

**Nombre de la aplicación: PokeUPSA****Desarrolladores: Daniel Díez Arias y Marcos Gomez de Quero Santos****Descripción**

**PokeUPSA** es una aplicación que nos permite obtener información acerca de elementos del mundo Pokemon como son los propios Pokemon y los objetos que son usados para capturarlos. ***Los Pokemon son seres ficticios correspondientes al mundo de la saga de anime y videojuegos "POKEMON".***

Con esta aplicación podremos obtener información detallada sobre ellos y así conocer mejor su mundo.

El usuario dispondrá de cuatro opciones. Dichas opciones serán mostradas en la ventana principal de la aplicación

**Opciones**

**-Buscar:** el usuario introducirá el ID del Pokemon que desea buscar. El rango comprende entre 1 y 400. El usuario tendrá que pulsar el botón "Buscar".

Si el usuario introduce un numero fuera de rango, la aplicación procesa el error y responde con un Toast indicándole el error cometido.

Si el usuario introduce un ID correcto se mostrará una nueva ventana que contendrá la información relativa al Pokemon.

Tendrá la opción dentro de esa vista de añadir el Pokemon a una lista personal si pulsa en el botón "Añadir a la lista personal".

**-Pokedex Aleatoria:** el usuario tendrá que hacer click sobre el botón "Pokedex aleatoria". Al pulsarse se abrirá una nueva ventana que mostrará un listado de 10 Pokemon.

El usuario podrá pulsar sobre los Pokemon que se han mostrado. Si así lo hace, se abrirá una nueva ventana mostrando información detallada del Pokemon seleccionado.

Tendrá la opción dentro de esa vista de añadir el Pokemon a una lista personal si pulsa en el botón "Añadir a la lista personal".

**-Ver Lista Personal:** el usuario tendrá que hacer click sobre el botón "Ver lista personal". Al pulsarse se abrirá una nueva ventana que mostrará un listado con todos los Pokemon que previamente había añadido a la lista.

Si la lista esta vacía no se muestra ninguna información.

El usuario podrá pulsar sobre los Pokemon que se han mostrado. Si así lo hace, se abrirá una nueva ventana mostrando información detallada del Pokemon seleccionado.

Tendrá la opción dentro de esa vista de borrar el Pokemon de la lista personal si pulsa sobre el botón “Borrar”.

**-Lista de Objetos:** el usuario tendrá que hacer click sobre el botón “Lista de objetos”. Al pulsarse se abrirá una nueva ventana que mostrará un listado de 10 objetos.

## Desarrollo

Para el desarrollo de la aplicación hemos utilizado suministrador de información una **API** llamada **PokeAPI**.

Hemos gestionado las llamadas a la API la creación de métodos asíncronos, los cuales se encargaban de recuperar la información desde la API.

Hemos utilizado como modelo de desarrollo Modelo-Vista-Controlador separando las funcionalidades de la aplicación en distintas clases.

También se ha utilizado el modelo Singleton para la creación de los servicios destacando una particularidad. En el caso de la lista aleatoria de Pokemon, el servicio permanecerá con la información siempre y cuando permanezcamos en la vista contendrá de los Pokemon aleatorios o la vista de su descripción detallada.

Si se vuelve a pulsar el botón “Pokedex aleatoria” el objeto correspondiente al servicio se resetea añadiendo nueva información. De esta manera podemos mantener la información el tiempo que requiere el usuario o modificarla si lo desea ahorrándonos creaciones de objetos.

**El proyecto desarrollado se compone de las siguientes clases, que a su vez subdividiremos por funcionalidad dentro de la aplicación:**

### ACTIVIDADES

**-MainActivity:** actividad principal de la aplicación. En ella se carga la vista principal de la aplicación y se preparan los datos para ser mostrados.

**-ObjetoActivity:** actividad encargada de cargar la vista contenedora de un RecyclerView. Mostrará una lista de objetos de tipo ObjetoDTO.

**-PokemonActivity:** actividad encargada de cargar la vista contenedora de un RecyclerView. Mostrará una lista de objetos de tipo PokemonDTO.

**-PokemonDescriptionActivity:** actividad encargada de cargar la vista que contendrá información relativa a un objeto PokemonDTO.

### VIEWMODEL

**-ObjetoViewModel:** clase encargada de preparar el servicio necesario para obtener la lista de objetos de tipo ObjetoDTO.

-**PokemonViewModel**: clase encargada de preparar el servicio necesario para obtener la lista de objetos de tipo PokemonDTO.

*Crearán el objeto servicio necesitado y además darán acceso a la lista de objetos que se especifique.*

## ADAPTADORES

-**ObjetoAdapter**: adaptador encargado de controlar y administrar el RecyclerView que contendrá la información correspondiente a los objetos de tipo ObjetoDTO.

-**PokemonAdapter**: adaptador encargado de controlar y administrar el RecyclerView que contendrá la información correspondiente a los objetos de tipo PokemonDTO.

## OBJETOS

-**ObjetoDTO**: objeto que contendrá la información correspondiente a un objeto de tipo Objeto.

-**PokemonDTO**: objeto que contendrá la información correspondiente a un objeto de tipo Pokemon.

## SERVICIOS

-**ObjetoService**: clase encargada de preparar la información que contendrá la lista de objetos de tipo ObjetoDTO. Responde a un patrón de diseño singleton.

-**PokemonService**: clase encargada de preparar la información que contendrá la lista de objetos de tipo PokemonDTO. Responde a un patrón de diseño singleton.

## CONEXIONES A LA API

-**ObjetoAsyncData**: clase encargada de proporcionar información acerca de un Objeto realizando una conexión a la API **PokeAPI**.

-**ObjetoAsyncImg**: clase encargada de resolver una dirección **URL** con la cual obtendremos una imagen correspondiente a un Objeto

-**PokemonAsyncData**: clase encargada de proporcionar información acerca de un Pokemon realizando una conexión a la API **PokeAPI**.

-**PokemonAsyncImg**: clase encargada de resolver una dirección **URL** con la cual obtendremos una imagen correspondiente a un Pokemon.

-**PokemonAsyncSingleData**: clase encargada de proporcionar información acerca de un único Pokemon realizando una conexión a la API **PokeAPI**.

