

HU - 06 Personalización avatar

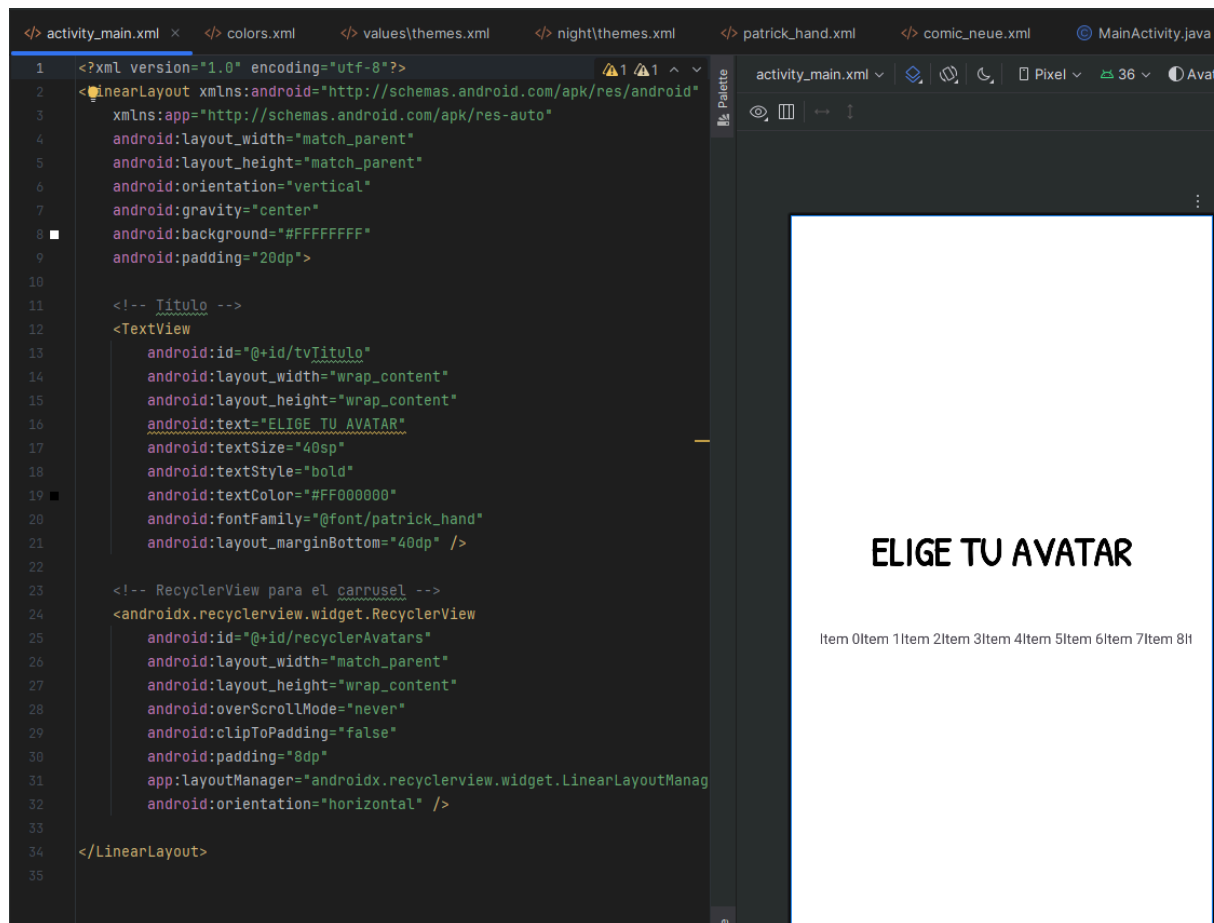
	Actividades
Act 1	Diseñar avatares
Act 2	Pantalla selección avatar
Act 3	Guardar avatar en perfil

Fecha estimada de realización				
L	M	M	J	V
22	23	24	25	26

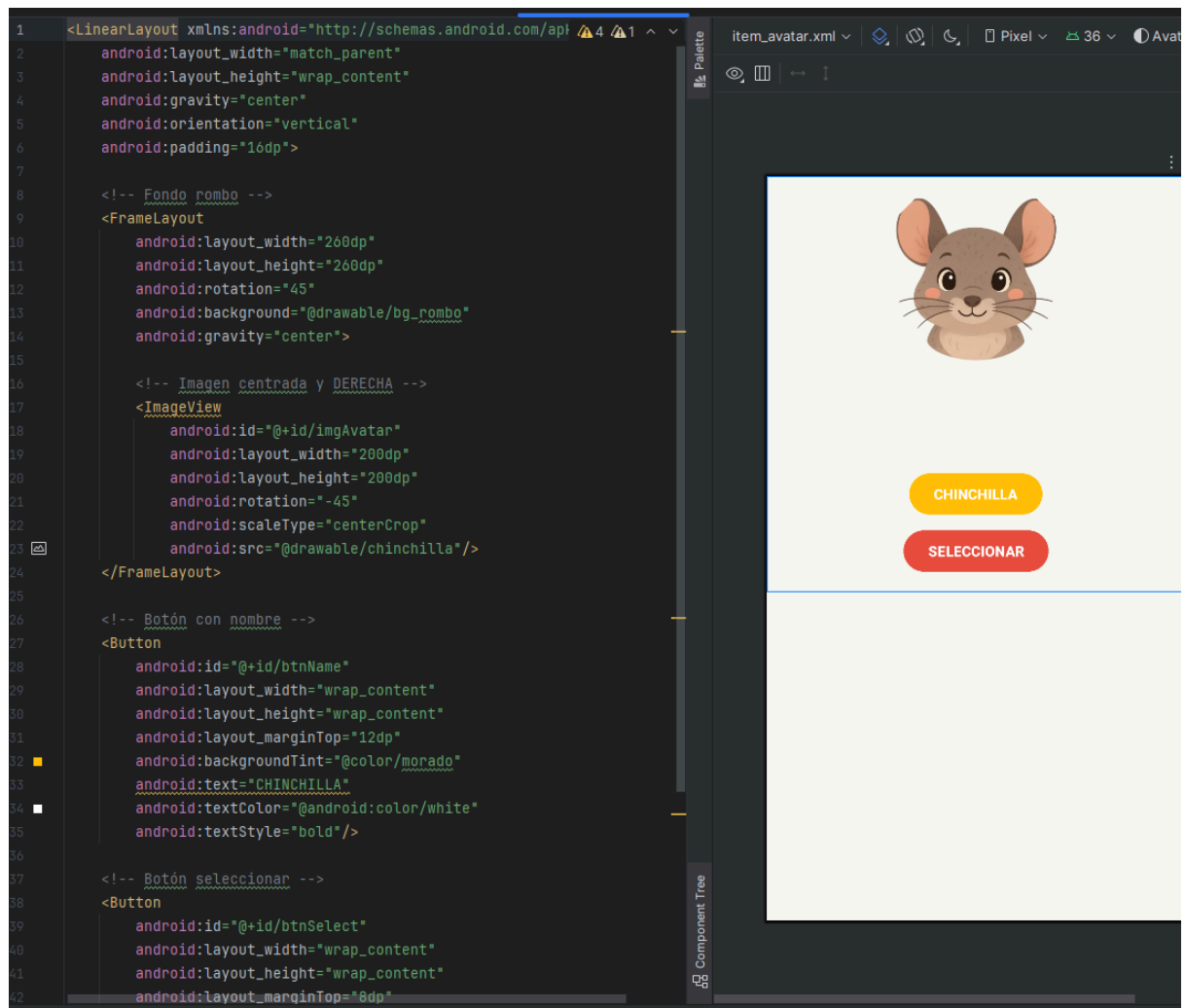
Fecha de desarrollo: 24 de septiembre

Tiempo utilizado para la elaboración: 2h

Para este día se realizó la elaboración y codificación de la pantalla principal de selección de avatar como se puede ver en la imagen, además que se pondrá cada uno de los avatares como opciones



luego se creo otra interfaz donde estará la imagen de cada animalito y su opción



Fecha de desarrollo: 25 de septiembre
Tiempo utilizado para la elaboración: 30min

Para este día se realizó las funcionalidades de cada interfaz tanto para la principal

```
© MainActivity.java x © AvatarItem.java </> activity_avatar_selector.xml build.gradle.kts (:app) </> item_avatar.xml </> bg_rombo.xml </> bg_a

1 package com.example.avatar;
2
3 import android.os.Bundle;
4 import android.widget.Toast;
5 import androidx.appcompat.app.AppCompatActivity;
6 import androidx.recyclerview.widget.LinearLayoutManager;
7 import androidx.recyclerview.widget.RecyclerView;
8
9 <?> public class MainActivity extends AppCompatActivity {
10
11     3 usages
12     private RecyclerView recyclerView;
13     2 usages
14     private AvatarAdapter adapter;
15
16     1 usage
17     private String[] nombres = {"Chinchilla", "Cóndor", "Gato", "Venado", "Vicuña", "Zorro"};
18     1 usage
19     private int[] imagenes = {
20         R.drawable.chinchilla,
21         R.drawable.condor,
22         R.drawable.gato,
23         R.drawable.venado,
24         R.drawable.vicuna,
25         R.drawable.zorro
26     };
27
28     @Override
29     protected void onCreate(Bundle savedInstanceState) {
30         super.onCreate(savedInstanceState);
31         setContentView(R.layout.activity_main);
32
33         recyclerView = findViewById(R.id.recyclerAvatars);
34         recyclerView.setLayoutManager(new LinearLayoutManager(context, LinearLayoutManager.HORIZONTAL, reverseLayout: false));
35
36         adapter = new AvatarAdapter(context, nombres, imagenes, new AvatarAdapter.OnAvatarSelectListener() {
37             1 usage
38             @Override
39             public void onAvatarSelected(String nombre) {
40                 Toast.makeText(context, MainActivity.this, "Seleccionaste: " + nombre, Toast.LENGTH_SHORT).show();
41             }
42         });
43     }
44 }
```

como también la parte interna de deslizar con sus botones de selección todo esto estara como AvatarAdapter

```
pill_background.xml button_background.xml circle_outer.xml circle_inner.xml circle_mask.xml bg_avatar.xml AvatarAdapter.java x
public class AvatarAdapter extends RecyclerView.Adapter<AvatarAdapter.AvatarViewHolder> {
    public AvatarViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View view = LayoutInflater.from(context).inflate(R.layout.item_avatar, parent, attachToRoot: false);
        return new AvatarViewHolder(view);
    }

    @Override
    public void onBindViewHolder(@NonNull AvatarViewHolder holder, int position) {
        holder.imgAvatar.setImageResource(images[position]);
        holder.btnName.setText(nombres[position]);

        // Botón seleccionar
        holder.btnSelect.setOnClickListener( View v -> {
            selectedPosition = position;
            notifyDataSetChanged();
            if (listener != null) {
                listener.onAvatarSelected(nombres[position]);
            }
        });

        // Marcar el avatar seleccionado con un borde especial
        if (selectedPosition == position) {
            holder.imgAvatar.setBackgroundResource(R.drawable.bg_avatar_circle_selected);
        } else {
            holder.imgAvatar.setBackgroundResource(R.drawable.bg_avatar_circle);
        }
    }

    @Override
    public int getItemCount() {
        return nombres.length;
    }

    4 usages
    public static class AvatarViewHolder extends RecyclerView.ViewHolder {
        4 usages
        ImageView imgAvatar;
        2 usages
        Button btnName, btnSelect;

        1 usage
        public AvatarViewHolder(@NonNull View itemView) {

```

y de esta manera ya tendríamos la funcionalidad de la selección del avatar que se vera asi, y que se verá como un carrusel de avatares para la comodidad del usuario

ELIGE TU AVATAR



Venado

SELECCIONAR

ELIGE TU AVATAR



Chinchilla

SELECCIONAR

Diagrama Burn Out Chart

Burn Out Chart

