



Universidad Mariano Galvez  
Ingeniería en sistemas

# Proyecto Final Crystal Collector

Programación 1

Mayo 29/2024



Profesor

**Walter Alexander  
Cordero Lara**

Alumno

**Carlos Humberto  
Interiano Garzona**  
0900-23-1399

## Contenido

Introducción.....	2
Análisis.....	3
Resumen del requerimiento:.....	3
En Crystal Collector, el jugador controla un avatar que se desplaza por distintos tableros con el objetivo de encontrar el portal que conduce al siguiente nivel. Para avanzar, el avatar debe recolectar cristales dispersos en cada tablero. A lo largo del juego, pueden aparecer Trolls o Enemigos que desafiarán al jugador con trivias. Si las respuestas son incorrectas, el jugador será obligado a reiniciar en el tablero anterior. ....	3
Entradas .....	4
Menú Principal .....	4
Juego .....	4
Salidas.....	5
Menú Principal .....	5
Juego .....	5
Procesos.....	6
Procesos principales del juego.....	6
- Start: Se ejecutará solo una vez para pedir nombre y genero al personaje e inicializará al jugador y el primer nivel.....	6
- Update: Sera el ciclo principal donde se ejecutará el juego y se actualizará cada vez que se detecte un input por parte del usuario.....	6
Subprocesos del Update .....	6
Restricciones .....	7
Diseño .....	8
Algoritmo en Pseudocodigo .....	8
Diagrama de flujo.....	12
Identificación de métodos .....	14

Diseño.....	16
Vista previa del laberinto .....	16
Vista previa de la trivia .....	17
Vista previa de pantalla de ganador .....	18
Vista previa de pantalla de perdedor .....	18
Referencias .....	19
Anexos .....	20

## Introducción

En este documento, se presenta el análisis y diseño de un sistema de juego de laberinto en consola, desarrollado en el lenguaje de programación C#. El objetivo del juego es que el jugador, representado por un personaje cuyo nombre y género se solicitan al inicio, recorra diferentes niveles de laberintos recolectando cristales y evitando obstáculos hasta encontrar el portal de salida.

El sistema incluye varios componentes clave: un menú principal donde el usuario puede iniciar el juego, una mecánica de juego que se actualiza con cada interacción del usuario, y niveles generados dinámicamente que desafían al jugador de diversas maneras. A lo largo del desarrollo, se han identificado y documentado las entradas, salidas, procesos y restricciones del juego para asegurar una implementación coherente y funcional.

El juego está diseñado para ser interactivo y desafiante, con características que incluyen la generación aleatoria de cristales y la aparición de trolls invisibles que agregan un elemento de sorpresa y dificultad. La interfaz del usuario en la consola mostrará el laberinto, el estado actual del jugador, y una caja recolectora de cristales, proporcionando una experiencia visual y textual completa.

Este documento detallará el análisis y diseño del sistema, presentando diagramas de flujo y pseudocódigo para ilustrar la lógica del juego. También se describirán los métodos identificados, su propósito y cómo se integran en la solución final. Además, se incluirán referencias y un manual de usuario para guiar tanto el uso del juego como su desarrollo. Finalmente, se proporcionará el código fuente debidamente documentado, junto con el ejecutable del programa, asegurando que el proyecto esté listo para su ejecución y evaluación.

## Análisis

### Resumen del requerimiento:

En Crystal Collector, el jugador controla un avatar que se desplaza por distintos tableros con el objetivo de encontrar el portal que conduce al siguiente nivel. Para avanzar, el avatar debe recolectar cristales dispersos en cada tablero. A lo largo del juego, pueden aparecer Trolls o Enemigos que desafiarán al jugador con trivias. Si las respuestas son incorrectas, el jugador será obligado a reiniciar en el tablero anterior.

## Entradas

### Menú Principal

- Enteros
  - Opción de usuario

### Juego

- Entero
  - Genero
    - 1 = Mujer
    - 2 = Hombre
  - Opción en trivia
- Cadena
  - Nombre de usuario
- Teclas
  - Axis X, Y
    - Izquierda (a o flecha izquierda)
    - Abajo (s o flecha abajo)
    - Derecha (d o flecha derecha)
    - Arriba (w o flecha arriba)
  - Escape

## Salidas

### Menú Principal

- Sin variable de salida

### Juego

- Puntaje
- Cristales faltantes
- Cristales recolectados
- Ubicación
- Nombre

## Procesos

### Procesos principales del juego

- Start: Se ejecutará solo una vez para pedir nombre y genero al personaje e inicializará al jugador y el primer nivel.
- Update: Sera el ciclo principal donde se ejecutará el juego y se actualizará cada vez que se detecte un input por parte del usuario.

### Subprocesos del Update

- DisplayLevel: Se encargará de imprimir el nivel.
- DisplayState: Se encargará de imprimir el estado del jugador.
- DisplayBox: Se encargará de imprimir la caja recolectora de cristales.
- AxisInput: Detectara la dirección a la que se busca mover a el jugador.
- AddPoints: Agrega o resta puntos dependiendo el evento.
- Movement: Permitirá o no el movimiento del jugador si hay una pared o portal inaccesible como en el nivel 1.
- GenerateNewLevel: Generara un nuevo nivel dependiendo lo que se le requiera



## Restricciones

1. El jugador debe recolectar todos los cristales he ir al portal
  - a. Si no tiene todos los cristales no puede acceder al portal y el portal se mueve exceptuando el nivel 1 donde el portal es fijo(ver la restricción 5).
2. En el primer nivel no aparecen trolls.
3. El personaje aparece aleatoriamente excepto en el nivel uno donde su posición será la (0, 0).
4. Los cristales se generarán de manera aleatoria y no podrán tocarse entre si exceptuando el nivel 1.
5. El portal de teletransporte se moverá de lugar si el jugador no tiene todos los cristales exepto en el nivel 1 donde aparecerá en la posición contraria al jugador.
6. Los trolls serán invisibles y en posición aleatoria
  - a. Los trols se mantienen activos hasta acabar el nivel solo cambian de posición si se responde correctamente
7. Los puntos pueden ser negativos.

## Diseño

El diseño del Pseudocodigo y diagrama de flujo está enfocado a los eventos posibles con los que se puede topar el jugador Como lo es un portal, obtener gemas y toparse con trolls que le harán trivias, Mientras que el apartado visual y funcional se enfocara en el diseño conceptual y el código final.

## Algoritmo en Pseudocodigo

### Pseudocodigo programa principal

Algoritmo CristalCollector

```
Definir op Como Entero
Mientras op <> 4 Hacer
    //Imprimir el menú
    Escribir "Menú";
    Escribir "1. Iniciar Partida";
    Escribir "2 Instrucciones..3 Informacion..4 Salir";
    Leer op;

    //Según la opción
    Segun op Hacer
        caso 1:
            //método Start
            Start(Argumentos);
        caso 2:
            Escribir "Información"
        caso 3:
            Escribir "Información";
        caso 4:
            Escribir "Saliendo";
    FinSegun
FinMientras
FinAlgoritmo
```

## Sub proceso del juego

SubProceso Start(Argumentos)

Definir nombre Como Caracter;

Definir genero Como Caracter;

Escribir "Nombre:";

Leer nombre;

Escribir "Avatar:";

Leer genero;

Definir play Como Logico;

Definir nivelActual Como Entero

Definir vidas Como Entero

Definir troll Como Entero;

Definir trivia Como Entero;

Definir gema Como Entero;

Definir cristales Como Entero;

Definir portal Como Entero;

play <- Verdadero;

nivelActual <- 1;

cristales <- 0;

Mientras play Hacer

//Aquí se imprimirá

//el Nivel

//Stats

//Caja recolectora

Escribir "Nivel";

Escribir "Estado del personaje";

Escribir "Caja Recolectora";

//Esto se encargará el programa no el usuario

Escribir "Hay portal 1. Si 2. no";

Leer portal;

Si portal == 1 Entonces

Si cristales >= 6 Entonces

nivelActual <- nivelActual+1;

FinSi

```

FinSi

Si portal <> 1 Entonces
    //Esto se encargará el programa no el usuario
    Escribir "Hay gema 1. Si 2. no";
    Leer gema;
    Si gema == 1 Entonces
        cristales <- cristales+1;
    FinSi
FinSi

Si portal <> 1 y gema <> 1 Entonces
    //Esto se encargará el programa no el usuario
    //Se realiza la trivía Si hay un troll en la posición
    Escribir "Hay un troll 1. Si 2. no";
    Leer troll;
    Si troll == 1 Entonces
        Escribir "Gano la trivía 1. Si 2. no";
        Leer trivia;
        Si trivia == 1 Entonces
            vidas <- vidas + 1;
            Si vidas >= 3 Entonces
                vidas <- 3;
            FinSi
        SiNo
            Si trivia == 2 Entonces
                vidas <- vidas-1;
                actual <- actual - 1;
            FinSi
        FinSi
    FinSi
FinSi

//Aquí se haría el conteo de puntos
//Aquí Si se permite el movimiento

//Si gano o perdió
Si(nivelActual > 5 o vidas < 0) Entonces
    play = Falso;

    Si(vidas<0) Entonces

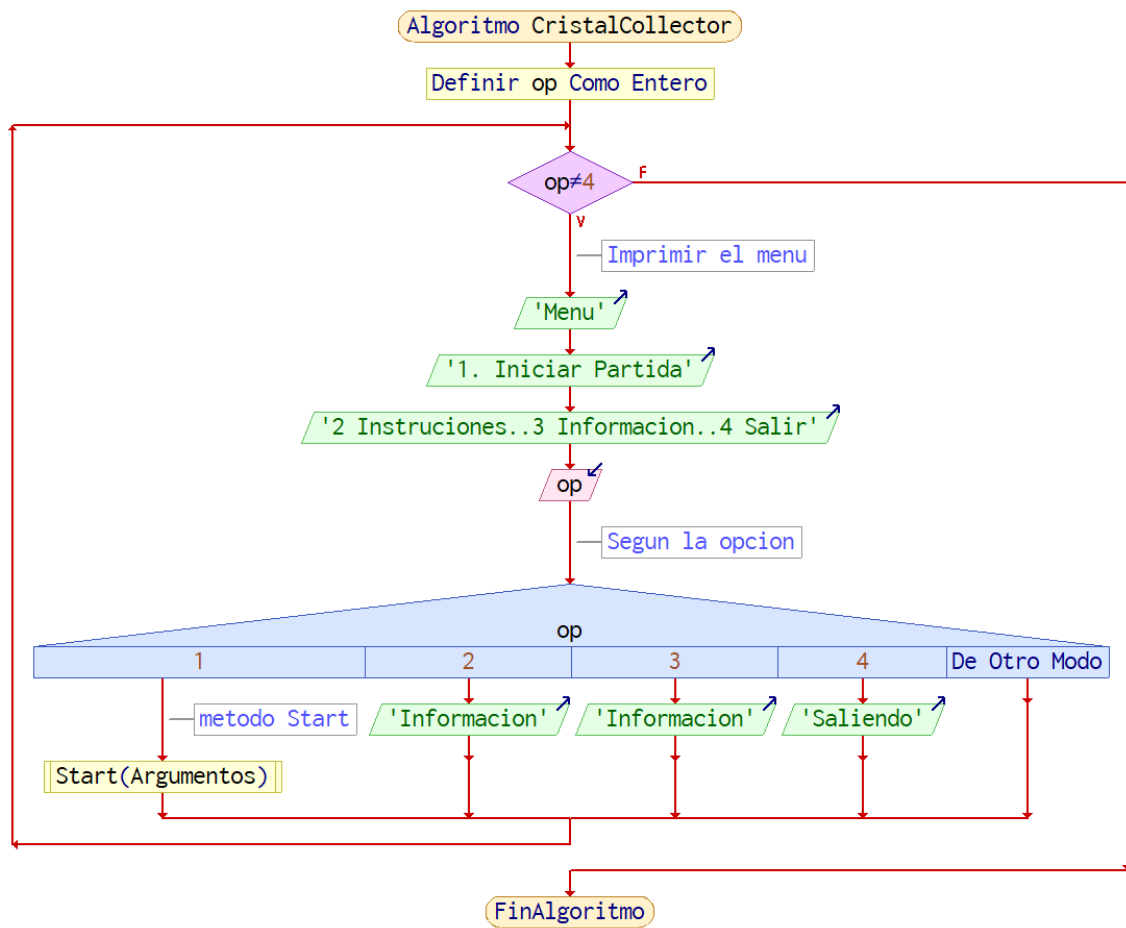
```

```
        Escribir "PERDISTE";  
    SiNo  
        Escribir "GANASTE";  
    FinSi  
    Escribir "Estado del personaje"  
FinSi  
FinMientras  
Fin SubProceso
```

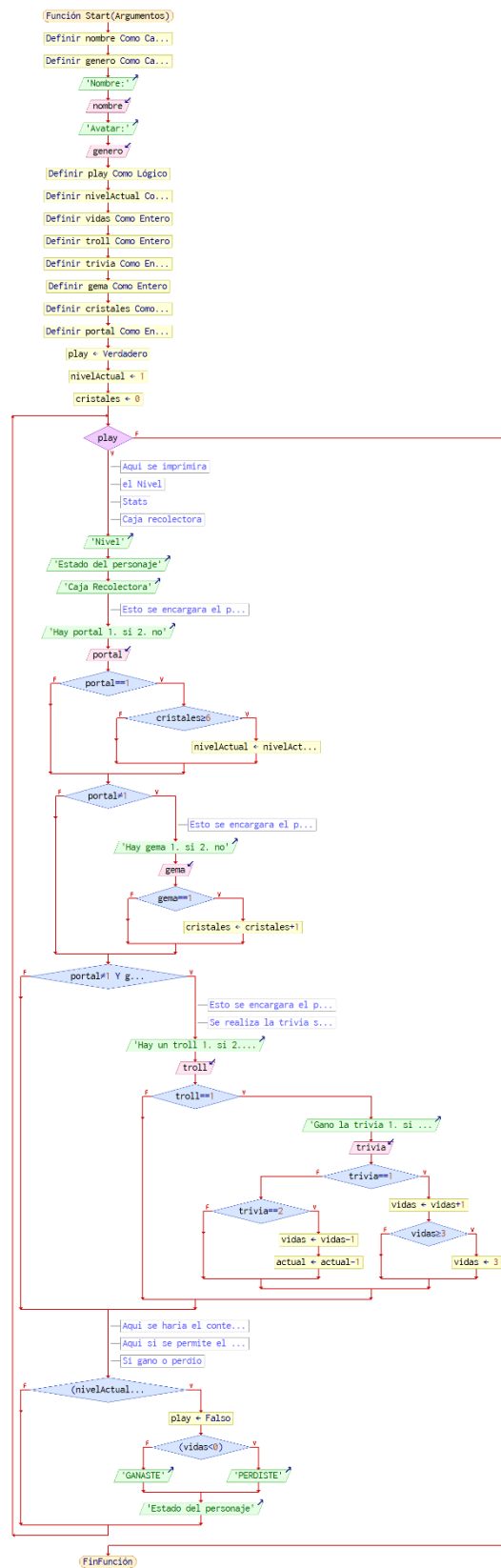
## Diagrama de flujo

### Programa principal

- Menú



## Juego Crystal Collector



## Identificación de métodos

### Start:

Descripción: Inicializa el juego solicitando el nombre y género del personaje.  
Configura el jugador y el primer nivel.

Entradas: Nombre de usuario, género (1 = Mujer, 2 = Hombre).

Salidas: Inicialización del jugador y nivel 1.

### Update:

Descripción: Ciclo principal del juego que se actualiza cada vez que se detecta un input del usuario.

Entradas: Teclas (Axis X, Y; Escape).

Salidas: Actualización del estado del juego y del jugador.

### Subprocesos del Update

#### DisplayLevel:

Descripción: Imprime el nivel actual en la pantalla.

Entradas: Estado del nivel.

Salidas: Representación visual del nivel.

#### DisplayState:

Descripción: Imprime el estado actual del jugador (puntaje, cristales faltantes, cristales recolectados, ubicación, nombre).

Entradas: Estado del jugador.

Salidas: Información del estado del jugador.

#### DisplayBox:

Descripción: Imprime la caja recolectora de cristales.



Entradas: Cristales recolectados.

Salidas: Representación de la caja recolectora.

AxisInput:

Descripción: Detecta la dirección en la que el jugador desea moverse.

Entradas: Teclas (Axis X, Y).

Salidas: Dirección del movimiento.

AddPoints:

Descripción: Modifica el puntaje del jugador según el evento.

Entradas: Puntos a agregar o restar.

Salidas: Actualización del puntaje del jugador.

Movement:

Descripción: Controla el movimiento del jugador, asegurándose de que no se mueva a través de paredes o portales inaccesibles.

Entradas: Dirección del movimiento, estado del nivel.

Salidas: Nueva posición del jugador (si el movimiento es permitido).

GenerateNewLevel:





Descripción: Genera un nuevo nivel con cristales, trolls y el portal en ubicaciones adecuadas.

Entradas: Parámetros para la generación del nivel.









Salidas: Nuevo estado del nivel.

Diseño









Vista previa del laberinto

-  = portal
-  = avatar hombre
-  = avatar mujer
-  = diamante

3x3

	0	1	2
0			
1			
2			

5x5

	0	1	2	3	4
0					
1					
2					
3					
4					

## Vista previa de la trivia



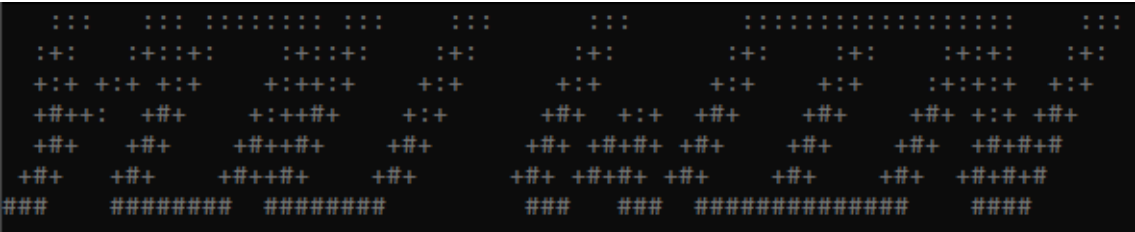
¡Oh no! Un enemigo

Si respondes correctamente te dejara pasar

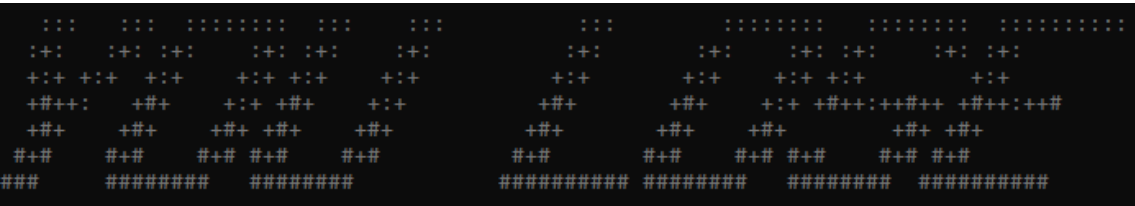
Pregunta¿?

1. Respuesta
2. Respuesta
3. Respuesta
4. Respuesta

Vista previa de pantalla de ganador



Vista previa de pantalla de perdedor



## Referencias

BillWagner. (s/f). *Operador ?: operador: operador condicional ternario*.

Microsoft.com. Recuperado el 31 de mayo de 2024, de

<https://learn.microsoft.com/es-es/dotnet/csharp/language-reference/operators/conditional-operator>

BitBoss [@BitBoss]. (2021, agosto 30). *Programación Orientada a Objetos*

(POO): Abstracción, Encapsulamiento, Herencia, Polimorfismo. Youtube.

<https://www.youtube.com/watch?v=SI7O81GMG2A>

dotnet-bot. (s/f-a). *Console.ForegroundColor Propiedad*. Microsoft.com.

Recuperado el 1 de junio de 2024, de <https://learn.microsoft.com/es-es/dotnet/api/system.console.foregroundcolor?view=net-8.0>

dotnet-bot. (s/f-b). *Console.OutputEncoding property*. Microsoft.com.

Recuperado el 1 de junio de 2024, de <https://learn.microsoft.com/en-us/dotnet/api/system.console.outputencoding?view=net-8.0>

dotnet-bot. (s/f-c). *Console.ResetColor method*. Microsoft.com. Recuperado el 1

de junio de 2024, de <https://learn.microsoft.com/en-us/dotnet/api/system.console.resetcolor?view=net-8.0>

dotnet-bot. (s/f-d). *String.Substring Método*. Microsoft.com. Recuperado el 31

de mayo de 2024, de <https://learn.microsoft.com/es-es/dotnet/api/system.string.substring?view=net-8.0>

## Anexos