



Componente formativo

## **Pruebas de funcionamiento y ajuste de mecánicas**

---

### **Breve descripción:**

Describe la importancia de la realización de pruebas de usabilidad para determinar la facilidad de uso de los videojuegos, utilizando los métodos, técnicas e instrumentos adecuados de acuerdo con el aspecto a probar.

### **Área ocupacional:**

Ciencias Naturales

---

**Mayo 2023**

## Tabla de contenido

Introducción.....	3
1. Metodología RITE.....	4
1.1. Herramienta de testeo en Unity.....	6
1.2. Herramienta de testeo incorporada a Unity .....	7
2. Modelos de iteración.....	11
2.1. Proceso Unificado Racional - PUR.....	16
2.2. Programación extrema - XP .....	20
Elementos XP .....	21
Características de EX .....	23
2.3. Scrum .....	25
Fases para el desarrollo de un videojuego en el modelo Scrum .....	26
3. Modelos de prueba de usabilidad .....	31
3.1. Modelo de inspección .....	32
3.2. Modelo de indagación.....	35
3.3. Modelo de test .....	37
3.4. Ejemplo de aplicación .....	38
Material complementario .....	49
Glosario.....	50
Síntesis .....	52
Referencias bibliográficas.....	53
Créditos.....	54

## Introducción

Bienvenido al componente formativo pruebas de funcionamiento y ajuste de mecánicas:

### Video 1. Pruebas de funcionamiento y ajuste de mecánicas



### Pruebas de funcionamiento y ajuste de mecánicas

#### **Síntesis del video:** Pruebas de funcionamiento y ajuste de mecánicas

Estimado aprendiz, en este componente formativo podrá apropiar metodologías de trabajo en equipo para desglosar su proyecto en miniproyectos con tareas y actividades específicas que deberán ser probadas para seguir avanzando hacia el producto final deseado.

Podrá aplicar metodologías y técnicas que le ayudarán a evaluar el cumplimiento de los requisitos de los usuarios, de acuerdo con los aspectos de calidad que exige el contexto actual, donde las aplicaciones de videojuegos están a la vanguardia por el desarrollo de la industria del “software” y las tendencias y tecnologías innovadoras del momento.

Determinar si un juego cumple o no con los estándares establecidos en materia de usabilidad, es tan importante como el diseño del juego mismo, ya que este debe satisfacer necesidades didácticas o de diversión de los usuarios.

Facilitando la ejecución de tareas a través de la comprensión de las mecánicas del juego para que pueda sacar el mayor provecho tanto en aspectos de aprendizaje como de esparcimiento.

Un proyecto de videojuego debe evaluarse de diferentes maneras para determinar si cumple con los objetivos que los mismos usuarios han planteado, respetando los principios que lo cataloguen como un producto de fácil uso que aporta a los sectores productivos del área de la educación y del entretenimiento, potencializando además competencias que los individuos pueden aplicar en la vida real para resolver problemas o situaciones de su diario vivir.

Tener en cuenta las condiciones de calidad al realizar las pruebas de usabilidad cuando se diseña e implementa un producto, es reducir costos de producción, mantenimiento y apoyo al evitar rediseños.

Por tal razón, la evaluación y aplicación de pruebas de usabilidad se considera como un elemento más que se debe integrar al diseño de los productos de la industria de los videojuegos.

## **1. Metodología RITE**

Las siglas RITE significan Rapid Iterative Testing Evaluation, es decir testeo rápido iterativo de evaluación. Por consiguiente, la metodología RITE consiste en un método para evaluar el desarrollo de un proyecto a través de ciclos rápidos de ejecución hasta alcanzar el resultado final, a estos ciclos se les llama iteraciones. Cada iteración representa un bloque de tareas que se realizan, se prueban y evalúan hasta lograr el perfeccionamiento del producto deseado; existe el método iterativo y el iterativo incremental los que se pueden observar en el video sobre este concepto.

## **Iterativo e incremental**

Existe el método iterativo y el iterativo incremental los que se pueden observar en el video sobre este concepto. [Ver video.](#)

De acuerdo con lo anterior, en el método iterativo para llevar a cabo un proyecto primero se define el concepto y alcance de este, después se desglosan las tareas que se realizarán en todo el proceso de desarrollo donde cada iteración es la construcción de una parte del concepto, ejecutándose y evaluándose repetidas veces si es necesario y al final se integran las partes construidas hasta lograr el concepto que se ha diseñado del producto final.

En el método iterativo e incremental se parte de la necesidad que se tiene y se estructura un proyecto definiendo tareas que se van ejecutando en cada iteración y evaluándose para aproximarse paulatinamente (incrementalmente) a lo que será el producto final como solución al problema identificado.

En el desarrollo de videojuegos el método que se utiliza es el iterativo incremental y en cada iteración se ejecuta una tarea que se iterará o repetirá probándose hasta que se pueda avanzar a la siguiente que será un progreso respecto a la iteración anterior; es decir, el producto va aproximándose a lo que se desea de él y se puede testear o probar cada iteración con el usuario para obtener la retroalimentación necesaria e ir perfeccionándose.

Lo anterior quiere decir que dentro del equipo desarrollador del videojuego se debe contar con un integrante que haga las veces de “tester” o testeador que:

1. Verifica constantemente las mecánicas que se van creando.
2. Hace los ajustes necesarios y los valida de nuevo.
3. Vuelve a realizar las mejoras documentando los fallos o errores con los cambios que se van llevando a cabo en el diseño o en la programación.
4. Aproxima cada producto parcial a lo que finalmente se desea obtener.

## 1.1. Herramienta de testeo en Unity

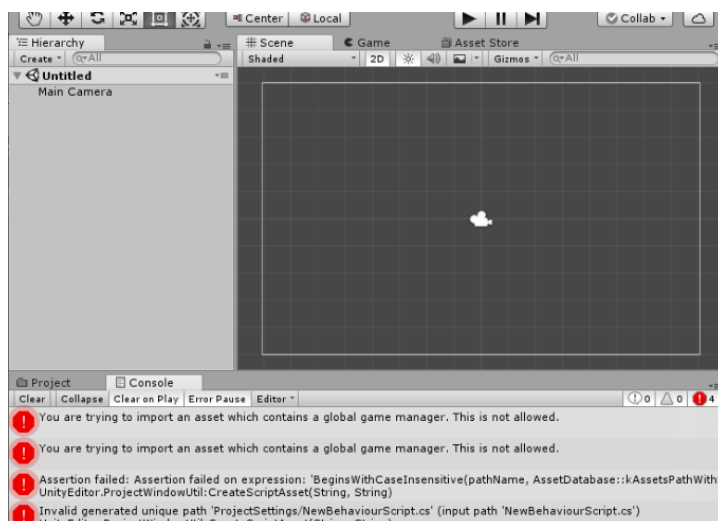
El “software” de videojuegos de Unity posee herramientas que pueden ayudar a testear las iteraciones para detectar los errores que se producen cuando se están ejecutando o desarrollando tareas en el videojuego, ya sea cuando se están escribiendo líneas de código o cuando se están llevando a cabo acciones con el menú para crear las escenas y mecánicas del videojuego.

Existen testeos rápidos que están incorporados en el “software” cuando se instala. Al conocer los errores que se producen en las iteraciones, el desarrollador puede tomar la decisión adecuada para dar solución; por ejemplo, corregir la sintaxis de programación necesaria o importar los archivos (asset) de formatos permitidos para incorporar en la escena.

### Mensajes de error en la consola

La consola es una pestaña que se encuentra dentro de la ventana de proyectos y es allí donde se van almacenando los mensajes de error al realizar iteraciones que ocasionen conflictos en la ejecución del videojuego. Estos mensajes permiten realizar acciones correctivas al diseñador para poder avanzar a otra fase de iteración.

**Figura 1.** Mensajes de error en la consola



Esta herramienta permite identificar los fallos está contenida en el “software”, es de rápido testeo para probar el funcionamiento del juego no solo para el desarrollador, sino para el usuario, puesto que informa exactamente qué está sucediendo cuando se ejecutan sus diferentes mecánicas.

## 1.2. Herramienta de testeo incorporada a Unity

Existen herramientas de testeo que se pueden importar al “software” y que en el momento de la instalación no son incorporadas. Las hay de descarga gratuita y también de pago que se incorporan al “software” en forma de assets o componentes que se integran en el menú del programa, para ser utilizadas en la identificación de incoherencias y fallos presentados en el desarrollo o rodaje de las escenas del videojuego. Cabe anotar que no se necesita ser un experto programador para utilizar estos componentes, puesto que son de fácil manejo y de ellos se encuentra documentación que explica la dinámica de uso.

### Unity Test Tool

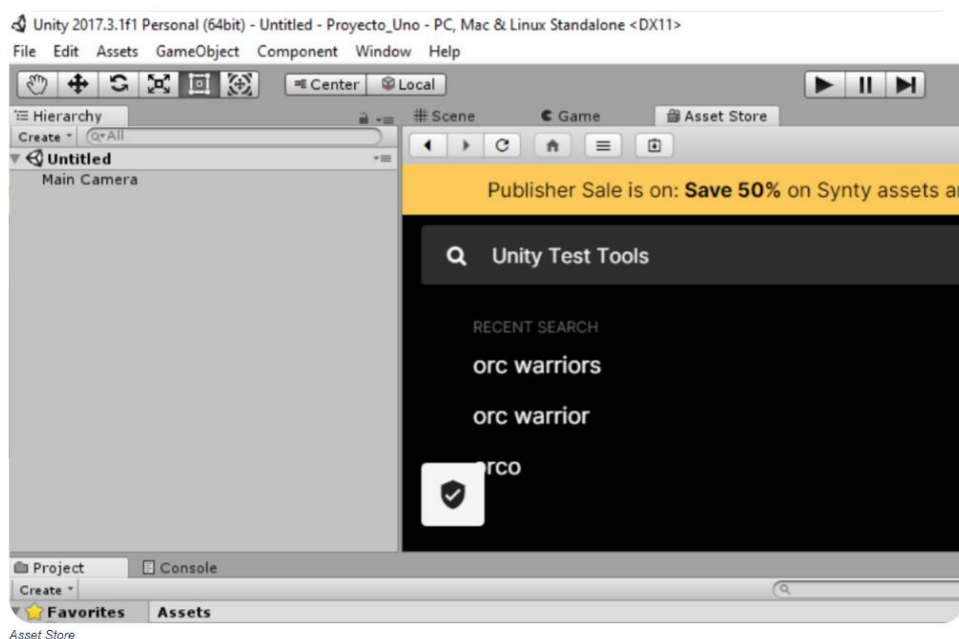
Este componente o asset permite realizar un testeo o evaluación rápida de las funcionalidades que se van incorporando al juego. Es una herramienta que se debe integrar desde la opción de Asset Store del menú de Unity ya que una vez se instala el programa no es importada de la nube automáticamente.

Pasos para incorporar esta herramienta dentro del programa de Unity:

**Paso 1:** hacer clic en el menú Window->Asset Store y el programa se direcciona a la URL <https://www.assetstore.unity3d.com/en/#!/content/13802>.

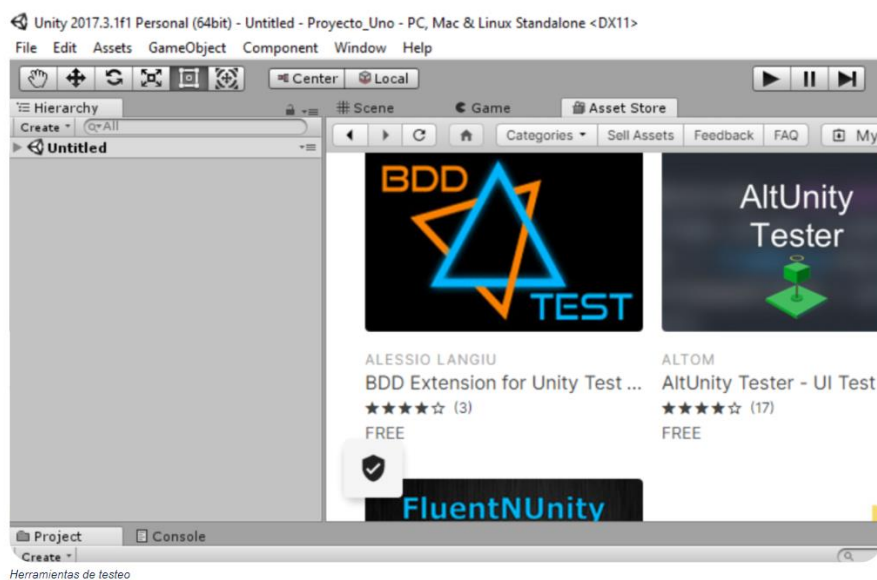
En la ventana de la escena aparece la página de UnityAssetStore y en el buscador se debe escribir “Unity Test Tools”.

**Figura 2. Paso 1**



**Paso 2:** luego aparecen los asset gratuitos y seleccionar uno de ellos, en este caso se seleccionará la herramienta BDD Test:

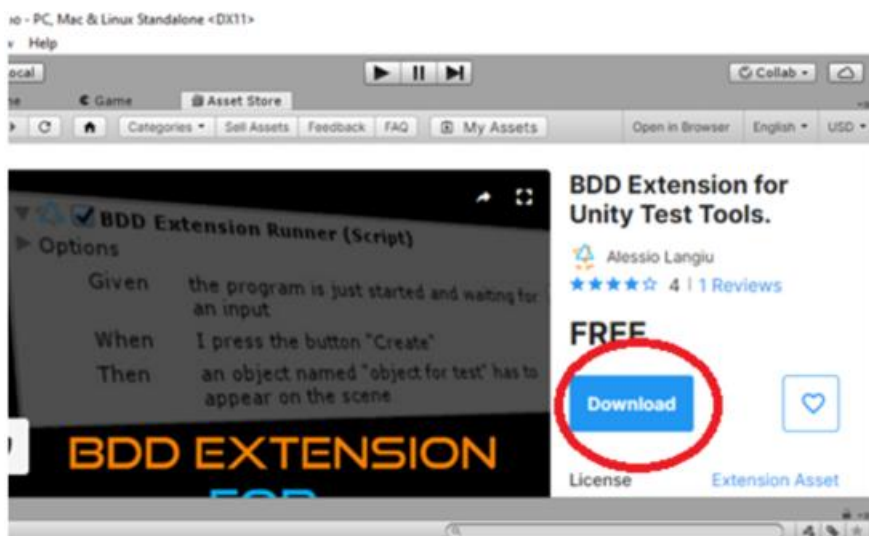
**Figura 3. Paso 2**



**Paso 3:** hacer clic en la opción “Download” para proceder a la descarga de la herramienta de testeo.

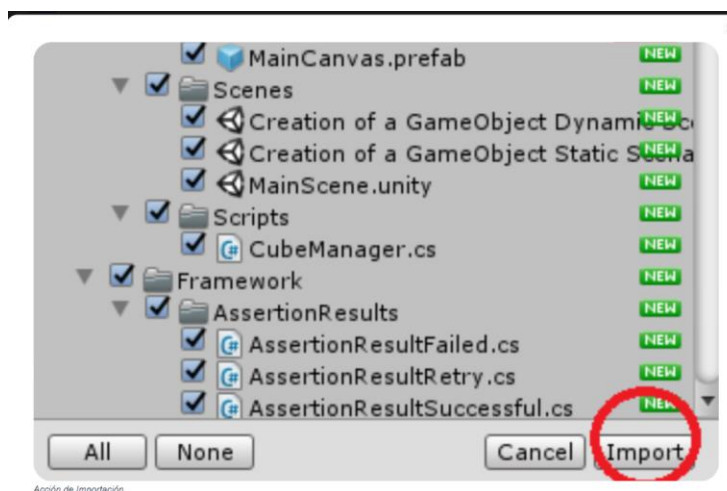


**Figura 4. Paso 3**



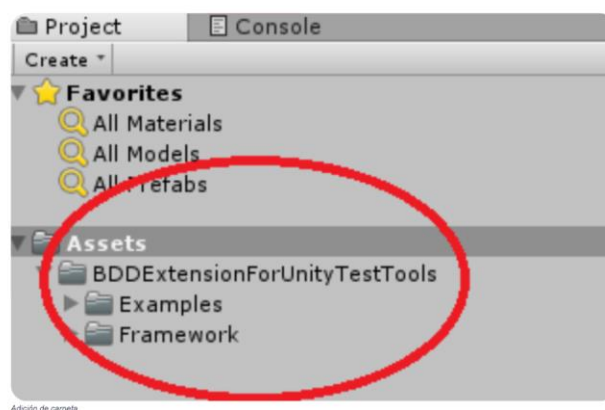
**Paso 4:** después que el programa hace la descarga, aparece la siguiente pantalla y se debe hacer clic en el botón “Import” que aparece en la esquina inferior derecha de la ventana.

**Figura 5. Paso 4**



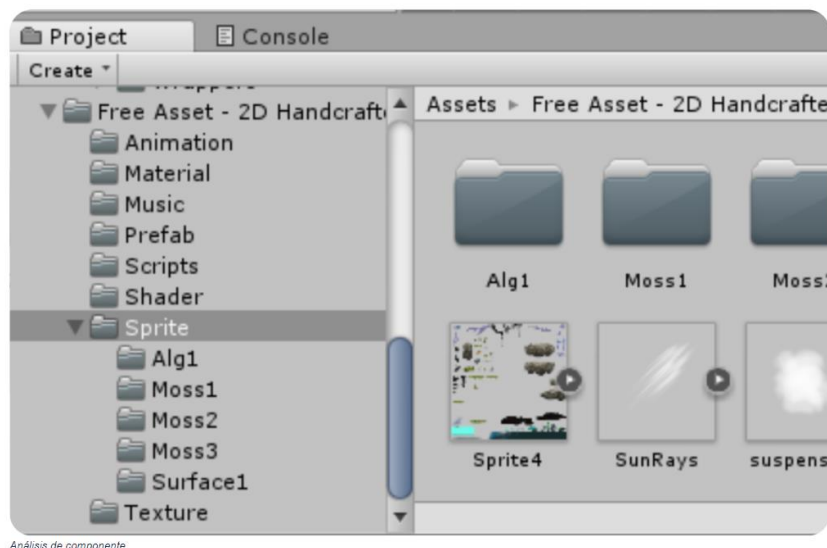
**Paso 5:** inmediatamente el componente se adiciona al programa Unity y, específicamente, al proyecto que se esté trabajando. En la ventana de proyecto se observa el componente de testeo adicionado para hacer las pruebas rápidas en las escenas y mecánicas del juego.

**Figura 6. Paso 5**



**Paso 6:** cuando se realiza cualquier acción en Unity por ejemplo, insertar un objeto a una escena, la herramienta *Unity Test Tools* se adiciona a este y analiza cualquier error en él o en la acción de copiarlo en la ventana de la escena. Por ejemplo, a continuación la acción que se va a realizar es arrastrar con el “mouse” el Sprite 3 que es un objeto de fondo de la ventana de *Project* a la ventana de la escena:

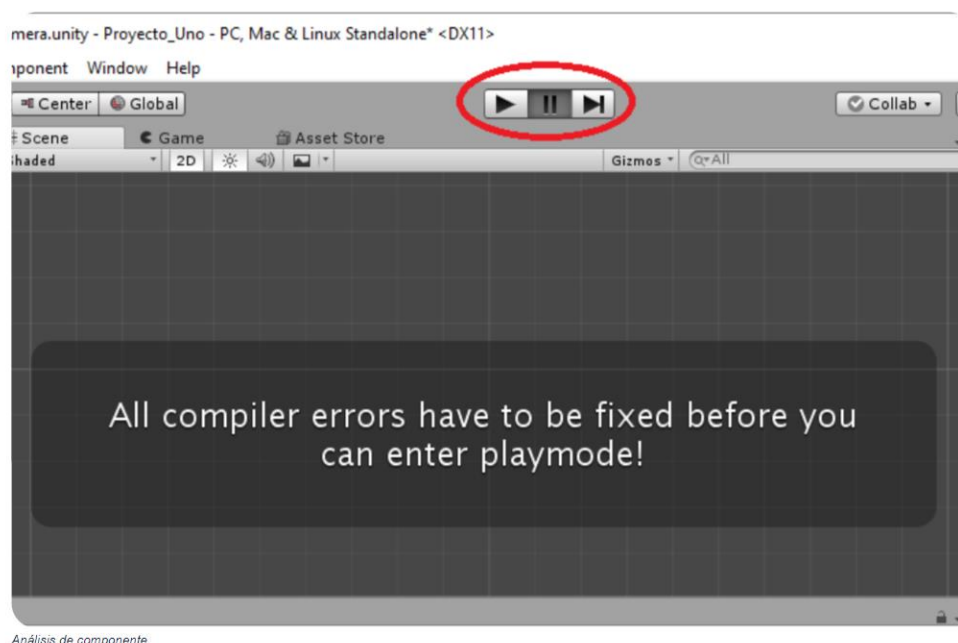
**Figura 7. Paso 6**



**Paso 7:** inmediatamente se activó la herramienta de testeo y mostró el siguiente mensaje en la ventana de la escena: All Compiler errors have to be fixed before you can enter play mode.

La acción no se pudo realizar ya que la escena se encontraba en modo de reproducción y el objeto no se puede compilar o procesar a nivel de software. Los botones encerrados en el círculo son los que indican el modo play o modo pausa, en el momento de copiar el elemento se encontraba en modo “play”.

**Figura 8. Paso 7**

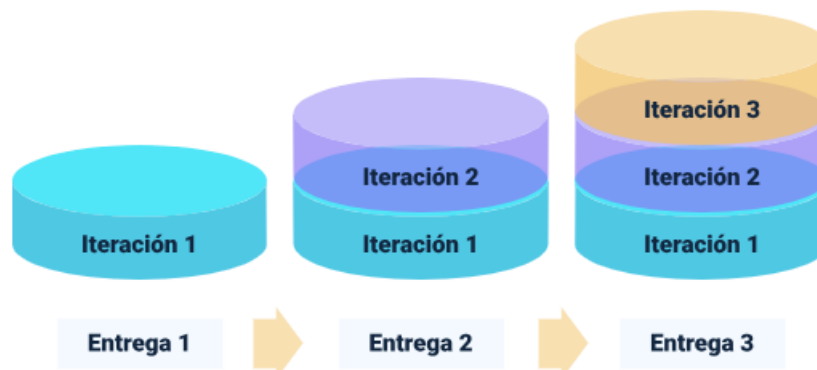


Para una mayor comprensión y profundización de las herramientas de testeo, se recomienda la siguiente lectura complementaria: [lectura de consulta](#).

## 2. Modelos de iteración

Como se mencionó antes, las iteraciones son paquetes de tareas que se reprocesan hasta lograr el perfeccionamiento de un producto, por consiguiente, un modelo iterativo según Cockburn, mencionado por Letelier (2013), corresponde a una metodología de re-trabajo para lograr mejorar un producto, donde cada iteración contiene un conjunto de actividades con el objetivo de entregar parte de las funcionalidades del producto al usuario.

**Figura 9.** Iteración incremental



Entonces, en una primera iteración se entrega una parte de la funcionalidad del producto, en una segunda iteración se entrega la parte que falta de la funcionalidad y en una tercera se puede entregar una nueva, es decir cada iteración se aproxima al producto final.

A este modelo también se le denomina incremental, puesto que se van obteniendo resultados parciales que al sumarse completan el producto final. Los usuarios van obteniendo beneficios de manera incremental a medida que se completan las tareas que deben incluir las pruebas, los ajustes y la documentación generada en cada proceso de entrega, esto quiere decir que no se deja para lo último la verificación de que el producto cumpla con los requisitos del proyecto.

Para citar un ejemplo sencillo de este modelo, en el desarrollo del juego se puede planificar que dos imágenes en escena, una quede en el fondo y la otra en el primer plano:

- a) El primer producto es colocar las imágenes en una escena; esta entrega consiste en dos iteraciones donde se realiza el mismo proceso:**

**Tabla 1.** Iteraciones para inserción

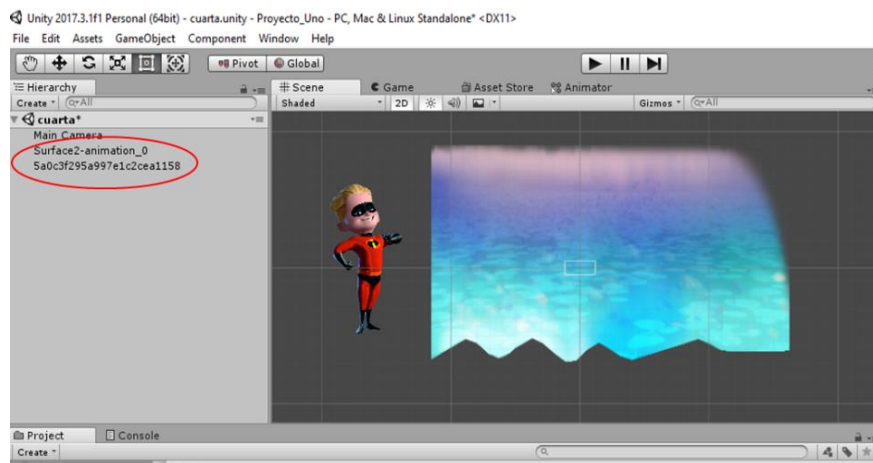
Iteración 1 para insertar fondo en ventana de escena	Iteración 2 para insertar superhéroe en ventana de escena
Ubicarse en la ventana de proyecto.	Ubicarse en la ventana de proyecto.
Abrir carpeta de <i>asset</i> fondos.	Abrir carpeta de <i>asset</i> fondos.
Seleccionar el fondo y arrástralo a la ventana de la escena.	Seleccionar el superhéroe y arrástralo a la ventana de la escena.

Obsérvese que en la ventana de jerarquía (parte izquierda) aparecen en orden de jerarquía las dos imágenes:

Primero el fondo - surface2-animation\_0

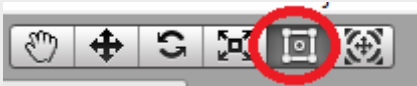

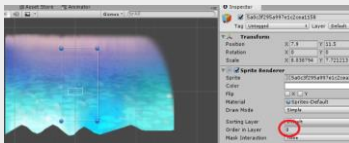
Segundo el superhéroe -5a0c3f295a997e1c2cea1158.png

**Figura 10.** Iteración 1 y 2



- b) El segundo producto es colocar la imagen del superhéroe sobre el fondo; esta entrega se compone de dos iteraciones:

**Figura 11.** Iteración 3 y 4

Iteración 3 para colocar el superhéroe en el mismo lugar del fondo	Iteración 4 para que el superhéroe quede sobre el fondo
<p>Hacer clic en el botón que está en el círculo.</p>  <p>Botones</p> <p>Hacer clic en el superhéroe.</p>  <p>Superhéroe</p>	<p>El superhéroe está seleccionado debajo del fondo; ya que se observa el recuadro con 4 puntos azules y en la ventana del inspector se aprecian los datos de escala, posición y en la parte de abajo la opción order layer con el valor de 0:</p>  <p>Order layer</p> <p>El valor de order layer se pone en 1 para que el superhéroe se coloque en el primer nivel.</p>

### Iteración 3 para colocar el superhéroe en el mismo lugar del fondo

Arrastrar el superhéroe a la posición del fondo para obtener el siguiente resultado:

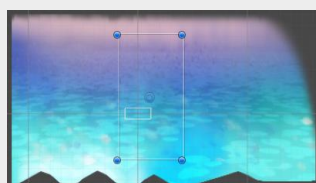


Imagen atrás

### Iteración 4 para que el superhéroe quede sobre el fondo



Imagen adelante

Para obtener el producto final se llevaron a cabo 4 iteraciones incrementales puesto que se tuvieron que realizar cuatro tareas para conseguir lo que se deseaba en el proyecto.

Los beneficios del modelo de iteración incremental radican en que:

1. Se pueden tomar decisiones en cada iteración, por ejemplo cambiar de personaje, de fondo, la escala de las imágenes, la rotación o la posición.
2. El equipo desarrollador tiene la posibilidad de validar si lo que ha realizado era lo que el usuario esperaba y hacer los ajustes necesarios.
3. No es necesario realizar una recopilación detallada de la información del producto ya que por el camino pueden surgir nuevos conceptos y se pueden ir afinando detalles.
4. Desde el comienzo se pueden obtener resultados y el usuario puede ir probando la usabilidad del producto aunque este no esté terminado.

5. Después de cada iteración se puede obtener el “feedback” del usuario y hacer los ajustes requeridos.
6. Debido a que el proyecto se evalúa después de cada iteración, se minimizan los riesgos y se optimizan los recursos, puesto que los fallos son detectados y superados a tiempo.
7. El proyecto se puede planificar de manera más acertada en los tiempos de ejecución, ya que se conoce con exactitud el avance de cada fase y por consiguiente proyectar su finalización.
8. Se minimiza el número de errores y se incrementa la calidad del producto.
9. Cada iteración entrega unos resultados concretos para el proyecto y el usuario de tal manera que se obtienen funcionalidades gradualmente y no hasta el final.

Algunos de los modelos iterativos que se están utilizando en este momento son: Proceso Unificado Racional (RUP), Programación Externa y SCRUM.

## **2.1. Proceso Unificado Racional - PUR**

El Proceso Unificado Racional – PUR es una metodología de desarrollo de “software” que se basa en componentes e interfaces para analizar, diseñar, codificar, implementar, probar y documentar sistemas orientados a objetos y se vale para ello de un Lenguaje Unificado de Modelado - UML.

### **Elementos del PUR**

1. Equipo de trabajo: define el rol de las personas que están a cargo del desarrollo del proyecto.
2. Actividades: son las tareas que realizan los miembros del equipo con los elementos del proyecto.
3. Artefactos: son los productos tangibles que se obtienen de cada fase e iteración del proyecto.



4. Flujo de actividades: secuencia de actividades que se llevan a cabo para la obtención de un artefacto.

## **Fases PUR**

Fases para el desarrollo de un videojuego en el Proceso Unificado Racional. Basado en Gustavo Torossi (s. f.).

### **1) Alcances y objetivos**

#### **Inicio**

En esta fase se define el alcance y los objetivos del proyecto de videojuego, determinando los requerimientos de los usuarios, el conjunto de funcionalidades, la estructura preliminar del diseño con la cantidad de elementos, personajes, características e iteraciones, se identifican los riesgos de mayor importancia y los costos del desarrollo.

### **2) Arquitectura**

#### **Elaboración**

En esta fase se planifica el prototipo del videojuego detallando los casos de uso es decir las funciones que va a tener, se mitigan los riesgos de mayor importancia, se planean las iteraciones del proyecto y las interfaces correspondientes a cada una de las escenas especificando los objetos, personajes, efectos, iluminaciones y animaciones.

### **3) Versión 1**

#### **Construcción**

En esta fase se llevan a cabo todas las iteraciones para el desarrollo del videojuego, teniendo en cuenta los requerimientos y el diseño.

- a) Requerimientos.
- b) Análisis y Diseño.
- c) Codificación: se construyen las líneas de código necesarias en los componentes y las funciones que se planificaron.

- d) Prueba: se realizan pruebas con los usuarios y se configuran controles sobre los productos entregables, los cambios sobre los mismos y las versiones que se vayan obteniendo.
- e) Gestión del proyecto: se gestionan los riesgos y las restricciones, culminando con una versión del producto.
- f) Gestión y configuración de cambios.
- g) Entregas incrementales: cada entrega se documenta hasta completar un manual.
- h) Iteración.

#### 4) Versión final

##### Transición

Aquí el producto se convierte en la primera versión y el usuario está usando de manera activa el videojuego. El equipo desarrollador puede extender las funciones de acuerdo con el “feedback” recibido y ajustar los elementos y características, es decir también se llevan a cabo iteraciones, pero con el producto en uso hasta alcanzar la versión final.

##### Características PUR

- a) Dirigido por casos de uso:** se refiere a que dentro del modelo se definen claramente las funcionalidades que el usuario espera encontrar en el producto para poder hacer uso de él.
- b) Centrado en la arquitectura:** se define la estructura completa del proyecto y los componentes principales que le dan la lógica de funcionamiento y ejecución.
- c) Iterativo incremental:** en cada fase del proyecto se definen los flujos de actividades que van constituyendo las partes del producto y sus funciones.

##### Ejemplo de PUR

El equipo de diseño de un videojuego debe desarrollar el caso de uso: **ingresar al videojuego, jugar y salir**.

Se deben codificar las funciones que debe realizar el sistema para que el jugador pueda ingresar, empezar a jugar y salir del juego. En la figura se ilustra el diagrama con los requerimientos funcionales.

También se pueden observar las funciones que debe desarrollar el programador para que un usuario pueda empezar a usar el juego que se propone.

Estas funciones hacen parte de una iteración que consiste en: **registro de usuario - carga escenario principal - salir** que conllevan la elaboración de una serie de tareas a codificar: **solicitud de datos al usuario, validación de los datos, creación de alias, ingreso del usuario, activación de las funciones del videojuego, proceso de desactivación y salida del videojuego**.

**Las funciones por codificar son:**

- a) Menú: el usuario puede ver las diferentes opciones para explorar los componentes del videojuego, con lo cual puede informarse acerca de él. Debe hacer clic en registrarse para ingresar.
- b) Ingresar: el jugador debe ingresar un alias para identificarse en el juego, el cual es validado por el sistema para determinar si ya existe o es nuevo.
- c) Jugador: el jugador debe escribir el nombre que ha validado el sistema para que se habiliten todas las funciones que puede utilizar.

- d) Cargar escenario: el sistema carga el escenario principal para que el jugador comience a demostrar sus habilidades y controla que este ascienda de nivel o se quede en el que está.
- e) Salir: el sistema controla la salida del jugador por pérdida de la partida o por que ha ganado, en ningún momento es igual a cerrar el juego, ya que si es así cuando ingrese lo dejará en el mismo punto.

El desarrollo de estas funciones corresponde a una iteración completa que, solo cuando esté terminada, se solicita la prueba a los usuarios para poder seguir con el siguiente caso de uso.

## **2.2. Programación extrema - XP**

La programación extrema - XP o Extreme Programming es una metodología de desarrollo de “software” que involucra a los usuarios desde el comienzo del proyecto hasta que finaliza siendo esta su mayor fortaleza, ya que asegura un desarrollo a la medida de acuerdo con las pruebas que estos van realizando a los productos entregados.

Durante el desarrollo del proyecto se pueden producir muchos cambios y esto garantiza una mayor aproximación del producto a lo que espera el usuario de él. Al igual que PUR posee elementos, trabaja en diferentes fases para lograr los resultados y tiene características que lo diferencian del anterior modelo.

## Elementos XP

**Equipo de trabajo: los diferentes roles son:** desarrolladores, usuarios, testadores y *personal de seguimiento*.

**Actividades:** son las tareas que hacen parte de la planificación del proyecto, agrupadas por iteraciones.

**Artefactos:** están representadas en las tarjetas de control o fichas que describen los elementos que se elaborarán como producto entregable.

**Flujo de actividades:** son las tareas que se van a desarrollar en secuencia para la obtención de un producto entregable.

**Fases:** corresponde a las diferentes etapas que se deben llevar a cabo para obtener el producto final.

## Fases XP

### Alcance y objetivos

**Planificación:** es la etapa donde se recogen los requisitos de los usuarios en cuanto a las funcionalidades que espera que tenga el producto con el que va a interactuar. El usuario establece las prioridades de cada requerimiento para que el equipo de ejecución planifique las iteraciones, los tiempos de programación y los costos de desarrollo.

### Prototipo- Pruebas de unidad y aceptación

**Diseño:** en esta etapa se define la estructura del proyecto con los elementos y componentes que debe tener para cumplir con los requerimientos, se elaboran las tarjetas que describen cada objeto que se pondrá en el producto, se realiza un glosario de términos especificando los nombres de los componentes.

También se mitigan los riesgos, se elaboran las pruebas de unidad con las que los desarrolladores verifican el funcionamiento del código y las pruebas de aceptación realizadas por los usuarios para verificar si, al final de una iteración, se cumplió con la funcionalidad del producto. Los entregables de esta etapa son el prototipo y las pruebas.

### **Integración de componentes/ Codificación de pruebas**

**Codificación:** es la fase de desarrollo del producto que se debe realizar por parejas para garantizar que se realice en un código universal y que cualquier persona pueda continuar el trabajo o hacer una modificación.

El resultado de esta etapa es la integración de los elementos creados en el diseño e igualmente la codificación de las pruebas (de unidad y aceptación) para verificar el cumplimiento de los estándares de programación y los requisitos de funcionalidad planteados en las anteriores etapas.

### **Unitarias y de aceptación**

**Prueba:** se elaboran dos tipos de pruebas: las unitarias y las de aceptación. Las primeras son diseñadas y aplicadas por los desarrolladores para verificar el funcionamiento del sistema con las líneas de código y las segundas son diseñadas y aplicadas por los usuarios para determinar el cumplimiento incremental de las funcionalidades que se van implementando en el proyecto y que son verificadas con datos reales.

### **Versiones del “software”**

**Lanzamiento:** corresponde al control de las versiones que se obtienen en el proceso de elaboración del producto y cuando se hacen modificaciones de este. Estas versiones se deben almacenar en un repositorio que conserve los históricos de lo que ha acontecido con el producto para gestionar las actualizaciones que se han llevado a cabo en una línea de tiempo.

## Características de EX

**Dirigido por historias de usuario:** se refiere a los requerimientos de los usuarios que se expresan como historias, de hecho, es él quien los elabora, por eso hay una colaboración activa por parte de ellos en todas las fases del proyecto.

**Diseño sencillo:** se realiza el diseño necesario para satisfacer las necesidades actuales y no las futuras.

**Iterativo incremental:** se realizan iteraciones o entregas pequeñas siendo la primera aquella que le proporcione la funcionalidad mínima al sistema, se van añadiendo paulatinamente funciones que le sumen a la anterior.

**Refactorización:** consiste en eliminar las líneas de código que pueden repetirse a lo largo del desarrollo del producto. Lo que se hace es reutilizar los fragmentos de programas que realicen procesos que se pueden aplicar en otras funciones.

### Ejemplo de PUR

El siguiente es un ejemplo de las iteraciones que se pueden obtener de acuerdo con las historias de usuario para ingresar a un videojuego, empezar a jugar y salir. Obsérvese que el usuario ha diseñado tres (3) historias que debe codificar el equipo desarrollador en la interfaz o pantalla de inicio.

Para los programadores cada historia que ha diseñado el usuario es una iteración que contiene líneas de código por desarrollar y es probada por el usuario inmediatamente se construye, luego utilizando XP se tienen tres (3) iteraciones a diferencia del proceso unificado que ha reunido el proceso en una sola iteración.

El siguiente formato es una ficha que han diseñado desarrollador y usuario para consignar la historia y tanto el uno como el otro se deben diligenciar conjuntamente.

**La historia de usuario # 1 de la interfaz (pantalla) de inicio del juego:** corresponde al requerimiento de poder entrar al juego y hacer el registro. La estimación es el grado de dificultad para el desarrollador la cual puede determinar en una escala de 1 a 10, siendo 1 la

menor dificultad. El valor es la prioridad o grado de importancia que le da el usuario dentro de las funciones del juego.

**Figura 12.** La historia de usuario # 1 en pantalla de inicio

Historia de usuario # 1		Pantalla de inicio
Como jugador quiero entrar al juego y poder registrarme.		
Estimación	Valor	Condiciones de satisfacción
1	10	Antes de iniciar el juego quiero que aparezca una pantalla con el nombre del juego y unos botones en la parte superior para registrarme.

**La historia de usuario # 2 de la interfaz (pantalla) de inicio del juego:** corresponde al requerimiento de entrar al juego con un alias que el sistema debe verificar para ser usado, de lo contrario buscar otro. El grado de dificultad para el desarrollador es superior al mínimo y la prioridad para el usuario es la máxima.

**Figura 13.** La historia de usuario # 2 en pantalla de inicio

Historia de usuario # 2		Pantalla de inicio
Como jugador quiero que al ingresar con mi alias se habiliten todas las funciones del juego.		
Estimación	Valor	Condiciones de satisfacción
4	10	Que el juego habilite todas las entradas a las diferentes acciones que puedo llevar a cabo en el juego.



**La historia de usuario # 3 de la interfaz (pantalla) de inicio del juego:** corresponde al requerimiento de salir sin bloquear el juego o afectar funciones. El grado de dificultad para el desarrollador es considerable puesto que es superior a la mitad de la escala, ya que debe construir líneas de programación que eviten que el programa presente un bloqueo de sus funciones y que pueda continuar donde estaba en el momento de cerrarlo, aunque el jugador cierre la ventana sin utilizar la opción salir. Para el jugador no es tan prioritaria esta opción, puesto que le ha dado un valor de “6”.

**Figura 14.** La historia de usuario # 3 en pantalla de inicio

Historia de usuario # 3		Pantalla de inicio
Como jugador quiero salir del juego en cualquier momento y poder continuar donde está cuando entre.		
Estimación	Valor	Condiciones de satisfacción
6	6	Que el juego presente la opción de salir sin tener que cerrar la ventana y sin importar qué esté haciendo. Volver a ingresar donde lo había dejado.

### 2.3. Scrum

Es una metodología de trabajo en equipo para el desarrollo de un proyecto. El término fue extraído del deporte Rugby en el que un número de jugadores de cada equipo hace una formación fija para entrar a disputar la tenencia del balón y cada vez que este sale del juego se hace un nuevo Scrum o formación. Esta metodología fue desarrollada por Nonaka y Takeuchi en 1986 para agilizar el desarrollo de productos comerciales en cualquier sector de

la economía. Es aplicable en la industria tecnológica para el desarrollo de “software” de diferentes categorías entre ellas los videojuegos.

Esta metodología está enfocada en el trabajo organizado de manera colaborativa y en lo que cada persona puede aportar al equipo para cumplir los objetivos. Implementa buenas prácticas para la obtención rápida de resultados, establece las entregas incrementales acorde con los requisitos, se miden los tiempos y se observa la retrospectiva de trabajo para mejorar las actividades que impidan alcanzar la mayor productividad.

### **Elementos Scrum**

**Equipo de trabajo:** está conformado por el cliente o dueño del producto, el Scrum manager o gestor del proyecto, el equipo Scrum que desarrolla el producto, los stakeholders o asesores del proyecto (internos o externos) y los usuarios que realizan las pruebas del producto.

**Pila de producto:** es la lista priorizada de requisitos para la elaboración del producto, en la que puede intervenir todo el equipo para su elaboración, pero solo el cliente y el director del equipo es responsable de decidir sobre ella.

**Artefactos:** están representados por la información clave generada en el proceso de desarrollo del producto; entre la que se encuentra la pila de producto, la planificación de la pila y la retrospectiva.

**Flujo de actividades:** es la lista de tareas de iteración que se deben ejecutar para el desarrollo de la pila del “sprint”.

**Fases:** corresponde a las diferentes etapas del proceso que se lleva a cabo para obtener el producto final.

### **Fases para el desarrollo de un videojuego en el modelo Scrum**

#### **Planificación de la iteración**

1. En la Planificación de la iteración el equipo desarrollador selecciona los requisitos de la lista que el cliente elaboró, los prioriza y despeja las dudas, realiza una lista de las tareas que se van a llevar a cabo en la iteración.

2. Planifica el tiempo de las tareas; el cual es generalmente de una semana (1) por actividad hasta completar un máximo de cuatro (4) por iteración es decir 30 días.
3. Se organizan en parejas o en las personas que sean necesarias para completarla.

### **Sincronizaciones diarias**

De acuerdo con las respuestas proporcionadas por los integrantes del equipo el Scrum, el manager (gestionador del proyecto) debe encargarse de hacer la inspección de los obstáculos y hacer las adaptaciones necesarias para eliminarlos, garantizando que el equipo pueda realizar las actividades para terminar la iteración.

En Sincronizaciones diarias el equipo realiza reuniones diarias normalmente de 15 minutos frente a un tablero para autoevaluarse frente a los avances en la tarea planteándose las siguientes preguntas:

¿Qué he hecho desde la última reunión de sincronización para aportar al cumplimiento de la tarea?

¿Qué voy a hacer a partir de este momento para ayudar al equipo a cumplir el objetivo de la tarea?

¿Qué obstáculos me impiden colaborar con el trabajo en equipo para conseguir el objetivo?

### **Retrospectiva**

1. El Scrum manager junto con el equipo realiza una retrospectiva del trabajo para identificar en que se puede mejorar la metodología y las técnicas utilizadas para incrementar la productividad.
2. De acuerdo con las observaciones recibidas por el cliente se inspeccionan de nuevo los obstáculos que impiden la optimización del producto obtenido y se hacen las adaptaciones necesarias.

3. En la **Retrospectiva**, es decir el día que se tiene previsto terminar la iteración, se hace nuevamente una reunión de 1 o 2 horas con el cliente para realizar una demostración del cumplimiento de los requisitos planteados inicialmente plasmados ahora sí en el producto incremental, resultado de la iteración completada.

### **Características de Scrum**

Este componente o *asset* permite realizar un testeo o evaluación rápida de las funcionalidades que se van incorporando al juego. Es una herramienta que se debe integrar desde la opción de *Asset Store* del menú de Unity ya que una vez se instala el programa no es importada de la nube automáticamente.

Pasos para incorporar esta herramienta dentro del programa de Unity:

**Elaboración de la pila de producto:** se refiere a los requerimientos que el cliente hace al equipo desarrollador, los cuales son elaborados en forma de pila o lista de requisitos, que se deben priorizar para su desarrollo.

**Planificación flexible:** el cliente puede aumentar o cambiar los requisitos durante el proceso de desarrollo de cada iteración, esto significa que lo definido en un comienzo no se toma como una camisa de fuerza y el equipo debe estar en capacidad de dar respuesta a las nuevas necesidades.

**Desarrollo incremental:** las iteraciones se planifican con el cliente de tal manera que el resultado sea un producto incremental, es decir, una parte de este hasta obtener el resultado final. Una vez desarrolladas las funciones con las que se llevan a cabo operaciones, se presentan al cliente para que éste pueda probarlas y hacer el “feedback” inmediatamente. El equipo inspecciona y hace las adaptaciones si son necesarias antes de terminar la iteración.

### **Ejemplo de Scrum**

El siguiente ejemplo es lo que se haría en la fase 1 de la situación planteada anteriormente sobre lo que se debe realizar en la o las iteraciones para que un usuario pueda ingresar a un videojuego.

En la siguiente tabla se observa que hay una sola iteración compuesta de seis (6) tareas a desarrollar, el equipo se puede repartir por parejas para ejecutar cada una de ellas. Cada pareja tiene determinados días para el desarrollo de la iteración.

**Tabla 2.** Ejemplo de Scrum

Lista de requisitos priorizada	Lista de tareas de la iteración	Planificación de iteración
Entrar al juego.	1. Codificación de Login/Logout (entrada/salida).	<b>Equipo:</b> 2 personas desarrollan. <b>Tiempo:</b> 5 días. <b>Producto:</b> interfaz (pantalla) de ingreso al juego, menú de inicio de sesión, alternativas de <i>Login</i> y <i>Logout</i> .
Hacer registro.	2. Codificación de formulario para registro de datos y alias.	<b>Equipo:</b> 2 personas desarrollan. <b>Tiempo:</b> 5 días. <b>Producto:</b> formulario para registro de datos, con nombre, apellido, alias, edad, contraseña.
Colocar alias.	3. Elaboración de base de datos de consulta de alias.	<b>Equipo:</b> 2 personas desarrollan. <b>Tiempo:</b> 5 días. <b>Producto:</b> formulario para registro de datos, con

Lista de requisitos priorizada	Lista de tareas de la iteración	Planificación de iteración
		nombre, apellido, alias, edad, contraseña.
Empezar a jugar.	4. Diseño de funciones del juego. 5. Codificación de instrucciones para activar funciones iniciales en la interfaz inicial del usuario.	<b>Equipo:</b> 2 personas desarrollan. <b>Tiempo:</b> 5 días. <b>Producto:</b> diseño funciones del juego.  <b>Equipo:</b> 2 personas desarrollan. <b>Tiempo:</b> 5 días. <b>Producto:</b> funciones de la interfaz del usuario activadas.
Interrumpir el juego.	6. Codificación de instrucciones para parar las funciones que se estén ejecutando y guardar la memoria donde quedaron para cuando se reinicie el juego.	<b>Equipo:</b> 2 personas desarrollan. <b>Tiempo:</b> 7 días. <b>Producto:</b> función de interrupción del juego activada.
Salir.	Codificación de Login/Logout (entrada/salida) ya se hizo en la tarea 1.	<b>Producto:</b> función salir activada por el usuario cuando desee abandonar

Si se quiere profundizar en estos temas, se recomienda la siguiente lectura complementaria:

Proyectos ágiles.org. (s.f). [Desarrollo iterativo e incremental.](#)

Pérez, O (2011). [Cuatro enfoques metodológicos para el desarrollo de “software”.](#)

### 3. Modelos de prueba de usabilidad

Probar la usabilidad de un proyecto, producto o, en este caso, un videojuego es, según Lorés (2006), evaluar que este tenga interacción con el usuario. Las pruebas de usabilidad comprenden la utilización de metodologías y técnicas para determinar en un producto su capacidad y facilidad de ser usado y así lograr una experiencia de usuario satisfactoria.

Lorés se refirió a la definición que hace la ISO 98 sobre usabilidad como la medida en la que un producto puede ser usado por los usuarios para lograr sus objetivos con efectividad y eficiencia en un contexto específico, pero este término está ligado al de experiencia de usuario (UX) el cual, según la International Standard ISO 9241-210 (2010), tiene que ver con “las percepciones y respuestas de la persona, resultantes del uso y/ o uso anticipado de un producto, sistema o servicio”.

En el proceso de diseño y desarrollo de videojuegos se debe evaluar permanentemente la experiencia del usuario al interactuar con el producto, de tal manera que llegue a tener plena satisfacción y de esta forma evitar rediseños posteriores que implique mayores recursos de los que se habían presupuestado.

Para lograr usabilidad y por consiguiente una UX satisfactoria en un videojuego se debe chequear entonces que la interacción del juego con el usuario fluya de manera natural, para ello se pueden hacer pruebas antes de empezar la fase de diseño y programación:

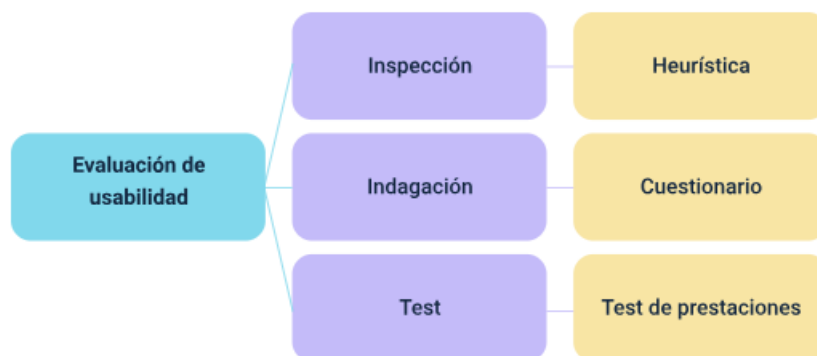
Traynor, V. (2015). [Pruebas de usabilidad con Verónica Traynor.](#)

El video refiere una prueba realizada individualmente a dos personas de edades diferentes en distintos momentos y de las cuales el diseñador ha observado y extraído los detalles respecto a las lógicas mentales que utilizan la una y la otra para realizar la misma acción.

En las pruebas de usabilidad se observan aspectos del usuario y del sistema. En los usuarios se observa la forma en que este usa el producto para conseguir el objetivo y en el sistema la respuesta que provee el programa ante las interacciones que el usuario realiza con el producto.

Lorés (2006) ha clasificado en modelos o metodologías la forma de evaluar la usabilidad y dentro de ellas menciona técnicas que se pueden utilizar para probarla:

**Figura 15.** Pruebas de usabilidad en videojuegos



En la figura anterior se observan claramente tres modelos o metodologías de prueba de usabilidad los que, a su vez, poseen técnicas que se pueden aplicar para probar el producto.

### 3.1. Modelo de inspección

Esta metodología comprende un conjunto de técnicas que se evalúan y se enfocan en la interfaz del usuario (UI), es decir se basa en los puntos de interacción y comunicación del usuario con los aspectos de visualización. En el caso del videojuego se evalúan los criterios que tienen que ver con elementos específicos de los escenarios con los que interactúa el jugador. Dentro de este método se encuentran las siguientes técnicas:



**Técnica de evaluación heurística:** consiste en el análisis de la interfaz con la cual interactúa el usuario para determinar si responde a los siguientes principios de usabilidad, donde cada principio corresponde a una “H” en la heurística:

### **H1 Visibilidad del estado del sistema**

El usuario debe recibir una retroalimentación del estado del producto en cada instante para saber qué pasa cuando interactúa con él.

Por ejemplo, en un videojuego se debe poder determinar si el programa se está cargando para empezar a jugar o si hay algún tipo de error.

### **H2 Relación del producto con el mundo real**

Se debe brindar al usuario toda la información necesaria para que no tenga dudas o cometa errores en la acción que va a realizar.

Por ejemplo, si se va a registrar en el juego debe poder ingresar sus datos a través de un formulario donde se le pregunte lo mínimo necesario como la edad, ya que todos los juegos no son aptos para menores.

### **H3 Libertad al usuario**

El sistema debe permitir que el usuario cancele un proceso que estaba realizando sin finalizarlo, de tal manera que esta acción no representa compromisos o penalizaciones que puedan llegar a afectar en cualquier aspecto de su estado emocional o material.

Por ejemplo, si el jugador intentó comprar el juego y después se arrepiente.

### **H4 Consistencia**

Utilizar un patrón de diseño de los elementos del producto no solo le proporciona coherencia sino claridad y facilidad de uso a la hora de ser utilizado por el usuario.

Por ejemplo, cuando se diseñan botones se deben diseñar del mismo color y tamaño.

### **H5 Prevención de errores**

Se deben dar instrucciones claras a los usuarios sobre cómo manejar el producto y las acciones que deben realizar para evitar su mal uso y, posteriormente, el abandono, debido a los errores que puede arrojar.

Por ejemplo, dibujar unas flechas en el camino para indicarle al jugador por dónde seguir.

## **H6 Recordación**

El producto debe proporcionar toda la información posible en aquellos momentos donde el usuario necesita recordar los detalles clave que le permitan confirmar la decisión que está a punto de tomar.

Por ejemplo si el jugador quiere adquirir el videojuego se le debe proporcionar el detalle del valor y no esperar a que él recurra a su memoria.

## **H7 Flexibilidad y eficiencia de uso**

Se refiere a la capacidad de utilización del producto independientemente del tipo de usuario; es decir, si a este se le facilita el uso de las herramientas tecnológicas o si, por el contrario, no le es sencillo ni familiar hacerlo.

Por ejemplo se puede poner un mensaje cuando pase el *mouse* por cada botón para indicar qué acción puede realizar en el juego.

## **H8 Diseño minimalista**

Es importante no saturar de información al usuario del producto, mostrándole contenido innecesario que lo distraiga y lo desvíe del objetivo de utilización del producto.

Por ejemplo, las imágenes de las escenas deben contener pocos elementos para que el usuario entienda mejor las mecánicas del juego.

## **H9 Reconocer y corregir errores**

Proporciona información al usuario de lo que está generando errores en la utilización del producto multimedia, y le brinda recomendaciones de las acciones que puede realizar para un buen uso del sistema.

Por ejemplo, si el jugador desea descargar el videojuego y se presenta un problema, el sistema debe indicarle cuál fue el error.

### **H10 Ayuda y documentación**

Se deben detectar las dudas más comunes de los usuarios cuando está usando el producto e incluir la información necesaria para que él mismo pueda encontrar la manera de resolverlas. De esta forma, el producto se hace auto gestionable.

Por ejemplo, incluir una explicación de cómo usar cada icono del juego haciendo clic en él.

Para aplicar esta técnica se deben definir las heurísticas a evaluar y se llevan a cabo métricas de evaluación a través de una tabla de relevancia que mide la severidad del error o falla del producto y una tabla de frecuencia que determina cuántas veces ha fallado el producto en las heurísticas seleccionadas. En el ejemplo que se expone más adelante se verá con más detalle su aplicación.

## **3.2. Modelo de indagación**

En este método se investiga acerca de lo que el usuario espera resolver con el producto, por eso es recomendable realizarlo en la fase de diseño y también en la fase de terminación, puesto que es en estos dos momentos, donde se verifica lo que el usuario necesita antes y lo que puede satisfacerlo cuando hace uso del producto.

Las técnicas utilizadas en este modelo tienen que ver directamente con el usuario, haciéndole preguntas verbales o escritas sobre el producto para obtener respuestas precisas que conduzcan a la generación de ideas de diseño y un posterior desarrollo a la medida.

### **Técnica de cuestionario**

En esta técnica se puede seleccionar un grupo amplio de personas que podrán con

testear las preguntas en tiempos diferentes y se puede realizar cuantas veces sea necesario; por ello es la más flexible de utilizar, incorporando varios tipos de preguntas:

**a) Preguntas de tipo general**

Son aquellas que ayudan a identificar el perfil del usuario como sexo, edad, nivel de educación, cargo que ocupa en la entidad, lugar de residencia, dirección, email, entre otras.

**b) Preguntas abiertas**

Permiten recoger información de apreciación sobre algo en particular, interviene la parte subjetiva por que la respuesta es dada de acuerdo con el juicio del usuario, son útiles para atacar otros puntos de vista del producto.

**c) Preguntas cerradas**

Permiten al usuario escoger entre varias opciones que se le presentan, por ello son más precisas y concretas, fáciles de tabular, graficar y analizar después, por ello se pueden obtener rápidamente los resultados.

**d) Preguntas tipo escala**

El usuario debe escoger una opción de una escala numérica. Por ejemplo: ¿la aplicación es comprensible?

**e) Preguntas con opción multiple**

Se presentan al usuario varias opciones de respuesta a una pregunta dada, para solamente seleccione una. Por ejemplo: ¿cuál de las siguientes opciones le resultó más fácil de utilizar?

**1. Pre-tarea**

Antes de que el usuario realice una tarea con el producto con el propósito de medir sus habilidades con el manejo de los sistemas informáticos.

**2. Post-tarea**

Después de que el usuario ha interactuado con alguna función del producto para medir su usabilidad.

### 3. Post-test

Después de que el usuario ha interactuado con todas las tareas que se le han planteado y se pueda medir su percepción global del producto.

Se pueden realizar cuestionarios en diferentes momentos.

#### 3.3. Modelo de test

Este método se utiliza cuando se tiene el prototipo del producto, para lo cual el diseñador selecciona un grupo de usuarios a los que se les asignan diferentes tareas a ser ejecutadas para determinar cómo el sistema entrega los resultados de lo que se le está solicitando. Los resultados de esta prueba son utilizados para determinar si el diseño de las interfaces puede soportar las tareas que los usuarios están realizando con el producto.

**Técnica de test de prestaciones:** esta técnica se utiliza para medir aspectos del uso del producto, por ello se debe llevar a cabo sobre el prototipo o sobre el producto terminado de tal forma que el evaluador pueda visualizar el número de errores que arroja el aplicativo, o las percepciones de los usuarios frente a lo que el producto lo puede beneficiar, por lo que se considera una técnica cuantitativa y cualitativa, en ella se debe:

- a) Seleccionar usuarios reales o potenciales que realmente usarán el producto.
- b) Entregar instrucciones claras acerca de cómo presentar la prueba.
- c) Elaborar una lista de tareas que los usuarios realizarán con el producto y analizar la manera en que lo utilicen y el tiempo que tardan en ejecutar cada tarea asignada.
- d) Observar el comportamiento del usuario durante la prueba.
- e) No entregar pistas ni orientaciones al usuario de cómo usar o recorrer la aplicación/producto.

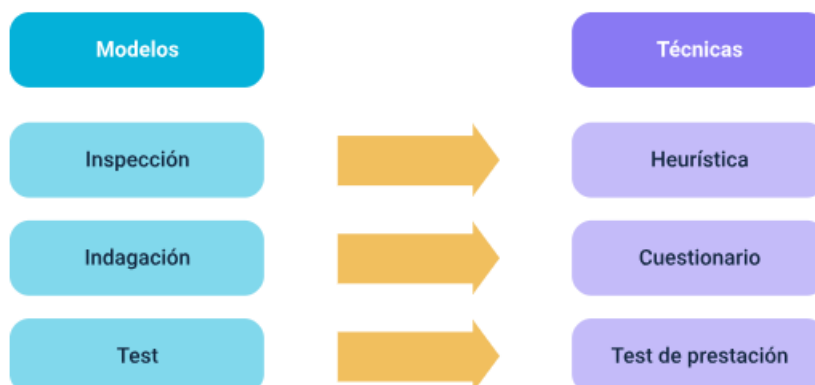
- f) Grabar las expresiones del usuario y la forma de usar la aplicación o el producto.
- g) Analizar los datos recolectados en la ejecución de las tareas, los problemas detectados para llevar a cabo soluciones que mejoren la usabilidad del producto.

### 3.4. Ejemplo de aplicación

Partiendo del supuesto que el equipo de diseño ha creado un juego llamado “La isla de los tesoros escondidos” en este, el jugador debe encontrar tesoros escondidos en diferentes puntos de la isla, a los que se debe llegar haciendo la búsqueda y superando obstáculos. Si el jugador está muy perdido puede solicitar pistas pagando mucho dinero.

El equipo ha decidido hacer pruebas de usabilidad aplicando los siguientes modelos y técnicas:

**Figura 16. Modelos y técnicas**



Antes de empezar las pruebas, el equipo debe definir con exactitud:

### **1. Objetivo de la prueba**

Es el propósito que condujo a la aplicación de la evaluación ya que hay muchos aspectos que se pueden medir en los productos o aplicativos informáticos en general y se debe especificar a cuál de ellos le apuntará las pruebas que se realicen.

### **2. Participantes**

Se deben definir las características o competencias que deben cumplir los usuarios para la realización de las pruebas, ya que estos deben ser seleccionados basados en ellas. Dichas características deben ser similares y se recomienda un número de participantes aproximadamente de ocho (8), ya que como dice Nielsen citado por Vega (s. f.) entre más usuarios participen más repetición de resultados se obtendrán, pero esto lo define el equipo de evaluadores de acuerdo con su conveniencia.

### **3. Descripción de la técnica utilizada**

La técnica que se llevará a cabo debe quedar clara para los participantes al igual que el instrumento que se utilice para la medición. Por ejemplo, las tareas a ejecutar deben ser completamente comprensibles para los usuarios, se deben definir con exactitud las H (heurísticas) que se van a evaluar y las preguntas que contestarán no pueden tener rastros de ambigüedad.

### **4. Análisis de resultados**

Después de aplicadas las pruebas el equipo debe analizar los resultados obtenidos y determinar cuáles son los aspectos que se deben mejorar en el producto para que la experiencia de usuario sea satisfactoria por cumplir con las condiciones de fácil uso.

Conocidos los aspectos para iniciar las pruebas se tiene para las tres técnicas que se va a utilizar el siguiente aspecto en común:

**Participantes:** se convocaron 4 usuarios que les llama la atención los videojuegos por lo que han jugado alguna vez y poseen entre 18 y 22 años, poseen habilidades tecnológicas por las ocupaciones de estudio y trabajo.

### **Primera técnica de usabilidad: heurística**

**Objetivo de la prueba:** en el videojuego de “La isla de los tesoros escondidos” el equipo de diseño y desarrollo deberá determinar si la interfaz de usuario presenta saturación de imágenes y si el jugador puede tomar el control del juego con facilidad a través de las diferentes funciones que posee.

**Descripción de la técnica:** para evaluar los aspectos definidos en el objetivo se utilizará la técnica heurística con la definición de las siguientes H:

#### **H4 Consistencia**

Con esta heurística se determina si el videojuego presenta un patrón de diseño en los botones, barras, pestañas y en general de los elementos de tal manera que le permitan al usuario tomar el control del juego y de las funciones con facilidad.

#### **H8 Diseño minimalista**

Con esta heurística se determinará si la interfaz del usuario está muy cargada de imágenes de tal manera que saturan y confunden.

Según Fernández (2019), la métrica de evaluación del método heurístico es la que utiliza Nielsen:

#### **Relevancia del problema**

En la figura se observa el grado de severidad de un problema presentado con el producto, siendo el más alto (4) cuando el problema es bloqueante o le impide al usuario seguir adelante con otra función, es grave (3) cuando el usuario puede utilizar alguna otra



función del producto, es menor (2) cuando puede operar el producto aunque presente algunos inconvenientes, es cosmético (1) cuando el producto es utilizable y funcional aunque requiera en algún momento de realizar cualquier tipo de actualización para su óptimo funcionamiento, y no supone problema (0) cuando el producto funciona correctamente.

**Figura 17. Relevancia del problema**

Relevancia del problema		
Severidad	Descripción	Prioridad
4	Problema bloqueante.	Imperativa
3	Problema grave.	Alta
2	Problema menor.	Baja
1	Problema cosmético.	-
0	No supone un problema.	-

### Frecuencia del problema

Aquí se observa la frecuencia con que la heurística no satisface al usuario y le asigna una calificación, es decir que si se obtuvo un 4 es porque más del 90% de las veces el producto presenta fallas, si es de 3 es porque entre el 50% y el 90% el producto no cumple con lo esperado por algún tipo de problemas, si es de 2 quiere decir el producto presenta problemas entre el 10% y 25% de las veces en que es utilizado, y si es 0 el producto falla máximo el 10% de las veces y se puede considerar que tiene un buen funcionamiento.

**Figura 18.** Frecuencia del problema

Frecuencia del problema	
Calificación	Porcentaje
4	$P > 90\%$
3	$50\% < P \leq 90\%$
2	$25\% < P \leq 50\%$
1	$10\% < P \leq 25\%$
0	$0\% < P \leq 10\%$

### Heurísticas

Las afirmaciones diseñadas para cada heurística también pueden ser preguntas si se desea. De acuerdo con las pruebas, el número de veces que se puede dar la falla en la toma del control del videojuego (H4) es cinco (5), ya que es la cantidad de veces que se prueba la afirmación y el número de veces que consideran los usuarios se falla por el exceso de imágenes es de tres (3). El porcentaje se halló dividiendo el número de veces de la falla entre el máximo número de fallas el cual es ocho (8). Si se trasladan los resultados a las tablas de relevancia y frecuencia se obtiene.

**Figura 19.** Heurísticas

ID	Heurística	N.º veces de la falla	%
H4	Los elementos de interacción permiten lograr control rápidamente del juego.	5	63%
H8	El videojuego presenta saturación de elementos en la interfaz de la isla.	3	38%

## Nivel de gravedad y número de reiteraciones

En la figura se observa que, de acuerdo con lo establecido respecto a la relevancia y la frecuencia, el problema con mayor prioridad está en la heurística H4 de consistencia con la falla para tomar el control del juego cuando se interacciona con él. La heurística H8 representa un problema menor para el producto, ya que se considera que solo el 38% de la muestra considera que hay exceso de imágenes. Lo anterior quiere decir que el aspecto prioritario a modificar en el producto está en la elaboración de controles más sencillos, siguiendo un patrón de colores o estilos para hacerlo más comprensible y funcional para el jugador.

**Figura 20.** Nivel de gravedad y número de reiteraciones

Relevancia del problema			Frecuencia del problema		
Severidad	Descripción	Prioridad	Calificación	Porcentaje	ID
4	Problema bloqueante.	Imperativa	4	$P > 90\%$	
3	Problema grave.	Alta	3	$50\% < P \leq 90\%$	H4
2	Problema menor.	Baja	2	$25\% < P \leq 50\%$	H8
1	Problema cosmético.	-	1	$10\% < P \leq 25\%$	
0	No supone un problema.	-	0	$0\% < P \leq 10\%$	

## Segunda técnica de usabilidad: cuestionario

**Objetivo de la prueba:** en el videojuego de “La isla de los tesoros escondidos” el equipo de diseño y desarrollo deberá determinar si la interfaz de usuario presenta una estructura sencilla con una interacción clara, buena calidad de imágenes sin saturación de ellas y sonidos acordes a las situaciones presentadas.

**Descripción de la técnica:** para evaluar los aspectos definidos en el objetivo se utilizará la técnica de cuestionario con la definición de las siguientes preguntas:

**Figura 21. Cuestionario**

**Cuestionario sobre el entorno audiovisual e interactivo del videojuego:**

**"La isla de los tesoros escondidos"**

1. La puntuación que considera se le debe dar a la estructura del videojuego de acuerdo con la claridad que presenta es (siendo 1 es el puntaje más bajo y 5 el puntaje más alto):

1	2	3	4	5
---	---	---	---	---

2. ¿En qué aspecto considera que puede mejorar la estructura general de la aplicación?

\_\_\_\_\_

\_\_\_\_\_

3. La cantidad de imágenes que se utilizan en la interfaz son:

Excesivas \_\_\_\_\_

Suficientes \_\_\_\_\_

Escasas \_\_\_\_\_

Deficientes \_\_\_\_\_

4. ¿Es buena la calidad de las imágenes que se presentan en la interfaz?

Sí \_\_\_\_\_

No \_\_\_\_\_

5. Los sonidos utilizados para cada situación o evento del juego son adecuados.

Muy de acuerdo \_\_\_\_\_

Muy en desacuerdo \_\_\_\_\_

De acuerdo \_\_\_\_\_

En desacuerdo \_\_\_\_\_

6. Los mensajes que utiliza el juego después de realizada una interacción, orientan con claridad al usuario respecto a lo que debe hacer después:

Siempre \_\_\_\_\_

Casi siempre \_\_\_\_\_

Algunas veces \_\_\_\_\_

Nunca \_\_\_\_\_

En el cuestionario anterior se observan diferentes tipos de preguntas; cerradas y abiertas para un análisis de tipo cuantitativo y cualitativo. Cuando se tiene las respuestas de los participantes, se hace una tabulación usando la herramienta ofimática de Excel para la obtención de los resultados. Al igual que el método heurístico se obtienen los porcentajes de las respuestas con preguntas cerradas y en las abiertas se tienen en cuenta todas las observaciones realizadas, por ello, el número de participantes debe ser pequeño. De acuerdo con los porcentajes y las observaciones se priorizan los aspectos con mayor

porcentaje de negatividad y deficiencia para realizar las modificaciones y se ejecutan los cambios que requiere cada comentario insatisfactorio de los usuarios.

### **Tercera técnica de usabilidad: test de prestación**

**Objetivo de la prueba:** en el videojuego “La isla de los tesoros escondidos” el equipo de diseño y desarrollo deberá identificar las fallas en las funciones de búsqueda, entrega de pistas, cancelación de una búsqueda, registrarse para obtener beneficios y conexión con otro jugador.

**Descripción de la técnica:** para evaluar los aspectos definidos en el objetivo se utilizará la técnica de test de prestación, utilizando un formato en el que los participantes diligenciarán datos personales básicos y estarán frente al producto terminado realizando la lista de tareas que se indica y anotando los eventos de éxito o de error que perciben en las funcionalidades que se le solicita probar:

**Tabla 3.** Test de prestación

<b>Test de prestación</b>
Método: test de usuario
Técnica: prestación
Instrumento: asignación de tareas
Producto: videojuego “La isla de los tesoros escondidos”
Encargado: XXXXXXXX
Fecha: DD/MM/AAAA
<b>Presentación</b>
<Nombre de usuario> gracias por su disposición para presentar la prueba de usabilidad del videojuego “La Isla de los tesoros escondidos”, la que nos ayudará a detectar los problemas y fallas de uso, para la implementación de mejoras.

## Test de prestación

### Datos del encuestado

Nombres y apellidos	
Dirección	
Teléfono	
Ocupación	
Experiencia con aplicaciones/productos similares	

### Instrucciones

1. Realizar cada tarea asignada y contestar la pregunta o preguntas después de finalizada.
2. Manifestar las dudas que le surjan del aplicativo/producto aunque no sean respondidas por el encargado (no puede hacerlo durante la prueba).
3. No hacer preguntas acerca de cómo hacer la tarea asignada al encargado durante la prueba.
4. El tiempo estimado para la realización de la prueba es de 30 minutos.

### Lista de tareas

Tarea 1: buscar el primer tesoro escondido.

Test de prestación
¿Fue fácil encontrar el tesoro que buscaba? (SÍ/NO) ¿Por qué? (argumente la respuesta).
Tarea 2: pedir una pista y hacer el recorrido hasta que encuentre un tesoro.
¿Pudo encontrar un tesoro con la pista dada por el juego? (SÍ/NO) manifieste cómo lo pudo hacer o por qué no lo pudo hacer.
Tarea 3: cancelar la búsqueda de un tesoro.
¿Pudo cancelar la búsqueda que estaba realizando? (SÍ/NO) ¿Por qué? (explicar qué pasó si hubo algún error en la cancelación).
Tarea 4: registrarse en el juego para obtener más beneficios.
¿Pudo realizar el registro para la obtención de beneficios? (SÍ/NO) ¿Por qué? (explicar qué pasó si hubo algún problema en el registro).
Tarea 5: conectarse con otro jugador para hacer una búsqueda conjunta del tesoro.
¿Pudo establecer contacto con otro jugador para realizar la búsqueda juntos? (SÍ/NO) por qué (explicar qué pasó si hubo algún problema).
¿El aplicativo arrojó algún mensaje de éxito o de error dependiendo de si pudo o no realizar el contacto? SÍ/NO (si la respuesta es SÍ/NO qué mensaje arrojó).

Los formatos diligenciados por los participantes son analizados y verificados por el equipo desarrollador para proceder a la corrección de las fallas presentados y volver a realizar la prueba que comprueben las mejoras en la funcionalidad del producto.

Si se quiere profundizar en las demás técnicas de cada modelo se recomienda la siguiente lectura complementaria:



## Material complementario

Tema	Referencia APA del Material	Tipo de material	Enlace del Recurso o Archivo del documento material
Metodología RITE	Enciendelaluz Agile E Innovación. (2018). <i>Iterativo e incremental</i> . [Video]. YouTube.	Video	<a href="https://youtu.be/_qUIL01th2s">https://youtu.be/_qUIL01th2s</a>
Metodología RITE	Unity3dtutorial. (2015). <i>Unity Test Tools</i> .	Artículo	<a href="https://unity3dtutorial.wordpress.com/2015/09/28/unity-test-tools/">https://unity3dtutorial.wordpress.com/2015/09/28/unity-test-tools/</a>
Modelos de iteración	Proyectos ágiles.org. (s.f). <i>Desarrollo iterativo e incremental</i> .	Artículo	<a href="https://proyectosagiles.org/desarrollo-iterativo-incremental/">https://proyectosagiles.org/desarrollo-iterativo-incremental/</a>
Modelos de Prueba de Usabilidad	Traynor, V. (2015). <i>Pruebas de usabilidad con Verónica Traynor</i> . [Video]. YouTube.	Video	<a href="https://youtu.be/uh4sZmQiRr4">https://youtu.be/uh4sZmQiRr4</a>
Modelos de Prueba de Usabilidad	Lorés, J. et al. (2006). <i>Evaluación</i> .	PDF	<a href="https://aipo.es/libro/pdf/04Evalua.pdf">https://aipo.es/libro/pdf/04Evalua.pdf</a>

## Glosario

**Indagación:** proceso para tratar de llegar al conocimiento de una cosa discuriendo o por conjeturas y señales (Lorés, 2006).

**Inspección:** nombre genérico para un conjunto de métodos basados en evaluadores que examinan aspectos relacionados con la usabilidad de la interfaz (Lorés, 2006).

**Interfaz de usuario (UI):** punto de interacción y comunicación usuario - dispositivo, que incluye aspectos de visualización como la pantalla, teclado, mouse, entre otros (Churchville, s.f.).

**Iteraciones:** miniproyectos donde se repite un proceso de trabajo similar para proporcionar un resultado completo sobre el producto final (proyectos ágiles.org, s.f.).

**PUR:** Proyecto Unificado Racional (PUR), es una metodología cuyo objetivo es ordenar y estructurar el desarrollo del software, en la cual se realizan un conjunto de actividades para transformar los requisitos del usuario en un sistema (Pérez, 2011).

**Rite:** Rapid Iterative Testing Evaluation, es decir testeo rápido iterativo de evaluación. RITE es una metodología que consiste en una técnica para evaluar el desarrollo de un proyecto a través de ciclos rápidos de ejecución hasta alcanzar el resultado final, a estos ciclos se les llama iteraciones.

**Scrum:** marco de trabajo basado en métodos ágiles cuyo objetivo es el control continuo sobre el estado actual de un software en el cual el cliente establece las prioridades y el equipo SCRUM se autoorganiza para determinar la mejor forma de entregar resultados (Pérez, 2011).

**Test:** método de usabilidad donde los usuarios representativos trabajan en tareas utilizando un prototipo (Lorés, 2006).

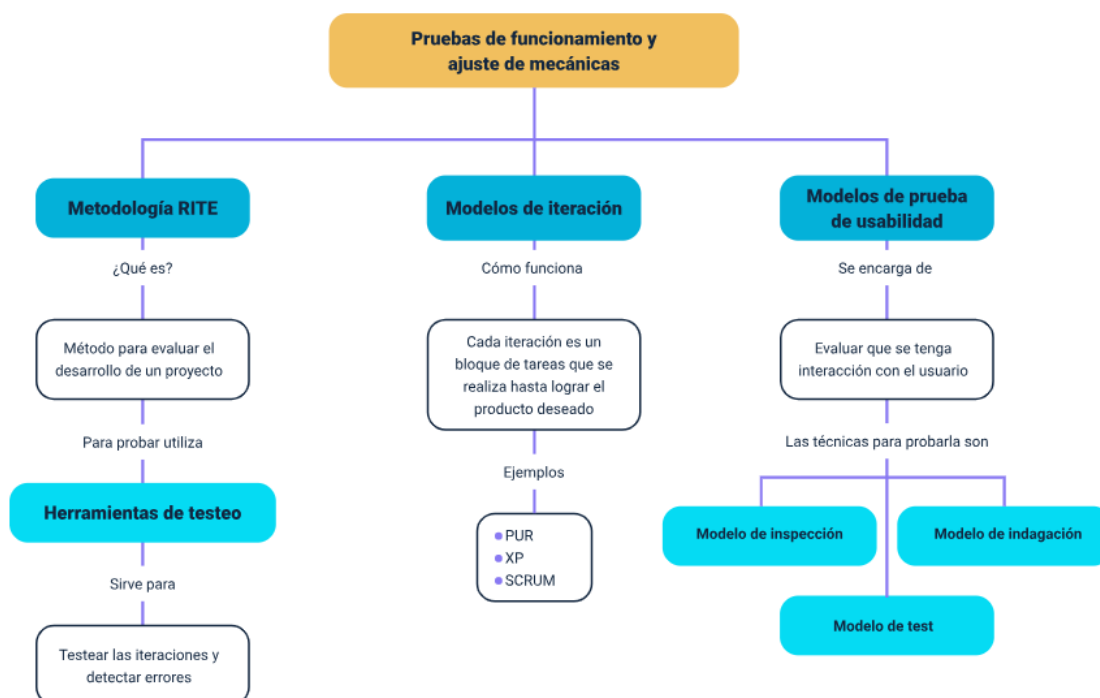
**Usabilidad:** medida en la que un producto puede ser usado por los usuarios para lograr sus objetivos con efectividad y eficiencia en un contexto específico (Lorés, 2006).

**XP:** programación Extrema (XP) es una disciplina de desarrollo de software basada en los métodos ágiles, donde se evidencian los principios de desarrollo incremental,

participación del cliente, el interés se centra en las personas y no en los procesos (Pérez, 2011).

## Síntesis

Manejar métodos, técnicas e instrumentos en la realización de pruebas de usabilidad que permitan determinar la facilidad de uso de los videojuegos creados. Todo esto visto en el componente formativo y que se visualiza en el siguiente esquema.



## Referencias bibliográficas

- Churchville, F. (s.f). *Interfaz de usuario (IU)*.  
<https://searchdatacenter.techtarget.com/es/definicion/Interfaz-de-usuario-UI>
- Fernández, C., P. E. (2019). *Usabilidad web: teoría y uso*. <https://www-ebooks7-24-com.bdigital.sena.edu.co/?il=9087>
- Letelier, P. (2013). *Desarrollo iterativo versus incremental...* [web log post]. *Blogspot*.  
<http://agilismoatwork.blogspot.com/2013/06/desarrollo-iterativo-versus-incremental.html>
- Lorés, J., et al. (2006). *Evaluación*. <https://aipo.es/libro/pdf/04Evalua.pdf>
- Pérez, O. A. (2011). *Cuatro enfoques metodológicos para el desarrollo de Software RUP – MSF – XP - SCRUM*. <https://core.ac.uk/download/pdf/230219821.pdf>
- Proyectos ágiles.org. (s. f.). *Desarrollo iterativo e incremental*.  
<https://proyectosagiles.org/desarrollo-iterativo-incremental/>
- Unity3dtutorial. (2015). *Unity Test Tools*.  
<https://unity3dtutorial.wordpress.com/2015/09/28/unity-test-tools/>

## Créditos

Nombre	Cargo	Regional y Centro de Formación
Claudia Patricia Aristizábal Gutiérrez	Responsable del equipo	Dirección General
Liliana Victoria Morales Gualdrón	Responsable de línea de producción	Centro de Gestión De Mercados, Logística y Tecnologías de la Información - Regional Distrito Capital
Rafael Neftalí Lizcano Reyes	Asesor Pedagógico Ecosistema RED	Regional Santander - Centro Industrial del Diseño y la Manufactura
Olga Lucía Mogollón Carvajal	Experto Temático	Centro para la Industria de la Comunicación Gráfica -CENIGRAF-
Luz Aida Quintero Velásquez	Diseñadora instruccional	Regional Distrito Capital - Centro de Gestión Industrial
Carolina Coca Salazar	Revisora Metodológica y Pedagógica	Regional Distrito Capital- Centro de Diseño y Metrología
José Gabriel Ortiz Abella	Corrector de estilo	Regional Distrito Capital - Centro para la Industria de la Comunicación Gráfica
Yuly Andrea Rey Quiñones	Diseñador web	Centro de Gestión de Mercados, Logística y Tecnologías de la Información - Regional Distrito Capital
Luis Jesús Pérez Madariaga	Desarrollador Fullstack	Centro de Gestión de Mercados, Logística y Tecnologías de la Información - Regional Distrito Capital
Laura Gisselle Murcia Pardo	Animación y Producción audiovisual	Centro de Gestión de Mercados, Logística y Tecnologías de la Información - Regional Distrito Capital
Carolina Coca Salazar	Evaluadora de contenidos inclusivos y accesibles	Centro de Gestión De Mercados, Logística y Tecnologías de la Información - Regional Distrito Capital
Lina Marcela Pérez Manchego	Validadora de recursos educativos	Centro de Gestión De Mercados, Logística y Tecnologías de la Información - Regional Distrito Capital

Castaño Pérez Leyson Fabian	Validador de recursos educativos	Centro de Gestión De Mercados, Logística y Tecnologías de la Información - Regional Distrito Capital
--------------------------------	-------------------------------------	--