



Universidad Pontificia Universidad Javeriana

Facultad de ingeniería

SRS

Proyecto de grado

Camilo José Narváez

Daniel Eduardo Castellanos

David Andres Duarte

Ingeniería de sistemas

Bogotá D.C

Mayo 2023

Tabla de Contenido

Tabla de Contenido	2
Lista de tablas	4
1 Descripción global.....	5
2 Descripción global.....	6
2.1 Modelo de dominio	6
2.2 Perspectiva de producto.....	8
2.2.1 Interfaces con el sistema.....	10
2.2.2 Operaciones.....	10
2.3 Funciones del producto	12
2.4 Características del usuario.....	12
2.5 Requisitos No Funcionales.....	14
2.6 Suposiciones y dependencias	14
3 Requisitos Específicos.....	16
3.1 Requisitos Funcionales.....	16
3.2 Requisitos de desempeño	17
3.3 Atributos del sistema software	18
4 Proceso de validación y verificación.....	20
4.1 Validación de requisitos	20
4.2 Verificación de requisitos.....	21
4.3 Modificación de requisitos.....	21
5 Anexos.....	21
6 Referencias.....	22

Lista de figuras

Ilustración 1 - Modelo dominio	6
Ilustración 2: Casos de uso	12
Ilustración 3 - BPMN validación, modificación y verificación.....	21

Lista de tablas

Tabla 1 - Componente juego	6
Tabla 2 - Componente jugador	7
Tabla 3 - Componente Personaje	7
Tabla 4 - Componente Ambiente.....	7
Tabla 5 - Componente evento.....	8
Tabla 6 - Componente NPC.....	8
Tabla 7 - Componente Inventario	8
Tabla 8 - Componente Objeto	8
Tabla 9 - Características del jugador	13
Tabla 10 - Características del desarrollador	13
Tabla 11 - Requisitos no funcionales	14
Tabla 12 - Requisito acceder inventario	16
Tabla 13 - Requisito guardar partida	16
Tabla 14 - Requisito acceder a un evento	16
Tabla 15 - Requisito interactuar con objetos	16
Tabla 16 - Requisito interactuar NPC.....	17
Tabla 17 - Requisito moverse en el mapa.....	17
Tabla 18 - Requisito gestionar inventario.....	17
Tabla 19 - Atributos de calidad	19

1 Descripción global

El SRS es un documento en donde se va a proporcionar una descripción completa del comportamiento de cada uno de los componentes del sistema que se va a desarrollar. Este documento incluye un conjunto de casos de uso que describe cada una de las interacciones que tendrán los jugadores con el videojuego y estos casos de uso vienen a ser los requisitos funcionales [1]. En cuanto a la calidad del producto se van a definir cada uno de los requisitos no funcionales, para así tener claridad sobre el producto final definidos por nosotros y los stakeholders, y así tener una mayor claridad de estos.

Además, se presentarán diferentes diagramas en los cuales se definen cada uno de los componentes del modelo del sistema y también se busca definir todo el sistema bien estructurado.

2 Descripción global

2.1 Modelo de dominio

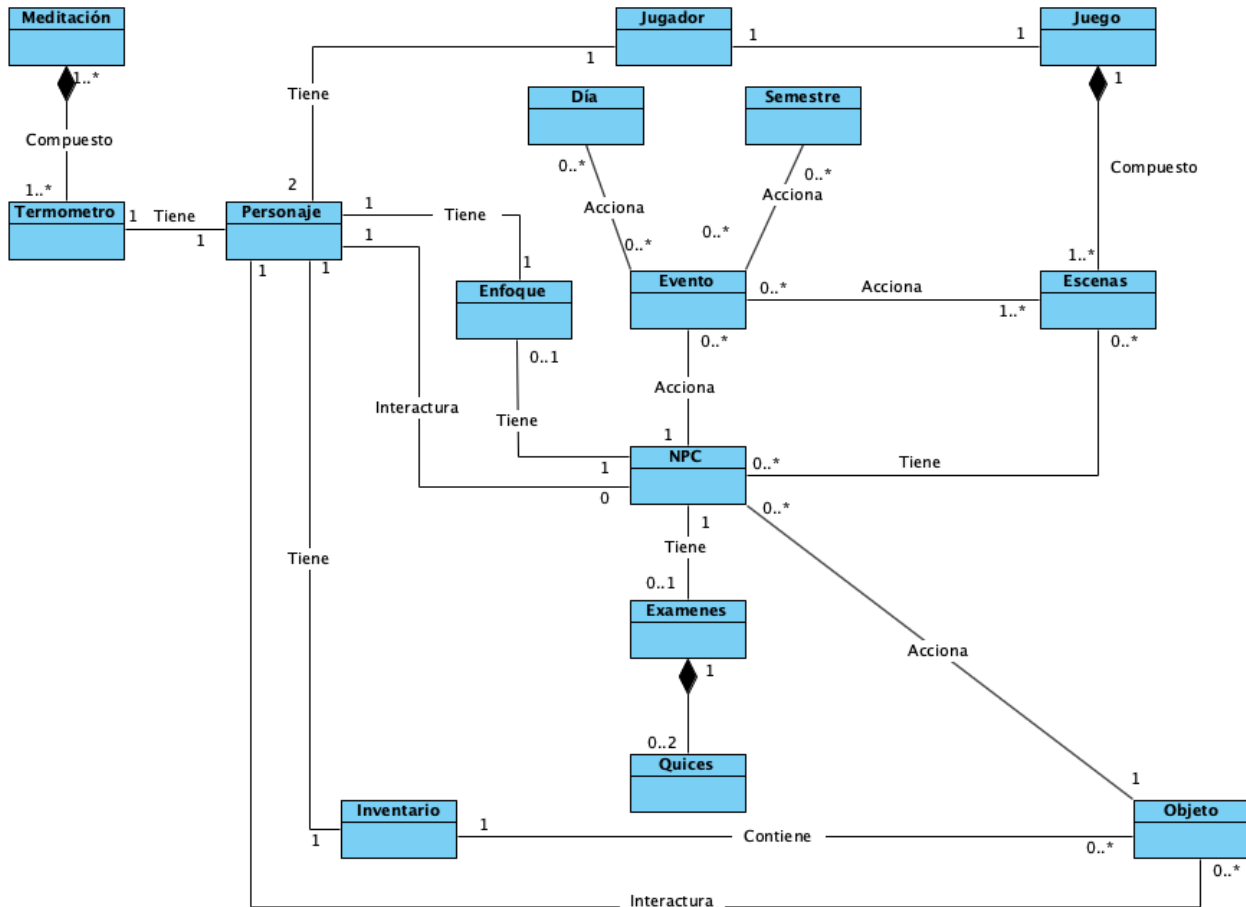


Ilustración 1 - Modelo dominio

ID	1	Componente del Dominio	Juego
Descripción	Representa el componente inicial del videojuego, el cual va a estar compuesto por diferentes ambientes y es el encargado de gestionar la partida del jugador.		
Elementos Asociados	Jugador y Ambiente.		

Tabla 1 - Componente juego

ID	2	Componente del Dominio	Jugador
Descripción	Representa al usuario como jugador del videojuego, el cual controla al personaje y cada una de las acciones que se toman en el desarrollo del juego, por medio de elementos de entrada como el teclado.		
Elementos Asociados	Juego y Personaje.		

Tabla 2 - Componente jugador

ID	3	Componente del Dominio	Personaje
Descripción	Representa a cada uno de los dos personajes del juego, los cuales interactuaran con distintos objetos o NPCs, además de poseer un inventario para guardar los objetos que este encuentre.		
Elementos Asociados	Inventario, Objeto, NPC y Jugador.		

Tabla 3 - Componente Personaje

ID	4	Componente del Dominio	Ambiente
Descripción	Representa el espacio en donde se podrán encontrar objetos y NPCs con los cuales se podrá interactuar. Además de que en los ambientes van a estar distintos eventos para el desarrollo de la historia.		
Elementos Asociados	Juego, Evento, Objeto y NPC.		

Tabla 4 - Componente Ambiente

ID	5	Componente del Dominio	Evento
Descripción	Representa los sucesos o eventos en los cuales se va a desarrollar el personaje a lo largo del juego. Estos eventos pueden ser activador por NPCs, el ambiente en el que personaje se encuentre o un objeto		

	del mismo ambiente.
Elementos Asociados	Ambiente, NPC.

Tabla 5 - Componente evento

ID	6	Componente del Dominio	NPC
Descripción	Representa a los personajes que forman parte de la historia del juego, pero no pueden ser utilizados por el jugador, por lo cual están en diferentes ambientes para dar indicaciones o como complementos.		
Elementos Asociados	Personaje, Evento y Ambiente.		

Tabla 6 - Componente NPC

ID	7	Componente del Dominio	Inventario
Descripción	Representa la opción de guardar cada uno de los objetos que el personaje encuentre a lo largo de la partida.		
Elementos Asociados	Personaje, Inventario y Ambiente.		

Tabla 7 - Componente Inventario

ID	8	Componente del Dominio	Objeto
Descripción	Representa a los objetos que se encuentre el personaje en cada uno de los ambientes, estos también pueden ser guardados en un inventario y pueden generar cualquier tipo eventos.		
Elementos Asociados	Personaje, Evento y Ambiente.		

Tabla 8 - Componente Objeto

2.2 Perspectiva de producto

Breaking limits nace como una propuesta para acompañar, y mostrar a los estudiantes de ingeniería de sistemas las diferentes técnicas usadas para el manejo de las emociones causadas por la frustración, en este caso frustración derivada de problemas con el entendimiento de las clases, con errores de código que complican su elaboración y comprensión y comúnmente inconvenientes en el diseño de proyectos, entre otras cosas.

Breaking limits es un videojuego Standalone que entraría a competir en el mercado de los juegos de aventura RPG, sin embargo, este está orientado a los estudiantes de ingeniería de sistemas que a causa de los problemas que se presentan a lo largo de la carrera sufren de frustración y de las emociones que derivan de esta. El producto no pretende eliminar la frustración en los estudiantes de ingeniería de sistemas, pues no está dentro de los márgenes de la investigación, sin embargo, busca presentarle estas herramientas para el manejo de esta, siendo un videojuego una experiencia artística existe la posibilidad de que los jugadores que la experimenten se puedan sentir relacionados con los problemas presentados en el mismo.

Este videojuego presenta el paso por diferentes etapas de la carrera de ingeniería de sistemas en la Pontificia Universidad Javeriana, y los posibles problemas que se van a encontrando los estudiantes a los largos de esta.

2.2.1 Interfaces con el sistema

El sistema en sí estará conectado con una plataforma donde se podrá descargar el producto teniendo en cuentas dos estados del desarrollo. En caso en el que el producto aun se encuentre en beta, producto será publicado a través de la plataforma de *Google Drive* a los usuarios se les entregara un enlace de descarga, donde estará disponible para descarga sin ningún tipo de sincronización con la versión que tenga el usuario, el manejo de datos entonces se hará de manera local en el espacio de almacenamiento proveído por la máquina del usuario. Al finalizar la experiencia de usuario se le pedirá al usuario rellenar un formulario que estará alojado en *Google Forms* donde se recogen los datos del impacto del videojuego. En el caso de que el producto se encuentre en su versión final se plantea que sea publicado en la plataforma de videojuegos *Steam* donde estará disponible para su descarga y el formulario post experiencia de juego se presentara dentro del mismo.

2.2.2 Operaciones

Periodos de inactividad:

El producto estará disponible para su descarga siempre y cuando los usuarios tengan conexión a internet y el servidor de descarga esté activo ya sea Google drive o Steam. Google dirve tiene alrededor de 800,000,000 de usuarios (Google, 2022) y se presentan alrededor de 50 reportes de errores en la operación de la plataforma. (downdetector.com, 2022). Steam siendo el candidato más viable para publicar el producto una vez esté finalizado cuenta con alrededor de 30,000,000 de usuarios conectados durante sus horas pico. (Steam, 2022) y diariamente se presentan alrededor de 3000 reportes de errores en la operación de la plataforma. (downdetector.com, 2022) por lo que se podría inferir que tiene una tasa de fallos del 0.0001% en su operación diaria. Dichas estas condiciones, se podría asumir que el servicio permanente está en línea.

Procesos de recuperación:

La partida de los jugadores se guardará cada vez que el usuario lo desee y se realizara desde el menú de juego. Si el juego está en su versión beta se guardarán los archivos de forma local por lo cual no se tendrá respaldo en caso de corromperse los archivos o de que sean eliminados. En caso de que el producto se encuentre en su versión final se espera contar con la opción de gradado en la nube que proporciona Steam, por lo cual mientras el usuario tenga conexión a internet podrá mantener sus datos de guardado sincronizados.

Funciones de soporte a procesamiento de datos:

El sistema no cuenta con una interfaz de pagos, esta estará tercerizada en la plataforma donde esté el producto disponible, sea Patreon, Kickstarter o BuyMeACoffe.

2.3 Funciones del producto

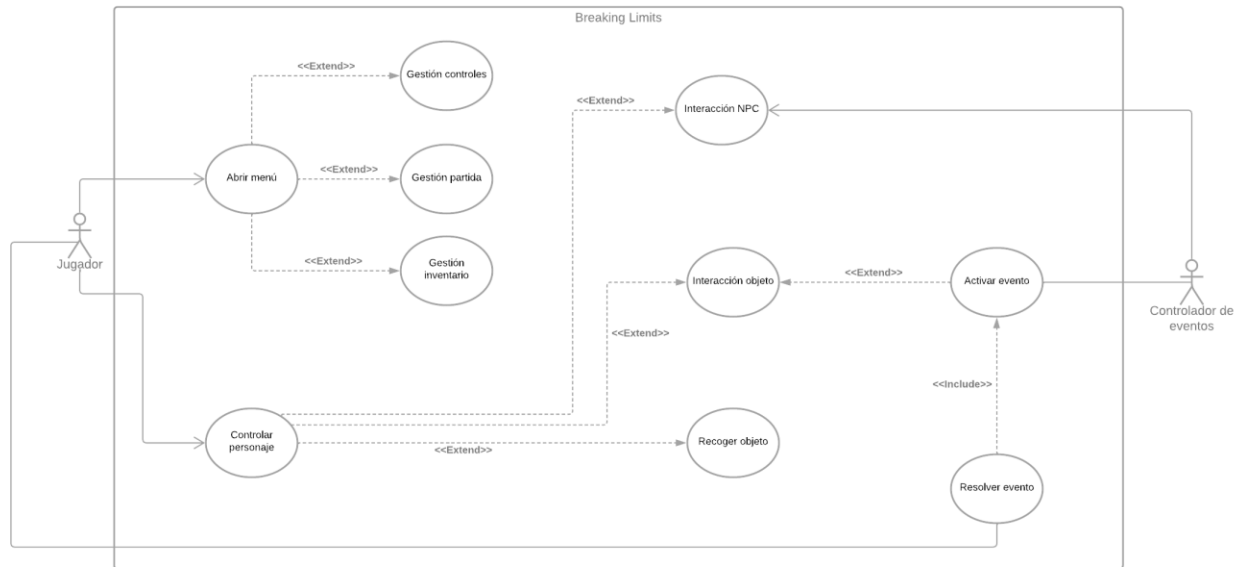


Ilustración 2: Casos de uso

2.4 Características del usuario

El sistema tendrá dos tipos de usuarios, estarán los jugadores, los cuales interactúan con la versión del producto para obtener la experiencia que este ofrece y los administradores, quienes se encargaran de mantener la versión actualizada del juego en el servidor, ya sea en versión beta en Google Drive o en su versión final en Steam.

Características del usuario	Descripción
Nivel de seguridad o de privilegios	Estándar, privilegios de acceso a contenido acorde al avance en la historia del juego. Únicamente tiene acceso a la versión compilada del producto, sin capacidad de ver el diseño o los archivos del sistema.
Rol	Jugador
Nivel de estudio o experiencia técnica	No requiere experiencia del sistema, a medida que progrese en el juego irá obteniendo información y desbloquearán

	mecánicas de juego junto a coleccionables a lo largo de la experiencia de juego
Frecuencia de uso	La frecuencia de uso está en manos del usuario, este hará uso del sistema desde el momento en que ejecute la aplicación en su computadora hasta que lo cierre. Podrá acceder a esta funcionalidad siempre y cuando tenga instalado los archivos del juego en su computador.

Tabla 9 - Características del jugador

Características del usuario	Descripción
Nivel de seguridad o de privilegios	Tiene todos los privilegios del sistema, puede acceder a los modelos y el sistema en el modo de desarrollo. Puede acceder a cualquier archivo o asset del sistema para modificarlos. El Desarrollador está en capacidad de subir nuevas versiones del sistema a la plataforma donde estará publicado siempre que éstas estén aprobadas por los líderes del proyecto.
Rol	Desarrollador
Nivel de estudio experiencia técnica	Requiere conocimiento en la herramienta <i>Unity</i> , así como estudio del diseño del sistema para saber cómo funcionan sus componentes.
Frecuencia de uso	El desarrollador usará el sistema cuando esté desarrollando elementos para la actualización el producto, agregar nuevas funcionalidades nuevas o arreglar errores. Los repositorios para que este desarrollador opere en el sistema estarán disponibles siempre y cuando la plataforma de <i>GitHub</i> este en línea.

Tabla 10 - Características del desarrollador

2.5 Requisitos No Funcionales

Descripción del requisito	Clasificación del requisito
El juego está ambientado en el campus de la universidad Javeriana sede en Bogotá y con acceso a algunos edificios.	Usabilidad
El usuario puede escoger entre un personaje masculino o femenino	Usabilidad
Los textos son legibles para el jugador	Usabilidad
El juego maneja términos asociados con la frustración asociada a la psicología en estudiantes universitarios	Usabilidad
Las narrativas tendrán texto claro y entendible.	Usabilidad
Las mecánicas asociadas al inventario y el manejo del personaje serán intuitivas.	Usabilidad
El sistema no tardará más de 20 segundos en almacenar el progreso	Desempeño
El sistema maneja transiciones fluidas entre ambientes	Desempeño
El juego correrá al menos en 30 frames por segundo	Desempeño
El sistema estará desarrollado para Windows utilizando Unity	Mantenibilidad
Los assets serán solo en formato mp3, obj, png y blend.	Mantenibilidad

Tabla 11 - Requisitos no funcionales

2.6 Suposiciones y dependencias

Lista de supuestos:

- La documentación es suficiente para el desarrollo del proyecto.
- Todos los integrantes tienen acceso a infraestructura para el desarrollo del proyecto.
- Los usuarios cuentan con la infraestructura para correr el proyecto.

Lista de dependencias:

- La licencia de las herramientas a usar es gratuita.
- La documentación de las herramientas estará disponible para la consulta siempre que se necesite.
- Las licencias no cambiarán a forma de pago durante el desarrollo del proyecto.

3 Requisitos Específicos

3.1 Requisitos Funcionales

# Requisito	1	Caso de uso asociado	Abrir Menú
Descripción	El usuario debe poder acceder al inventario mientras esté en el mapa		
Razón	El usuario debe poder interactuar con los coleccionables o los elementos del mapa que generan eventos.		
Criterio de medición	El usuario puede abrir y cerrar el inventario.		
Prioridad	1	Modulo asociado	

Tabla 12 - Requisito acceder inventario

# Requisito	2	Caso de uso asociado	Gestionar Partida
Descripción	El usuario debe poder guardar y almacenar el progreso del juego.		
Razón	En el juego, el usuario debe poder guardar en cualquier momento su progreso para poder continuar luego.		
Criterio de medición	El usuario puede accionar el botón de guardar y cargar partida.		
Prioridad	1	Modulo asociado	Persistencia

Tabla 13 - Requisito guardar partida

# Requisito	3	Caso de uso asociado	Resolver Evento
Descripción	El usuario debe poder acceder a un evento.		
Razón	Los eventos son el centro del juego, el usuario debe poder hacer uso de los eventos y de los objetos que componen los ambientes.		
Criterio de medición	El usuario puede activar un evento.		
Prioridad	1	Modulo asociado	

Tabla 14 - Requisito acceder a un evento

# Requisito	4	Caso de uso asociado	Resolver Evento
Descripción	El usuario debe poder interactuar con los objetos que existen en los eventos		
Razón	El usuario podrá guardar e interactuar con objetos dentro de los eventos.		
Criterio de medición	El usuario guarda objetos en el inventario y puede activar acciones asociadas a objetos.		
Prioridad	1	Modulo asociado	

Tabla 15 - Requisito interactuar con objetos

# Requisito	5	Caso de uso asociado	Interactuar NPC
Descripción	El usuario debe poder interactuar con los NPCs, y acceder a los eventos que estos generen		
Razón	Los NPC son entidades importantes del desarrollo, ya que son generadores de eventos para el jugador		
Criterio de medición	El usuario puede hablar y acceder a instancias de eventos generados por los NPC		
Prioridad	1	Modulo asociado	

Tabla 16 - Requisito interactuar NPC

# Requisito	6	Caso de uso asociado	Controlar Personaje
Descripción	El usuario podrá moverse por el mapa y acceder a edificios en el campus		
Razón	Es necesario que exista movimiento del personaje para hacer uso de las situaciones que el juego propone para el jugador.		
Criterio de medición	El usuario puede moverse a través del mapa y acceder a edificios.		
Prioridad	1	Modulo asociado	

Tabla 17 - Requisito moverse en el mapa

# Requisito	7	Caso de uso asociado	Gestión de Inventario
Descripción	El usuario podrá gestionar su inventario, añadiendo y modificando artículos.		
Razón	El jugador necesita acceder, añadir y modificar objetos de su inventario.		
Criterio de medición	El jugador puede almacenar y modificar objetos del inventario.		
Prioridad	2	Modulo asociado	

Tabla 18 - Requisito gestionar inventario

3.2 Requisitos de desempeño

La ejecución del producto únicamente soporta las entradas para un usuario. Sin embargo, el sistema debe concurrentemente recibir múltiples entradas y superponer estas entradas en jerarquías para controlar las animaciones que se presentan dependiendo de la situación.

Se espera que el producto corra a 30 FPS en su versión final, el tiempo de renderizado de imagen sería de unos 30 ms, si además tenemos en cuenta el tiempo de respuesta de un teclado estándar que esta entre los 30 a 60ms, el tiempo final de respuesta esperado para que cada entrada de teclado se vea reflejada en pantalla es de aproximadamente 60 a 90 ms.

Analizando los controles de desplazamiento y las posibilidades de acciones extra que puede tener el usuario, se espera que en el peor de los casos haya un total de 3 entradas simultaneas, dos entradas para movimiento, en caso de que el jugador desee moverse de manera diagonal, más una entrada en caso de que el jugador este tratando de interactuar con un objeto.

3.3 Atributos del sistema software

Nuestro proyecto al desarrollar un videojuego prima la seguridad y comodidad del usuario, para que este pueda jugar sin ningún tipo de inconveniente interno, o sea sin ningún malware que pueda llegar a dañar o vulnerar el computador donde se descargue el videojuego. También se busca que el jugador en el transcurso del juego se dé por enterado de las técnicas de superación en cuanto a cada uno de los sentimientos que genera la frustración y que pueda pasar un rato entretenido con cada uno de los eventos que se van a llegar a cabo. En conclusión, el trabajo en equipo y el de investigación van a ser primordiales para la búsqueda de factores que puedan llegar a afectar al jugador mientras este juegue.

Los atributos de calidad los cuales van a ser parte fundamental del desarrollo están definidos en la siguiente tabla:

Seguridad	Breaking Limits está libre de todo tipo de virus, malware que pueda afectar su rendimiento y el del computador del usuario. Además, al exportar el videojuego de Unity, se busca que el jugador no tenga acceso al código fuente del videojuego.
Portabilidad	Breaking Limits solo estará soportado para correr en el sistema operativo de Windows, ya que la mayoría de nuestro público utilizan este sistema. Además, de que el código estará desarrollado en Windows.
Mantenibilidad	Breaking Limits estará basado de manera que nos permita actualizar o modificar cada uno de los componentes por separado y así poder generar una menor dependencia entres los componentes. Por otro lado, la calidad de cada uno de los cambios que se realicen se espera que sigan las pautas y cada una de las normativas, para así facilitar el desarrollo de cada una de las actividades y las revisiones des estas mismas, estos cambios deben estar documentados a detalle.
Disponibilidad	Breaking Limits estará disponible, siempre y cuando el usuario tenga todos los archivos necesarios para el correcto funcionamiento del videojuego. Además, para el

	funcionamiento de este, no será necesario cualquier tipo de conexión a internet, ya que no aportará, ni afectará a la jugabilidad de este.
Confiabilidad	Como se ha mencionado nuestro videojuego no afectara al del jugador del dispositivo en el que se va a descargar y utilizar en circunstancias de juego. Además, se busca que en el momento de suceder algún tipo de error o bug en el videojuego se intentara recuperar o como caso extremo se cerrara el videojuego y tocara volverlo a iniciarlo en el último guardado de partida.

Tabla 1919 - Atributos de calidad

4 Proceso de validación y verificación

4.1 Validación de requisitos

Si tenemos en cuenta las necesidades de los jugadores, los desarrolladores y la lista de requerimientos que anteriormente se presentaron, se decidió utilizar una serie de criterios para determinar si el requerimiento es válido o no. Los criterios decididos para la validación de requerimientos fueron los siguientes:

1. **Atómico:** Debe ser único, no debe poderse dividir en más requerimientos.
2. **Completo:** No se deben encontrar ni interrupciones en su construcción ni falta de procesos adicionales.
3. **Consistente:** No debe contradecirse y tampoco hacer el trabajo de otro requerimiento.
4. **Acertado:** Debe surgir a partir de una necesidad del usuario.
5. **No ambiguo:** No debe interpretarse de más de una manera, es decir, evita subjetividades
6. **No redundante:** Debe evitar palabras innecesarias o que aclaren la misma idea más de una sola vez.
7. **Importancia:** Debe solucionar un problema y ser esencial en el proyecto.
8. **Verificable:** Que exista algún modo de saber si está cumpliendo con las necesidades esperadas por el usuario y el grupo de trabajo.
9. **Modificable:** Que algún cambio hecho en un requerimiento no signifique la modificación completa del requerimiento o del sistema.
10. **Trazable:** Debe ser posible de documentar y llevar un historial del proceso hecho en cualquier momento del tiempo.
11. **Preciso:** No deben ser extensas, solamente debe estar escrito lo fundamental.

4.2 Verificación de requisitos

Una vez el requisito es validado pasa al proceso de verificación donde se evalúa si el requisito es alcanzable o viable, en caso de que la evaluación sea positiva se acepta el requisito, en caso de que no, deberá ser modificado, luego validado y por último verificado nuevamente.

4.3 Modificación de requisitos

En caso de que uno de los requisitos previamente validados deba ser modificado, se evaluará la necesidad de dicho requisito y será evaluado nuevamente por el proceso de validación.

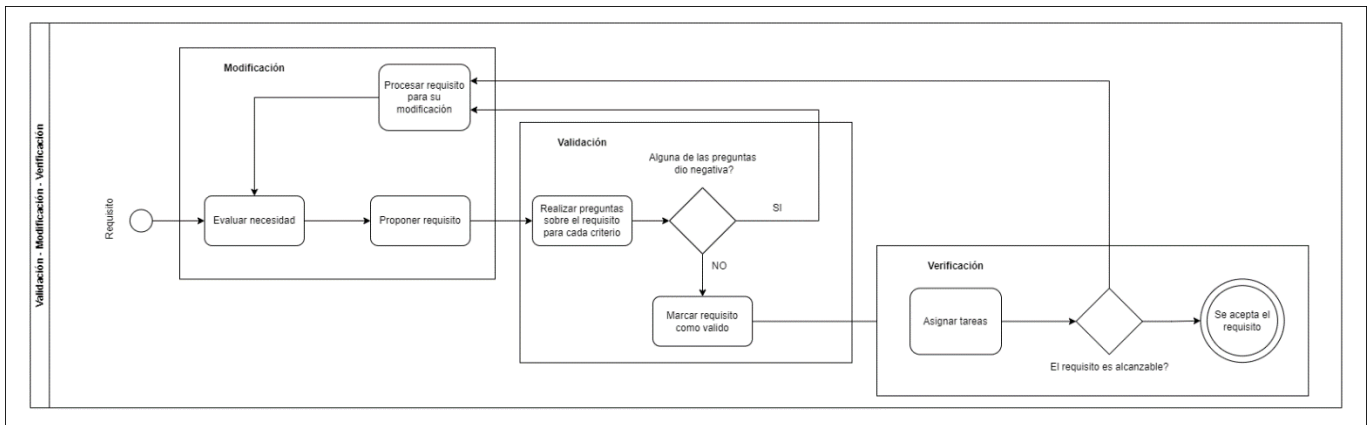


Ilustración 3 - BPMN validación, modificación y verificación

5 Anexos

1. Diagrama del modelo de dominio
2. Diagrama de casos de uso
3. Documentación de los casos de uso

6 Referencias

[1] Características de un buen SRS (Software Requirements Specification) especificación de requerimientos de software. (2022, mayo 2). Tech-Quality-simplicity.

<https://deapharma.com/caracteristicas-de-un-buen-srs-software-requirements-specification-especificacion-de-requerimientos-de-software/>

[2] Usuarios de Google Drive. Recuperado 14 de noviembre de 2022, de

<https://es.statista.com/grafico/9471/google-en-cifras/>

[3] Usuarios diarios de Steam. Recuperado 14 de noviembre de 2022, de

<https://store.steampowered.com/charts/?l=spanish>

[4] Disponibilidad de servicios downdetector. Recuperado 14 de noviembre de 2022, de

<https://downdetector.com/>.

[5] Ingeniería del software. (s. f.). Google Books. Recuperado 15 de noviembre de 2022, de

https://books.google.com/books/about/Ingenier%C3%ADa_del_software.html?id=gQWd49zSut4C