|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **FACULTAD/DEPENDENCIA:**  Ingeniería | | | **FECHA DE ENTREGA** | | | |
| **PROGRAMA ACADÉMICO/ ÁREA:**  TECNOLOGÍA EN SISTEMAS | | | **DD**  06 | | **MM**  09 | **AAAA**  204 |
| **ASIGNATURA**:  Construcción de software IV | | **PROFESOR:**  Juan Guillermo Duque Galvis | | | | |
| **TIPO DE EVALUACIÓN** | Práctica | **PORCENTAJE:**  15% | | **CALIFICACIÓN:** | | |
| Nombre Alumno: Documento  Manuela Ruiz Bedoya cc. 1017222185 | | | | | | |

**Entrega**: Enviar documento con URL del repositorio al correo jduqueg2@tdea.edu.co Con asunto: Entregable 02 - CS4

**Lenguaje: C#**

**Punto 1:**

La desarrolladora Game Freak está solicitando una nueva instancia del mítico juego Pokemon pero con unas obligaciones específicas.

Cada pokémon tiene un nombre, un tipo, 3 ataques con un puntaje de entre 0 y 40, No hay valores decimales y una defensa entre 10 y 35 no hay valores decimales. Además de tener las habilidades de atacar y defender.

La habilidad atacar va seleccionar uno de los ataques de forma aleatoria y su puntaje lo va multiplicar con 1 o 0 y devuelve el valor resultante.

La habilidad defensa va multiplicar la defensa por 1 y 0.5 y va devolver el resultado. Los valores de los pokemones serán ingresados a través de consola.

Para realizar un combate en el escenario principal, deben estar los dos pokemones creados y se harán 3 turnos donde cada 1 de los pokemons va atacar y defender.

Al final deben decir si gano el Pokemon 1 o el Pokemon dos o si hay empate.

**Obligaciones**:

1. Usar 1 Interfaz

2. Usar 1 Clase abstracta

3. Usar 1 Clases que hereden

4. Usar constructores

5. Todas las propiedades privadas

6. Ejecutar el programa la batalla

Se puede hacer en cualquier editor que prefieran.

Punto 2:

Se solicita un programa que seleccione jugadores automáticos para un partido de basket de 3 vs 3. El software tiene una lista de jugadores registrados que serán un máximo de 6. Los datos de los jugadores serán nombre, posición y rendimiento, El rendimiento será un número entre 0 y 10. La selección del jugador será una mecánica aleatoria donde obtendrá 1 de la lista de jugadores. Este jugador se agrega al equipo a través de la acción agregar jugador.

Una vez distribuidos los jugadores en los equipos, se decide el ganador del partido comparando el equipo que tenga mayor puntaje en la sumatoria del rendimiento de sus jugadores.

Requisitos:

1. Usar Mínimo una Interfaz

2. Usar Clases

3. Usar constructores

4. Todas las propiedades privadas

5. Ejecutar el programa de selección y partido