

    <TextView
        android:id="@+id/ganador"
        android:layout_width="411dp"
        android:layout_height="254dp"
        android:text="FELICIDADES, GANASTE!!"
        android:textColor="#FFFFFF"
        android:textSize="30sp" />
```

En el TextView tenemos lo que es el Inflate o paso de datos, y un background con una imagen de un ganador.

Luego, de vuelta en el "GameActivity" tenemos el toast que nos va a permitir ver el ícono de victoria por un pequeño tiempo, y luego desaparece, y tanto la puntuación como las cartas se voltean automáticamente.



```

if (allMatched) {
    this.conteo = 0;
    val inflater = inflater
    val layout = inflater.inflate(R.layout.ganador, root: null)

    val text = layout.findViewById<TextView>(R.id.ganador)
    text.text = "FELICIDADES, GANASTE!"

    val toast = Toast(applicationContext)
    toast.duration = Toast.LENGTH_SHORT
    toast.view = layout
    toast.show()
    Handler(Looper.getMainLooper()).postDelayed({
        recreate()
    }, delayMillis: 2000)
}
}
}
}

```

Y así como se desarrolló esta aplicación.