

EJEMPLO POO A REALIZAR EN CLASE

Estás encargado de desarrollar una **aplicación Android** para una empresa de transporte. La aplicación permitirá a los usuarios gestionar diferentes tipos de vehículos como **autos, camiones y motocicletas**, mostrando información sobre ellos y simulando su comportamiento. Los vehículos deben compartir características comunes, pero también tienen comportamientos específicos según el tipo.

Parte 1: Clases base, herencia e interfaces

1. Crea una **interfaz** llamada Conducible que defina los siguientes métodos abstractos:

- o fun arrancar()
- o fun detener()

El método arrancar() imprimirá un mensaje indicando que el vehículo está arrancando, mientras que detener() imprimirá un mensaje indicando que el vehículo se ha detenido.

2. Crea una **clase base abstracta** llamada Vehiculo, que implemente la interfaz Conducible y que contenga las siguientes propiedades comunes:

- o marca: String
- o modelo: String
- o anio: Int
- o velocidadMaxima: Int

La clase Vehiculo implementará de forma genérica los métodos de la interfaz:

- o arrancar(): Imprime "El vehículo [marca] ha arrancado."
 - o detener(): Imprime "El vehículo [marca] se ha detenido."
3. Además, define un método abstracto acelerar() en la clase Vehiculo, que cada subclase deberá implementar.
 4. Crea tres clases que hereden de Vehiculo:
 - o Auto: Con una propiedad adicional numPuertas: Int.
 - o Camion: Con una propiedad adicional capacidadCarga: Double (en toneladas).
 - o Motocicleta: Con una propiedad adicional esDeportiva: Boolean.
 5. En cada una de estas subclases, sobrescribe el método acelerar() para que el mensaje de aceleración sea específico para cada tipo de vehículo. Ejemplos:
 - o En Auto: "El auto [marca] está acelerando."
 - o En Camion: "El camión [marca] está acelerando con una carga de [capacidadCarga] toneladas."
 - o En Motocicleta: "La motocicleta [marca] está acelerando a toda velocidad."

Parte 2: Polimorfismo y simulación

1. Crea una función simularConduccion(vehiculo: Conducible) que reciba un objeto que implemente Conducible y realice lo siguiente:
 - o Llamar a arrancar().
 - o Si el objeto también es de tipo Vehiculo, llamar a acelerar() (utilizando la palabra clave is en Kotlin para verificar el tipo).
 - o Llamar a detener().

Parte 3: Aplicación Android

1. Implementa una **Activity principal** en la aplicación Android que tenga la siguiente interfaz gráfica:
 - o Un **Spinner** o **Dropdown** que permita seleccionar entre los distintos tipos de vehículos: **Auto**, **Camion**, y **Motocicleta**.
 - o Un botón "**Simular Conducción**" que, al ser presionado, muestre los mensajes correspondientes para el arranque, aceleración y detención del vehículo seleccionado.
2. Cuando el usuario seleccione un tipo de vehículo en el **Spinner** y presione el botón, se deberá:
 - o Crear una instancia del vehículo correspondiente (Auto, Camion o Motocicleta).
 - o Mostrar los mensajes de arranque, aceleración y detención en un **TextView** o **Dialog** (puedes usar un Toast, AlertDialog o actualizar un TextView con el texto).
3. Usa el polimorfismo para garantizar que la función `simularConduccion()` funcione para cualquier tipo de vehículo seleccionado.

Parte 4: Interactividad avanzada (Opcional)

1. Agrega campos adicionales en la interfaz gráfica que permitan al usuario ingresar datos específicos para cada vehículo, como:
 - o Marca y modelo del vehículo.
 - o Número de puertas en el caso de un auto.
 - o Capacidad de carga en el caso de un camión.
 - o Indicar si la motocicleta es deportiva o no.
2. Muestra estos datos personalizados en los mensajes de simulación. Por ejemplo:
 - o "El auto Toyota Corolla ha arrancado."
 - o "El camión Volvo ha arrancado con una carga de 10 toneladas."