

Propuesta de proyecto de fundamentos de programación

Universidad Centroamericana José Simeón Cañas

Facultad de Ingeniería y Arquitectura

Departamento de Electrónica e Informática



Integrantes

Nombre	Código
Escobar Menjivar, Jorge Alberto	00072325
Flores Crisóstomo, Juan Manuel	00135025
Pablo Antonio Brito Elías	00179025
González García, Diego Alejandro	00072825

Antiguo Cuscatlán, Martes 3 de junio de 2025.

Índice

Introducción.....	3
Descripción del juego.....	4
Temática.....	4
Mecánica principal.....	4
Idea General de la Jugabilidad.....	4
Objetivos del jugador.....	5
Aplicación de temas del curso.....	5
Consideraciones Técnicas del Desarrollo.....	5
Estructura del código:.....	5
Programación Orientada a Objetos (POO).....	6
Tipos de Datos y Estructuras.....	6
Control de Flujo.....	6
Diagramas y Planificación.....	6
Mockups.....	6
Menú principal.....	6
Duelo:.....	7
Conclusión.....	8
Referencias.....	9

Introducción

El presente documento tiene como propósito detallar la propuesta formal para el desarrollo del videojuego que constituirá el proyecto final de la asignatura Fundamentos de Programación. A lo largo del curso, se han abordado los conceptos esenciales para comprender y aplicar los principios de la programación, tanto desde una perspectiva teórica como práctica. Entre los temas estudiados se incluyen las metodologías de programación, el uso y características de los lenguajes de bajo y alto nivel, la estructura de un programa, tipos de datos, estructuras de control de flujo, y el uso de estructuras de datos estáticas como arreglos, matrices y estructuras (structs), entre otros.

El lenguaje seleccionado para el desarrollo del proyecto es C++, una elección adecuada por su capacidad para soportar tanto programación estructurada como orientada a objetos, permitiendo así integrar todos los fundamentos vistos en clase. El proyecto consistirá en el desarrollo de un videojuego de batallas de tipo Pokémon, ejecutado completamente en consola, donde los jugadores podrán enfrentarse en combates por turnos seleccionando ataques según el tipo de su Pokémon.

A continuación, se expone en detalle la propuesta de diseño del videojuego, junto con la justificación técnica de cada una de sus partes, y la manera en que los contenidos del curso serán aplicados a lo largo del proceso de desarrollo.

Descripción del juego

Temática

Videojuego por turnos ambientado en el universo de Pokémon, diseñado para ejecutarse en consola. Los jugadores asumen el rol de entrenadores Pokémon que se enfrentan entre sí o contra la cpu, utilizando sus criaturas en combates estratégicos. Cada Pokémon posee un tipo (fuego, agua, planta, etc.) que influye en la efectividad de sus ataques, replicando las mecánicas clásicas de la franquicia.

La ambientación es completamente textual, utilizando interfaz por consola. A través de menús y mensajes, se representa el flujo de combate, la información de los Pokémon, y las decisiones del jugador.

Mecánica principal

El juego se basa en combates por turnos entre dos Pokémon. Al inicio, el jugador selecciona uno de dos modos:

- Jugador vs Jugador
- Jugador vs CPU

Cada jugador elige un Pokémon y comienza el combate. En cada turno, los jugadores seleccionan un ataque. El tipo del Pokémon determina la efectividad del ataque sobre el oponente. Los puntos de vida disminuyen según los daños calculados. El combate continúa hasta que un Pokémon pierde toda su vida.

El juego es completamente textual y basado en consola. Los jugadores interactúan mediante entradas por teclado, seleccionando opciones del menú, ataques y visualizando el estado de la batalla.

Idea General de la Jugabilidad

- Menú de inicio con opciones de modo de juego.
- Selección de Pokémon por parte de los jugadores.
- Combate por turnos gestionado por una clase Partida.
- Ataques basados en el tipo del Pokémon (implementado con clases derivadas).
- El juego muestra mensajes en consola indicando el estado del combate.
- Determinación automática del ganador al finalizar la partida.

Objetivos del jugador

- Derrotar al oponente agotando completamente los puntos de vida de su Pokémon.
- Tomar decisiones estratégicas sobre qué ataque usar según el tipo del oponente.

- En el caso del modo CPU, vencer al algoritmo contra el que se está enfrentando

Aplicación de temas del curso

A continuación, se detalla cómo se aplicarán los principales temas abordados en clase durante el desarrollo del videojuego. En primer lugar, se emplearán variables y tipos de datos como int, float, bool, char y string para representar aspectos fundamentales del juego, como la vida de los Pokémon, sus nombres, tipos y estados durante la batalla. Las estructuras condicionales como if, else y switch-case se utilizarán para gestionar la lógica del menú principal, la selección de ataques y las decisiones que afectan el desarrollo de cada partida. En cuanto a los ciclos, se aplicarán bucles for, while para manejar la repetición de turnos, validar entradas del usuario y controlar el flujo del juego hasta que uno de los jugadores gane la partida.

Asimismo, se hará un uso intensivo de funciones para modularizar el código y separar la lógica en bloques reutilizables, como la selección de ataques, la verificación del estado del juego o la visualización del menú. Las estructuras de datos estáticas como arreglos, matrices, structs y clases serán fundamentales para modelar a los Pokémon, sus ataques, y la gestión de las partidas. Por último, toda la interfaz del juego será textual, desarrollada completamente en consola, lo que permitirá una interacción simple y clara con el usuario a través de menús, mensajes y visualización de estados.

Consideraciones Técnicas del Desarrollo

- Lenguaje de programación: C++
- Paradigma principal: Programación orientada a objetos
- Entorno de desarrollo: Visual Studio Code / Dev-C++ / Code::Blocks (dependiendo del equipo)
- Compilador sugerido: g++

Estructura del código:

- Uso de clases Pokemon, PokemonAgua, PokemonFuego, Partida, etc.
- Módulos separados: lógica de batalla, manejo de entrada/salida, clases y estructuras de datos.
- Buenas prácticas aplicadas:
 - Comentarios claros y precisos en funciones clave.
 - Nombres significativos y consistentes en variables, clases y funciones.
 - Código modular, legible y bien indentado.

Programación Orientada a Objetos (POO)

- Creación de una clase base Pokemon y clases derivadas como PokemonFuego, PokemonAgua, etc., para modelar diferentes tipos.
- Aplicación de herencia, constructores, encapsulamiento.
- Una clase Partida administrará el control de turnos y condiciones de victoria.

Tipos de Datos y Estructuras

- Uso de tipos de datos primitivos (int, char, float, bool).
- Manejo de strings para nombres y descripciones.
- Implementación de arrays o vectores para registrar ataques.
- Posible uso de structs para representar el estado del juego si se considera conveniente.

Control de Flujo

- Implementación de estructuras condicionales: if, if-else, switch.
- Uso de bucles (while, do-while, for) para gestionar turnos, validar entradas, y repetir acciones hasta cumplir condiciones de fin de juego.

Diagramas y Planificación

- Representación del flujo del programa mediante diagramas de flujo.

Mockups

Menú principal

```
=====
                    POKEMON GAME
=====

    [1] Un Jugador
    [2] Multijugador
=====

Seleccione una opción: 2
Has seleccionado: Multijugador
```

Duelo:

```
Jugador 1:                               Jugador 2:
  _/\t                                   _/
  ( . . )                               ( . . )
  / > ⚡                                / > 🔥

Vida P1: 100                             Vida P2: 100

Jugador 1, elige tu ataque:
1. Rayo ⚡ (alto daño)
2. Fuego 🔥 (moderado daño)
Opción: 
```

```
Jugador 1:                               Jugador 2:
  _/\t                                   _/
  ( . . )                               ( . . )
  / > ⚡                                / > 🔥

Vida P1: 100                             Vida P2: 89

Presiona Enter para continuar...
```

```
Jugador 1:                               Jugador 2:
  _/\t                                   _/
  ( . . )                               ( . . )
  / > ⚡                                / > 🔥

Vida P1: 85                             Vida P2: 60

Jugador 1, elige tu ataque:
1. Rayo ⚡ (alto daño)
2. Fuego 🔥 (moderado daño)
Opción: 1
Jugador 2, elige tu ataque:
1. Llama 🔥 (alto daño)
2. Trueno ⚡ (moderado daño)
Opción: 
```

Conclusión

El desarrollo del videojuego de batallas Pokémon en C++ como proyecto final del curso de Fundamentos de Programación constituye una experiencia integradora que permite poner en práctica de manera concreta y significativa los conocimientos teóricos y prácticos adquiridos. Este proyecto no solo busca implementar una aplicación funcional, sino también fomentar el pensamiento lógico, la planificación estructurada, la reutilización del código y el uso de buenas prácticas de programación.

A lo largo del diseño y desarrollo del juego, se aplicarán de forma directa los conceptos de programación estructurada mediante el uso correcto de estructuras de control de flujo, funciones y estructuras de datos.

Además, el uso de herramientas como diagramas de flujo permitirá representar visualmente el comportamiento del sistema antes de implementarlo, lo cual es esencial para evitar errores lógicos y mantener la coherencia en la ejecución del programa. La organización del código en módulos, la documentación del mismo y la validación continua del funcionamiento de las distintas partes garantizarán la calidad y mantenibilidad del software desarrollado.

En suma, este proyecto refleja no solo el dominio técnico alcanzado, sino también su capacidad de aplicar el conocimiento de forma creativa, resolver problemas de manera autónoma y trabajar con metodologías propias del desarrollo de software. Con ello, se establece una base sólida para enfrentar retos más complejos en asignaturas futuras y proyectos del mundo real.

Referencias

- CodeWithHarry. (2021, Julio 25). *Learn c++*. c++ cheatsheet. Retrieved Junio 3, 2025, from <https://www.codewithharry.com/blogpost/cpp-cheatsheet>
- guide, s. (2012, Noviembre 13). *Learn C++ Programming*. Programiz. Retrieved June 3, 2025, from <https://www.programiz.com/cpp-programming>
- Jain, S. (2025, May 12). *C++ Programming Language*. GeeksforGeeks. Retrieved June 3, 2025, from <https://www.geeksforgeeks.org/c-plus-plus/>
- W3schools. (2015, Abril 14). *C++ Tutorial*. W3Schools. Retrieved June 3, 2025, from <https://www.w3schools.com/cpp/>
- Deitel, H. M., & Deitel, P. J. (2016). *C++ Cómo Programar* (10ª ed.). Pearson Educación.
- Malik, D. S. (2018). *C++ Programming: From Problem Analysis to Program Design*. Cengage Learning.
- Stroustrup, B. (2013). *The C++ Programming Language* (4th ed.). Addison-Wesley.
- Downey, A. (2015). *Think C++: How to Think Like a Computer Scientist*. Green Tea Press.
- Ullman, J. D., & Aho, A. V. (1992). *Compilers: Principles, Techniques and Tools*. Addison-Wesley.
- IEEE (2022). *IEEE Standard Glossary of Software Engineering Terminology*.