MetProg URJC GAME

Análisis Funcional

Empresa:

- MetProg URJC S.L.

Autores Proyecto:

- **Analista Funcional:** Eloy de Sande de las Heras
- **Analista Programador:** Olga Somalo Serrano
- **Ingeniero de Desarrollo:** Jorge Padilla Rodríguez
- **QA:** Alejandro Martín Carrera
- **Jefe de Proyecto:** José Víctor García Llorente

ÍNDICE

Objetivo

El propósito principal de este documento es recoger la mayor cantidad de información posible relativa a los requisitos especificados por el cliente sobre el diseño de la aplicación, con el fin de optimizar el diseño y la implementación de la misma.

El objetivo principal del proyecto es la creación de una aplicación de ocio en el cual se enfrenten diferentes personajes, creados por los usuarios. Para lograr una explicación más concisa, se explicarán los detalles más generales y se llegará hasta lo más específico de acuerdo a la siguiente estructura:

- Registro/Borrado de usuarios.
- Elementos del juego.
- Persistencia.

Nicho de mercado

La aplicación (o juego) no está enfocada a un sector poblacional concreto, si bien se requiere o busca que el usuario tenga nociones básicas sobre la ficción gótica, es decir, que al usuario no le sean ajenos determinados conceptos sobre los que sustenta la temática del juego.

Fuente de datos

El programa contará con dos fuentes de datos: la introducida por el administrador y la proporcionada por los usuarios en el desarrollo del juego.

El administrador se encargará de subir a la base de datos aquellos requerimientos técnicos del juego, como son los posibles personajes, características del juego y de la jugabilidad.

Los usuarios irán nutriendo al juego con los avances que realicen en el mismo; tras cada batalla se irá actualizando el ranking, el login de nuevos usuarios y el registro de batallas.

Información básica

Análisis del problema

La empresa MetProg URJC S.L. desea realizar una aplicación en la cual distintos personajes se enfrenten combatiendo entre sí. Cada personaje tendrá una serie de información y características, todas ellas podrán ser modificadas por los usuarios de la aplicación. Se requerirá incluir diferentes habilidades y debilidades (que se llaman modificadores) y también armas y armaduras (llamados equipos). Además los personajes se enfrentarán entre ellos en combates, precedidos por un desafío (el usuario desafía a otro usuario, el personaje combate con otro personaje). Se incluirá un sistema de persistencia para el juego, de forma que toda la información de un usuario se guarde al terminar la ejecución.

Se requiere que la aplicación sea sencilla y fácil de utilizar, ya que el usuario o cliente de la aplicación tiene que usarla de forma casi intuitiva, puesto que es una aplicación de entretenimiento. La aplicación podrá tener varios usuarios. Cada uno con un número de registro distinto. Se deberá introducir la información del usuario (nombre, contraseña…) por teclado.

Preguntas para el cliente

Al realizar el análisis funcional del texto base, nos han surgido dudas o dilemas acerca de determinados conceptos de la práctica. A continuación detallaremos estas cuestiones:

- ¿Cuántos personajes tiene un usuario? El texto proporcionado vislumbra que cada usuario tendrá un único personaje, lo cual confronta con la dicotomía usuario-personaje, pues si el personaje fuese único no haría falta separarlo del usuario. Finalmente optamos por la elección de que un usuario tiene un único personaje, pues para esta fase del proyecto es más fácil conceptualmente y en cualquier momento se podría modificar sin mucho esfuerzo.

- ¿Cuál es la diferencia entre combate y ronda? A la hora de entender el problema dado, no terminamos de entender si un desafío se compone a su vez de combates y estos de rondas… o si por el contrario el término ronda hace referencia sinonímica a combate. Esta última interpretación es la que optamos por elegir, pues es la que nos parece más adecuada.

- ¿El personaje viene inicialmente equipado por defecto? En los requisitos especificados por el cliente, se mencionan pocos detalles sobre cómo debe ser la creación de un nuevo usuario, y por lo tanto, las características que por defecto debe tener un personaje. Lo más lógico sería que un nuevo personaje no tenga equipación alguna, aunque sí pueda ser desafiado. Esta será la opción que llevaremos a cabo.

- ¿Cuál es la finalidad del oro? La intuición nos dice que el oro de un personaje nos sirve para desbloquear nuevas habilidades, armas y esbirros. Pero no se explica en ningún momento para qué sirve. Por lo tanto, a no ser que se diga lo contrario, utilizaremos el oro como recurso para desbloquear nuevas características de un personaje. Siguiendo esta lógica, creemos conveniente la creación de una "tienda" que permita a los usuarios poder acceder a la compra de armas, armaduras, esbirros y habilidades especiales con las que poder mejorar sus personajes.

- ¿Qué ocurre cuándo un personaje se queda a 0 de oro tras un desafío? Al pensar la especificación del problema, nos ha surgido la cuestión de qué ocurre cuándo un personaje se queda a 0 de oro, pues no podría realizar ningún nuevo desafío. Este problema se podría solventar dándole a cada participante tras finalizar una cantidad prefijada de oro, para que pudiese realizar un nuevo desafío.

- ¿Cuáles son las habilidades especiales?¿Cuántas hay? El texto proporcionado indica que existen habilidades especiales para los personajes, pero no se detalla cuáles son. En esta fase del proyecto no abordaremos cuáles son; lo dejaremos para la fase de codificación.

- ¿Qué ocurre si el usuario desafiado no dispone de saldo suficiente en caso de rechazar un desafío? El texto proporcionado especifica que la declinación de un desafió conlleva la pérdida de oro del 10% de la cantidad apostada por el desafiante, pero puede darse la circunstancia de que el usuario desafiado no disponga de dicha cantidad (e.g. El atacante apuesta 100 de oro pero el desafiado solo dispone de 8). Ante esta circunstancia, sobrentendemos que se debe verificar antes de validar el ataque que el desafiado dispone de dicha cantidad.

- ¿Para qué sirve, en el apartado de esbirros, los niveles de lealtad de los humanos, así como la dependencia de los ghouls? Según leemos en el texto, la funcionalidad de los esbirros es restar de sus puntos de salud el daño que sufra el personaje, de aquí que surge la duda de qué aportan en esto los niveles de lealtad y dependencia.

Conceptos básicos

A continuación pasaremos a desarrollar de forma sucinta los conceptos básicos que nos ayudarán a entender la mecánica del juego; es decir, los elementos que interactuarán en las diferentes fases del juego y que nos serán de gran utilidad para la correcta interpretación de las relaciones del programa.

Actor

Se denominará “Actor” a cualquier entidad externa que de una forma u otra haga interactúe con el programa. Por ejemplo, los clientes y los operadores (usuarios) del juego son actores.

Entidad

Una “Entidad” será un elemento básico e indispensable para que sea posible utilizar el juego. Así, se puede pensar que las entidades del juego son los personajes, los usuarios, las habilidades, los equipos etc.

Elemento

Llamaremos “Elemento del juego” a cualquier entidad interna que forme parte del programa y sea necesaria para su correcto funcionamiento. Por ejemplo, los personajes del juego son elementos.

Usuario

Serán los distintos “clientes” de la aplicación. Cada usuario tendrá un personaje, el cual podrá customizar con el arsenal de objetos del que disponga; este arsenal se podrá ampliar de acuerdo al desarrollo del juego, pues con cada combate ganado obtendrá recompensas (oro) que podrá canjear por “Equipos” para su personaje. Los usuarios son un caso especial dentro de la aplicación, ya que son tanto actores como entidades (ya que cada usuario representa un cliente o un operador de la aplicación, pero a su vez es indispensable para el funcionamiento correcto del juego).

Personaje

Es el “alter égo” ficticio con el que cada usuario podrá jugar. Son personales e intransferibles. Cada usuario tendrá un único personaje propio. Atenderán a distintos roles: vampiro, licántropo o cazadores. Además, cada personaje contará con diferentes atuendos, que podrá alternar a conveniencia para cada “Desafío”. Así mismo, podrá ir acompañado de singulares mascotas, “Esbirros”, que le proporcionarán mejores características o habilidades.

Esbirros

Serán las “mascotas” de los personajes, a los que proporcionarán puntos extra de salud. Existen tres tipos de esbirros: los humanos, los ghouls y los demonios, cada uno de ellos con características especiales.

Habilidades especiales

Servirán para mejorar las características de los “personajes”. Cada personaje tendrá una habilidad activa, que le servirá para potenciar (o no) sus ataques y defensas. Solo se podrá usar cuando el personaje haya superado un determinado umbral de activación, el cual se irá recuperando tras cada “combate”.

Las habilidades de los vampiros se llamarán disciplinas, las de los licántropos dones y las de los cazadores talentos (actualmente sin ninguna función definida).

Disclaimer: similar a los ataques cargados de Pokemon Go.

Equipo

Serán los atuendos y equipaciones de cada personaje. Se dividirán en armas y armaduras. Aportarán un valor de ataque o de defensa al personaje.

Modificadores

Son circunstancias inherentes que aportarán una ventaja o inconveniente a cada personaje. Es decir, cuando se dé determinada circunstancia, nuestro personaje será más vulnerable o más feroz contra el atacante.

Aplicación

Será el programa a codificar en Java. Deberá poder cumplir con todos los requerimientos dados por el cliente.

Desafíos

Los desafíos son los retos en los que se enfrentarán los personajes de usuarios distintos. El usuario retado deberá validar si acepta o declina el desafío (le supondrá cierto coste). Los desafíos se componen de combates.

Los desafíos tienen gran cantidad de requisitos que tendrá que codificar el programador. Estos requisitos se detallarán más adelante.

Combate

En cada combate (o enfrentamiento) se irán alternando el ataque-defensa entre ambos personajes. Así pues, si en el primer combate le tocó atacar al personaje 1 y defenderse al personaje 2, en el siguiente combate se cambiarán los papeles. Esta alternancia se prolongará hasta que uno de los personajes muera (pierda).

Actores

En este apartado se describen los dos únicos actores que intervienen en los procesos que tienen lugar en el sistema. Sin tener en cuenta actores ajenos al funcionamiento más básico de la aplicación como la empresa encargada de la gestión de la base de datos o la empresa de gestión de ayuda on-line etc.

Actor	Cliente (Usuario)
Descripción	Cualquier persona física con la capacidad de crear un usuario en la aplicación.
Casos de uso	CU-01 CU-04 CU-06 CU-07 CU-08

Actor	Operador (Usuario)
Descripción	Cualquier usuario autorizado a realizar modificaciones críticas en el sistema.
Casos de uso	CU-01 CU-02 CU-03 CU-04 CU-05 CU-06 CU-07 CU-08

Casos de uso

En esta sección se explican y se enumeran los casos de uso principales para los actores de la aplicación.

Caso de uso	Registrar usuario
ID	CU-01
Descripción	Da de alta a un usuario. El sistema valida si es un usuario válido y qué tipo de usuario es.
Actores	Clientes y Operadores
Entrada	Un usuario (cliente u operador)
Salida	Validación correcta del sistema
Precondiciones	Nick del usuario no repetido
Poscondiciones	No hay

Caso de uso	Banear usuario
ID	CU-02
Descripción	Inhabilita el uso de la aplicación a un usuario.
Actores	Operadores
Entrada	Un usuario (cliente u operador)
Salida	Usuario deshabilitado
Precondiciones	Usuario previamente validado
Poscondiciones	No hay

Caso de uso	Desbanear usuario
ID	CU-03
Descripción	Habilita el uso de la aplicación a un usuario que previamente estaba baneado.
Actores	Operadores
Entrada	Un usuario (cliente u operador)
Salida	Usuario habilitado
Precondiciones	Usuario previamente validado y baneado
Poscondiciones	No hay

Caso de uso	Login
ID	CU-04
Descripción	Inicia sesión un usuario. Cargando toda la información relativa al mismo.
Actores	Clientes y Operadores
Entrada	Un usuario (cliente u operador)
Salida	Validación correcta de credenciales del usuario en el sistema y comprobación de usuario habilitado
Precondiciones	Usuario previamente validado
Poscondiciones	No hay

Caso de uso	Validación de combate
ID	CU-05
Descripción	Valida si el combate entre dos usuarios se puede realizar o no.
Actores	Operadores
Entrada	Dos usuarios habilitados (cliente u operador)
Salida	Validación correcta por parte del operador y envío de una notificación al usuario desafiado
Precondiciones	Usuarios previamente validados
Poscondiciones	No hay

Caso de uso	Modificar Personaje
ID	CU-06
Descripción	Modifica el personaje de un usuario. Modificar incluye editar cualquier característica del personaje y añadir armas, armaduras, fortalezas etc.
Actores	Clientes y Operadores
Entrada	Un usuario (cliente u operador)
Salida	Validación correcta de modificación válida y las correspondientes modificaciones del personaje en cuestión.
Precondiciones	Usuario previamente validado y habilitado
Poscondiciones	No hay

Caso de uso	Consulta de información global
ID	CU-07
Descripción	Da información a tiempo real del juego (por ejemplo el ránking global).
Actores	Clientes y Operadores
Entrada	Solicitud de vista de información
Salida	Validación correcta de visualización de información global y devolución de los datos solicitados.
Precondiciones	Usuario previamente validado y habilitado
Poscondiciones	No hay

Caso de uso	Consulta de información privada
ID	CU-08
Descripción	Consulta de información de un usuario en concreto. Solo puede ser utilizada por el propio usuario o por los operadores del juego.
Actores	Clientes y Operadores
Entrada	Un usuario cualquiera y otro que realiza la consulta (pueden ser clientes u operadores)
Salida	Datos solicitados si la validación es correcta.
Precondiciones	Usuarios previamente validados
Poscondiciones	No hay

Diagramas de casos de uso

Diagrama de caso de uso de registro de usuario

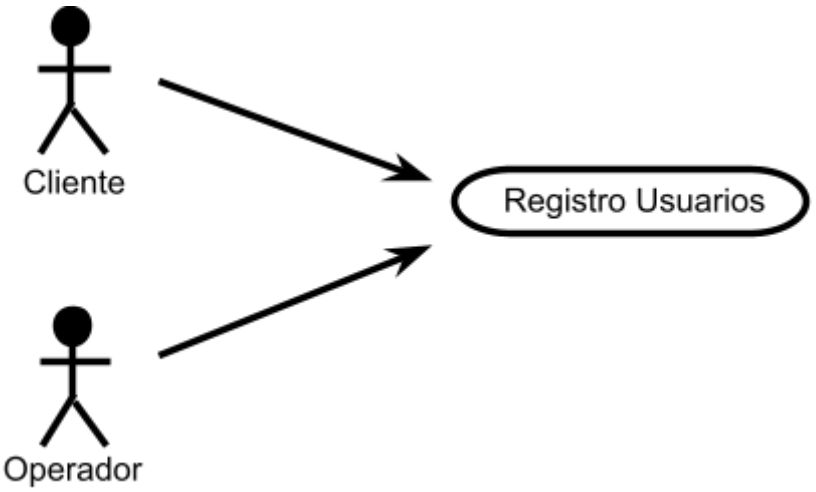
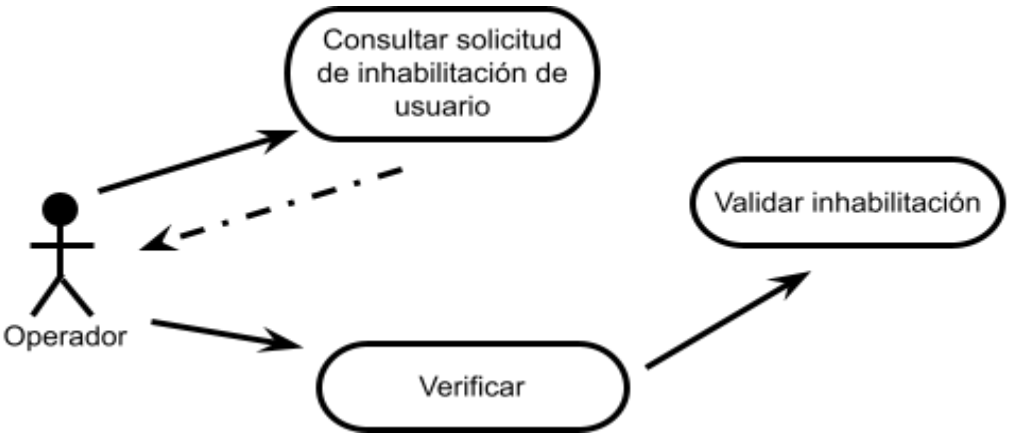


Diagrama de caso de uso CU-02 (análogo a CU-03)



Entidades

En este apartado realizaremos una descripción pormenorizada de cada uno de los agentes que participan en el juego.

Personaje

La entidad personaje tiene los siguientes atributos:

- Nombre: cadena de caracteres
- Habilidad especial: habilidad especial
- Conjunto de armas: cadena de caracteres
- Armas activas (1 o 2). Como máximo 2 armas de una mano o 1 de dos manos.
- Conjunto de armaduras: cadena de caracteres
- Armadura activa
- Conjunto de esbirros
- Cantidad de oro: entero
- Salud: entero entre 0 y 5
- Poder: entero entre 1 y 5
- Potencial de ataque: entero
- Debilidades
- Fortalezas

Los tipos de personaje que hay son:

- Vampiro. Los vampiros para poder activar su disciplina (habilidad especial), deben robar y acumular (hasta un máximo de 10) la sangre de sus víctimas (contrincantes). Cada disciplina tendrá un coste de entre 1 y 3 puntos de sangre, y proporcionará una ventaja para la defensa o ataque en cada combate. Las disciplinas específicas se desconocen por el momento.
- Licántropo. Los licántropos para poder activar su don (habilidad especial), necesitan un valor de rabia mínimo (no especificado), que tras usarse el don disminuye. Los puntos de rabia aumentan en 1 punto tras cada combate perdido, inicializándose el desafío con 0 puntos.
- Cazadores. Poseen puntos de voluntad, que son enteros entre 0 y 3. Inician el combate con 3 y van disminuyendo en 1 punto cada vez que pierde salud.

Habilidad especial

La entidad habilidad especial tiene como atributos:

- Nombre: cadena de caracteres
- Valor de ataque: entero entre 1 y 3
- Valor de defensa: entero entre 1 y 3

Los tipos de habilidades especiales son:

- Disciplinas. Propias de los Vampiros. Su uso gasta puntos de sangre en función del valor del ataque de su disciplina.
- Dones. Propias de los Licántropos. Requiere de un valor mínimo de rabia.
- Talentos. Propias de los Cazadores.

Equipo

Además, tenemos a la entidad equipo con los siguientes atributos:

- Nombre: cadena de caracteres
- Modificador al ataque: entero entre 1 y 3
- Modificador a la defensa: entero entre 1 y 3

Los tipos de equipo son:

- Armas. Las hay de 1 mano (2 posibles) y de 2 manos (1 posible).
- Armaduras.

Esbirro

También está la entidad esbirro, cuyas propiedades son:

- Nombre: cadena de caracteres
- Salud: entero entre 1 y 3

Los tipos de esbirro son:

- Humanos. Poseen un valor de lealtad, medida en ALTA, NORMAL o BAJA. Esto hará que la eficacia de la ayuda al personaje sea mayor (lealtad ALTA) o menor (lealtad BAJA). Los humanos no pueden en ningún caso ser esbirros de vampiros.
- Ghouls. Mediante un entero entre 1 y 5 se mide la dependencia de los ghouls con sus amos, siendo 1 la más baja y 5 la más alta.
- Demonios. Los demonios se convierten en esbirros de sus amos a través de un Pacto, este Pacto se formaliza con una descripción. Los demonios se diferencian de los otros esbirros en que pueden tener a su vez otros esbirros de cualquier tipo (si sus esbirros fueran demonios, estos también podrían tener esbirros).

Los esbirros ayudan a los personajes restando sus puntos de salud antes que los del jugador. Esto será así hasta que se acaben los puntos de vida de todos los esbirros y, a partir de entonces, se empezará a restar la salud del personaje.

Usuario

La entidad usuario tiene los siguientes atributos:

- Nombre: cadena de caracteres
- Nick: cadena de caracteres
- Password: cadena de entre 8 y 12 caracteres

Tipos de usuario:

- Cliente. Se identifican por un número de registro, que es único, se genera de forma automática y es una cadena de caracteres de la forma LNNLL (siendo L una letra y N un número natural).
- Operador del sistema. Tendrán otras funciones, que se detallarán más adelante.

Desafío

La entidad desafío tiene como propiedades:

- Usuario desafiante: cadena de caracteres (número de registro)
- Usuario desafiado: cadena de caracteres (número de registro)
- Usuario vencedor: cadena de caracteres (número de registro)
- Oro ganado: entero
- Identificador del combate: cadena de caracteres

Combate

La entidad combate tiene como propiedades:

- Identificador del combate: cadena de caracteres
- Fecha: cadena de caracteres.
- Rondas empleadas: entero

Modificador

La entidad modificador tiene los siguientes atributos:

- Nombre: cadena de caracteres
- Valor: entero entre 1 y 5

Los tipos de modificadores son:

- Debilidades. Cuanto mayor es el valor de la debilidad, este número se restará al potencial de ataque en combate. Cada personaje tendrá unas debilidades que se verán presentes dependiendo del combate.
- Fortalezas. El valor de las fortalezas se sumarán al potencial de combate según estas estén presentes o no en el combate.

Requisitos

Requisitos funcionales

En este apartado se tratará el desarrollo del juego, es decir, todas las fases o procesos que irán aconteciendo en el programa.

Principalmente, el programa permitirá realizar a los usuarios las siguientes acciones:

Creación de usuario (y personaje)

El programa no deberá sufrir inconsistencias, evitando la duplicidad de usuarios (o nickname), campos obligatorios vacíos, …

Esta información debe ser almacenada en una base de datos, que permita a cada usuario acceder a su personaje al introducir sus credenciales.

El sistema deberá permitir que al iniciar el programa una persona, tenga opción de iniciar sesión o de crear un nuevo usuario. Si la persona elige la creación de un nuevo usuario, el sistema pedirá información para la creación del perfil como nombre de usuario (deberá verificar en la base de datos que el nombre no esté en uso), contraseña, rol de su personaje (deberá elegir entre vampiro, licántropo o cazador). El resto de propiedades o atributos del personaje (oro, armas activas, equipo, vida, nº de esbirros, …) se inicializarán con valores por defecto. Estas propiedades se irán modificando de forma automática tras cada desafío o bien por la acción del usuario, que podrá acceder a la “tienda” para la compra de equipos, esbirros, … que adquirirá por oro.

Iniciación y desarrollo de los desafíos

1. Solicitud de un desafío

   Los desafíos comenzarán cuando un usuario decida atacar (o desafiar) a otro usuario. Para ello, el atacante deberá escribir el nickname del usuario al que desea atacar.

   El sistema comprobará que dicho usuario existe y que no tiene “ningún escudo activo” (es decir, no ha sido atacado por el mismo usuario en las 24 horas anteriores).

   Posteriormente, el atacante escribirá la cantidad de oro que desea apostar. El sistema se asegurará de que el atacante dispone de esa cantidad de oro y validará el desafío. También, el sistema verificará que el usuario desafiado dispone de al menos el 10% de la cantidad apostada. En caso contrario, el sistema avisará al atacante de que no se puede procesar la solicitud de desafío.

Por último, el atacante eligirá el/las arma/s, armadura, habilidad especial y número de esbirros (el sistema deberá verificar que el usuario dispone de dicho número de esbirros) que desea que porte su personaje para el desafío.

2. Aceptación del desafío

Tras la aceptación del sistema del desafío, al usuario desafiado nada más iniciar sesión le saldrá un aviso de desafío pendiente. El usuario solo podrá aceptar o declinar el desafío.

2.1. El usuario retado no acepta el desafío

La declinación del reto conlleva una pérdida de saldo por rendirse. Esta pérdida será del 10% de la cantidad apostada por el desafiante.

Tras la rendición, el programa volverá a la pantalla inicial.

2.2. El usuario retado acepta el desafío

El sistema solicitará al usuario que seleccione el/las arma/s, armadura, habilidad especial y número de esbirros con los que desea que su personaje acuda al desafío. El sistema verificará la información proporcionada y dará comienzo el desafío.

3. Desarrollo del desafío

El sistema se encargará de recrear el desafío. Cada desafío se compondrá de diferentes combates, hasta que un usuario resulte derrotado.

En cada combate el rol de atacante-defensor se irá alternando por rondas.

El programa se encargará de crear aleatoriamente distintas circunstancias en cada combate que propicien mejores ataques o defensas en relación a cada tipo de personaje (determinadas circunstancias mejorarán los ataques/defensas de los personajes. E.g. la luna llena beneficia a los licántropos; la luz solar perjudica a los vampiros).

El sistema deberá actualizar de la base de datos los cambios realizados en cada combate (puntos de salud perdidos, número de esbirros muertos, valor de rabia, …)

Tras la “muerte” de uno o de ambos personajes al acabar una ronda el desafío termina, determinando el resultado del combate.

4. Finalización del desafío

   El sistema deberá informar a cada contrincante el resultado de la batalla. Además deberá mostrar un informe que identifique el desafío, participantes, cantidad de oro apostada, número de combates realizados, …

   El sistema deberá actualizar el "ranking universal" con el nuevo desafío y "proteger" al usuario perdedor de nuevos ataques del usuario vencedor durante las 24 horas posteriores.

Creación, modificación y consulta del ranking de desafíos

El sistema creará un ranking en relación al número de desafíos ganados. Este ranking deberá actualizarse tras cada desafío, mostrando de manera descendente el nickname de los jugadores y el número de batallas ganadas que acumulan.

Cada usuario desde el menú tendrá acceso al ranking y al puesto en el que se encuentra.

Este ranking deberá almacenarse en una base de datos, de forma que se acumulen todas las batallas jugadas.

Consulta y modificación del personaje, equipo disponible y propiedades del personaje (salud, oro, debilidades, fortalezas, …)

El sistema deberá garantizar a cada usuario poder consultar desde el menú principal toda la información relativa a su personaje.

En relación al equipo (arma/s y armadura), podrá intercambiar el arma/s activa/s por otra que posea en el "arsenal". Lo mismo debe ocurrir para la armadura.

Tienda

El sistema deberá garantizar que cada usuario pueda acceder a la "tienda" desde el menú principal.

En la tienda podrá comprar con su oro disponible armas, armaduras, esbirros, habilidades especiales (del tipo que sea su personaje), …

El sistema deberá actualizar la cantidad de oro del usuario, y almacenar en la base de datos los objetos comprados por el usuario.

Banear/Desbanear usuario

El sistema deberá comprobar que ningún usuario ataque a otro al que ha atacado en las 24 horas anteriores.

Transcurrido dicho tiempo, el sistema debe permitir poder atacar de nuevo (desbanear).

Requisitos no funcionales

Requisitos de interfaz: menús, ventanas, mensajes de error y notificaciones.

La interfaz del programa será básica, mostrando el programa por consola y sin el uso de una GUI, dado que el cliente muestra indiferencia respecto al uso de interfaz gráfica o de ventanas.

Se permitirá la implementación de mensajes de error, únicamente en los casos que sea indispensable. Pero por lo general se procurará evitar implementarlos y no se permitirá su uso, es decir, no se podrán producir mensajes de error.

Se tendrá en cuenta que cuando un usuario desafíe a otro, este último deberá recibir una notificación justo al entrar en el sistema, avisando de que ha sido desafiado.

Requisitos operacionales: modos de operación, back-ups.

La gestión de la información y back-ups queda subrogada a las correspondientes empresas que prestan su servicio de gestión de base de datos.

Requisitos de documentación: idiomas, ayuda on-line.

No se implementará un sistema de traducción de ningún tipo. El único idioma utilizado será el castellano.

La gestión de la ayuda on-line queda subrogada a las correspondientes empresas que prestan su servicio de gestión de ayuda on-line.

Requisitos de seguridad: niveles de acceso.

El sistema de seguridad debe ser sencillo: Cada usuario tiene su propio nick, nombre, identificador único y contraseña.

El usuario es único y posee todos los permisos, salvo los permisos del administrador.

Requisitos de portabilidad y mantenimiento.

La portabilidad y el mantenimiento de la aplicación será gestionado por las correspondientes empresas. En nuestro caso, el programa debe ser ejecutable únicamente desde el dispositivo propio del usuario.

Requisitos de recursos: memoria, almacenamiento.

Durante la ejecución del programa se usarán estructuras dinámicas y estáticas de acuerdo a reducir lo máximo posible el uso de memoria. Además se contará con un sistema de persistencia que permitirá almacenar la información de los usuarios en el dispositivo al terminar la ejecución del programa.

Análogamente este sistema de serialización será de utilidad para las empresas que gestionan la base de datos de la aplicación.

Requisitos de rendimiento: tiempo de respuesta, número de usuarios.

No se ha establecido un número límite de usuarios en total. Sin embargo, en el mismo programa se podrá gestionar un máximo de dos usuarios.

El tiempo de respuesta será el mínimo posible. Para ello se debe pensar un diseño óptimo de la aplicación.