**Enunciado del Examen: Desarrollo de una App Android con MVP, Retrofit, Descarga de Imágenes con Threads patrón Builder y Filtros con Spinner**

|  |
| --- |
| **Asignatura:** Programación Multimedia y Dispositivos Móviles **Curso:** 2024-2025 **Duración del examen:** 90 minutos **Formato del examen:** Individual **Recursos permitidos:** Instalador de PostgreSQL (se proporciona en un pen drive) y una plantilla de proyecto MVP incompleta. **Objetivo:** Comprobar la autoría del trabajo realizado y el aprendizaje de los conceptos impartidos. |

**Descripción del Proyecto**

Desarrollar una **aplicación Android** que muestre una lista de productos en un **RecyclerView** y permita filtrarlos mediante un **Spinner**. Los datos de los productos y las categorías deben almacenarse en una **base de datos PostgreSQL local** y ser consumidos a través de una **API REST con Retrofit**. Las imágenes de los productos deben descargarse utilizando **Threads** implementando el **patrón Builder**.

**Requerimientos del Examen**

1. **Base de Datos PostgreSQL (10 minutos)**
   * Crea una base de datos llamada tienda.
   * Crea las siguientes tablas en PostgreSQL:

|  |
| --- |
| CREATE TABLE categorias (  id SERIAL PRIMARY KEY,  nombre VARCHAR(50) NOT NULL  );  CREATE TABLE productos (  id SERIAL PRIMARY KEY,  nombre VARCHAR(100) NOT NULL,  descripcion TEXT,  precio DECIMAL(10, 2),  imagen\_url VARCHAR(255),  categoria\_id INTEGER REFERENCES categorias(id)  );  -- Insertar categorías  INSERT INTO categorias (nombre) VALUES  ('Instrumentos'),  ('Álbumes'),  ('Accesorios de Música');  -- Insertar productos  INSERT INTO productos (nombre, descripcion, precio, imagen\_url, categoria\_id) VALUES  ('Guitarra Eléctrica', 'Guitarra con cuerpo sólido y 6 cuerdas.', 399.99, 'https://example.com/guitarra.jpg', 1),  ('Piano Digital', 'Piano con 88 teclas y sonido de alta calidad.', 999.99, 'https://example.com/piano.jpg', 1),  ('Álbum Rock Clásico', 'Incluye 12 canciones icónicas de rock clásico.', 19.99, 'https://example.com/album\_rock.jpg', 2),  ('Auriculares Profesionales', 'Auriculares con calidad de estudio.', 129.99, 'https://example.com/auriculares.jpg', 3),  ('Cuerdas de Guitarra', 'Juego de cuerdas para guitarra acústica.', 9.99, 'https://example.com/cuerdas.jpg', 3); |
| CREATE TABLE categorias (  id SERIAL PRIMARY KEY,  nombre VARCHAR(50) NOT NULL  );  CREATE TABLE productos (  id SERIAL PRIMARY KEY,  nombre VARCHAR(100) NOT NULL,  descripcion TEXT,  precio DECIMAL(10, 2),  imagen\_url VARCHAR(255),  categoria\_id INTEGER REFERENCES categorias(id)  );  -- Insertar categorías  INSERT INTO categorias (nombre) VALUES  ('Ficción'),  ('No Ficción'),  ('Accesorios de Lectura');  -- Insertar productos  INSERT INTO productos (nombre, descripcion, precio, imagen\_url, categoria\_id) VALUES  ('Novela de Aventuras', 'Una historia emocionante llena de giros inesperados.', 14.99, 'https://example.com/novela.jpg', 1),  ('Biografía Inspiradora', 'La vida de un emprendedor visionario.', 19.99, 'https://example.com/biografia.jpg', 2),  ('Lámpara de Lectura', 'Lámpara portátil con luz ajustable.', 12.99, 'https://example.com/lampara.jpg', 3),  ('Marcadores de Páginas', 'Set de marcadores con diseños únicos.', 5.99, 'https://example.com/marcadores.jpg', 3),  ('Libro de Historia', 'Una mirada detallada a los eventos históricos.', 24CREATE TABLE categorias (  id SERIAL PRIMARY KEY,  nombre VARCHAR(50) NOT NULL  );  CREATE TABLE productos (  id SERIAL PRIMARY KEY,  nombre VARCHAR(100) NOT NULL,  descripcion TEXT,  precio DECIMAL(10, 2),  imagen\_url VARCHAR(255),  categoria\_id INTEGER REFERENCES categorias(id)  );  -- Insertar categorías  INSERT INTO categorias (nombre) VALUES  ('Consolas'),  ('Videojuegos'),  ('Accesorios Gaming');  -- Insertar productos  INSERT INTO productos (nombre, descripcion, precio, imagen\_url, categoria\_id) VALUES  ('Consola de Nueva Generación', 'Consola con gráficos avanzados y almacenamiento SSD.', 499.99, 'https://example.com/consola.jpg', 1),  ('Videojuego de Aventura', 'Explora mundos abiertos con gráficos de última generación.', 59.99, 'https://example.com/juego\_aventura.jpg', 2),  ('Teclado Mecánico', 'Teclado RGB con interruptores mecánicos.', 79.99, 'https://example.com/teclado.jpg', 3),  ('Silla Gaming', 'Silla ergonómica para largas sesiones de juego.', 199.99, 'https://example.com/silla.jpg', 3),  ('Auriculares Inalámbricos', 'Auriculares con micrófono para gaming.', 99.99, 'https://example.com/auriculares\_gaming.jpg', 3);.99, 'https://example.com/historia.jpg', 2); |
|  |

**API REST con node.js y express (20 minutos)**

* Desarrolla una API REST usando **Node.js con Express** para exponer los datos de los productos y las categorías.
  1. Crea tres endpoints:
     + GET /productos: Devuelve todos los productos.
     + GET /categorias: Devuelve todas las categorías.
     + GET /productos/categoria/:id : Filtrar productos por categoría.

Ayuda:

|  |
| --- |
| **app.get("/productos/categoria/:id", async (req, res)** => { const categoriaId = req.params.id; |

* Asegúrate de que los productos incluyan el id, nombre, descripcion, precio, imagen\_url y categoria\_id.
* En el filtro recibirá el id del Spinner. El usuario seleccionará una categoría pero mandaremos al endpoint el id.

ENTREGABLE:

Deja preparado los test en THUNDER de los 3 ENPOINTS y llama al profesor

|  |  |
| --- | --- |
| **APARTADO** | APTO/NO APTO |
| **1. Obtener todos los productos**: |  |
| **2. Obtener todas las categorías:** |  |
| **3. Filtrar productos por una categoría específica (por ejemplo, categoría con ID 1):** |  |

**Aplicación Android retrofit 2. (60 minutos)**

* Utiliza la **plantilla MVP proporcionada** y completa las partes faltantes.
* **RecyclerView**: Muestra la lista de productos con su **nombre**, **descripción**, **precio** y la **imagen** descargada desde la URL usando **Threads** con el **patrón Builder**.
  + En caso de no tener conexión a internet, utiliza la imagen del pen USB.
  + Deja preparado el código para cuando tengas INTERNET.
* **Spinner**: Carga las categorías desde la base de datos y permite filtrar los productos por categoría.

AYUDA:

|  |
| --- |
| spinnerCategorias = findViewById(R.id.spinnerCategorias);  apiService = RetrofitClient.getRetrofitInstance().create(ApiService.class);  private void cargarCategorias() {  Call<List<Categoria>> call = apiService.getCategorias();  call.enqueue(new Callback<List<Categoria>>() {  @Override  public void onResponse(Call<List<Categoria>> call, Response<List<Categoria>> response) {  if (response.isSuccessful() && response.body() != null) {  List<Categoria> listaCategorias = response.body();  ArrayAdapter<Categoria> adapter = new ArrayAdapter<>(  MainActivity.this,  android.R.layout.simple\_spinner\_item,  listaCategorias  );  adapter.setDropDownViewResource(android.R.layout.simple\_spinner\_dropdown\_item);  spinnerCategorias.setAdapter(adapter);  } else {  Toast.makeText(MainActivity.this, "Error al cargar categorías", Toast.LENGTH\_SHORT).show();  }  }  @Override  public void onFailure(Call<List<Categoria>> call, Throwable t) {  Toast.makeText(MainActivity.this, "Error: " + t.getMessage(), Toast.LENGTH\_SHORT).show();  }  });  }  } |

* **Retrofit**: Utiliza Retrofit para hacer llamadas a la API y obtener los productos y categorías.
* **Patrón Builder**: Implementa el patrón Builder para gestionar la descarga de imágenes de los productos en segundo plano CON THREADS.

AYUDA:

|  |
| --- |
| private void initSVTeamImageView() {  String myUrl = "https://lh4.googleusercontent.com/proxy/JG73x59mTNYlvZ5cQsk9mBag4NiNL\_O58q4YC0DtyiSsEw8W2iQcAZhYTyEPbfr1DM3CWbT-LgJ8T8QDy6tKjRGHODw4UDQgN9ZF-rta-ifXdWwCmqw";  SAMBA\_ANDROID.Builder builder = new SAMBA \_ANDROID.Builder(myUrl, imagen);  builder.build().load();  } |
| public class SAMBA\_ANDROID {  private String url;  private ImageView imageView;  public SAMBA\_ANDROID (Builder builder) {  this.url = builder.url;  this.imageView = builder.imageView;  }  static Bitmap bitmap;  public void load(){  new Thread(() ->{  try {  URL url = new URL("https://lh4.googleusercontent.com/proxy/JG73x59mTNYlvZ5cQsk9mBag4NiNL\_O58q4YC0DtyiSsEw8W2iQcAZhYTyEPbfr1DM3CWbT-LgJ8T8QDy6tKjRGHODw4UDQgN9ZF-rta-ifXdWwCmqw");  HttpURLConnection connection = (HttpURLConnection) url.openConnection();  connection.connect();  InputStream input = connection.getInputStream();  bitmap = BitmapFactory.decodeStream(input);  // Actualizar el ImageView en el hilo principal  imageView.post(() -> imageView.setImageBitmap(bitmap));  } catch (Exception e) {  throw new RuntimeException(e);  }  }).start();  }  public static class Builder{  private String url;  private ImageView imageView;  private boolean generateImage = false;  public Builder(String url, ImageView imageView){  this.url = url;  this.imageView = imageView;  }  public Builder generateImage() {  this.generateImage = true;  return this;  }  public SAMBA\_ANDROID build(){  return new SAMBA\_ANDROID (this);  }  }  } |

ENTREGABLE:

Casos de uso e imágenes con THREADS y llama al profesor:

|  |  |
| --- | --- |
| **APARTADO** | APTO/NO APTO |
| **1. Obtener todos los productos -> recyclerview** |  |
| **2. Obtener todas las categorías -> spinner** |  |
| **3. Filtrar productos por una categoría específica (por ejemplo, categoría con ID 1): -> filtro actualizando recyclerview** |  |