*Salvador Costilla Caballero*

*ESCOM / IPN*

*tarea 3*

# "Desarrollo de una Aplicación Móvil Nativa con Consumo de API REST"

**Ejercicio 1: Creación de un backend básico y conexión con Android}**

Primero se implementó un servicio REST básico usando Spring Boot, para eso se importó el proytecto generado por spring initializr a IntelliJ y posteriormente se modificó el pom.xml

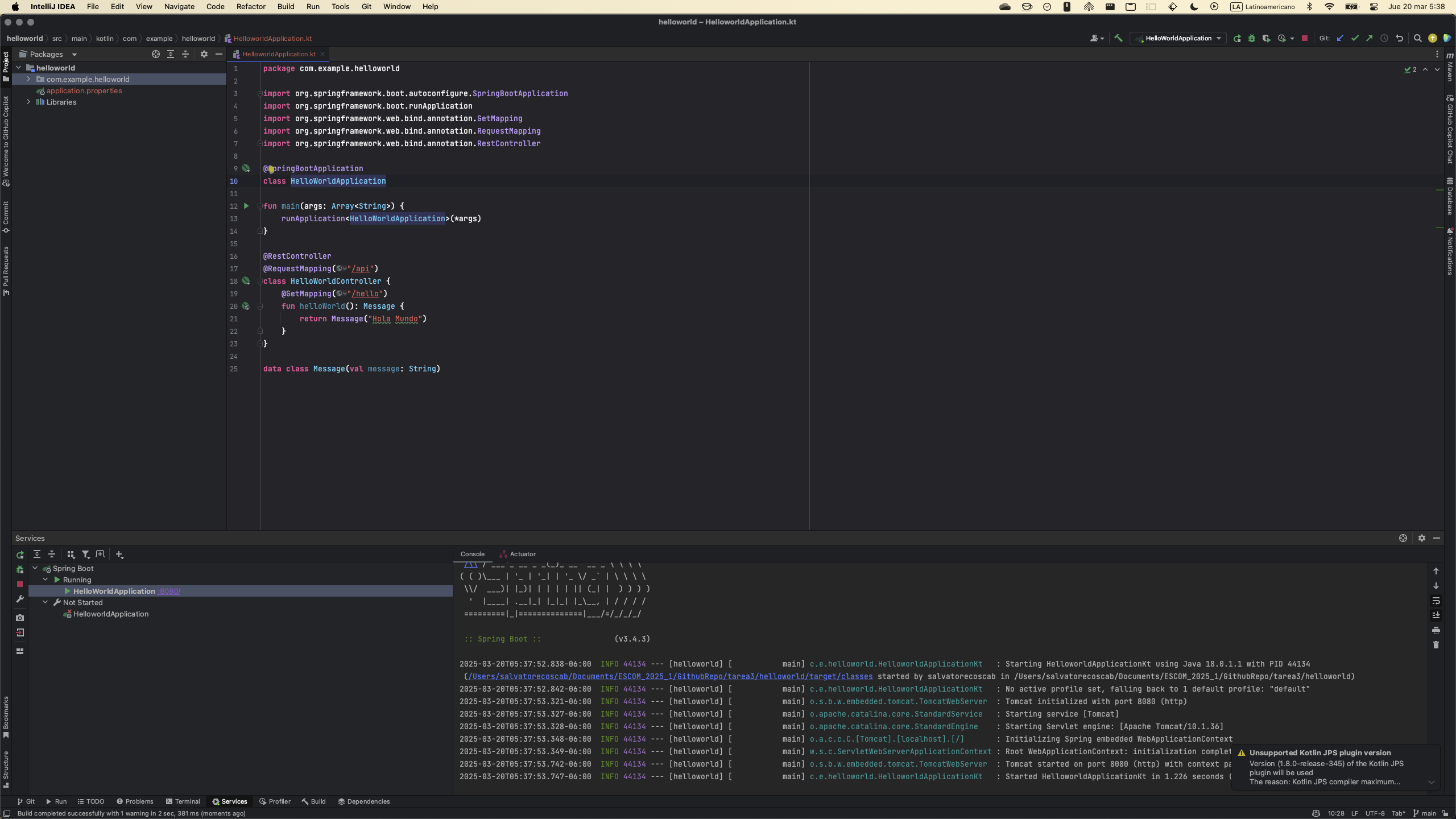
Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Posteriormente se programó y se inicializó el servicio red y se probó que el servidor funcionara en local, con la URL localhost:8080/api/hello



Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Posteriormente se configuró la aplicación de Android y se configuraron los permisos en el Android Manifest y se implementó el código para realizar la petición HTTP al servicio REST creado utilizando la biblioteca Retrofit.

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

El código realizado se muestra en el repositorio: <https://github.com/Salvatorecoscab/AplicacionesMovilesNativas/tree/main/tarea3/APIRest>

**Ejercicio 2**

Se realizó la selección de api: Rick and Morty API

* **Base URL:** https://rickandmortyapi.com/api/
* **Endpoints (ejemplos):**
  + **Personajes:** /character (para obtener una lista de personajes)
    - /character/{id} (para obtener detalles de un personaje específico por su ID)
    - /character/?name={name} (para buscar personajes por nombre)
  + **Episodios:** /episode (para obtener una lista de episodios)
    - /episode/{id} (para obtener detalles de un episodio específico por su ID)
  + **Locaciones:** /location (para obtener una lista de locaciones)
    - /location/{id} (para obtener detalles de una locación específica por su ID)

**Documentación de la API:** <https://rickandmortyapi.com/>

Posteriormente se realizó el diseño de la interfaz de usuario para permitir la búsqueda de personajes y mostrar resultados en una lista grid utilizando Recycler viewer y se muestran los detalles al seleccionar un elemento de la lista. Se incluyen elementos visuales de imágenes que regresa la API y un diseño responsivo que funciona en diferentes tamaños de pantalla.

Se crearon las clases necesarias para mapear las respuestas JSON de la API y se implementó el servicio de conexión utilizando Retrofit.

Se utilizaron los headers necesarios y se implementó la paginación para hacer la búsqueda de los personajes.

Se mejora la experiencia de usuario al añadir elementos de carga durante las peticiones y se configuró el manejo de errores con mensajes descriptivos.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.Texto

El contenido generado por IA puede ser incorrecto.

Pantalla de computadora con un teclado de computadora

El contenido generado por IA puede ser incorrecto.