

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()

if 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 pokémon")  
        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.")

if 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 pokémon")
        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 displays a Windows desktop environment with two code editors and a terminal window.

- Left Editor:** Shows the beginning of the 'main.py' file. It imports 'Pokemon' from 'clases.pokemon'. A 'NameError' is visible in the terminal below it, indicating that 'Pokemon' is not defined. The code continues with a 'capturar_pokemon()' function definition.
- Right Editor:** Shows the full 'main.py' file. It includes a 'menu_principal()' function that prints a menu and handles user input. It also contains a 'capturar_pokemon()' function. The terminal window below shows the execution of the script and its output.
- Terminal Window:** Displays the following output:


```

      --- 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: 342
      ¡Ha aparecido un yael salvaje! 🌈
      Nombre del Pokémon: yael
      Tipo del Pokémon (ej. Fuego, Agua, Planta): fuego
      
```

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