Tercera entrega Proyecto final

Licenciatura en ingeniería de Software

Facultad de matemáticas UADY

14 de junio de 2020

Estudiante:

William Cetina

Gener Echeverria

Jesús Estrella

Profesor:

Edwin León

Curso:

Programación estructurada

Contenido

[Descripción del problema 3](#_Toc43049412)

[Requerimientos funcionales 4](#_Toc43049413)

[Generales 4](#_Toc43049414)

[Función **Pong** 4](#_Toc43049415)

[Función **Asteroids** 5](#_Toc43049416)

[Función **Preguntas** 6](#_Toc43049417)

[Función **Gato** 6](#_Toc43049418)

[Requerimientos no funcionales 7](#_Toc43049419)

[Casos de usos 8](#_Toc43049420)

[Flujo de eventos: Preguntas 8](#_Toc43049421)

[Flujo de eventos: Gato 11](#_Toc43049422)

[Flujo de eventos: Asteroids 15](#_Toc43049423)

[Flujo de eventos: Puntuaciones 19](#_Toc43049424)

[Flujo de eventos: Salir 19](#_Toc43049425)

[Pantallas del módulo. 20](#_Toc43049426)

[Plan de trabajo 26](#_Toc43049427)

[Repositorio 28](#_Toc43049428)

[Comunicación 28](#_Toc43049429)

[Monitonitoreo 28](#_Toc43049430)

[Herramientas 29](#_Toc43049431)

# Descripción del problema

El proyecto consiste en un juego llamado “Quicker” cuya mecánica principal es obtener el mayor puntaje de minijuegos completados en una sucesión rápida. Dichos minijuegos incrementan su dificultad a medida que aumenta el puntaje de minijuegos completados. Cuando el jugador falla en completar el minijuego, pierde una vida. Finalmente, el juego termina cuando el jugador pierde 3 vidas.

El juego tendrá 4 minijuegos:

* **Pong:** simulará un tenis donde el jugador tenga control de una paleta que se mueve verticalmente en la parte izquierda de la pantalla y competirá contra otra paleta del lado derecho. El jugador utiliza la raqueta para pegarle a la pelota y devolverla al otro lado. El jugador pierde si no logra devolver la pelota.
* **Preguntas:** presentará una pregunta sobre operaciones matemáticas en el cual el jugador seleccionar la respuesta. Pierde una vida si no responde en el dentro del tiempo límite o si su respuesta es errónea. Disminuye el tiempo límite para responder la pregunta a medida que aumenta el puntaje del jugador.
* **Asteroids:** El jugador controla de manera horizontal y vertical una nave cuyo objetivo es esquivar los asteroides que irán apareciendo en la parte superior de la pantalla con dirección descendente. El jugador obtiene un punto si logra esquivar los asteroides durante 10 segundos, de lo contrario, pierde una vida.
* **Gato:** también conocido como tres en línea o Tic Tac Toe, consiste en un tablero de 3x3 en donde el jugador selecciona la casilla donde colocará su símbolo (X), el jugador obtiene un punto si logra realizar una línea recta o diagonal de su símbolo, y pierde una vida si el programa logra formar una recta o diagonal con el símbolo ‘O’. En caso de surgir un empate, la vida y el puntaje no cambian.

Al terminar cada minijuego, se desplegará una pantalla de 2 segundos de duración que muestra la vida y el puntaje actual del jugador, y pasará a otro minijuego mientras el jugador tenga vidas.

# Requerimientos funcionales

## Generales

Al iniciar el programa, se despliega el menú principal del juego, el cual contiene las opciones **Empezar**, **Puntuaciones** y **Salir**,

En el menú principal, el programa reconocerá las teclas **W y S** para seleccionar (**W** = arriba**, S** = abajo) y **ENTER** para confirmar las opciones.

En el menú principal, al seleccionar la opción **Empezar** se ejecutará la función **Selec** para comenzar la partida.

En el menú principal, al seleccionar la opción **Puntuaciones** se desplegará un listado con los tres mejores puntajes obtenidos en el juego.

En el menú principal, al seleccionar la opción **Salir**, se termina el programa.

Al ejecutar la función **Selec,** se inicializa la vida en 3 y el puntaje en 0 y el programa comenzará un ciclo que mostrará pantalla de puntaje y se escogerá de manera aleatoria el minijuego (Función: **Pong**, **Asteroid**, **Preguntas** o **Gato**), que se ejecutará.

El programa aumentará en uno el puntaje del jugador cuando complete exitosamente un minijuego.

El programa disminuirá en uno el número de vidas del jugador cuando falle en completar un minijuego.

El programa cambiará el color de los caracteres en pantalla a amarillo si el número de vidas es dos.

El programa cambiará el color de los caracteres en pantalla a rojo si el número de vidas es uno.

Si la vida del jugador llega a cero, el programa detendrá el ciclo de minijuegos y **Selec** ejecutará la función **game over** seguido de la función **posiciónPuntaje**

La función **Game over** mostrará el puntaje total de minijuegos completados y el mensaje “Game over”

La función **posiciónPuntaje** guardará en un archivo .txt la puntuación obtenida en el juego si es mayor que alguna de las tres previamente guardadas.

## Función **Pong**

El programa imprime dentro de un campo rectangular de 75 columnas por 21 filas: una raqueta derecha de trece filas de altura, una izquierda de diez filas de altura y una pelota. Debajo del campo, imprime el tiempo restante.

El jugador moverá la raqueta izquierda hacia arriba al presionar la tecla **W** y hacia abajo al presionar la tecla **S**.

El programa moverá continuamente la raqueta derecha de arriba hacia abajo y viceversa.

El programa moverá la pelota de manera diagonal de un lado a otro del campo cada vez que una raqueta la toque.

El programa cambiará el tamaño de la raqueta izquierda a ocho filas si el puntaje del jugador es entre 3 y 5.

El programa cambiará el tamaño de la raqueta izquierda a seis filas si el puntaje del jugador es entre 6 y 8.

El programa cambiará el tamaño de la raqueta izquierda a cuatro filas si el puntaje del jugador es entre 9 y 11.

El programa cambiará el tamaño de la raqueta izquierda a tres filas si el puntaje del jugador es mayor a 11.

La función **Pong** termina y el programa aumenta en uno el puntaje del jugador si la pelota toca el límite derecho del campo o transcurren 30 segundos sin que la pelota toque el límite izquierdo.

La función **Pong** termina y el programa disminuye en uno el número de vidas del jugador si la pelota toca el límite izquierdo del campo

## Función **Asteroids**

El programa imprime dentro de un campo rectangular de 25 columnas y 60 filas: una nave situada en la parte inferior y un asteroide cada seis filas que desciende desde la parte superior del campo. Debajo del campo, imprime el tiempo restante.

El programa desaparece los asteroides a medida que tocan el límite inferior del campo.

El jugador moverá la nave hacia arriba con la tecla **W,** hacia abajo con **S,** hacia la izquierda con **A** y a la derecha con **D**.

El programa genera un asteroide cada cinco filas de la pantalla si el puntaje del jugador es 2 o 3.

El programa genera un asteroide cada cuatro filas de la pantalla si el puntaje del jugador es 4 o 5.

El programa genera un asteroide cada tres filas de la pantalla si el puntaje del jugador es 6 o 7.

El programa genera un asteroide cada dos filas de la pantalla si el puntaje del jugador es 8 o 9.

El programa genera un asteroide en cada fila de la pantalla si el puntaje del jugador es 10 o mayor.

La función **Pong** termina y aumenta en uno el puntaje del jugador si la nave no toca algún asteroide durante 10 segundos.

La función **Pong** termina y disminuye en uno el número de vidas del jugador si la nave toca algún asteroide.

## Función **Preguntas**

El programa muestra en pantalla de manera aleatoria una de diez preguntas con sus respectivas tres opciones de respuesta y el tiempo restante.

El jugador escogerá su respuesta usando tecla **W** para subir entre las opcionesy la tecla Spara bajar.

El jugador confirmará su respuesta escogida al presionar la tecla **Enter.**

Cada pregunta tiene una única respuesta correcta.

La función **Preguntas** termina y el programa aumenta en uno el puntaje del jugador si la respuesta seleccionada es correcta.

La función **Preguntas** termina y el programa disminuye en uno el número de vidas del jugador si su respuesta es errónea o no contesta en 10 segundos.

## Función **Gato**

El programa imprime un tablero 3x3 que se actualiza después de cada turno.

El programa coloca la ficha ´X´ en la casilla disponible que el jugador ha seleccionado.

El programa coloca la ficha ´O´ en alguna de las casillas disponibles.

El programa evalúa en cada turno si se ha formado alguna recta o diagonal de la misma ficha.

La función **gato** termina y el programa aumenta en uno el puntaje del jugador si logra formar una recta o diagonal con tres símbolos ‘X’.

La función **gato** termina y el programa disminuye en uno las vidas del jugador si se forma una recta o diagonal con tres símbolos ‘O’.

La función **gato** termina y el programa mantiene el puntaje y la vida del jugador previo al minijuego si no se forma ninguna recta o diagonal del mismo símbolo.

# Requerimientos no funcionales

El programa podrá ser ejecutado en equipos con compiladores de lenguaje C.

El programa únicamente podrá funcionar si el compilador cuenta con todas las librerías incluidas en el código.

El programa es desarrollado con las herramientas que la programación en C ofrece.

El programa debe estar disponible para todo aquel interesado en experimentar su contenido.

El programa necesita ser ejecutado en la misma ubicación que el archivo Puntuaciones.txt para poder acceder a la función **Puntuaciones**

Los textos que aparezcan en la pantalla deberán ser totalmente legibles y entendibles para el usuario que esté ejecutando el programa.

El programa se encuentra en idioma español.

# Casos de usos

## Flujo de eventos: Preguntas

|  |  |  |  |
| --- | --- | --- | --- |
| **PASO** | **ACTOR** | **DESCRIPCIÓN** | **EVENTO DEL SISTEMA** |
| 1 | Máquina | Se activa la pantalla principal. | Se despliega la pantalla principal con el logo del juego (QUICKER) y se muestran 3 opciones: Empezar, Puntuaciones y Salir.  **Pantalla 1.1.** |
| 2 | Usuario | El usuario escoge la opción **Empezar.** | El usuario mueve las teclas hasta la opción “Empezar” y presiona la tecla “Enter”.  **Pantalla 1.1.** |
| 3 | Máquina | El usuario seleccionó la opción **Empezar.** | La función **Selec** despliega la función **Puntaje.** |
| 4 | Máquina | Se muestra el puntaje y las vidas actuales. | La función **Selec** despliega la función **puntaje** y sus valores iniciales.  **Pantalla 2.** |
| 5 | Máquina | Selecciona el juego que ejecutara. | Mediante un algoritmo, la función **Selec** escoge aleatoriamente uno de los cuatros minijuegos. |
| 6 | Máquina | **Selec** selecciona el minijuego **Preguntas** | La función **Selec** selecciona la opción “0”, la cual, corresponde al minijuego de preguntas, luego ejecuta la función **Preguntas.** |
| 7 | Máquina | Comienza el minijuego **Preguntas**. | El programa ejecuta la función **Preguntas** y muestra aleatoriamente una de 10 preguntas con tres opciones de respuesta y activa un cronómetro de 10 segundos.  **Pantalla 3.1.1.** |
| 8 | Usuario | Escoge su respuesta | El usuario escoge una de las tres opciones y presiona la tecla **Enter.**  Si la respuesta es correcta aparece el mensaje “Correcto”  **Pantalla 3.1.2.**  Si la respuesta es incorrecta aparece el mensaje “Incorrecto”  **Pantalla 3.1.3.**  Si el usuario no responde en 10 segundos aparece el mensaje “Se acabó el tiempo”.  **Pantalla 3.1.4.** |
| 9 | Máquina | La máquina comprueba la respuesta seleccionada por el jugador. | Tras recibir la respuesta del usuario, el programa comprueba si es la correcta o no. Posterior a esto regresa a la pantalla de puntaje |
| 10 | Máquina | La máquina verifica el resultado final y modifica su vida o puntaje | Dependiendo al resultado del juego el programa efectuara una de estas acciones:  1)Si el usuario acierta se le agrega un punto a su puntaje  **Pantalla 4.**  2)Si el usuario falla se le resta una de sus vidas  **Pantalla 5 y 6**  Si el usuario mantiene como mínimo una de sus 3 vidas entonces se regresa la función **selec** para seleccionar otro minijuego (paso 3), en caso contrario se activa la función **Game over.** |
| 11 | Máquina | Se muestra durante cuatro segundos el mensaje “Game over” con la puntuación final obtenida. | El sistema ejecuta la función **Game over** y muestra el número de minijuegos completados (Puntaje).  Se activa la función **posición Puntaje.**  **Pantalla 7.** |
| 12 | Máquina | Verificar puntuaciones | La función **posición Puntaje,** comprueba si el puntaje obtenido es mayor que alguna de las tres guardadas y se realiza una de las dos opciones:  1)Si su puntaje es mayor, se solicita el nombre del jugador y se actualiza la tabla de Puntuaciones.  2)Si su puntaje es menor, el programa regresa a la pantalla principal (Paso 1) |
| 13 | Máquina | El programa solicita el nombre del usuario. | El programa abre una entrada por teclado.  **Pantalla 8.** |
| 14 | Usuario | El usuario escribe su nombre. | El usuario teclea el nombre que quiera grabar y presione la tecla **Enter.** |
| 15 | Máquina | Guarda su nombre, y muestra durante cuatro segundos la tabla de Puntuaciones actualizada. | La máquina graba su nombre, lo imprime en la tabla de posiciones de acuerdo a la posición que este alcanzó por su puntaje y guarda los datos en el archivo Puntuaciones.txt, posterior a esto regresa al menú principal (caso 1).  **Pantalla 9.** |

## Flujo de eventos: Gato

|  |  |  |  |
| --- | --- | --- | --- |
| **PASO** | **ACTOR** | **DESCRIPCIÓN** | **EVENTO DEL SISTEMA** |
| 1 | Máquina | Se activa la pantalla principal | Se despliega la pantalla principal con el logo del juego (QUICKER) y se muestran 3 opciones Empezar, Puntuaciones y Salir  **Pantalla 1.1** |
| 2 | Usuario | El usuario escoge la opción **Empezar.** | El usuario mueve las teclas hasta la opción “Empezar” y presiona la tecla “Enter”.  **Pantalla 1.1** |
| 3 | Máquina | El usuario seleccionó la opción **Empezar.** | La función **Selec** despliega la función **Puntaje.** |
| 4 | Máquina | Se muestra el puntaje y las vidas actuales. | La función **Selec** despliega la función **puntaje** y sus valores iniciales.  **Pantalla 2.** |
| 5 | Máquina | Selecciona el juego que ejecutara. | Mediante un algoritmo, la función **Selec** escoge aleatoriamente uno de los cuatros minijuegos. |
| 6 | Máquina | **Selec** selecciona el minijuego **Gato** | La función **Selec** selecciona la opción “1”, la cual, corresponde al minijuego de preguntas, luego ejecuta la función **Gato.** |
| 7 | Máquina | Se genera la tabla para el juego Gato y se abre una entrada por teclado. | La función **Gato** genera un tablero de 3x3. El programa coloca una ficha ‘O’ en alguna de las casillas disponibles de la tabla y abre una entrada por teclado para el usuario.  **Pantalla 3.2.1.** |
| 8 | Usuario | El usuario coloca su ficha | El usuario escribe el número de la casilla en donde quiere colocar su ficha ‘X’ y presiona la tecla “Enter” para actualizar la tabla.  **Pantalla 3.2.2** |
| 9 | Máquina | El programa verifica si se forman líneas de tres fichas iguales. | El programa compara si se ha formado alguna recta con tres fichas iguales.   1. Si el usuario forma primero una línea con 3 fichas ‘X’, usuario gana.   **Pantalla 3.2.3.**   1. Si la máquina forma primero una línea de 3 fichas ‘O’, el usuario pierde.   **Pantalla 3.2.4.**   1. Si se llenan las casillas y ninguno formó una línea de 3 entonces se considera empate.   **Pantalla 3.2.5.**  Si no se ha formado una recta, se repiten el paso 7 y 8 hasta un máximo de tres veces. |
| 10 | Máquina | La máquina verifica el resultado final y modifica su vida o puntaje | Dependiendo al resultado del juego el programa efectuara una de estas acciones:  1)Si el usuario ganó se le agrega un punto a su puntaje  **Pantalla 4.**  2)Si el usuario perdió se le resta una de sus vidas  **Pantalla 5 y 6**  3)Si el usuario empato no se modifica ni la vida ni la puntuación.  Si el usuario mantiene como mínimo una de sus 3 vidas entonces se regresa la función **selec** para seleccionar otro minijuego (paso 3), en caso contrario se activa la función **Game over.** |
| 11 | Máquina | Se muestra durante cuatro segundos el mensaje “Game over” con la puntuación final obtenida. | El sistema ejecuta la función **Game over** y muestra el número de minijuegos completados (Puntaje).  Se activa la función **posición Puntaje.**  **Pantalla 7.** |
| 12 | Máquina | Verificar puntuaciones | La función **posición Puntaje,** comprueba si el puntaje obtenido es mayor que alguna de las tres guardadas y se realiza una de las dos opciones:  1)Si su puntaje es mayor, se solicita el nombre del jugador y se actualiza la tabla de Puntuaciones.  2)Si su puntaje es menor, el programa regresa a la pantalla principal (Paso 1) |
| 13 | Máquina | El programa solicita el nombre del usuario. | El programa abre una entrada por teclado.  **Pantalla 8.** |
| 14 | Usuario | El usuario escribe su nombre. | El usuario teclea el nombre que quiera grabar y presione la tecla **Enter.** |
| 15 | Máquina | Guarda su nombre, y muestra durante cuatro segundos la tabla de Puntuaciones actualizada. | La máquina graba su nombre, lo imprime en la tabla de posiciones de acuerdo a la posición que este alcanzó por su puntaje y guarda los datos en el archivo Puntuaciones.txt, posterior a esto regresa al menú principal (caso 1).  **Pantalla 9.** |

## Flujo de eventos: Asteroids

|  |  |  |  |
| --- | --- | --- | --- |
| **PASO** | **ACTOR** | **DESCRIPCIÓN** | **EVENTO DEL SISTEMA** |
| 1 | Máquina | Se activa la pantalla principal | Se despliega la pantalla principal con el logo del juego (QUICKER) y se muestran 3 opciones Empezar, Puntuaciones y Salir  **Pantalla 1.** |
| 2 | Usuario | El usuario escoge la opción **Empezar.** | El usuario mueve las teclas hasta la opción “Empezar” y presiona la tecla “**Enter**”.  **Pantalla 1.1** |
| 3 | Máquina | El usuario seleccionó la opción **Empezar.** | La función **Selec** despliega la función **Puntaje.** |
| 4 | Máquina | Se muestra el puntaje y las vidas actuales. | La función **Selec** despliega la función **puntaje** y sus valores iniciales.  **Pantalla 2.** |
| 5 | Máquina | Selecciona el juego que ejecutara. | Mediante un algoritmo, la función **Selec** escoge aleatoriamente uno de los cuatros minijuegos. |
| 6 | Máquina | **Selec** selecciona el minijuego **Asteroids** | La función **Selec** selecciona la opción “2”, la cual, corresponde al minijuego de preguntas, luego ejecuta la función **Asteroid.** |
| 7 | Máquina | Corre el juego **Asteroids** | El programa ejecuta la función **Asteroid,** donde imprime una navecita que inicia en la parte baja de la pantalla, y un asteroide cada seis líneas que va descendiendo desde la parte superior de la pantalla. Se inicia un cronómetro de 10 segundos.  **Pantalla 3.3** |
| 8 | Usuario | El usuario mueve la nave. | Con las teclas **A, W, S** y **D** el usuario mueve la nave hacia la izquierda, arriba, abajo y derecha respectivamente. |
| 9 | Máquina | Verificación del Estado del juego. | Durante 10 segundos los asteroides descienden en la pantalla y se comprueba que la nave del usuario no toque a ninguno.  Si se termina el tiempo y la nave no tocó ningún meteorito, se detiene el minijuego y se cuenta como victoria; de lo contrario, se termina el minijuego y el jugador pierde |
| 10 | Máquina | La máquina verifica el resultado final y modifica su vida o puntaje | Dependiendo al resultado del juego el programa efectuara una de estas acciones:  1)Si el usuario ganó se le agrega un punto a su puntaje  **Pantalla 4.**  2)Si el usuario perdió se le resta una de sus vidas  **Pantalla 5 y 6**  3)Si el usuario empato no se modifica ni la vida ni la puntuación.  Si el usuario mantiene como mínimo una de sus 3 vidas entonces se regresa la función **Selec** para seleccionar otro minijuego (paso 3), en caso contrario se activa la función **Game over.** |
| 11 | Máquina | Se muestra durante cuatro segundos el mensaje “Game over” con la puntuación final obtenida. | El sistema ejecuta la función **Game over** y muestra el número de minijuegos completados (Puntaje).  Se activa la función **posición Puntaje.**  **Pantalla 7.** |
| 12 | Máquina | Verificar puntuaciones | La función **posición Puntaje,** comprueba si el puntaje obtenido es mayor que alguna de las tres guardadas y se realiza una de las dos opciones:  1)Si su puntaje es mayor, se solicita el nombre del jugador y se actualiza la tabla de Puntuaciones.  2)Si su puntaje es menor, el programa regresa a la pantalla principal (Paso 1) |
| 13 | Máquina | El programa solicita el nombre del usuario. | El programa abre una entrada por teclado.  **Pantalla 8.** |
| 14 | Usuario | El usuario escribe su nombre. | El usuario teclea el nombre que quiera grabar y presione la tecla **Enter.** |
| 15 | Máquina | Guarda su nombre, y muestra durante cuatro segundos la tabla de Puntuaciones actualizada. | La máquina graba su nombre, lo imprime en la tabla de posiciones de acuerdo a la posición que este alcanzó por su puntaje y guarda los datos en el archivo Puntuaciones.txt, posterior a esto regresa al menú principal (caso 1).  **Pantalla 9.** |

## Flujo de eventos: Puntuaciones

|  |  |  |  |
| --- | --- | --- | --- |
| **PASO** | **ACTOR** | **DESCRIPCIÓN** | **EVENTO DEL SISTEMA** |
| 1 | Máquina | Se activa la pantalla principal | Se despliega la pantalla principal con el logo del juego (QUICKER) y se muestran 3 opciones Empezar, Puntuaciones y Salir  **Pantalla 1.1** |
| 2 | Usuario | El usuario escoge la opción **Puntuaciones** | El usuario mueve las teclas hasta la opción “Puntuaciones” y presiona la tecla “Enter”.  **Pantalla 1.2** |
| 3 | Máquina | El usuario seleccionó la opción Puntuaciones | La función Selec despliega la función **imprimir Puntuaciones.** |
| 4 | Máquina | Se muestra los mejores tres puntajes junto con el nombre de su respectivo jugador (usuario). | La función imprimir Puntuaciones despliega los datos guardados en el archivo Puntuaciones.txt  **Pantalla 10** |
| 5 | Usuario | El usuario presiona cualquier tecla. | El usuario presiona cualquier tecla para cerrar la función **imprimir Puntaje.** |
| 6 | Máquina | El programa regresa al Menú principal | El programa regresa al Menú principal (Paso 1). |

## Flujo de eventos: Salir

|  |  |  |  |
| --- | --- | --- | --- |
| **PASO** | **ACTOR** | **DESCRIPCIÓN** | **EVENTO DEL SISTEMA** |
| 1 | Máquina | Se activa la pantalla principal | Se despliega la pantalla principal con el logo del juego (QUICKER) y se muestran 3 opciones Empezar, Puntuaciones y Salir  **Pantalla 1.1.** |
| 2 | Usuario | El usuario escoge la opción Salir | El usuario mueve las teclas hasta la opción “Salir” y presione la tecla “**Ente**r”.  **Pantalla 1.3.** |
| 3 | Máquina | Se cierra el programa. | Finaliza el programa.  **Pantalla 11.** |

## Pantallas del módulo.



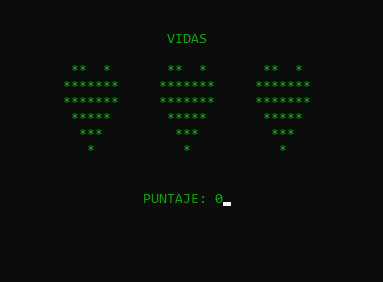
Pantalla 1.1



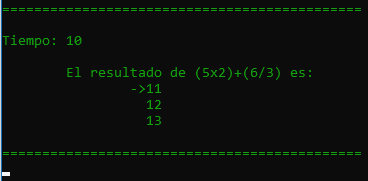
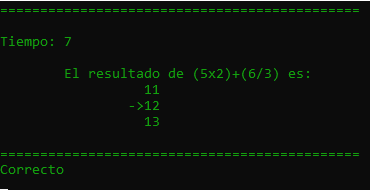
Pantalla 1.2



Pantalla 1.3

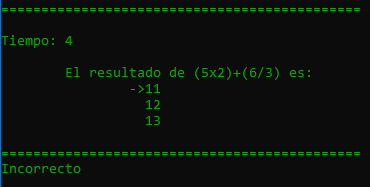
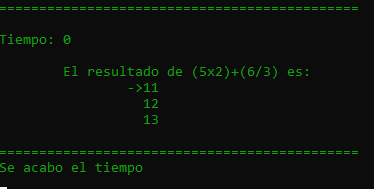


Pantalla 2



Pantalla 3.1.2

Pantalla 3.1.1

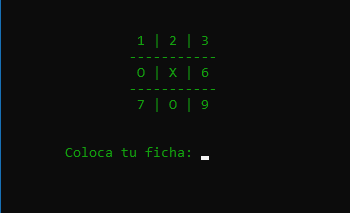


Pantalla 3,1.1

Pantalla 3.1.4

Pantalla 3.1.3

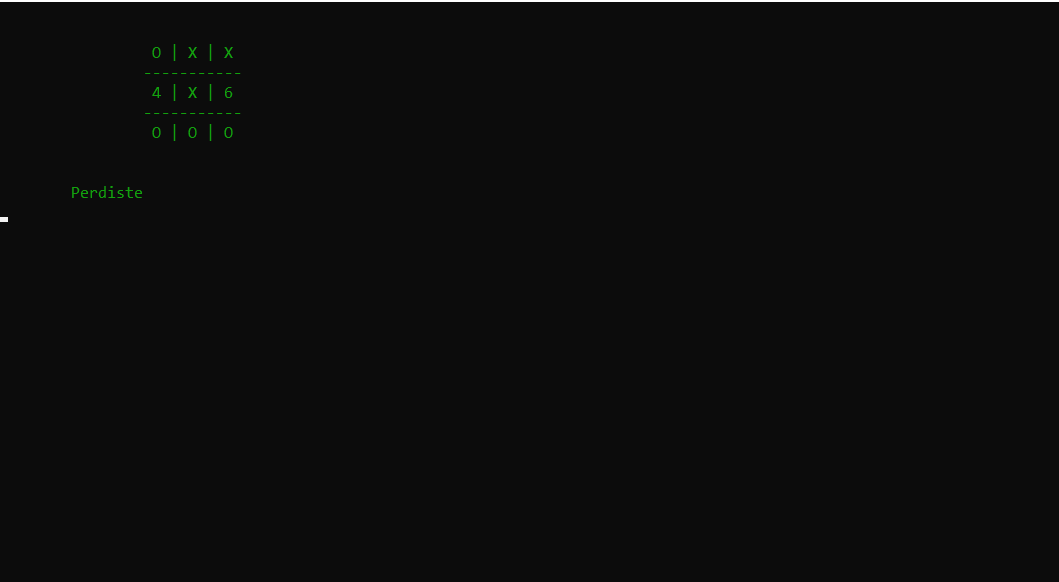




.

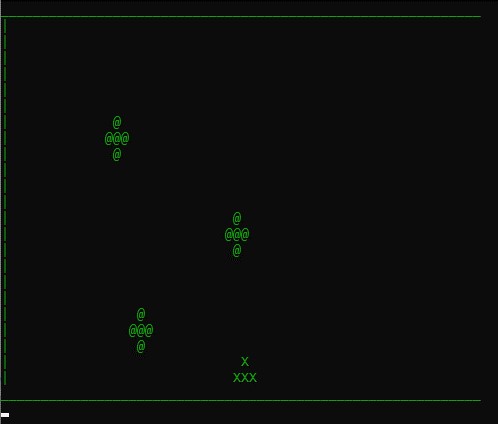
Pantalla 3.2.2

Pantalla 3.2.1



Pantalla 3.2.4

Pantalla 3.2.3





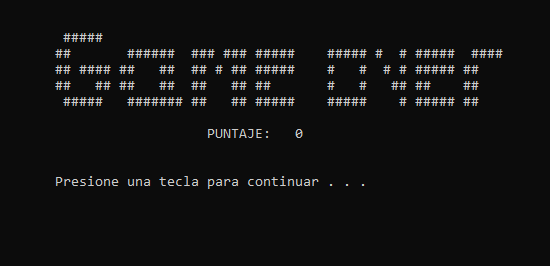
Pantalla 3.2.5

Pantalla 3.3.



Pantalla 3.4



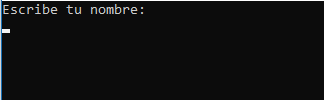


Pantalla 7

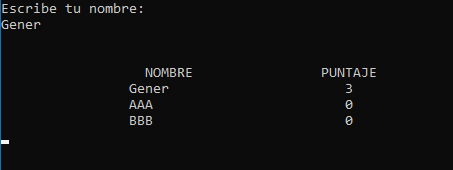
Pantalla 6

Pantalla 5

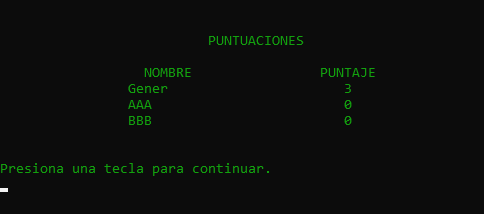
Pantalla 4



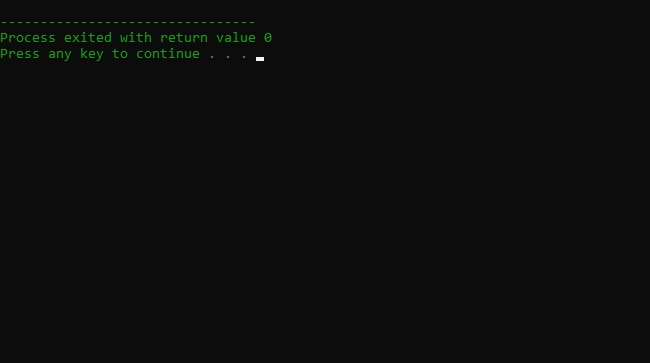
Pantalla 8.



Pantalla 9.



Pantalla 10.



Pantalla 11.

# Plan de trabajo

## Calendarización

|  |  |  |
| --- | --- | --- |
| Actividades | Productos/Artefactos resultantes | Responsable |
| Junta para definir roles y variables  17 de febrero |  | Todo el equipo. |
| Desarrollo de pantalla de inicio y estructura general (Función main y selec).  18 abril | Función main.  Función selec | Gener Echeverría |
| Junta para revisión de Función main y división de los minijuegos.  19 abril | Documentación Función main. | Todo el equipo |
| Programación funciones anexas  21 abril. | Función Pantalla inicio  Función Puntaje.  Función GameOver | Gener Echeverría |
| Programación de cada minijuego.  21 a 30 de abril. | Función Pong.  Función Asteroids.  Función Preguntas.  Función Gato. | Pong: Jesús Estrella  Asteroids: William Cetina  Preguntas: Gener Echeverría  Gato: Gener Echeverría |
| Junta para revisión de minijuegos.  1 de mayo de marzo. | Documentación Pong.  Documentación Asteroids.  Documentación Preguntas  Documentación Gato | Junta: Todo el equipo |
| Actualización función Asteroid  5 de mayo | Segunda versión de Asteroid (movimiento vertical) | William Cetina |
| Integración de funciones anexas a función main.  11 de mayo | Versión sin minijuegos. | Gener Echeverría |
| Actualización función Pong  12 de mayo. | Segunda versión de Pong (exportar resultado de la partida) | Jesús Estrella |
| Integración de minijuegos a función main.  12 de mayo | Primera versión jugable | Gener Echeverría |
| Revisión de primera versión jugable  13 de mayo | Actualización de primera versión jugable | Jesús Estrella |
| Revisión de documentación  14 de mayo | Actualización del documento | Todo el equipo |
| Junta de testeo.  20 de mayo. | Feedback. | Todo el equipo |
| Nueva función: Puntuaciones  26 de mayo. | Función Puntuaciones. | Gener Echeverría |
| Actualización de Asteroids  10 de junio | Función Asteroid actualizada | William Cetina |
| Actualización de Preguntas  11 de junio. | Función Preguntas actualizada | Gener Echeverría y William Cetina |
| Actualización de Pong  12 de junio | Función Pong actualizada | Gener Echeverría |
| Prueba Quicker  12 de junio. | Versión de entrega de Quicker. | Todo el equipo. |
| Manual de usuario.  7-14 de junio. | Manual de Usuario | Información: Gener Echeverría y Jesús Estrella  Edición: William Cetina |
| Correcciones Documento de Especificación de Requerimientos  7-14 de junio. | Documento de Especificación de Requerimientos | Información: William Cetina y Gener Echeverría.  Edición: Gener Echeverría |

## Repositorio

Se decidió el uso de GitHub como repositorio ya que es una herramienta profesional que permite trabajar en colaboración con otras personas, planificar proyectos y realizar un seguimiento del trabajo. En el apartado del proyecto se agregaron todos aquellos documentos que se requerirán para elaborar el trabajo, así como links de vídeos, productos y toda aquella información que nos ayude en el proceso.

<https://github.com/GenerEcheverria/Estructurada>

## Comunicación

Para expresar opiniones, compartir ideas o discutir puntos clave del proyecto, se busca una interacción eficaz entre los diferentes miembros del grupo mediante el uso de las plataformas descritas en el apartado de herramientas y por medio de las reuniones indicadas en el calendario. Se utilizó un grupo de WhatsApp, uno de Discord y reuniones presenciales para manifestar comentarios, resolver dudas o hacer aclaraciones que puedan surgir entre reuniones con el objetivo de reducir errores y retrasos.

## Monitoreo

Las diferentes tareas y actividades requeridas para cumplir con nuestros objetivos se asignan a cada integrante con base en sus habilidades y áreas de oportunidad. Esto se podrá adecuar y modificar de acuerdo con la opinión grupal y coevaluación que se realiza en cada reunión.

De igual forma, el calendario de actividades servirá como guía principal en el orden de las actividades que se lleven acaba. Sin embargo, si el equipo decide que es necesario realizar modificaciones al mismo, se harán las modificaciones necesarias.

## Herramientas

* Google forms: (Para trabajar de manera simultánea documentación del proyecto).
* WhatsApp: (Aplicación utilizada para mantener contacto grupal las 24 horas y resolver dudas menores).
* Discord <https://discord.gg/4W83yP> (El objetivo de usar esta plataforma es mantener comunicación eficaz durante la elaboración de documentos de manera no presencial).
* Trello h[ttps://trello.com/invite/b/cGjtn6SA/7185b1827e89084c7b39e801114292bc/estructurada](https://trello.com/invite/b/cGjtn6SA/7185b1827e89084c7b39e801114292bc/estructurada) (El uso de esta herramienta surge por la necesidad de llevar una mejor organización de las actividades a realizar, asignar roles y estar al tanto del estatus de cada tarea).