

Historias de usuario Infinity Jump

1)

| | |
|---|---|
| <p>Como: Jugador</p> <p>Quiero: Que haya muchos obstáculos</p> <p>Para poder: Retarme a superar diferentes obstáculos</p> | |
| <p>Validación:</p> <ul style="list-style-type: none">• Crear diferentes obstáculos• Comprobar la aleatoriedad de los obstáculos para que se usen todos• Comprobar que se puedan utilizar todos los obstáculos | <p>Valor (1-100): 100</p> <p>Prioridad: 1</p> <p>Estimación: 3 días</p> |

2)

| | |
|---|--|
| <p>Como: Jugador</p> <p>Quiero: Que el personaje tenga mejoras permanentes</p> <p>Para poder: Sentir que progreso después de cada jefe</p> | |
| <p>Validación:</p> <ul style="list-style-type: none">• Implementar mejoras permanentes (vida, salto, daño)• Verificar que se activen al vencer un jefe• Registrar visualmente las mejoras obtenidas | <p>Valor (1-100): 95</p> <p>Prioridad: 2</p> <p>Estimación: 3 días</p> |

3)

| | |
|---|--|
| <p>Como: Encargado de Videojuego</p> <p>Quiero: Que se puedan agregar nuevos enemigos</p> <p>Para poder: Expandir el contenido del juego fácilmente</p> | |
| <p>Validación:</p> <ul style="list-style-type: none">• Diseñar clases reutilizables para | <p>Valor (1-100): 85</p> <p>Prioridad: 3</p> |

| | |
|--|---------------------------|
| <p>enemigos</p> <ul style="list-style-type: none"> • Comprobar que los nuevos enemigos funcionen con el sistema actual • Probar nuevos comportamientos sin romper el juego | <p>Estimación: 4 días</p> |
|--|---------------------------|

4)

| | |
|--|--|
| <p>Como: Encargado de Base de Datos Quiero: Guardar las estadísticas de los jugadores Para poder: Llevar un registro de su desempeño</p> | |
| <p>Validación:</p> <ul style="list-style-type: none"> • Crear estructura en BD para guardar datos de cada partida • Asegurar integridad y consistencia de los datos • Verificar escritura al finalizar cada partida | <p>Valor (1-100): 80 Prioridad: 4 Estimación: 2 días</p> |

5)

| | |
|--|---|
| <p>Como: Usuario Quiero: Que el juego tenga pausa Para poder: Tomar descansos sin perder progreso</p> | |
| <p>Validación:</p> <ul style="list-style-type: none"> • Agregar funcionalidad para pausar y reanudar el juego • Asegurar que el estado del personaje se congele • Verificar que los inputs no funcionen al estar en pausa | <p>Valor (1-100): 85 Prioridad: 5 Estimación: 1 día</p> |

6)

| | |
|--|--|
| <p>Como: Encargado de Web Quiero: Que los assets carguen correctamente Para poder: Evitar errores de renderizado en el navegador</p> | |
|--|--|

| | |
|---|---|
| Validación: <ul style="list-style-type: none"> • Comprobar paths de recursos • Probar en distintos navegadores • Verificar ausencia de errores 404 | Valor (1-100): 90 Prioridad: 6 Estimación: 2 días |
|---|---|

7)

| | |
|--|---|
| Como: Jugador Quiero: Tener diferentes tipos de enemigos Para poder: Enfrentarme a retos variados | |
| Validación: <ul style="list-style-type: none"> • Diseñar enemigos con distintas mecánicas • Verificar comportamiento individual de cada enemigo • Probar en distintos niveles | Valor (1-100): 88 Prioridad: 7 Estimación: 3 días |

8)

| | |
|---|--|
| Como: Encargado de Videojuego Quiero: Agregar niveles con distintos fondos Para poder: Diferenciar visualmente el progreso del jugador | |
| Validación: <ul style="list-style-type: none"> • Asignar fondos específicos por nivel • Verificar carga y transición entre niveles • Ajustar el tamaño del canvas según el fondo | Valor (1-100): 80 Prioridad: 5 Estimación: 1 día |

9)

| | |
|--|-----------------------------------|
| Como: Jugador Quiero: Ver mis estadísticas al final del juego Para poder: Evaluar mi rendimiento | |
| Validación: <ul style="list-style-type: none"> • Mostrar enemigos derrotados, | Valor (1-100): 92 Prioridad: 9 |

| | |
|---|--------------------|
| mejoras y nivel alcanzado <ul style="list-style-type: none"> • Asegurar formato visual claro y atractivo • Probar que se actualicen correctamente los datos | Estimación: 2 días |
|---|--------------------|

10)

| | |
|--|---|
| Como: Encargado de Base de Datos Quiero: Eliminar datos de prueba Para poder: Mantener limpia la base de datos | |
| Validación: <ul style="list-style-type: none"> • Agregar script para depurar datos por rango de fechas • Verificar que no afecte partidas reales • Ejecutar pruebas en entorno de staging | Valor (1-100): 70 Prioridad: 10 Estimación: 1 día |

11)

| | |
|--|--|
| Como: Encargado de Web Quiero: Que los scripts estén bien organizados Para poder: Dar mantenimiento fácilmente al proyecto | |
| Validación: <ul style="list-style-type: none"> • Reestructurar carpetas por módulos (player, enemigos, interfaz) • Verificar que no se rompa ninguna funcionalidad • Documentar la organización | Valor (1-100): 78 Prioridad: 11 Estimación: 3 días |

12)

| | |
|--|--|
| Como: Jugador Quiero: Que haya jefes finales con ataques únicos Para poder: Tener batallas más desafiantes | |
| Validación: <ul style="list-style-type: none"> • Implementar ataques especiales para cada jefe | Valor (1-100): 95 Prioridad: 12 Estimación: 4 días |

| | |
|---|--|
| <ul style="list-style-type: none"> • Probar dificultad progresiva • Verificar comportamiento del jefe según su vida | |
|---|--|

13)

| | |
|--|--|
| Como: Encargado de Videojuego Quiero: Usar delta time correctamente Para poder: Asegurar que el juego funcione igual en todas las computadoras | |
| Validación: <ul style="list-style-type: none"> • Aplicar delta time en movimiento y físicas • Comparar el comportamiento en distintos equipos • Ajustar la escala de tiempo si es necesario | Valor (1-100): 90 Prioridad: 13 Estimación: 2 días |

14)

| | |
|--|--|
| Como: Encargado de Web Quiero: Tener el código bien comentado y organizado Para poder: Encontrar cosas fácilmente y entender mejor el código | |
| Validación: <ul style="list-style-type: none"> • Separar en carpetas específicas • Comentar en las funciones para saber que hacen • Ser claros con los nombres de las variables | Valor (1-100): 80 Prioridad: 14 Estimación: 3 días |

15)

| | |
|--|--|
| Como: Jugador Quiero: Que se guarde mi progreso automáticamente Para poder: Continuar mi partida después | |
| Validación: <ul style="list-style-type: none"> • Guardar datos en localStorage • Probar carga automática al regresar al juego • Verificar integridad del progreso | Valor (1-100): 93 Prioridad: 15 Estimación: 2 días |

16)

| | |
|--|---|
| <p>Como: Encargado de Base de Datos</p> <p>Quiero: Integrar la base de datos con gráficos</p> <p>Para poder: Mostrar visualmente el desempeño del jugador</p> | |
| <p>Validación:</p> <ul style="list-style-type: none">• Agregar exportación de datos a JSON o CSV• Conectar con librerías de visualización• Probar estadísticas cargadas en dashboard | <p>Valor (1-100): 80</p> <p>Prioridad: 16</p> <p>Estimación: 3 días</p> |

17)

| | |
|---|---|
| <p>Como: Usuario</p> <p>Quiero: Una recompensa al derrotar a los jefes</p> <p>Para poder: Sentir que me gane algo por mi esfuerzo</p> | |
| <p>Validación:</p> <ul style="list-style-type: none">• Ofrecer una mejora al ganarle a un jefe• Asegurar que se implemente bien• Asegurar que se quede guardada | <p>Valor (1-100): 90</p> <p>Prioridad: 17</p> <p>Estimación: 3 días</p> |

18)

| | |
|---|---|
| <p>Como: Encargado de Web</p> <p>Quiero: Monitorear errores del juego</p> <p>Para poder: Corregirlos rápidamente</p> | |
| <p>Validación:</p> <ul style="list-style-type: none">• Implementar sistema de logs• Enviar errores a un endpoint• Probar capturas de errores en consola | <p>Valor (1-100): 75</p> <p>Prioridad: 18</p> <p>Estimación: 2 días</p> |

19)

Como: Jugador
Quiero: Que los controles sean responsivos
Para poder: Tener mejor experiencia de juego

Validación:

- Optimizar detección de teclas
- Probar en distintas velocidades de hardware
- Ajustar input lag si existe

Valor (1-100): 88
Prioridad: 19
Estimación: 2 días

20)

Como: Encargado de Videojuego
Quiero: Agregar nuevos power-ups
Para poder: Diversificar estrategias del jugador

Validación:

- Diseñar nuevos efectos de power-up
- Probar compatibilidad con los ya existentes
- Ajustar balance del juego

Valor (1-100): 85
Prioridad: 20
Estimación: 3 días

21)

Como: Encargado de Base de Datos
Quiero: Validar los usuarios de partidas
Para poder: Evitar duplicados en los registros

Validación:

- Generar usuarios únicos por partida
- Probar colisiones y duplicaciones
- Validar desde backend y frontend

Valor (1-100): 70
Prioridad: 21
Estimación: 1 día

22)

Como: Jugador
Quiero: Ver las mejoras que tengo en un inventario

| | |
|---|--|
| Para poder: Tener un indicador visual de las mejoras que tengo | |
| Validación: <ul style="list-style-type: none"> • Agregar en el canvas el inventario • Hacer que se actualice cuando agarre otra mejora • Poner un límite | Valor (1-100): 86 Prioridad: 22 Estimación: 1 días |

23)

| | |
|---|--|
| Como: Jugador Quiero: Consultar mis estadísticas generales desde el menú principal Para poder: Evaluar mi progreso sin jugar una nueva partida | |
| Validación: <ul style="list-style-type: none"> • Mostrar pantalla de estadísticas generales • Incluir totales acumulados (enemigos derrotados, mejoras obtenidas, niveles completados) • Acceso desde botón en la pantalla de inicio | Valor (1-100): 84 Prioridad: 23 Estimación: 2 días |

24)

| | |
|---|---|
| Historia de usuario 24 Como: Jugador Quiero: Que haya una pantalla final con créditos Para poder: Reconocer a los creadores | |
| Validación: <ul style="list-style-type: none"> • Crear escena de créditos • Activar después de vencer jefe final • Incluir nombres, música y efectos | Valor (1-100): 65 Prioridad: 24 Estimación: 1 día |

25)

| | |
|---|--|
| Como: Encargado de Videojuego Quiero: Que las plataformas móviles regresen a su posición | |
|---|--|

| | |
|--|--|
| Para poder: Evitar que el jugador quede atrapado | |
| Validación: <ul style="list-style-type: none"> ● Implementar lógica de rebote o reinicio ● Probar que se mantenga el patrón ● Verificar sincronización con delta time | Valor (1-100): 85 Prioridad: 25 Estimación: 2 días |

Requerimientos Funcionales

Sistema de obstáculos aleatorios. El juego debe generar obstáculos variados y aleatorios durante la partida.

Implementación de mejoras permanentes. El personaje debe poder recibir y conservar mejoras al vencer jefes.

Soporte para múltiples tipos de enemigos. Se deben implementar enemigos con distintas mecánicas de ataque.

Agregar jefes con ataques únicos. Jefes con comportamientos distintos y progresión de dificultad.

Sistema de pausa. El jugador debe poder pausar y reanudar el juego sin perder el estado actual.

Guardar progreso automáticamente: El progreso del jugador debe almacenarse en localStorage al avanzar.

Visualización de estadísticas al final del juego: Al terminar, se deben mostrar estadísticas como plataformas alcanzadas, niveles alcanzados y mejoras obtenidas.

Visualización de estadísticas generales desde el menú: El jugador debe poder ver estadísticas acumuladas desde el menú principal.

Inventario visual de mejoras: Debe mostrarse en pantalla un inventario con las mejoras adquiridas.

Plataformas móviles deben regresar a su posición original. Las plataformas en movimiento deben reiniciar su ciclo para evitar errores.

Agregación de nuevos power-ups. El sistema debe aceptar nuevos power-ups sin romper el balance del juego.

Ofrecer mejoras como recompensa por jefes. Al vencer un jefe, debe darse una recompensa funcional al jugador.

Delta time para físicas y movimiento. El juego debe calcular movimientos basados en delta time para evitar inconsistencias en distintos dispositivos.

Guardar estadísticas de partida en base de datos. Se debe registrar cada partida en la BD con su desempeño.

Validación de usuarios únicos por partida. Cada jugador/partida debe tener un identificador único para evitar duplicados.

Requerimientos no funcionales:

Carga correcta de assets en distintos navegadores. Las imágenes, videos y sonidos deben cargarse sin errores 404.

Organización modular del código (por carpetas y scripts). El código debe estar separado por componentes para facilitar mantenimiento.

Código comentado y claro. Las funciones y variables deben estar bien nombradas y documentadas.

Eliminación de datos de prueba. Scripts para limpieza de registros falsos en BD.

Visualización de estadísticas en gráficas. Exportación a JSON/CSV y conexión con dashboards visuales.

Sistema de monitoreo de errores. Registro de errores en consola y envío a un endpoint.

Controles responsivos y sin input lag. Optimización para que el juego responda bien sin importar el hardware.

Pantalla final con créditos. Mostrar reconocimiento del equipo al terminar el juego.

Cambio de fondos por nivel. Fondos visuales únicos por nivel, pero no afectan a la jugabilidad

