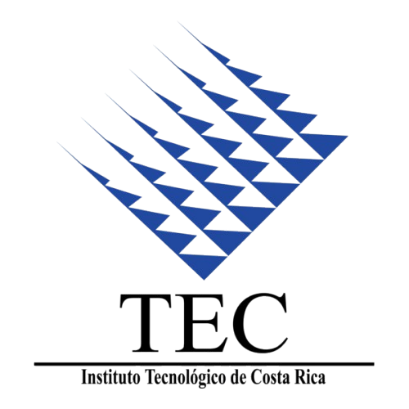
Instituto Tecnológico de Costa Rica

Unidad de Computación



“Space Invanders”

April Nayhara Meléndez Vásquez y

Marlon Fabián Aragón Arce

Sede San Carlos

Miércoles 11 de junio 2025

Tabla de contenidos

[Introducción 3](#_Toc200493792)

[Análisis del problema 3](#_Toc200493793)

[Solución del problema 4](#_Toc200493794)

[1. Etapas de la solución 4](#_Toc200493795)

[2. Pseudocódigo de la lógica central 5](#_Toc200493796)

[3. Código desarrollado 6](#_Toc200493797)

[Análisis de resultados 14](#_Toc200493798)

[Conclusiones 15](#_Toc200493799)

[Recomendaciones 15](#_Toc200493800)

[Referencias bibliográficas 16](#_Toc200493801)

[Participación del grupo 17](#_Toc200493802)

# Introducción

El presente documento expone el desarrollo de una versión personalizada del clásico videojuego Space Invaders, realizado como proyecto de la asignatura actual. La actividad se planteó con el objetivo de aplicar de forma práctica los conocimientos adquiridos durante el curso, utilizando como herramienta principal el lenguaje de programación Python, asignado por el profesor para guiar el aprendizaje. Python fue seleccionado no solo por ser uno de los lenguajes más populares y utilizados en la industria actual, sino también por su capacidad de adaptarse tanto a proyectos sencillos como a sistemas complejos, lo que lo convierte en una excelente opción para quienes se están iniciando en el mundo de la programación.

A través del desarrollo de este proyecto se fortalecieron diversas habilidades técnicas y blandas. En el ámbito técnico, los estudiantes enfrentaron desafíos relacionados con la lógica de programación, la estructuración del código, el manejo de eventos y la implementación de interfaces visuales. Además, la programación de un videojuego como Space Invaders permitió ejercitar la capacidad de análisis, el pensamiento lógico y la resolución de problemas en tiempo real, habilidades fundamentales para cualquier desarrollador.

Por otra parte, este proyecto representó una valiosa oportunidad para potenciar el trabajo colaborativo, la comunicación efectiva y la organización con el otro, aspectos clave en el desarrollo de software en ambientes profesionales. La colaboración constante permitió distribuir tareas, compartir ideas y solucionar obstáculos de forma conjunta, enriqueciendo tanto el proceso como el resultado final.

En las siguientes secciones se detalla el proceso de desarrollo, las decisiones de diseño tomadas, los elementos personalizados integrados al juego y el análisis de los aprendizajes obtenidos mediante esta experiencia.

# Análisis del problema

El proyecto se enfoca en la elaboración de una versión a medida del videojuego Space Invaders, utilizando los principios esenciales del software development. La meta principal es convertir las necesidades del juego en una solución práctica, organizada y consistente, utilizando el lenguaje de programación Python como herramienta principal de implementación, tal como se explicó en el curso.

La problemática inicial se enfoca en cómo convertir las características fundamentales del juego —tales como el desplazamiento del jugador, la aparición y desplazamiento de los adversarios, el lanzamiento de proyectiles y la identificación de colisiones— en una estructura lógica y programable. Esto implica reconocer los elementos fundamentales del sistema (jugador, adversarios, disparos, interfaz gráfica, etc.) y su adecuada interacción a través del código.

Para tratar este problema, se utilizó la metodología modular, segmentando el programa en secciones manejables que desempeñan funciones concretas. Se empleó programación basada en objetos para ilustrar las entidades del juego, implementando ciclos, condiciones, listas y gestión de eventos para enriquecer la dinámica del juego.

Adicionalmente, se recurrió a herramientas gráficas elementales en Python, como bibliotecas para la generación de interfaces visuales, con el objetivo de replicar la experiencia del juego inicial. Este procedimiento no solo requería el manejo del lenguaje, sino también la habilidad para organizar el código de forma nítida y eficaz.

Desde una perspectiva metodológica, se llevó a cabo una dinámica de cooperación entre los alumnos, recreando un ambiente laboral auténtico en proyectos de software. Esta colaboración simplificó la organización de tareas, la revisión de código y la solución conjunta de fallos, potenciando competencias fundamentales como la comunicación técnica y la toma de decisiones en grupo.

# Solución del problema

Basándonos en el análisis efectuado, se propuso una solución organizada que facilitara el manejo eficiente del desarrollo de una versión personalizada del videojuego Space Invaders empleando Python. La estrategia implementada se fundamenta en dividir el problema en elementos funcionales, implementando buenas prácticas de diseño de software y tácticas características del desarrollo orientado a objetos.

### Etapas de la solución

1. **Definición de requerimientos del juego:** Se establecieron las funcionalidades esenciales a replicar: movimiento horizontal del jugador, generación continua de enemigos, disparos controlados por el usuario, detección de colisiones, contador de puntaje y condiciones de victoria o derrota.
2. **Diseño de la arquitectura del sistema:** Se optó por un diseño modular, en el cual cada componente del juego (jugador, enemigos, proyectiles, interfaz gráfica) se representa como una clase o módulo separado. Esta decisión facilita la mantenibilidad y escalabilidad del código.
3. **Selección de herramientas gráficas:** Se utilizó la biblioteca pygame, una herramienta especializada en la creación de videojuegos en Python. Esta permite manejar fácilmente eventos del teclado, dibujar gráficos en pantalla, manejar sonidos y actualizar elementos en tiempo real.
4. **Implementación iterativa del código:** Se desarrolló el código por etapas, comenzando por las funcionalidades básicas (movimiento del jugador y enemigos) y posteriormente integrando nuevas características (disparos, colisiones, puntaje, interfaz personalizada). Cada etapa fue probada de forma independiente para garantizar su correcto funcionamiento.
5. **Pruebas y ajustes finales:** Se realizaron pruebas funcionales y de usuario para asegurar que el juego respondiera correctamente a las interacciones. También se ajustaron aspectos visuales y de jugabilidad para mejorar la experiencia.

### Pseudocódigo de la lógica central

función bucle\_principal():

mientras el juego no haya terminado:

capturar\_eventos\_teclado()

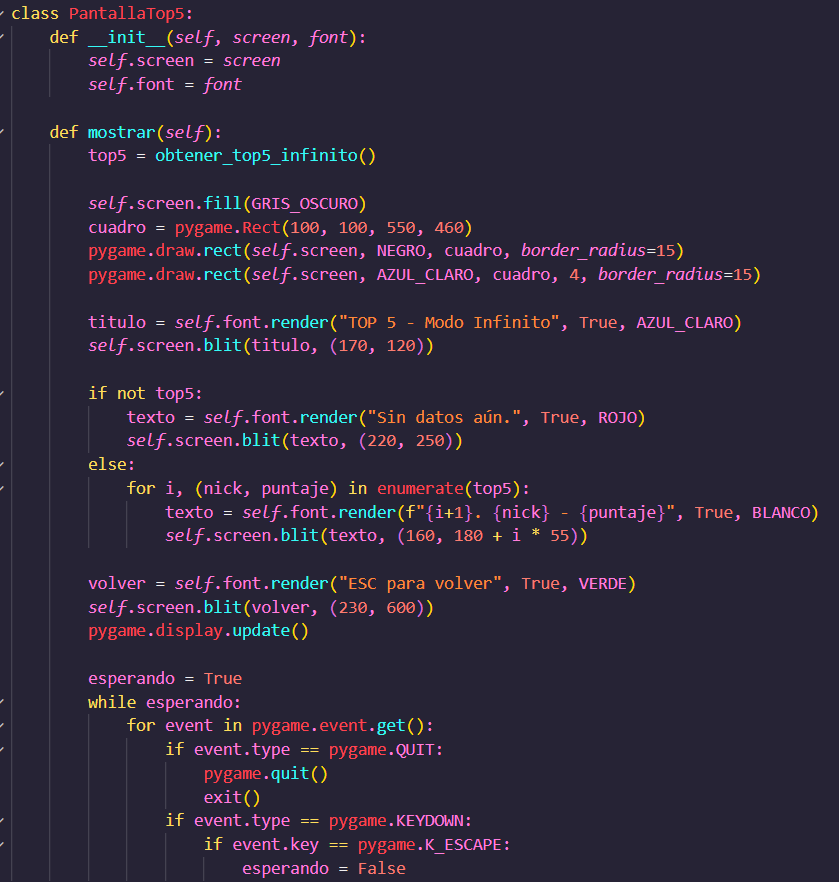
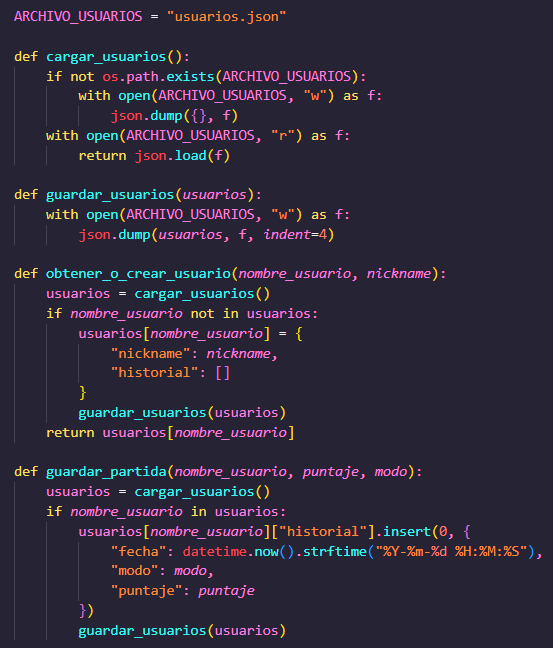
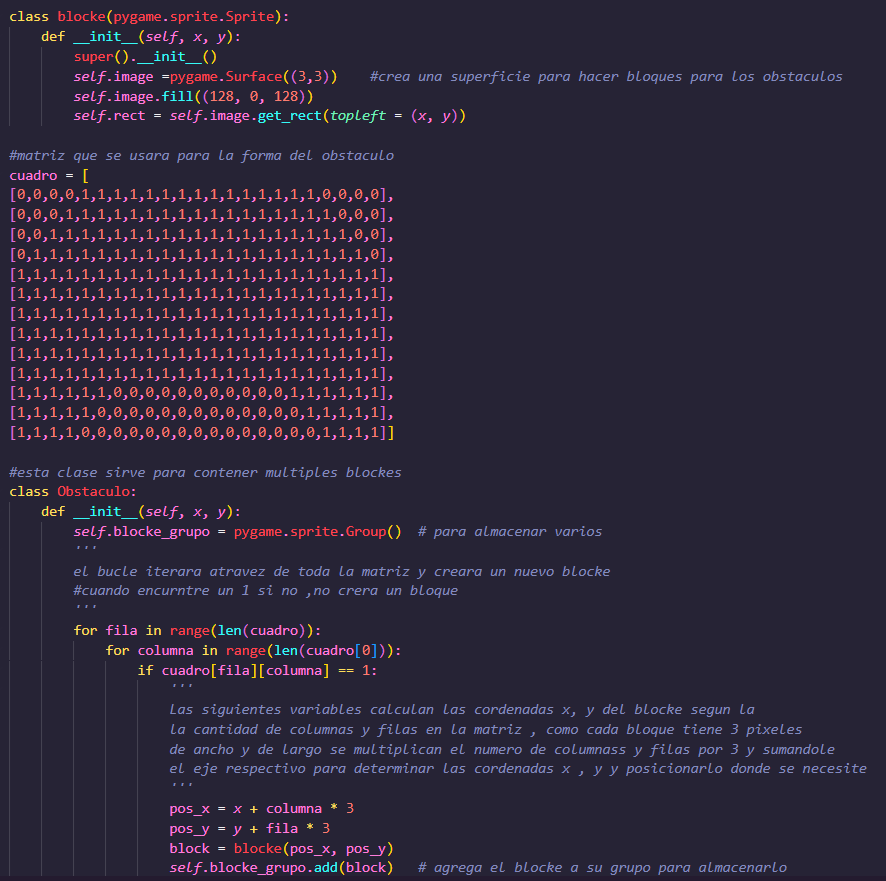
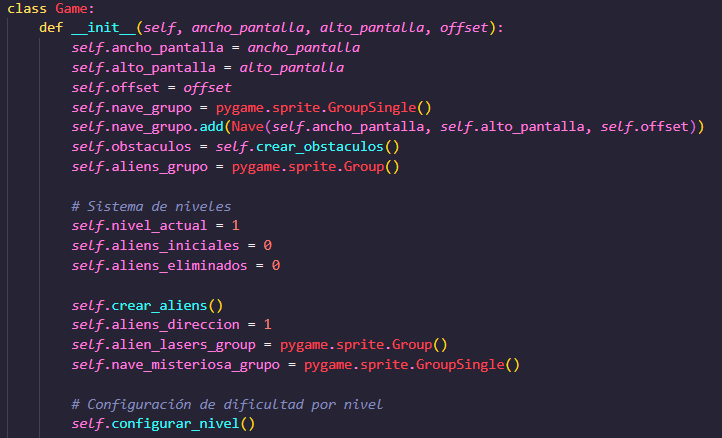
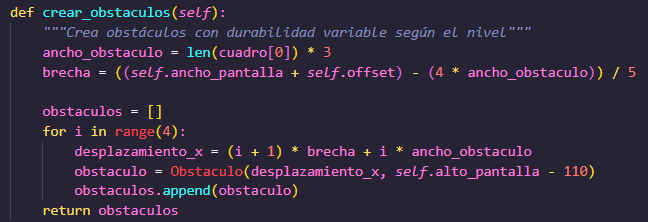
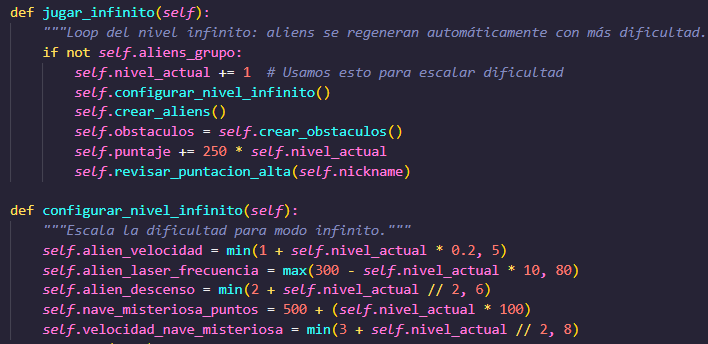
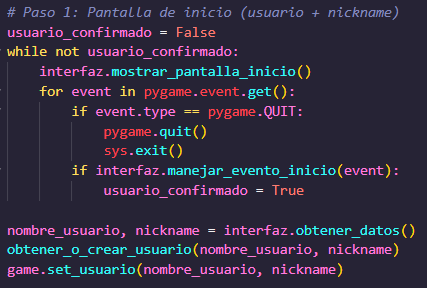
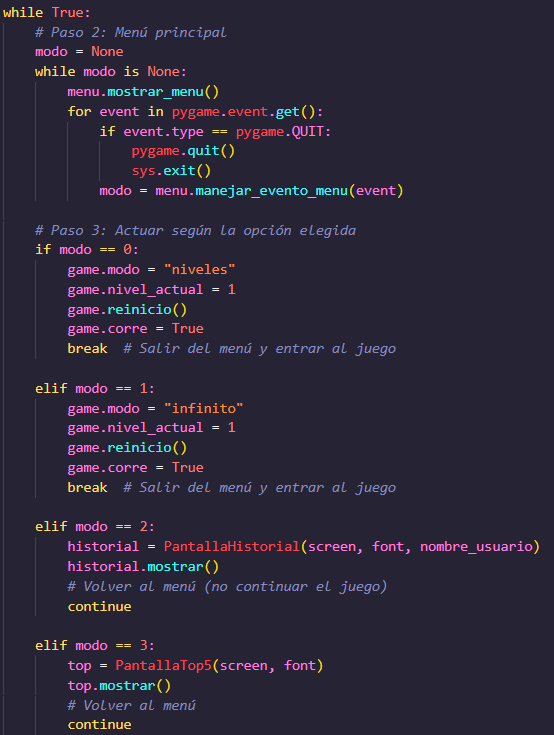
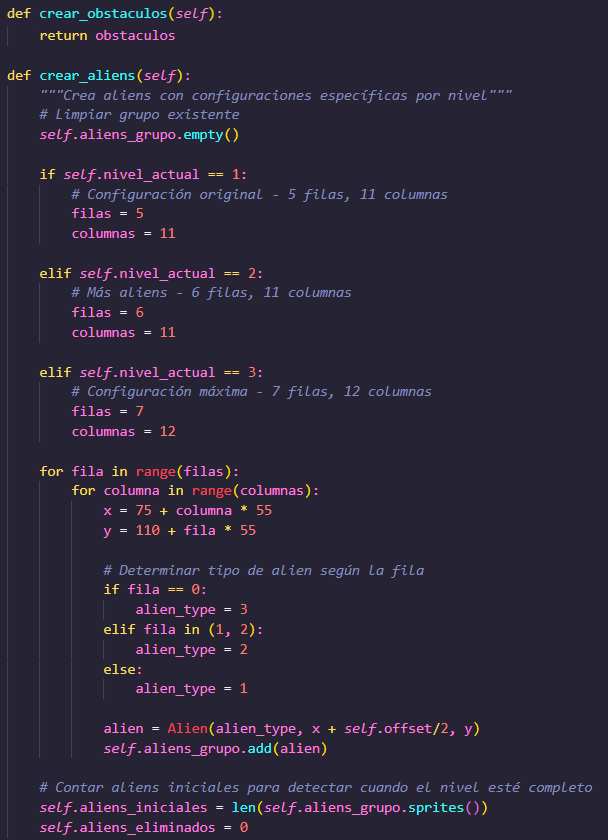
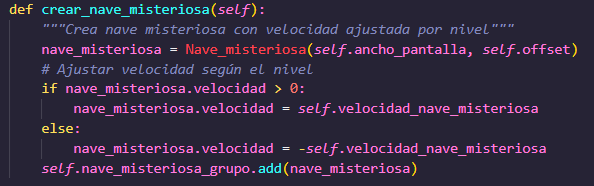
actualizar\_posiciones()

verificar\_colisiones()

dibujar\_elementos\_en\_pantalla()

actualizar\_pantalla()

### Código desarrollado

1. **Top 5 mejores puntajes:** 
2. **Historial:** ****
3. **Guardar usuario y su registro de las partidas:** ****
4. **Creación de superficies en bloques para los obstáculos:** 
5. **Niveles y dificultad por nivel: ** ****
6. **Creación de bstáculos:** Dentro de la clase Game en el archivo game.py 
7. **Nivel infinito:** Dentro de la clase Game en el archivo game.py
8. **Pantalla de inicio:** En el archivo main.py 
9. **Menú principal:** En el archivo main.py 
10. **Creación de aliens:** Dentro de la clase Game en el archivo game.py 
11. **Creación de nave misteriosa:** Dentro de la clase Game en el archivo game.py 

# Análisis de resultados

|  |  |  |
| --- | --- | --- |
| **Tarea / Requerimiento** | **Estado** | **Observaciones** |
| Movimiento del jugador | Completo | Responde correctamente a las teclas de dirección. |
| Generación y movimiento de enemigos | Completo | Se generan en formación y descienden progresivamente. |
| Disparo de proyectiles | Completo | Los proyectiles del jugador se activan y recorren la pantalla hacia arriba. |
| Detección de colisiones | Completo | Las colisiones entre proyectiles y enemigos están correctamente implementadas. |
| Sistema de puntaje | Completo | Se suma el puntaje al eliminar enemigos, se muestra en pantalla. |
| Condiciones de fin del juego | Completo | El juego termina al eliminar todos los enemigos o si estos llegan al jugador. |
| Nivel infinito | Completo | Los puntos se van acumulando hasta que el jugador pierda. |
| Mejores puntujes | Completo | Se guardan las puntuaciones más altas de los diferentes usuarios. |
| Interfaz visual básica (pygame) | Completo | Se muestra correctamente el entorno gráfico del juego. |
| Personalización del diseño | Completo | Se incluyeron elementos visuales diferenciados y sonidos opcionales. |
| División del código en módulos | Completo | El código está organizado en archivos y clases separadas. |

# Conclusiones

Mediante la puesta en marcha de una versión personalizada del videojuego Space Invaders empleando Python, se demostró la eficacia del método modular y orientado a objetos en la creación de sistemas interactivos. La segmentación del proyecto en clases claramente establecidas proporcionó una mayor claridad en la administración de responsabilidades, simplificando tanto el mantenimiento como la ampliación del código.

El empleo de Python, el lenguaje base asignado por el curso, se evidenció como una herramienta apropiada para este tipo de proyectos gracias a su sintaxis clara, la accesibilidad a bibliotecas especializadas como pygame, y su habilidad para fusionar elementos gráficos y lógicos sin una elevada curva de complejidad. Esto corroboró la utilidad de Python en la creación de videojuegos sencillos y didácticos, particularmente en entornos de capacitación técnica.

Además, los hallazgos logrados indican que la estructura lógica de eventos, colisiones y ciclos de juego puede ser gestionada de manera eficaz a través de un ciclo principal que gestiona la entrada del usuario, la actualización de estados y el renderizado gráfico en tiempo real. Esta metodología es apropiada para proyectos que demandan una interacción continua y un procesamiento gráfico.

Desde un punto de vista funcional, se consiguió implementar con éxito todos los requisitos establecidos: gestión del jugador, aparición y desplazamiento de adversarios, disparos, choques, sistema de calificación y condiciones de finalización. La solución sugerida cumplió con creces con las expectativas del proyecto, corroborando tanto el diseño propuesto como la implementación técnica de este.

# Recomendaciones

Considerando los resultados obtenidos en el desarrollo del proyecto, se sugieren los siguientes puntos para ampliar su alcance y enriquecer futuras implementaciones:

1. **Incluir el almacenamiento de datos de larga duración:** La implementación de funciones como almacenar las puntuaciones máximas o estadísticas del jugador a través de archivos o bases de datos puede contribuir a la gestión de persistencia de datos, un campo crucial en aplicaciones reales.
2. **Investigar la incorporación de la inteligencia artificial fundamental:** La implementación de patrones de movimiento más sofisticados para los adversarios o respuestas automatizadas podría dar paso a ideas básicas de Inteligencia Artificial, muy útiles tanto en videojuegos como en sistemas más sofisticados.
3. **Mejorar la eficiencia del motor de juego:**  Analizar el efecto del empleo de recursos gráficos y lógica de colisiones en el rendimiento posibilitaría la comparación de distintos métodos de implementación, lo que resulta crucial en el desarrollo de software con limitaciones de tiempo o hardware.
4. **Ampliar la interfaz visual con menús y ajustes:**  Incorporar pantallas de inicio, alternativas de configuración y menús interactivos potenciaría la experiencia del usuario y facilitaría el trabajo con estructuras de interfaz y navegación más sofisticadas en videojuegos.

Estas sugerencias están dirigidas a alumnos y programadores que quieran explorar en profundidad el campo de la programación de videojuegos, fusionando elementos técnicos, lógicos y visuales en un único ambiente de desarrollo.

# Referencias bibliográficas

Garrido, J. M. (31 de Enero de 2022). *Space Invaders con Python y Pygame*. JorgeMartinez.Space. <https://jorgemartinez.space/posts/computaticas/space-invaders-con-python-y-pygame/>

Pitchers, J. (2022). Obtenido de <https://spaceinvaders.viperfish.com.au/>

RiverView. (15 de Abril de 2020). *Crear Space Invaders con Python* [Video]. Bing Videos. <https://www.bing.com/videos/riverview/relatedvideo?q=crear+space+invaders+python&mid=748589D36FD0927BF61A748589D36FD0927BF61A&FORM=VIRE>

Educálix. (27 de Agosto de 2023). *Creando Space Invaders con Python*. Blog Educálix. <https://blog.educalix.com/creando-space-invaders-con-python/>

Clear Code. (2021, enero 10). *Space Invaders with Python | Full Game Tutorial* [Video]. YouTube. <https://youtu.be/f4coFYbYQzw>

Python-3.com. (2024-2025). *Crear Space Invaders con Python* [Artículo]. <https://es.python-3.com/?p=2538>

# Participación del grupo

|  |  |  |
| --- | --- | --- |
| **Rubro** | **April Meléndez Vásquez** | **Fabián Aragón Arce** |
| Registro de usuarios | 50% | 50% |
| Diseño de niveles | 50% | 50% |
| Mejores puntajes | 50% | 50% |
| Nivel infinito | 50% | 50% |