



Cátedra Tecnología de Sistemas

[Introducción a la Programación]

Código: [00831]

Tarea 1. Valor 1%

Temas de Estudio

Tema 1

Subtemas

1. Introducción a las computadoras y a C++
2. Introducción a la programación en C++, entradas/salidas y operadores
3. Introducción funciones miembro y cadenas

Tema 2

Subtemas

1. Desarrollo de algoritmos e instrucciones de control: Parte 1
2. Instrucciones de control; Parte 2: operadores lógicos

Objetivo

Resolver un problema, con un programa en el lenguaje de C++ aplicando lo aprendido en los temas de estudio

Software de Desarrollo

CodeBlocks, en la plataforma MOODLE está disponible las instrucciones para su instalación

Desarrollo

Se debe desarrollar un programa en **C++** que permita a un entrenador Pokémon gestionar su equipo y mejorar sus estadísticas mediante entrenamientos.

El sistema contará con un **menú principal** con las siguientes opciones:

Menú Principal

1. Registrar un Pokémon
2. Entrenar un Pokémon
3. Mostrar el equipo Pokémon
4. Mostrar el Pokémon más fuerte
5. **Salir del programa**

1. Registrar un Pokémon

El usuario podrá registrar un Pokémon ingresando los siguientes datos:

- **Nombre del Pokémon** (*Ejemplo: Pikachu, Charmander, Bulbasaur, etc.*)
- **Número de identificación** (6 dígitos numéricos)
- **Tipo de Pokémon:** El usuario debe seleccionar el tipo de Pokémon de una lista predefinida, tipos disponibles (*1.Fuego, 2.Agua, 3.Planta, 4.Eléctrico, etc.*). El programa debe permitir la selección ingresando el número correspondiente.
- **Nivel de Poder Inicial** (*Valor entre 1 y 100*)

Validaciones:

- ✓ El **nombre** no puede estar vacío.
- ✓ El **ID** debe contener exactamente **6 dígitos**.
- ✓ Si el ID ya existe en el equipo Pokémon, debe mostrar un mensaje indicando que el Pokémon ya está registrado y no permitir el ingreso duplicado.
- ✓ El **tipo de Pokémon** debe ser una opción válida de la lista.
- ✓ El **nivel de poder** debe estar entre **1 y 100**.

2. Entrenar un Pokémon

Cada Pokémon puede entrenar para mejorar su poder. El usuario deberá ingresar:

- **ID del Pokémon**
- **Tipo de entrenamiento:** El usuario debe seleccionar una opción de la siguiente lista (*1.Combate en gimnasio, 2.Batalla con otro entrenador, 3.Práctica de habilidades, etc.*). El programa debe validar que el usuario ingrese un número válido de la lista.
- **Dificultad del entrenamiento** (*Valor entre 1 y 100*)

Reglas:

- ✓ Si el **nivel de poder** del Pokémon es **mayor o igual** a la dificultad, el entrenamiento es **exitoso** y el Pokémon **gana +10 puntos**.
- ✓ Si el entrenamiento **falla**, el Pokémon no gana puntos y se muestra un mensaje motivador.

Ejemplo de resultado:

¡Entrenamiento exitoso! Pikachu ha ganado +10 puntos de poder.

o

Entrenamiento fallido. Pikachu necesita más práctica antes de intentarlo de nuevo.

Validaciones:

- ✓ Si el **ID ingresado no existe** en el equipo Pokémon, el programa debe mostrar un mensaje de error y no permitir el entrenamiento.

3. Mostrar el equipo Pokémon

Se listarán todos los Pokémon registrados con el listado siguiente:

=====

EQUIPO POKÉMON

=====

- Nombre: Pikachu

ID: 102345

Tipo: Eléctrico

Nivel de Poder: 75

- Nombre: Charmander

ID: 103222

Tipo: Fuego

Nivel de Poder: 65

Si no hay Pokémon registrados, se mostrará:

No hay Pokémon registrados aún.

4. Mostrar el Pokémon más fuerte

El programa buscará el Pokémon con **mayor nivel de poder** y lo mostrará así:

=====

POKÉMON MÁS FUERTE

=====

Nombre: Pikachu

Tipo: Eléctrico

Nivel de Poder: 85

=====

Si no hay registros:

No hay Pokémon registrados aún.

5. Salir del programa

El programa seguirá ejecutándose hasta que el usuario seleccione "**Salir**".

Ejemplo de Ejecución

=====

SISTEMA DE ENTRENAMIENTO POKÉMON

=====

1. Registrar un Pokémon

2. Entrenar un Pokémon
3. Mostrar el equipo Pokémon
4. Mostrar el Pokémon más fuerte
5. Salir del programa

=====

Seleccione una opción: 1

Ingrese el nombre del Pokémon: Pikachu

Ingrese el número de identificación (6 dígitos): 102345

Seleccione el tipo de Pokémon: Eléctrico

Ingrese el nivel de poder inicial (1-100): 75

¡Pokémon registrado exitosamente!

=====

Seleccione una opción: 2

Ingrese el ID del Pokémon: 102345

Seleccione el tipo de entrenamiento: Combate en gimnasio

Ingrese la dificultad del entrenamiento (1-100): 50

¡Entrenamiento exitoso! Pikachu ha ganado +10 puntos de poder.

=====

Seleccione una opción: 3

EQUIPO POKÉMON:

- Nombre: Pikachu

ID: 102345

Tipo: Eléctrico

Nivel de Poder: 85

=====

Seleccione una opción: 4

POKÉMON MÁS FUERTE:

Nombre: Pikachu

Tipo: Eléctrico

Nivel de Poder: 85

=====

6. Almacenamiento de datos: Toda la información del equipo Pokémon se almacenará temporalmente en memoria, utilizando vectores o arreglos.

Honestidad Académica



<https://audiovisuales.uned.ac.cr/play/player/23048>

Nota Importante

Cada estudiante es responsable del contenido que entrega, si no es el archivo correcto, no podrá entregarlo posterior a la fecha establecida.

Si el contenido del archivo coincide con algún otro estudiante, o se comprueba que no es de su autoría, se aplicaría lo indicado en la plataforma en el documento [Lineamientos ante casos de plagio](#)

Indicaciones Importantes

- Es obligatorio que incluya todo el directorio donde se encuentra el proyecto1
- El < proyecto1 > debe estar desarrollado en [CodeBlocks] que es la herramienta oficial del curso.
- El programa debe ser modular, utilizando de la mejor manera funciones definidas por usted.
- Los trabajos deben realizarse en forma individual. Dentro del código del programa debe de indicar la documentación que explique cómo fue realizado el programa.
- Si utiliza código de algún ejemplo del libro, o de otra fuente que no sea de su autoría, debe de indicarlo.
- Comprima todos los archivos en un solo archivo .zip o .rar.
- Nombre del archivo que envía: debe ser nombre y primer apellido del estudiante, y nombre de la tarea. **Ejemplo: JuanRojas-proyecto1.**
- La entrega del <proyecto1> en las fechas establecidas en la plataforma de aprendizaje en línea Moodle en el apartado que se indique.

Rúbrica de Evaluación

Criterio	Cumple a satisfacción (Puntaje Máximo)	Cumple medianamente	Cumple en contenido y formato pero los aportes no son significativos	No cumple o no presenta lo solicitado
Formato: Nitidez y presentación del código, incluyendo Redacción - Ortografía // Documentación interna dentro del código	5 - Código limpio, bien estructurado, con comentarios claros y explicativos. Buena ortografía y redacción en nombres de variables y funciones.	3 - Código con buena estructura, pero con comentarios mínimos o desorganizados. Algunos errores ortográficos.	1 - Código con deficiente presentación y mínima documentación. Se dificulta la lectura.	0 - Código desorganizado, sin comentarios ni estructura clara.
Orden y claridad en el planteamiento (lógica). Uso adecuado de indentación, nombres significativos de variables y funciones.	10 - Código bien estructurado con lógica clara, indentación uniforme y nombres de variables significativos.	5 - Código con estructura aceptable pero con alguna desorganización en la lógica.	3 - Lógica confusa, nombres poco descriptivos, indentación inconsistente.	0 - Código sin estructura lógica clara, difícil de entender.
Estructuras de control – Secuenciales. Implementación correcta de if, if/else y switch.	25 - Uso correcto de estructuras de control. Se evalúa: <ul style="list-style-type: none"> • Evaluación correcta de opciones del menú. • Implementación correcta de condiciones en el entrenamiento Pokémon. • Lógica clara y sin errores de secuencia. 	15 - Uso adecuado de estructuras de control, pero con errores menores en la evaluación de opciones.	10 - Uso limitado de if/switch, con fallas en la lógica de las condiciones.	0.1 - Código no implementa correctamente las estructuras de control, o no están presentes.
Estructuras de control - iterativas. Implementación de while, do/while y for.	25 - Uso correcto de ciclos para el manejo del menú, ingreso de datos y validaciones.	15 - Uso adecuado de ciclos, pero con errores en la ejecución o sin	10 - Uso ineficiente de ciclos, con algunas repeticiones incorrectas o	0.1 - No usa estructuras de repetición o el código entra en bucles infinitos.

	<ul style="list-style-type: none"> • No hay ciclos infinitos. • Se repite correctamente hasta que el usuario decida salir. 	un control óptimo.	mal implementadas.	
Validaciones. Implementación de las validaciones requeridas en el enunciado.	20 - Se validan correctamente los siguientes aspectos: <ul style="list-style-type: none"> • El ID del Pokémon debe tener 6 dígitos. • El nivel de poder está entre 1 y 100. • El usuario debe seleccionar un tipo de Pokémon válido. • Evita errores en el ingreso de datos. 	10 - La mayoría de las validaciones están presentes, pero hay errores en la detección de datos inválidos.	5 - Se implementan algunas validaciones, pero no son suficientes o presentan fallos graves.	0.1 - No hay validaciones o el código acepta cualquier entrada sin restricciones.
Impresión de información en pantalla. Presentación de datos clara y organizada.	10 - La salida de datos es clara, bien organizada y tabulada. Se muestran los Pokémon registrados, entrenamientos y Pokémon más fuerte con buena legibilidad.	5 - La salida de datos es comprensible pero desordenada o con errores en la visualización.	2 - La información no está bien estructurada, dificultando su comprensión.	0 - La presentación es confusa o no se imprimen los datos correctamente.
Interfaz de usuario (NO GUI) – Aplicación intuitiva y fácil de usar.	5 - Menú bien diseñado, intuitivo y con respuestas claras para el usuario. Permite interacción fluida y sin errores.	3 - El menú y la interfaz son funcionales, pero no intuitivos o presentan errores menores.	1 - Interfaz poco clara, con errores en las opciones o falta de mensajes adecuados.	0 - Difícil de usar, no responde correctamente a la interacción del usuario.
TOTAL	100 puntos			