

**POKEMON casi GO****A. PARTE 1**

Crea la clase **Pokemon**:

- **Atributos**

**String nombre**

**Double fuerza**

- **Métodos**

**Pokemon(int huevo)**: constructor que asigna los datos del Pokemon en función del valor del huevo (tiene que ser un valor entre 0 y 2):

Huevo	Nombre	Fuerza
0	Pikachu	Número aleatorio entre 30 y 60
1	Squirtle	Número aleatorio entre 40 y 50
2	Charmander	Número aleatorio entre 45 y 55

**combatir()**: devuelve el valor del atributo fuerza.

**entrenar()**: aumenta y devuelve la fuerza del pokemon según las ganas (fuerza += ganas). Ganas podría ser aleatorio entre 0 y 5.

**nivel()**: devuelve el nivel según la tabla.

Fuerza	Nivel
[0 y 20]	1
[21 y 40]	2
[41 y 50]	3
[51 y 60]	4
>60	5

**B. PARTE 2**

Comprueba el funcionamiento de la clase siguiendo las indicaciones dadas:

Crea un ArrayList de Pokemon y muestra un menú con las opciones:

1. **Encontrar un Pokemon nuevo**: (crea un nuevo Pokemon usando el método encontrarPokemon)
2. **Entrenar**: (entrena al Pokemon que se elija de los contenidos en el ArrayList)
3. **Combatir**: (enfrenta a dos Pokemon de los contenidos en el ArrayList, informa del ganador)
4. **Ver Pokedex**: (muestra toda la información de los Pokemon del ArrayList)
5. **Salir**: (sale del programa)

Crea los siguientes métodos estáticos:

***encontrarPokemon(ArrayList<Pokemon>)***: que usa el constructor de la clase Pokemon para crear una nueva instancia y lo añade al ArrayList de Pokemon.

***entrenarPokemon(ArrayList<Pokemon>)***: solicita el Pokemon a entrenar y, si existe, lo entrena.

***Combatir(Pokemon pA, Pokemon pB)***: enfrenta a dos Pokemons (si existen) y devuelve el Pokemon ganador.

***verPokedex(ArrayList<Pokemon>)***: muestra en pantalla todos los Pokemons del ArrayList (nombre, fuerza, nivel).

Cuando sea necesario lanza excepciones y captúralas.