

# Proyecto final PacMan

AUTORES:

FARYD STEBAN WALTEROS MOSQUERA

JUAN PABLO GOMEZ SEQUEDA

*Universidad Nacional De Colombia, Departamento de  
ingeniería .*

*Bogota D.C - Colombia*

**Resumen:** -A lo largo del semestre hemos visto y practicado diferentes tipos de habilidades relacionados a la programación de los computadores , en busca de engrandecer y retroalimentar los conocimientos previos o en algunos casos nulos sobre los computadores , desde lo más básico como el lenguaje de programación y cómo trabaja una computadora hasta las funciones y capacidades que tienen estas .

Términos claves : Programación , Computadores , Git , Github , Juego , Pacman , Menú de usuario , Interfaz Gráfica , sistema de puntos .

Abstract - Throughout the semester we have seen and practiced different types of skills related to computer programming, seeking to enhance and feed back previous or in some cases null knowledge about computers, from the basics of programming language and how a computer works to the functions and capabilities of computers.

Key terms : Programming , Computers , Git , Github , Game , Pacman , User menu , Graphical interface , Point system .

## ***I. introducción.***

-Este documento busca reflejar los conocimientos adquiridos a lo largo del semestre; Teniendo en cuenta las herramientas, habilidades e intelecto que se fueron trabajando en la materia **Programación de**

**computadores** a cargo del docente *Nestor German Bolivar Pulgarin*

*-Mediante un trabajo escrito con el que se busca complementar y dar fundamentos a un proyecto final .*

*-El proyecto final consiste en un juego completamente funcional con un menú de usuario , un sistema de puntos y evidentemente una interfaz gráfica.*

*-El juego toma inspiración de PacMan .*

## ***II. Desarrollo.***

### ***A. ¿CÓMO ES PACMAN ?***

**-PAC-Man** es un clásico videojuego arcade lanzado en 1980. El jugador controla a Pac-Man, una figura amarilla que debe recorrer un laberinto, comiendo puntos mientras evita a los fantasmas que lo persiguen. Al consumir "píldoras energéticas", Pac-Man puede comer a los fantasmas temporalmente. El objetivo es comer todos los puntos en el laberinto sin ser atrapado por los fantasmas. El juego se vuelve más difícil a medida que avanzan los niveles, con los fantasmas volviéndose más rápidos y agresivos.

### ***B. LIBRERÍAS:***

¿que es una librería?

En programación una librería es un conjunto de funciones que permite al usuario llevar a cabo nuevas tareas que antes no eran posibles, es una implementación que permite codificar el lenguaje creando una interfaz nueva e independiente. ()

#### ***Librerías a usar:***

Utilizaremos el lenguaje de programación python por ende usaremos librerías en este lenguaje,

librerías que nos permitan la posibilidad de generar colisiones, sonido, inteligencia artificial de los fantasmas, movimiento y potenciales (que son los que permiten al jugador tener la posibilidad de eliminar a los fantasmas) las cuales son:

1. **INTERFAZ:** AL ABRIR EL JUEGO QUEREMOS REPRESENTAR DE INICIO UNA INTERFAZ O MENÚ QUE SEA FÁCILMENTE ENTENDIBLE DONDE LE DEMOS AL JUGADOR DIFERENTES OPCIONES TANTO DE INICIAR EL JUEGO COMO DE MIRAR LA CONFIGURACIÓN, EL SCORE Y LA POSIBILIDAD DE SALIR DEL JUEGO SI ES QUE SE DESEA, ESTA INTERFAZ QUEREMOS QUE SEA SIMPLE POR ESO UTILIZAREMOS PYGAME Y LA EXTENSIÓN DE ESTA LLAMADA PYGAME GUI, LA CUAL ESTÁ ESPECIALIZADA EN EL DISEÑO DE INTERFACES EN BASE AL LENGUAJE DE PROGRAMACIÓN PYTHON Y LA LIBRERÍA PYGAME.

#### **VENTAJAS:**

- Utilizamos esta interfaz puesto que gran parte del proyecto se planea diseñar en pygame y utilizar esta extensión nos facilita el trabajo.
- Da facilidades en la personalización.
- Presenta una interfaz fácil de entender permitiendo un diseño óptimo.
- Nos permite tener control total del programa.
- La dificultad al utilizar esta librería es relativamente fácil lo que nos permite trabajar de mejor manera teniendo en cuenta nuestro nivel de programación

#### **DESVENTAJAS:**

- Limitación frente a la búsqueda de diseños únicos y diferentes, creando una brecha de originalidad.
- Fuerte dependencia de pygame.
- Interfaz gráfica de usuario limitada, esta es básica y

poco potente lo cual puede generar problemas en el desarrollo del videojuego.

A pesar de estas desventajas consideramos que es lo más óptimo tomando en cuenta nuestro nivel de programación y el proyecto a realizar, si presentamos problemas con esta librería planteamos la posibilidad de utilizar otras como arcade o tkinter.

2. **Colisiones y hitbox:** Para las colisiones utilizaremos PyGame, esta librería es muy reconocida ya que proporciona un conjunto de funciones y módulos que facilitan el desarrollo de videojuegos 2D.

#### ***¿Por qué utilizar esta y no otras librerías?***

Consideramos que para el desarrollo de pac-Man necesitamos una librería de fácil acceso y fácil entendimiento que sea eficiente en el término de desarrollo de videojuegos en 2d, pygame es perfecto para esto además de darnos un control total acerca de las físicas que sostendremos en el desarrollo del videojuego, encontrando así abundantes recursos inclusive más de los necesarios para el proyecto.

Utilizaremos la librería principalmente para la detección de “hitbox” y colisiones siendo muy importante para el desarrollo del juego ya que el requerimiento que deseamos para que se pueda avanzar de nivel es la NO colisión de pac-Man con los fantasmas, junto a este requerimiento, planteamos la posibilidad de que pac-Man pueda al comerse un punto potenciador tenga la posibilidad de eliminar a los fantasmas, pero que son las colisiones y el “hitbox”.

#### ***¿Qué es el Hitbox?***

El Hitbox básicamente es un espacio que ocupa nuestro personaje y en este caso los

fantasmas que buscamos evadir, este espacio suele ser representado como un cuadrado o caja de allí el termino “hitbox”, aquí es donde entra pygame con esta librería podemos definir una hitbox para el personaje de pac-Man y otras para los fantasmas de forma sencilla con la función `pygame.rect` como ejemplo:

```
## Crear un rectángulo para el jugador x y ancho al
player = pygame.Rect(50, 50, 40, 40)

## Crear un rectángulo para los fantasmas
fantasma1 = pygame.Rect(50, 50, 40, 40)
fantasma2 = pygame.Rect(50, 50, 40, 40)
fantasma3 = pygame.Rect(50, 50, 40, 40)
```

Ya creadas las hitboxes queremos que en nuestro juego el choque o colisión de la hitbox del personaje con el de los fantasmas sea el fin del juego.

### ¿Qué son las Colisiones?

una colisión es una intersección de dos o más objetos en pygame es el mecanismo computacional que verifica la intersección utilizamos la función `.colliderect` para verificar si hay una colisión, como ejemplo:

```
##verificar la colicion con fantasmas
if player.colliderect (fantasma1):
    print ("colicion deterctada")
```

esto es esencial para la dinámica del juego a presentar luego de definir la colisión y la hitbox () desarrollaremos más acerca del movimiento.

3. **Movimiento:** Para el movimiento tanto de nuestro personaje como de los fantasmas también utilizamos la librería pygame pues consideramos que es la mas optima para complementar este proyecto, la librería de pygame cuenta con diferentes funciones de movimiento tanto para el personaje como el de los fantasmas un pequeño ejemplo:

```
## Configurar la velocidad del personaje
## adicionando a lo que anteriormente fue los datos
## del personaje

velocidad = 5

## Capturar teclas presionadas
teclas = pygame.key.get_pressed()
if teclas[K_UP]:
    player.y -= velocidad
if teclas[K_DOWN]:
    player.y += velocidad
if teclas[K_LEFT]:
    player.x -= velocidad
if teclas[K_RIGHT]:
    player.x += velocidad
```

4. **Desarrollo de niveles:** también utilizaremos pygame para el desarrollo del juego, esta librería incluye funciones para:

- Renderizar imágenes y sprites.
- Reproducir sonidos o música de fondo.
- Dibujar formas básicas como rectángulos, círculos, y líneas para construir niveles minimalistas.

5. **Datos y optimización:** para los datos y su registro utilizaremos pygame y Cython. las utilizaremos para el registro de datos de la partida con puntaje e historial de este mismo, guardar configuraciones como la dificultad, o movimientos.

En conclusión nuestro proyecto toca fuertes bases en la librería de python pygames puesto que consideramos que es la más completa de acuerdo con nuestro nivel de programación, esta entrega control total del proyecto, además de ayudarnos tanto en físicas como en menú, diseño de niveles, puntaje score o memoria e interfaz, es una librería muy completa y simple que no nos presenta muchos problemas en su uso y entendimiento .

## C. GIT Y GITHUB

### ❖ ¿Qué es Git?

Git es un software de control de versiones, pensado en la eficiencia, la confiabilidad y compatibilidad del mantenimiento de versiones de aplicaciones cuando esta tiene un gran número de

archivos de código fuente.

git se basa en el flujo de trabajo usando de manera óptima:

1. el uso del historial.
2. la seguridad y continuidad.
3. el registro continuo.
4. la función eficiente.

### ***Historia:***

Una de las grandes cualidades de Git es que te permite tomar decisiones en el último momento. Puede decidir qué archivos entran en juego antes de comprometerse con el área de ensayo, puedes decidir que no querías estar trabajando en algo todavía, y puedes reescribir confirmaciones que ya hayan pasado haciendo parecer que fueron hechas de diferente manera.

### ***Seguridad:***

Cada usuario esencialmente tiene una copia de seguridad completa del servidor principal. Cada una de estas copias podría ser empujada para reemplazar el servidor principal en caso de un bloqueo o corrupción. Esto representa una ventaja frente a otros softwares de control de versiones, puesto que en vez de verificar y modificar el servidor principal este genera una copia que es distribuida a todos los usuarios y modificadores del proyecto, evitando así problemas sin retorno o el estropeo del servidor principal.

Debido a esta misma naturaleza el programa git permite casi que un flujo de trabajo infinito, dotando de variedad a la forma de trabajo.

El modelo de datos que utiliza Git garantiza la integridad criptográfica de cada bit del proyecto, También es imposible cambiar cualquier archivo, fecha, mensaje de confirmación o cualquier otro datos en un repositorio Git sin cambiar los ID de todo después de él.

### ***Registro:***

El registro continuo en Git implica una serie de prácticas y pasos que aseguran que los cambios en el código se registren de manera regular y detallada. donde puedes agregar archivos, registrar commits y trabajar en ramas, Para mantener el flujo de trabajo

organizado, es una buena práctica trabajar en ramas. Pues se puede crear una nueva rama para cada nueva característica o corrección que queremos trabajar.

También cuenta con la posibilidad de generar un “push” siendo esta la función básica para subir archivos a un repertorio o al servidor principal, esté “push” suele generarse en un trabajo remoto para luego ser subido al servidor principal.

Al igual que podemos subir o hacer un “push” también podemos traer la información a un dispositivo remoto esta función es nombrada “pull” donde descarga y procesa lo que anteriormente fue subido ya sea por nosotros mismos como otro participante del proyecto esto para hacer cambios o revisiones.

### ***❖ ¿Qué es Github?***

GitHub es una plataforma que permite a desarrolladores gestionar proyectos de software mediante el uso de Git, un sistema de control de versiones. Es una herramienta clave en el desarrollo colaborativo, ya que facilita la organización, el seguimiento de cambios en el código, y la colaboración entre equipos o con la comunidad global.

***Características:*** algunas de las características más importantes de github son:

- ***Repositorios:*** Espacios donde se almacena el código, junto con su historial de cambios. Pueden ser públicos (accesibles para todos) o privados.
- ***Control de versiones:*** Permite registrar cada modificación, revertir cambios no deseados y gestionar diferentes ramas de un proyecto para trabajar en paralelo.
- ***Colaboración:*** Funciones como “pull request” ayudan a proponer cambios, revisarlos y discutirlos antes de integrarlos al proyecto principal.
- ***Gestión de proyectos:*** Ofrece herramientas como issues para rastrear errores o tareas pendientes y tableros Kanban para visualizar el progreso.

Esta herramienta es muy importante para el trabajo

en equipo puesto que presenta muchas virtudes, tanto como el trabajo colaborativo como el hecho de descargar y generar repositorios donde de manera autónoma cada uno de los participantes puede poseer un rol frente al proyecto que se desea realizar, y puede mostrar en forma detallada el avance el tiempo de trabajo de cada componente facilitando así el trabajo realizado.

Aparte de estas virtudes también comprende una gran biblioteca donde la comunidad puede exponer sus trabajos o información para que el resto de la comunidad puedan documentar, complementar o estudiar en base a esta, tiene la capacidad de generar muchas ramas de trabajo donde la comunidad puede desarrollarse para resolver problemas, generar proyectos, experimentar códigos y conocer los lenguajes.

## **D. FIRESTORE Y REALDATE**

### **❖ ¿Que es firestore?**

Firestore es una base de datos flexible y escalable que permite a los desarrolladores almacenar y sincronizar datos en tiempo real para aplicaciones web, móviles y de servidor. Utiliza un modelo de datos basado en documentos que facilita la organización y consulta de la informa

Firestore funciona mediante la colección de documentos. Una colección es un grupo de documentos de base de datos, cada uno con un identificador único. Los documentos contienen campos clave-valor que representan los datos almacenados. Esta estructura jerárquica facilita la organización y consulta eficiente de datos.

- **Documentos:** Cada documento en Firestore tiene un identificador único y contiene campos con sus respectivos valores. Esto permite una estructura flexible y escalable para almacenar datos.
- **Colecciones:** Las colecciones son grupos de documentos que comparten un tema o tipo similar de datos. Las consultas pueden realizarse a nivel de colección para recuperar múltiples documentos relacionados.

- **Escucha en tiempo real:** Una característica destacada de Firestore es la capacidad de sincronización en tiempo real, lo que significa que los cambios realizados en la base de datos se reflejan instantáneamente en todas las instancias conectadas a ella.

### **❖ ¿Qué es realtime?**

Se refiere a cualquier sistema de procesamiento de información con componentes de software y hardware que realiza funciones de aplicación en tiempo real y puede responder a los eventos con limitaciones de tiempo predecibles y específicas. Entre los ejemplos comunes de sistemas en tiempo real se incluyen los sistemas de control de tráfico aéreo, los sistemas de control de procesos y los sistemas de conducción autónoma.

Para que un sistema en tiempo real sea capaz de realizar cálculos en tiempo real, debe cumplir con dos requisitos:

- **Puntualidad:** capacidad de producir el resultado esperado en un plazo específico.
- **Sincronización de tiempo:** capacidad de los agentes de coordinar los relojes independientes y operar en conjunto en unísono.
- **Latencia:** medición del tiempo entre dos eventos.
- **Inestabilidad informática:** variación de la latencia entre iteraciones.

### **¿Que diferencia existe entre estos dos ?**

Firestore y Realtime Database son dos servicios de bases de datos en tiempo real ofrecidos por Firebase, una plataforma de Google para el desarrollo de aplicaciones. Aunque ambos se usan para almacenar y sincronizar datos entre dispositivos y usuarios en tiempo real, tienen diferencias clave en términos de arquitectura, funcionalidad y casos de uso.

1. **Estructura de Datos.**
2. **Consultas y Filtrado.**

3. *Escalabilidad.*
4. *Sincronización y Rendimiento.*
5. *Seguridad.*

### ***III. Historias de usuario con sus respectivos requerimientos***

Historias Usuarios	Requerimientos
-Como jugador quiero primordialmente un menú básico y sencillo muy fácil de entender.	<p>- Se necesita diseñar una interfaz gráfica (Menú) .</p> <p><i>1.Para esto podemos realizarlo mediante PyGame .</i></p> <p>-La interfaz gráfica debe tener un mínimo de 3 botones;</p> <p><i>1.Jugar</i></p> <p><i>2.Controles</i></p> <p><i>3.Salir</i></p> <p>Donde si presionamos el botón <i>Jugar</i> se abran otras opciones , como;</p> <p><i>1.Dificultad</i></p> <p><i>2.Mapas</i></p> <p>Si presionamos el botón de controles se mostrará en la ventana los respectivos controles , que en nuestro caso se podrá jugar con dos sets de controles ;</p> <p><i>1.W,A,S,D ;</i></p> <p><i>W:arriba</i></p> <p><i>A:izquierda</i></p> <p><i>S:abajo</i></p> <p><i>D:derecha</i></p> <p><i>2.con las flechitas del teclado</i></p>

	Si presionamos el botón de <i>SALIR</i> el juego automáticamente se cerrará y dejará de funcionar.
-Como jugador quiero una variedad de mapas	<p>- Al presionar el botón de <i>mapas</i> se van a desplegar 3 imágenes (.jpg) , donde se muestran los mapas a disposición;</p> <p><i>1.Mapa unal:Este mapa va a ser basado en la universidad nacional de colombia.</i></p> <p><i>2.Mapa Colombia:Este mapa va a estar basado en el país colombia , con elementos alusivos a su cultura.</i></p> <p><i>3.Mapa Rio:Mapa inspirado en Río de Janeiro.</i></p>
-Como jugador quiero ver la puntuación que tengo en cada partida	<p>-Se coloca un <i>score point</i> que recolecta la cantidad de puntos conseguidos en cada partida.</p> <p>-Los puntos se ganan mediante</p> <p><i>1.Coins</i></p> <p><i>2.Fruits</i></p> <p><i>3.Ghost</i></p>

#### ***Extras***

Como agregado final queremos agregar un mapa adicional que se va a desbloquear al llegar a más de 150 puntos , en este mapa el laberinto va a cambiar , va a cambiar su dirección , ya no va a hacer de manera cuadrada con puntos distribuidos aleatoriamente en el mapa , si no que va a hacer un mapa vertical en el cual se tiene que llegar en

menos de 3 minutos a la meta que está en la parte de abajo. con fantasmas más inteligentes y rápidos.

## SOPORTES

- Este video va a ser usado como guía :  
[https://www.youtube.com/watch?v=hebtgq99sBg&t=394s&ab\\_channel=CódigoEspinoza-DesarrollodeVideoJuegos](https://www.youtube.com/watch?v=hebtgq99sBg&t=394s&ab_channel=CódigoEspinoza-DesarrollodeVideoJuegos)
- Stack Overflow es un sitio web de preguntas y respuestas en el que los desarrolladores pueden plantear sus dudas y obtener respuestas de otros miembros de la comunidad. El funcionamiento se basa en un sistema de reputación, donde los usuarios acumulan puntos por su participación y la calidad de sus respuestas

## Términos Claves

-Programación: Formar programas, previa declaración de lo que se piensa hacer y anuncio de las partes de que se ha de componer un acto o espectáculo o una serie de ellos.

-Computadores: Máquina electrónica que, mediante determinados programas, permite almacenar y tratar información, y resolver problemas de diversa índole.

-Git : Git es un sistema de control de versiones distribuido, gratuito y de código abierto, diseñado para gestionar con rapidez y eficacia desde proyectos pequeños a muy grandes.

-GitHub: GitHub es una plataforma de control de versiones a la que se accede de forma online, es decir, que no lo instalas en tu ordenador, si no que se accede desde Internet.

-Juego :Programa informático que sirve para jugar en un dispositivo electrónico adecuado.

-PacMan: Juego japonés.

-Menú de usuario: La **interfaz de usuario en juegos** es todo lo que permite al jugador interactuar con el juego. Esto incluye la interfaz y cualquier componente que los jugadores puedan usar para navegar, encontrar información y lograr sus objetivos.

-Interfaz Gráfica:Las interfaces gráficas de usuario abarcan desde barras de salud, indicadores de recursos hasta mapas, menús y mensajes. Proporcionan datos vitales sobre la progresión del juego, el estado del personaje, misiones actuales y otros elementos esenciales.

-Sistema de puntos:Es una interfaz gráfica que le informa al usuario su puntuación en juego.

## REFERENCIAS

- Programar : <https://Rae.es/programar>
- Computadores : <https://dle.rae.es/computador>
- Git : <https://git.com>
- Github : <https://docs.github.com/es>
- Juego : <https://dle.rae.es/videojuego?m=form>
- PacMan : <https://es.wikipedia.org/wiki/Pac-Man>
- Menú de usuario : [https://es.wikipedia.org/wiki/Menú\\_Inicio](https://es.wikipedia.org/wiki/Menú_Inicio)
- interfaz : [https://es.wikipedia.org/wiki/Interfaz gráfica de usuario](https://es.wikipedia.org/wiki/Interfaz_gráfica_de_usuario)
- Sistemadepuntos : <https://dictionary.net/definition/score+points>

## Key Terms

-Programming: Forming programmes, after stating what is to be done and announcing the parts of which an act or performance or series of acts or performances is to be composed.

-Computers: Electronic machines that, by means of certain programs, allow information to be stored and processed, and problems of various kinds to be solved.

-Git: Git is a free, open source, distributed version control system designed to manage small to very large projects quickly and efficiently.

-GitHub: GitHub is a version control platform that is accessed online, i.e. you don't install it on your computer, you access it from the Internet.

-Game: A computer programme used to play games on a suitable electronic device.

-PacMan: Japanese game.

-User interface: The user interface in games is everything that allows the player to interact with the game. This includes the interface and any components that players can use to navigate, find information and achieve their goals.

-Graphical user interfaces range from health bars and resource indicators to maps, menus and messages. They provide vital

data on game progression, character status, current quests and other essential elements.

Point system: A graphical interface that informs the user of their in-game score.

## REFERENCES

- Programar : <https://Rae.es/programar>
- Computadores : <https://dle.rae.es/computador>
- Git : <https://git.com>
- Github : <https://docs.github.com/es>
- Juego : <https://dle.rae.es/videojuego?m=form>
- PacMan : <https://es.wikipedia.org/wiki/Pac-Man>
- Menú de usuario : [https://es.wikipedia.org/wiki/Menú\\_Inicio](https://es.wikipedia.org/wiki/Menú_Inicio)
- interfaz :
- [https://es.wikipedia.org/wiki/Interfaz\\_gráfica\\_de\\_usuario](https://es.wikipedia.org/wiki/Interfaz_gráfica_de_usuario)
- Sistemadepuntos :
- <https://dictionary.net/definition/score+points>