



Pokédex

Aplicación Android basada en el patrón Modelo - Vista - Controlador (MVC)

Autora: Ana Núñez

Curso: Desarrollo de Aplicaciones Multiplataforma (DAM)

Fecha: Octubre 2025

Proyecto Android Studio – Lenguaje Java

github.com/AnaNunezRejon/PokedexAndroidStudiojava

1. Descripción general

Aplicación Android basada en el patrón **Modelo - Vista - Controlador (MVC)**.

Autora: Ana Núñez

Curso: Desarrollo de Aplicaciones Multiplataforma (DAM)

Fecha: Octubre 2025

Proyecto: Android Studio – Lenguaje Java

Repositorio: github.com/AnaNunezRejon/PokedexAndroidStudio_java

2. Descripción de la aplicación

Pokédex es una app Android desarrollada en **Java** siguiendo el patrón **MVC**. Consume la API pública PokeAPI para mostrar información de los Pokémon: nombre, número, imagen, tipos, altura, peso, habilidad, categoría y descripción (todo en español).

La interfaz incluye:

- Buscador con sugerencias en tiempo real (nombre o número).
 - Chips de tipos y debilidades.
 - Carga por bloques con paginación.
 - Traducción al español de habilidades, categorías y descripciones.
-

3. Objetivo general

Permitir **consultar** y **explorar** Pokémon como ejemplo práctico de:

- Uso del patrón MVC en Android.
 - Consumo de APIs REST (HTTP + JSON).
 - Diseño de interfaces dinámicas en XML.
 - Buenas prácticas de UI y asincronía (uso de hilos, `runOnUiThread`, `ExecutorService`).
-

4. Requisitos funcionales

- **RF1:** Pantalla de inicio – Mostrar el logo de Pokédex y redirigir a la pantalla de tipos.
 - **RF2:** Pantalla de tipos – Permitir seleccionar un tipo de Pokémon y realizar búsquedas por nombre o número.
 - **RF3:** Pantalla de lista – Mostrar los Pokémon del tipo seleccionado en cuadrícula con paginación (30 por bloque).
 - **RF4:** Pantalla de detalle – Mostrar ficha completa con imagen, número, altura, peso, habilidad, categoría, descripción, tipos y debilidades.
 - **RF5:** Buscador – Realizar búsquedas dinámicas (nombre o número) con sugerencias automáticas.
 - **RF6:** Traducción – Mostrar todos los textos traducidos al español.
 - **RF7:** Carga asíncrona – Las llamadas a la API deben ejecutarse en hilos separados para no bloquear la interfaz.
 - **RF8:** Interfaz dinámica – Generar chips de tipo y debilidad de forma programática con color por tipo.
 - **RF9:** Manejo de errores – Mostrar mensajes adecuados si no hay conexión o el Pokémon no existe.
 - **RF10:** Fluidez – Cargar imágenes y datos sin retrasos visibles mediante asincronía.
-

5. Funcionamiento general

1. **Pantalla de inicio:** muestra el logo de Pokédex.
 2. **Pantalla de tipos:** botones de tipos (agua, fuego, planta, etc.) y buscador superior.
 3. **Pantalla de lista:** muestra los Pokémon del tipo elegido en un `GridLayout`.
 4. **Pantalla de detalle:** muestra la ficha completa del Pokémon: imagen, número, altura, peso, habilidad, categoría, descripción, tipos y debilidades.
-

6. Arquitectura interna (MVC)

6.1. Modelo (`com.example.intentopokedex3.model`)

- **Pokemon.java:** Clase que representa un Pokémon individual. Contiene: nombre, número, imageUrl, tipos, altura, peso y habilidad.
 - **PokedexApi.java:** Gestiona las peticiones HTTP a la API.
-

6.2. Controlador (`com.example.intentopokedex3.controller`)

Controla la lógica entre las vistas y el modelo (`PokedexApi`). Gestiona hilos y comunicación entre UI y datos. En este proyecto, las *Activities* realizan funciones de controlador directo (por claridad académica).

6.3. Vista (`com.example.intentopokedex3.view`)

Archivo	Descripción
ActivityInicio.java	Pantalla inicial con el logo.
ActivityTipos.java	Muestra botones de tipos y buscador.
ActivityLista.java	Lista de Pokémon del tipo seleccionado, con paginación.
ActivityDetalle.java	Ficha detallada del Pokémon con información traducida al español.

7. Conexión con la API

Base URL: `https://pokeapi.co/api/v2/`

Endpoints utilizados:

- `type/{tipo}` → lista de Pokémon por tipo.
- `pokemon/{nombre|id}` → datos principales.
- `pokemon-species/{id}` → categoría y descripción.
- `ability/{nombre}` → nombre de la habilidad en español.

Formato: JSON – Conversión manual mediante JSONObject y JSONArray.

8. Flujo interno del programa

ActivityInicio → ActivityTipos → ActivityLista → PokedexApi → ArrayList<Pokemon> → ActivityDetalle.

- **Pantalla de inicio:** Muestra el logo y redirige con Intent.
 - **Pantalla de tipos:** Traduce tipo a inglés para la API.
 - **Pantalla de lista:** Carga Pokémon por tipo con paginación (30 en 30).
 - **Pantalla de detalle:** Carga información extendida y chips de tipo/debilidad.
-

9. Internacionalización (Español)

Elemento	Fuente	Traducción
Habilidad	/ability/{nombre}	names[language=es].name
Categoría	/pokemon-species/{id}	genera[language=es].genus
Descripción	/pokemon-species/{id}	flavor_text_entries[language=es].flavor_text
Tipos (UI)	Mapeo interno	fire→Fuego, grass→Planta, etc.

10. Características técnicas

- **Lenguaje:** Java
 - **IDE:** Android Studio
 - **Arquitectura:** MVC
 - **API:** PokeAPI
 - **Diseño:** XML + Drawables personalizados
 - **Permisos:** INTERNET
 - **Librerías:** AndroidX, FlexboxLayout
-

11. Funcionamiento del buscador

Basado en `EditText + TextWatcher`. Realiza llamadas a `PokedexApi.buscarPorNombre()` en un hilo separado. Si el texto cambia durante la búsqueda, se descarta el resultado anterior (antirrebote). Permite buscar por nombre parcial o número. Muestra una lista flotante (`ListView`) con resultados.

12. Rendimiento y asincronía

Uso de `ExecutorService` y `Handler` para evitar bloqueos en la UI. Carga por bloques (paginación) y descarga de imágenes en hilos independientes. Manejo de valores seguros (sin null).

13. Diseño visual

Temática Pokédex: fondo morado degradado y logo inferior. Cajas de detalle con esquinas redondeadas. Botones personalizados y chips de tipo/debilidad generados dinámicamente con colores.

14. Estructura de carpetas

```
com.example.intentopokedex3
  model
    Pokemon.java
  controller
    PokedexApi.java
  view
    ActivityInicio.java
    ActivityTipos.java
    ActivityLista.java
    ActivityDetalle.java
  res
    layout/
    drawable/
    values/
```

15. Pruebas realizadas

Búsqueda por nombre (ej. “char” → Charmander, Charizard). Búsqueda por número (ej. “25” → Pikachu). Traducciones correctas (habilidad, descripción, categoría). Paginación fluida (bloques de 30). Antirrebote funcional. Carga de debilidades exacta.

16. Mejoras futuras

- Implementar RecyclerView en lugar de GridLayout.
 - Añadir favoritos offline (Room Database).
 - Cachear imágenes (Glide o LruCache).
 - Modo oscuro y accesibilidad.
 - Pruebas unitarias de modelo y UI (Espresso).
-

17. Mockups y referencias visuales



Figura 1: *
Pantallas de la aplicación Pokédex Android

18. Créditos

Proyecto académico desarrollado por **Ana Núñez**, como parte del módulo *Desarrollo de Aplicaciones Multiplataforma (DAM)*. Datos obtenidos de la API oficial PokeAPI.

Pokédex Android – Octubre 2025

Desarrollado en Java + Android Studio.

Diseño y código original por Ana Núñez.