

Pokemon

Main.

```
from clases.pokemon import Pokemon
```

```
pokedex = []
```

```
def capturar_pokemon():
```

```
    print("\n--- Capturar Nuevo Pokémon ---")
```

```
    nombre = input("Nombre del Pokémon: ")
```

```
    tipo = input("Tipo del Pokémon (ej. Fuego, Agua, Planta): ")
```

```
    try:
```

```
        ataque = int(input("Puntos de ataque: "))
```

```
        defensa = int(input("Puntos de defensa: "))
```

```
        salud = int(input("Puntos de salud: "))
```

```
        nuevo_pokemon = Pokemon(nombre, tipo, ataque, defensa, salud)
```

```
        pokedex.append(nuevo_pokemon)
```

```
        print(f"¡Has capturado a {nombre} con éxito! ")
```

```
    except ValueError:
```

```
        print("Error: El ataque, la defensa y la salud deben ser números.")
```

```
def ver_pokedex():
```

```
    if not pokedex:
```

```
        print("\nNo tienes ningún Pokémon capturado. ")
```

```
        return
```

```
    print("\n--- Mis pokemons ---")
```

```
    for i, pokemon in enumerate(pokedex):
```

```
print(f"{i + 1}. {pokemon.nombre}")
```

```
try:
```

```
    eleccion = int(input("Elige un Pokémon para ver su información (0 para volver): "))
```

```
    if 0 < eleccion <= len(pokedex):
```

```
        pokedex[eleccion - 1].mostrar_info()
```

```
    elif eleccion == 0:
```

```
        return
```

```
    else:
```

```
        print("Opción fuera de rango.")
```

```
except ValueError:
```

```
    print("Opción no válida.")
```

```
def entrenar_pokemon():
```

```
    if not pokedex:
```

```
        print("\nNo tienes ningún Pokémon para entrenar. ")
```

```
    return
```

```
print("\n--- Entrenar Pokémon ---")
```

```
for i, pokemon in enumerate(pokedex):
```

```
    print(f"{i + 1}. {pokemon.nombre}")
```

```
try:
```

```
    eleccion = int(input("Elige un Pokémon para entrenar (0 para volver): "))
```

```
    if 0 < eleccion <= len(pokedex):
```

```
        pokemon_a_entrenar = pokedex[eleccion - 1]
```

```

        pokemon_a_entrenar.entrenar()
    elif eleccion == 0:
        return
    else:
        print("Opción fuera de rango.")
except ValueError:
    print("Opción no válida.")

def liberar_pokemon():
    if not pokedex:
        print("\nNo tienes ningún Pokémon para liberar. ")
        return

    print("\n--- Liberar Pokémon ---")
    for i, pokemon in enumerate(pokedex):
        print(f"{i + 1}. {pokemon.nombre}")

    try:
        eleccion = int(input("Elige un Pokémon para liberar (0 para volver): "))
        if 0 < eleccion <= len(pokedex):
            pokemon_a_liberar = pokedex.pop(eleccion - 1)
            print("Pokémon liberado correctamente.")
        elif eleccion == 0:
            return
        else:
            print("Opción fuera de rango.")

```

```
except ValueError:
```

```
    print("Opción no válida.")
```

```
def menu_principal():
```

```
    while True:
```

```
        print("\n--- Simulador de Pokémon ---")
```

```
        print("1. Capturar un nuevo Pokémon ")
```

```
        print("2. Ver mis pokemon")
```

```
        print("3. Entrenar un Pokémon ")
```

```
        print("4. Liberar un Pokémon ")
```

```
        print("5. ver pokemons")
```

```
        print("6. Salir")
```

```
    opcion = input("Elige una opción: ")
```

```
    if opcion == "1":
```

```
        capturar_pokemon()
```

```
    elif opcion == "2":
```

```
        ver_pokedex()
```

```
    elif opcion == "3":
```

```
        entrenar_pokemon()
```

```
    elif opcion == "4":
```

```
        liberar_pokemon()
```

```
    elif opcion == "5":
```

```
        Pokemon.total_pokemons()
```

```
    elif opcion == "6":
```

```

        print("juego terminado")

        break

    else:

        print("Opción no válida. Inténtalo de nuevo.")


if __name__ == "__main__":

    menu_principal()

pokemon.

from clases.pokemon import Pokemon

pokedex = []


def capturar_pokemon():

    print("\n--- Capturar Nuevo Pokémon ---")

    nombre = input("Nombre del Pokémon: ")

    tipo = input("Tipo del Pokémon (ej. Fuego, Agua, Planta): ")

    try:

        ataque = int(input("Puntos de ataque: "))

        defensa = int(input("Puntos de defensa: "))

        salud = int(input("Puntos de salud: "))

        nuevo_pokemon = Pokemon(nombre, tipo, ataque, defensa, salud)

        pokedex.append(nuevo_pokemon)

        print(f"¡Has capturado a {nombre} con éxito! ")

    except ValueError:

        print("Error: El ataque, la defensa y la salud deben ser números.")

```

```

def ver_pokedex():
    if not pokedex:
        print("\nNo tienes ningún Pokémon capturado. ")
        return
    print("\n--- Mis pokemons ---")
    for i, pokemon in enumerate(pokedex):
        print(f"{i + 1}. {pokemon.nombre}")

    try:
        eleccion = int(input("Elige un Pokémon para ver su información (0 para volver): "))
        if 0 < eleccion <= len(pokedex):
            pokedex[eleccion - 1].mostrar_info()
        elif eleccion == 0:
            return
        else:
            print("Opción fuera de rango.")
    except ValueError:
        print("Opción no válida.")

```

```

def entrenar_pokemon():
    if not pokedex:
        print("\nNo tienes ningún Pokémon para entrenar. ")
        return

    print("\n--- Entrenar Pokémon ---")
    for i, pokemon in enumerate(pokedex):

```

```
print(f"{i + 1}. {pokemon.nombre}")
```

```
try:
```

```
eleccion = int(input("Elige un Pokémon para entrenar (0 para volver): "))
```

```
if 0 < eleccion <= len(pokedex):
```

```
    pokemon_a_entrenar = pokedex[eleccion - 1]
```

```
    pokemon_a_entrenar.entrenar()
```

```
elif eleccion == 0:
```

```
    return
```

```
else:
```

```
    print("Opción fuera de rango.")
```

```
except ValueError:
```

```
    print("Opción no válida.")
```

```
def liberar_pokemon():
```

```
    if not pokedex:
```

```
        print("\nNo tienes ningún Pokémon para liberar. ")
```

```
    return
```

```
print("\n--- Liberar Pokémon ---")
```

```
for i, pokemon in enumerate(pokedex):
```

```
    print(f"{i + 1}. {pokemon.nombre}")
```

```
try:
```

```
    eleccion = int(input("Elige un Pokémon para liberar (0 para volver): "))
```

```
    if 0 < eleccion <= len(pokedex):
```

```

        pokemon_a_liberar = pokedex.pop(eleccion - 1)

        print("Pokémon liberado correctamente.")

    elif eleccion == 0:

        return

    else:

        print("Opción fuera de rango.")

except ValueError:

    print("Opción no válida.")


def menu_principal():

    while True:

        print("\n--- Simulador de Pokémon ---")

        print("1. Capturar un nuevo Pokémon ")

        print("2. Ver mis pokemon")

        print("3. Entrenar un Pokémon ")

        print("4. Liberar un Pokémon ")

        print("5. ver pokemons")

        print("6. Salir")


    opcion = input("Elige una opción: ")


    if opcion == "1":

        capturar_pokemon()

    elif opcion == "2":

        ver_pokedex()

    elif opcion == "3":

```



```
    entrenar_pokemon()
```

```
elif opcion == "4":
```

```
    liberar_pokemon()
```

```
elif opcion == "5":
```

```
    Pokemon.total_pokemons()
```

```
elif opcion == "6":
```

```
    print("juego terminado")
```

```
    break
```

```
else:
```

```
    print("Opción no válida. Inténtalo de nuevo.")
```

```
if __name__ == "__main__":
```

```
    menu_principal()
```

The screenshot shows a Python IDE with two files open: `main.py` and `pokemon.py`. The `main.py` file contains a menu function `menu_principal()` that prints options for the simulator. The `pokemon.py` file contains a `Pokemon` class and a `capturar_pokemon()` function. The terminal output shows the execution of the simulator, displaying a menu of options and the details of a captured Pokémon named 'yael'.

```
Archivo Editar Selección Ver Ir ... POO
main.py x pokemon.py
unidad 3 > trabajo 1 > main.py > ...
1 from clases.pokemon import Pokemon
2
3 pokedex = []
4
5 def capturar_pokemon():
6     print("\n--- Capturar Nuevo Pokémon ---")
7     nombre = input("Nombre del Pokémon: ")
8     tipo = input("Tipo del Pokémon (ej. Fuego, Agua, Planta): ")
9     try:
10         ataque = int(input("Puntos de ataque: "))
11         defensa = int(input("Puntos de defensa: "))
12
13     except ValueError:
14         print("Puntos de ataque y defensa deben ser números enteros.")
15
16     pokemon = Pokemon(nombre, tipo, ataque, defensa)
17     pokedex.append(pokemon)
18     print(f"¡Has capturado un {nombre} de tipo {tipo}!")
19
20
21 def menu_principal():
22     print("\n--- Simulador de Pokémon ---")
23     print("1. Capturar un nuevo Pokémon")
24     print("2. Ver mi Pokédex")
25     print("3. Entrenar un Pokémon")
26     print("4. Liberar un Pokémon")
27     print("5. Ver total de Pokémon creados")
28     print("6. Salir")
29     opcion = input("Elige una opción: ")
30
31     if opcion == "1":
32         capturar_pokemon()
33     elif opcion == "2":
34         print(pokedex)
35     elif opcion == "3":
36         print("Entrenar un Pokémon")
37     elif opcion == "4":
38         print("Liberar un Pokémon")
39     elif opcion == "5":
40         print(f"Total de Pokémon creados: {len(pokedex)}")
41     elif opcion == "6":
42         print("¡Gracias por jugar!")
43         exit()
44
45 if __name__ == "__main__":
46     menu_principal()
```

```
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
NameError: name 'Pokemon' is not defined. Did you mean: 'pokemon'?
PS C:\Users\Desconocido\Documents\POO> & C:\Users\Desconocido\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/Desconocido/Documents/POO/unidad 3/trabajo 1/main.py"

--- Simulador de Pokémon ---
1. Capturar un nuevo Pokémon
2. Ver mi Pokédex
3. Entrenar un Pokémon
4. Liberar un Pokémon
5. Ver total de Pokémon creados
6. Salir
Elige una opción: 1

--- Capturar Nuevo Pokémon ---
Nombre del Pokémon: yael
Tipo del Pokémon (ej. Fuego, Agua, Planta): fuego
Puntos de ataque: 23
Puntos de defensa: 32
Puntos de salud: 342
¡Ha aparecido un yael salvaje!
Nombre del Pokémon: yael
Tipo del Pokémon (ej. Fuego, Agua, Planta): fuego
```

Lin. 109, Col. 21 (3510 seleccionada) Espacios: 4 UTF-8 CRLF Python 3.13.1

<https://github.com/yael-byte/poo>