**FORMATO PARA EL DESARROLLO DE COMPONENTE FORMATIVO**

| PROGRAMA DE FORMACIÓN | Desarrollo de videojuegos y entornos interactivos |
| --- | --- |

| COMPETENCIA | 220501087. Planear el videojuego de acuerdo con procedimientos y requisitos técnicos. | RESULTADOS DE APRENDIZAJE | 220501087-03. Validar el funcionamiento de las mecánicas del videojuego a través del prototipo físico.  220501087-04. Ajustar las mecánicas del videojuego de acuerdo con el proceso de validación. |
| --- | --- | --- | --- |

| NÚMERO DEL COMPONENTE FORMATIVO | CF007 |
| --- | --- |
| NOMBRE DEL 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. |
| PALABRAS CLAVE | RITE, iteraciones, PUR, XP, SCRUM, usabilidad. |

| ÁREA OCUPACIONAL | 5 - Arte, cultura, esparcimiento y deportes |
| --- | --- |
| IDIOMA | Español |

1. **TABLA DE CONTENIDO**
2. **Metodología RITE**
   1. Herramienta de testeo en Unity
   2. Herramienta de testeo incorporada a Unity
3. **Modelos de iteración**
   1. Proceso Unificado Racional -PUR
   2. Programación Extrema -XP
   3. SCRUM
4. **Modelos de prueba de usabilidad**

3.1 Modelo de inspección

3.2 Modelo de indagación

3.3 Modelo de test

3.4 Ejemplo de aplicación

3.4.1. Primera técnica de usabilidad: heurística

3.4.2. Segunda técnica de usabilidad: cuestionario

3.4.3. Tercera técnica de usabilidad: test de prestación

1. **DESARROLLO DE CONTENIDO**

**INTRODUCCIÓN**

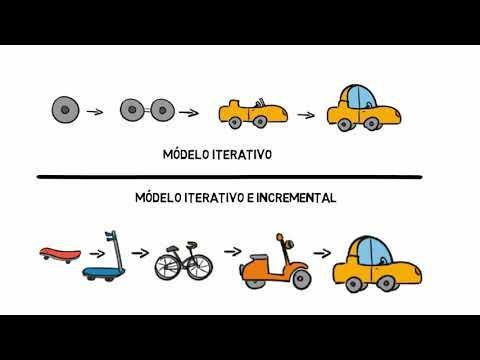
Bienvenido al componente formativo pruebas de funcionamiento y ajuste de mecánicas:

| Video  Anexo: CF007\_video\_introduccion |
| --- |

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*

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. **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.

En la imagen se observan cuatro mensajes de error en la consola; errores que se han cometido en la manipulación del programa, por ejemplo:

| Gráfico interactivo  CF007\_1.1.\_ mensajes\_error\_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. **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:

| Pasos B  CF007\_1.2.\_ Unity\_Test\_Tool |
| --- |

Para una mayor comprensión y profundización de las herramientas de testeo, se recomienda la siguiente lectura complementaria:

| Lectura de consulta: | UNITY3DTUTORIAL. (2015). *Unity Test Tools.* <https://unity3dtutorial.wordpress.com/2015/09/28/unity-test-tools/> |
| --- | --- |

1. **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.

| 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. | *Iteración incremental* |
| --- | --- |

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:

1. El primer producto es colocar las imágenes en una escena; esta entrega consiste en dos iteraciones donde se realiza el mismo proceso:

| Iteración 1  para insertar fondo en ventana de escena | Iteración 2  para insertar superhéroe en ventana de escena |
| --- | --- |
| * Ubicarse en la ventana de proyecto. * Abrir carpeta de *asset* fondos. * Seleccionar el fondo y arrástralo a la ventana de la escena. | * Ubicarse en la ventana de proyecto. * Abrir carpeta de *asset* fondos. * 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 | *Iteraciones* |
| --- | --- |

Estas iteraciones corresponden a la primera entrega incremental.

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

| Iteración 3 para colocar el superhéroe en el mismo lugar del fondo | Iteración 4 para que el superhéroe quede sobre el fondo |
| --- | --- |
| * Hacer clic en el botón que está en el círculo.   *Botones*   * Hacer clic en el superhéroe.   *Superhéroe*   * Arrastrar el superhéroe a la posición del fondo para obtener el siguiente resultado:   *Imagen atrás* | * 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:   D:\SENA\DISEÑO INSTRUCCIONAL\228108 (Videojuegos_Entornos_Interactivos)\1. PCI\Contenido\CF007\Anexos\Imagenes\order layer.png  *Order layer*   * El valor de *order layer* se pone en 1 para que el superhéroe se coloque en el primer nivel.   *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:

* Se pueden tomar decisiones en cada iteración, por ejemplo cambiar de personaje, de fondo, la escala de las imágenes, la rotación o la posición.
* El equipo desarrollador tiene la posibilidad de validar si lo que ha realizado era lo que el usuario esperaba y hacer los ajustes necesarios.
* 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.
* Desde el comienzo se pueden obtener resultados y el usuario puede ir probando la usabilidad del producto aunque este no esté terminado.
* Después de cada iteración se puede obtener el *feedback* del usuario y hacer los ajustes requeridos.
* 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.
* 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.
* Se minimiza el número de errores y se incrementa la calidad del producto.
* 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.

* 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.

| Presentación Interactiva  CF007\_2.1\_ PUR |
| --- |

* 1. **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.

| Presentación Interactiva  CF007\_2.2\_ XP |
| --- |

* 1. **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.

| Presentación Interactiva  CF007\_2.3\_ Scrum |
| --- |

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

| Lecturas de consulta | Proyectos ágiles.org. (s.f). *Desarrollo iterativo e incremental.* <https://proyectosagiles.org/desarrollo-iterativo-incremental/>  Pérez, O (2011). *Cuatro enfoques metodológicos para el desarrollo de software.* <https://core.ac.uk/download/pdf/230219821.pdf> |
| --- | --- |

1. **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.* [Video]. YouTube. <https://youtu.be/uh4sZmQiRr4>

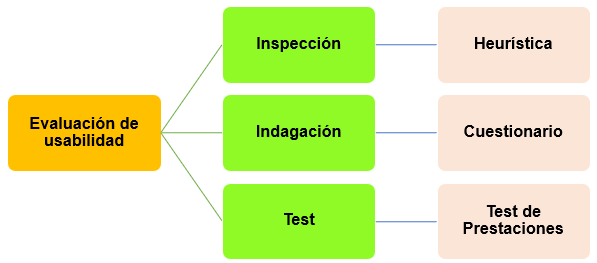
El video refiere una prueba realizada individualmente a dos personas de edades diferentes en distintos momentos y de las cuales el diseñador ha observado y extraído los detalles respecto a las lógicas mentales que utilizan la una y la otra para realizar la misma acción.

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 1**

*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.

* 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:

| Pestañas C  CF007\_3.1\_ Tecnica evaluacion\_heuristica |
| --- |

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.

* 1. **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:

1. **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.

1. **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.

1. **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.

1. **Preguntas tipo escala**

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

| 1 | | 2 | | 3 | 4 | 5 |
| --- | --- | --- | --- | --- | --- | --- |
| Poco |  | |  | |  | Mucho |
|  |  | |  | |  |  |

1. **Preguntas con opción múltiple**

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?

| Iconos | X |
| --- | --- |
| Botones |  |
| Barra de desplazamiento |  |
| Pestañas |  |

| **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. | **Post-tarea**  Después de que el usuario ha interactuado con alguna función del producto para medir su usabilidad. | **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:

* 1. **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:

1. Seleccionar usuarios reales o potenciales que realmente usarán el producto.
2. Entregar instrucciones claras acerca de cómo presentar la prueba.
3. 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.
4. Observar el comportamiento del usuario durante la prueba.
5. No entregar pistas ni orientaciones al usuario de cómo usar o recorrer la aplicación/producto.
6. Grabar las expresiones del usuario y la forma de usar la aplicación o el producto.
7. Analizar los datos recolectados en la ejecución de las tareas, los problemas detectados para llevar a cabo soluciones que mejoren la usabilidad del producto.
   1. **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:

| **Modelos** | **Técnicas** |
| --- | --- |
| Inspección | Heurística |
| Indagación | Cuestionario |
| Test | Test de prestación |

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.

* 1. **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.

* 1. **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.

* 1. **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.

* + 1. **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:

| **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. | **Diseño minimalista**  Con esta heurística se determinará si la interfaz del usuario está muy cargada de imágenes de tal manera que saturan y confunden al usuario con distractores que lo desvíen del objetivo de encontrar los tesoros escondidos |
| --- | --- |

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

| Pestañas A  Anexo: CF007\_3.4\_ejemplo\_heuristicas |  |
| --- | --- |

* + 1. **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:

| 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 | | --- | --- | --- | --- | --- |  1. ¿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\_\_\_   1. ¿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.

**3.4.3. 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:

| **Test de prestación**  **Método :** Test de usuario  **Técnica :** Prestación  **Instrumento :** Asignación de tareas  **Producto :** Videojuego **“La isla de los tesoros escondidos”**  **Encargado :** XXXXXXX  **Fecha :** DD/MM/AAAA  **Presentación**  *<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.  **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.   | ¿Fue fácil encontrar el tesoro que buscaba? (SÍ/NO) porqué (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 que 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:

| Lectura de consulta: | Lorés, J., et al. (2006). *Evaluación*. <https://aipo.es/libro/pdf/04Evalua.pdf> |
| --- | --- |

1. **ACTIVIDADES DIDÁCTICAS (OPCIONALES SI SON SUGERIDAS)**

| DESCRIPCIÓN DE ACTIVIDAD DIDÁCTICA | |
| --- | --- |
| Nombre de la Actividad |  |
| Objetivo de la actividad |  |
| Tipo de actividad sugerida | Interfaz de usuario gráfica  Descripción generada automáticamente |
| Archivo de la actividad  (Anexo donde se describe la actividad propuesta) |  |

1. **MATERIAL COMPLEMENTARIO:**

| Tema | Referencia APA del Material | Tipo de material  (Video, capítulo de libro, artículo, otro) | Enlace del Recurso o  Archivo del documento o material |
| --- | --- | --- | --- |
| 1. Metodología RITE | Enciendelaluz Agile E Innovación. (2018). *Iterativo e incremental.* [Video]. YouTube. <https://youtu.be/_qUlL01th2s> | [Video]. YouTube. | <https://youtu.be/_qUlL01th2s> |
| 2. Metodología RITE | Unity3dtutorial. (2015). *Unity Test Tools.* <https://unity3dtutorial.wordpress.com/2015/09/28/unity-test-tools/> | [Artículo]. | <https://unity3dtutorial.wordpress.com/2015/09/28/unity-test-tools/> |
| 3. Modelos de iteración | Pérez, O. A. (2011). *Cuatro enfoques metodológicos para el desarrollo de Software RUP- MSF-XP-SCRUM.* [*https://core.ac.uk/download/pdf/230219821.pdf*](https://core.ac.uk/download/pdf/230219821.pdf) | [Artículo]. | [*https://core.ac.uk/download/pdf/230219821.pdf*](https://core.ac.uk/download/pdf/230219821.pdf) |
| 4. Modelos de iteración | Proyectos ágiles.org. (s.f). *Desarrollo iterativo e incremental.* <https://proyectosagiles.org/desarrollo-iterativo-incremental/> | [Artículo]. | <https://proyectosagiles.org/desarrollo-iterativo-incremental/> |
| 5. Modelos de Prueba de Usabilidad | Traynor, V. (2015). *Pruebas de usabilidad con Verónica Traynor.* [Video]. YouTube. <https://youtu.be/uh4sZmQiRr4> | [Video]. YouTube. | <https://youtu.be/uh4sZmQiRr4> |
| 6. Modelos de Prueba de Usabilidad | Lorés, J. et al. (2006). *Evaluación.* <https://aipo.es/libro/pdf/04Evalua.pdf> | [Archivo PDF]. | <https://aipo.es/libro/pdf/04Evalua.pdf> |

1. **GLOSARIO:**

| TÉRMINO | SIGNIFICADO |
| --- | --- |
| Indagación | proceso para tratar de llegar al conocimiento de una cosa discurriendo 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). |

1. **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*](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/>

1. **CONTROL DEL DOCUMENTO**

|  | Nombre | Cargo | Dependencia  *(Para el SENA indicar Regional y Centro de Formación)* | Fecha |
| --- | --- | --- | --- | --- |
| Autor (es) | Rafael Neftalí Lizcano Reyes | Asesor Pedagógico Ecosistema RED | Regional Santander - Centro Industrial del Diseño y la Manufactura | Julio 2021 |
| Olga Lucía Mogollón Carvajal | Experto Temático | Centro para la Industria de la Comunicación Gráfica -CENIGRAF- | Septiembre 2021 |
| Luz Aida Quintero Velásquez | Diseñadora instruccional | Regional Distrito Capital - Centro de Gestión Industrial | Septiembre 2021 |
| Carolina Coca Salazar | Revisora Metodológica y Pedagógica | Regional Distrito Capital- Centro de Diseño y Metrología | Octubre 2021 |
|  | José Gabriel Ortiz Abella | Corrector de estilo | Regional Distrito Capital - Centro para la Industria de la Comunicación Gráfica. | Octubre del 2021. |

1. **CONTROL DE CAMBIOS**

**(Diligenciar únicamente si realiza ajustes a la Unidad Temática)**

|  | Nombre | Cargo | Dependencia | Fecha | Razón del Cambio |
| --- | --- | --- | --- | --- | --- |
| Autor (es) |  |  |  |  |  |