Miniproyecto 1

Desarrolla un simulador de batallas Pokémon utilizando **Java** y aplicando todos los conceptos de **Programación Orientada a Objetos (POO)**. En este proyecto, deberán crear una simulación en la que dos entrenadores Pokémon se enfrenten en una batalla. El proyecto debe ser interactivo y permitir la creación de objetos de la clase Pokémon, entrenadores y ataques mediante el uso de la clase **Scanner** para ingresar la información desde la consola.

Requisitos:

1. **Clases Principales:**
    - **Pokémon**: Cada Pokémon debe tener atributos como:
        - Nombre
        - Tipo (Fuego, Agua, Planta, etc.)
        - Puntos de Salud (HP)
        - Ataques disponibles (pueden ser hasta 4)
        - Debe permitir realizar ataques y recibir daño.
    - **Ataque**: Cada ataque debe tener atributos como:
        - Nombre
        - Tipo de daño (Físico o Especial, esto no lo usaremos en este proyecto para nada, pero haganlo)
        - Potencia de ataque
    - **Entrenador**: Un entrenador debe tener:
        - Nombre
        - Equipo de Pokémon (puede ser una lista de hasta 3 Pokémon)
        - Método para elegir un Pokémon para la batalla.
        - No se permite intercambiar Pokémon
        - El primer pokémon que realiza el movimiento es el Pokémon con menos salud
2. **Interacción con el Usuario:**
    - El simulador debe utilizar **Scanner** para que el usuario ingrese información como:
        - Nombre de los entrenadores.
        - Selección del Pokémon para la batalla. El usuario podrá crear su propio equipo manualmente (crear pokemon por pokemon, con todos los detalles) o una opción donde se le asignará un equipo totalmente aleatorio
        - Elección de ataques durante la batalla.
3. **Lógica de la Batalla:**

- El combate debe ser turnado, donde cada entrenador elige un ataque para su Pokémon y ambos Pokémon intercambian golpes hasta que uno se quede sin puntos de salud.
- Cuando un pokémon ataca al otro, el daño se calcula restando la vida del pokémon atacado menos la potencia del ataque. Si hay una ventaja de tipos (investigar esto), la potencia del ataque aumenta 30%
- La batalla termina cuando uno de los Pokémon tiene 0 puntos de salud.

4. **POO Aplicada:**

    - Uso de herencia para tipos de Pokémon específicos (por ejemplo, clase **PokémonFuego** que hereda de la clase **Pokémon**).
    - Uso de polimorfismo para que los ataques y las batallas puedan adaptarse a distintos tipos de Pokémon.
    - Encapsulamiento para proteger los atributos de las clases y permitir su manipulación a través de métodos apropiados.
    - Uso de clases y objetos de manera eficiente para representar las relaciones entre entrenadores, Pokémon, y ataques.
    - Usen lo visto en clase, si usan cosas por fuera, les podría bajar

Entrega:

- Deben subir el código a github. Tendré en cuenta para la calificación:
    - Buenas practicas
    - Comentarios en codigo
    - Quien hace commits, quien revisa, que usen el tablero kanban. Lo visto en clase
- Los integrantes deben ir en el README. Txt del github. El que no aparezca ahí no tendrá nota (la primera vez que suceda esto, se les bajara a la mitad la calificación. Para una segunda vez no tendrá nota quien no aparezca en el README). No vale que lo envíes en el campus, que aparece en el código, etc… debe aparecer en el README.