



ESCUELA POLITÉCNICA NACIONAL
ESCUELA DE FORMACIÓN DE TECNÓLOGOS

ALGORITMOS Y ESTRUCTURAS DE DATOS



ASIGNATURA:

Algoritmos y Estructuras de Datos

PROFESOR:

Ing. Loarte Byron

PERÍODO ACADÉMICO:

2025-B

INFORME PROYECTO FINAL

TÍTULO:

Desarrollo de Videojuego Interactivo con Pygame

INTEGRANTES

- KARLA QUIMBITA
- JORDAN ZUÑA
- HAZIEL MONCAYO
- JAVIER PINARGOTE
- SELENA TORRES

FECHA DE REALIZACIÓN: 02/02/2026

FECHA DE ENTREGA: 04/02/2026

CALIFICACIÓN OBTENIDA:

FIRMA DEL PROFESOR:

1 PROPÓSITO DEL PROYECTO

El propósito de este proyecto es desarrollar un videojuego interactivo que permita aplicar los conocimientos aprendidos en la materia de Algoritmos y Estructuras de Datos. A través de la creación del videojuego, se busca comprender mejor cómo funciona la programación y cómo puede utilizarse para resolver problemas de manera práctica.

Este proyecto también tiene como objetivo mejorar las habilidades del estudiante en lógica, concentración y resolución de problemas, utilizando la programación de una forma más dinámica y motivadora. Además, se pretende demostrar que el aprendizaje puede ser más interesante cuando se aplica en actividades reales, como el desarrollo de un videojuego, fortaleciendo así el trabajo en equipo y el uso responsable de la tecnología.

2 OBJETIVOS ESPECÍFICOS

- Aplicar los conocimientos básicos de programación en Python para el desarrollo de un videojuego interactivo.
- Utilizar algoritmos y estructuras de datos para controlar el funcionamiento del videojuego, como el movimiento del jugador, los enemigos y el sistema de puntajes.
- Desarrollar habilidades de lógica y resolución de problemas mediante la implementación de funciones y estructuras de control.
- Fomentar el trabajo en equipo y la organización durante el desarrollo del proyecto.
- Demostrar que la programación puede ser utilizada de forma práctica y dinámica para crear soluciones tecnológicas.

3 DESARROLLO Y RESULTADOS

3.1 Organización y trabajo en equipo

El presente proyecto fue realizado de manera grupal, cumpliendo con lo indicado en la asignatura. Para el desarrollo del videojuego, el grupo se organizó distribuyendo las actividades de forma equitativa, permitiendo que todos los integrantes participen activamente en el proceso.

Cada integrante colaboró en distintas etapas del proyecto, como la planificación, la programación, la prueba del videojuego y la corrección de errores. Durante el trabajo se mantuvo una comunicación constante, lo que ayudó a resolver dificultades y a tomar decisiones en conjunto. Esta forma de trabajo permitió reforzar el compañerismo, la responsabilidad y el respeto por las ideas de los demás.

El trabajo en equipo fue fundamental para cumplir con los objetivos del proyecto y para lograr un producto funcional dentro del tiempo establecido.

3.2 Problemática a solucionar

En el aprendizaje de la programación, muchos estudiantes presentan dificultades para comprender cómo aplicar los conceptos vistos en clase a situaciones reales. Cuando la enseñanza se basa únicamente en la teoría, el aprendizaje se vuelve más complicado y poco motivador, lo que genera desinterés y problemas al momento de desarrollar programas.

3.3 Solución de la problemática

Como solución a esta problemática, se desarrolló un **videojuego interactivo utilizando Python y la librería Pygame**. Este videojuego permite aplicar los conocimientos de programación de forma práctica, ayudando a los estudiantes a comprender mejor el uso de algoritmos, estructuras de datos y la lógica de programación, al mismo tiempo que hace el aprendizaje más dinámico y entretenido.

3.4 Selección de la tecnología

Para el desarrollo del proyecto se eligió utilizar **Python con la librería Pygame**, cumpliendo con lo solicitado en el documento del examen. Esta tecnología fue seleccionada porque permite crear videojuegos de forma sencilla y clara, además de facilitar la aplicación de los conceptos aprendidos en la asignatura. Durante el desarrollo del videojuego se utilizaron estructuras condicionales, ciclos, listas y funciones, así como el manejo de eventos del teclado y del mouse. También se implementó un sistema de puntajes, lo cual permitió guardar y mostrar los resultados obtenidos por el jugador. El uso de Pygame permitió desarrollar un videojuego interactivo y funcional, demostrando que la tecnología seleccionada es adecuada para dar solución a la problemática planteada.

3.5 Funcionamiento y desarrollo del videojuego

El videojuego desarrollado es de tipo interactivo y permite al usuario controlar una nave mediante el teclado. Durante el juego, el jugador puede desplazarse y disparar para eliminar a los enemigos que aparecen en la pantalla. A medida que avanza el juego, la dificultad aumenta, lo que exige mayor atención y rapidez por parte del jugador.

El sistema responde de manera inmediata a las acciones del usuario, actualizando el puntaje y controlando el avance de los niveles. El videojuego finaliza cuando el jugador colisiona con un enemigo o cuando un enemigo alcanza el límite inferior de la pantalla. Este funcionamiento permite una interacción constante entre el usuario y el sistema.

3.5 Fragmentos importantes del código

1. **Creación del jugador:** Esta parte del código crea al jugador del videojuego. Se encarga de cargar la imagen de la nave, definir su tamaño, su posición inicial en la pantalla y su velocidad. También prepara una lista para los disparos y controla el tiempo entre cada disparo.

```
class Nave:
    def __init__(self):
        self.imagen_original = pygame.image.load("nave.png").convert_alpha()
        self.imagen = pygame.transform.scale(self.imagen_original, (50, 40))
        self.rect = self.imagen.get_rect()
        self.rect.centerx = ANCHO // 2
        self.rect.bottom = ALTO - 10
        self.velocidad = 5
        self.lasers = []
        self.cool_down = 0
```

- Movimiento del jugador con el teclado:** Permite mover la nave a la izquierda y a la derecha usando el teclado. Además, evita que la nave salga de los límites de la pantalla. Esta parte hace que el videojuego sea interactivo, ya que responde a las acciones del jugador.

```
def mover(self, teclas):  
    if teclas[pygame.K_LEFT] and self.rect.left > 0:  
        self.rect.x -= self.velocidad  
    if teclas[pygame.K_RIGHT] and self.rect.right < ANCHO:  
        self.rect.x += self.velocidad
```

- Disparo del jugador:** Este código permite que la nave dispare cuando el jugador presiona la tecla correspondiente. También controla el tiempo entre disparos para evitar que se dispare de forma continua sin control.

```
def disparar(self):  
    if self.cool_down == 0:  
        nuevo_laser = Laser(self.rect.centerx - 2, self.rect.top, -10)  
        self.lasers.append(nuevo_laser)  
        self.cool_down = 20
```

- Creación y movimiento de los enemigos:** En esta parte del proyecto se implementó la creación de los enemigos, definiendo su posición, tamaño y movimiento dentro del videojuego. Los enemigos se desplazan hacia abajo en la pantalla, aumentando la dificultad conforme avanza el juego, lo que representa el principal desafío para el jugador.

```
class Enemigo:  
    def __init__(self, x, y):  
        try:  
            self.imagen_orig = pygame.image.load("enemigo.png").convert_alpha()  
            self.imagen = pygame.transform.scale(self.imagen_orig, (40, 30))  
            self.usa_imagen = True  
        except:  
            self.usa_imagen = False  
  
        self.rect = pygame.Rect(x, y, 40, 30)  
        self.color = (random.randint(150, 255), 50, 50)  
  
    def mover(self, vel_y):  
        self.rect.y += vel_y
```

- Detección de colisiones (disparo vs enemigo):** Detecta cuando un disparo impacta a un enemigo o cuando un enemigo choca con el jugador. Si un enemigo es alcanzado, se elimina y se suman puntos. Si el enemigo toca al jugador, el juego termina.

```
if e.rect.colliderect(jugador.rect) or e.rect.bottom > ALTO:  
    corriendo_juego = False  
for l in jugador.lasers[:]:  
    if l.rect.colliderect(e.rect):  
        enemigos.remove(e)  
        jugador.lasers.remove(l)  
        puntos += 10  
        break
```

6. **Sistema de puntajes guardado en archivo:** Guarda el puntaje más alto del jugador en un archivo de texto. Esto permite que el récord se conserve incluso cuando el juego se cierra.

```
def obtener_max_puntaje():
    try:
        with open("puntaje.txt", "r") as f:
            return int(f.read())
    except:
        return 0

def guardar_puntaje(nuevo):
    max_actual = obtener_max_puntaje()
    if nuevo > max_actual:
        with open("puntaje.txt", "w") as f:
            f.write(str(nuevo))
```

4 CONCLUSIONES

- Mediante el desarrollo del videojuego interactivo se logró aplicar de forma práctica los conocimientos básicos de programación en Python, cumpliendo el objetivo de utilizar la programación como una herramienta funcional y no solo teórica.
- La implementación del movimiento de la nave, los enemigos y el sistema de puntajes permitió fortalecer la lógica y la capacidad de resolución de problemas, ya que fue necesario analizar errores y realizar mejoras durante el desarrollo del proyecto.
- El uso de algoritmos y estructuras de datos facilitó la organización del código y el correcto funcionamiento del videojuego, lo que permitió comprender mejor cómo estructurar un programa completo.
- El trabajo realizado en equipo contribuyó a una mejor organización y distribución de tareas, permitiendo cumplir con los objetivos planteados dentro del tiempo establecido.
- A través del desarrollo del proyecto se adquirieron nuevos conocimientos relacionados con el uso de la librería Pygame y el manejo de eventos, lo cual reforzó el aprendizaje práctico visto en la clase anterior.