

PRÁCTICA METODOLOGÍA DE LA PROGRAMACIÓN  
2022-23

## ANÁLISIS FUNCIONAL

NOMBRE	APELLIDOS	DNI	GRADO
Lucía	Domínguez Rodrigo	49452469-P	GII+GIS
Ángel	Marqués García	34295108-S	GII+GIS
Marcos	Jiménez Pulido	04860135-M	GII+GIS
Sergio	De Oro Fernández	54712181-B	GII+GIS

LISTA DE DISTRIBUCIÓN DE ROLES

NOMBRE	ROL	ORGANIZACIÓN
Sergio	Jefe de Proyecto	URJC
Lucía	Analista Funcional	URJC
Marcos	Analista Programador	URJC
Sergio	Ingeniero de Desarrollo	URJC
Ángel	Quality Assurance Tester	URJC

## Contenido

OBJETIVOS DEL DOCUMENTO .....	3
ANÁLISIS DEL PROBLEMA .....	3
OBJETIVO DE LA APLICACIÓN .....	3
CONCEPTOS BÁSICOS .....	3
IDENTIFICAR USUARIOS POTENCIALES.....	4
IDENTIFICAR FUENTE DE DATOS .....	4
RECOPILAR INFORMACIÓN Y HECHOS .....	4
PREGUNTAS DIRIGIDAS HACIA EL CLIENTE .....	4
REQUISITOS .....	5
Requisitos funcionales.....	5
Explicación en detalle de cada requisito: .....	6
Requisitos no funcionales .....	11
Requisitos operacionales .....	12
Requisitos de documentación .....	13
Requisitos de seguridad .....	13
Requisitos de portabilidad y mantenimiento.....	13
Requisitos de recursos .....	13
Requisitos de rendimiento .....	14
ENTIDADES DE LA APLICACIÓN .....	14
ROLES DE USUARIO .....	20
PROCESOS DEL SOFTWARE .....	22
Combate .....	22
Desafío.....	23
Persistencia .....	24
Anexo 1: Opciones y funcionamiento de los menús .....	25
Usuario Anónimo .....	25
Jugador .....	25
Administrador .....	27

## OBJETIVOS DEL DOCUMENTO

Este documento establece los requisitos que debe de seguir el sistema para que se ejecute el software que la empresa *MetProg URJC S.L.* ha pedido desarrollar. Se definirán dichos requisitos, así como las entidades y los conceptos clave necesarios para poder realizar el proyecto de manera óptima y llegar a cumplir sus objetivos.

## ANÁLISIS DEL PROBLEMA

Se debe crear un videojuego que cumpla una serie de características marcadas (las cuales se expresarán en forma de requisitos y análisis del sistema en este mismo documento). Dicho videojuego será ejecutado en un sistema, y el usuario controlará las acciones a realizar para poder ejecutar el programa de forma correcta.

## OBJETIVO DE LA APLICACIÓN

El videojuego deberá de dejar a los usuarios retar a otros oponentes para luchar contra ellos. Dichos combates se realizarán con un personaje, creado y equipado por el usuario.

## CONCEPTOS BÁSICOS

A continuación, se definirán algunos conceptos claves que se repetirán a lo largo de este documento:

- **ARCHIVO:** fichero organizado por un conjunto de unidades (bytes), necesarios para almacenar la información requerida por el programa, así como información de consulta para los usuarios.
- **USUARIO:** se define como cualquier persona que pueda interactuar con el sistema. Se distinguirá entre *usuarios anónimos*, *jugadores*, y *administradores*.
- **DARSE DE ALTA:** acción que realizarán los usuarios finales para poder registrarse en el sistema. Este término también aparecerá referenciado en este documento como *registrarse* en el sistema.
- **DARSE DE BAJA:** acción que realizarán los usuarios finales para eliminar su registro del sistema. Con dicha acción, se eliminará al usuario de los registros del programa.
- **INICIAR SESIÓN:** acción que realizará el usuario para entrar al sistema a través de un *nick* y una contraseña. Para poder realizar esta acción, el usuario deberá de haberse dado de alta en el sistema antes.
- **NICK:** alias o pseudónimo que se utilizará para hacer referencia a un determinado usuario del software.

## IDENTIFICAR USUARIOS POTENCIALES

El videojuego estará enfocado para un sector de población concreta (más específicamente estudiantes y profesorado de la Universidad Rey Juan Carlos), que por lo general estará bastante familiarizado con la manera de ejecutar del videojuego, así como con el funcionamiento de dicho programa.

Sin embargo, ya que el proyecto estará en un repositorio público en GitHub, cualquiera podrá acceder a él.

La aplicación tendrá distintos tipos de usuarios, que dependerán fundamentalmente del tipo de funcionalidades que pueda ejecutar cada uno. Este rol vendrá dado por el tipo de permisos que la aplicación conceda a cada uno de ellos, los cuales son: administrador, jugador, y usuario anónimo.

## IDENTIFICAR FUENTE DE DATOS

La información necesaria para la ejecución del programa se dividirá en dos grandes ramas. La primera será la información que el usuario deberá completar manualmente, y que será almacenada en archivos. Por otro lado, la aplicación utilizará y recabará información de manera automática, que utilizará el mismo tipo de almacenamiento que la rama anterior.

## RECOPILAR INFORMACIÓN Y HECHOS

Se han planificado reuniones con el intermediario (el profesor), y la empresa que requiere el software para poder comprender mejor la lógica que debe seguir la operativa del programa, así como dudas que puedan surgir. Dichas dudas se podrán plantear y resolver, de manera análoga, a través correo electrónico, o de manera presencial.

## PREGUNTAS DIRIGIDAS HACIA EL CLIENTE

1. A la hora de apostar una cantidad de oro, ¿se contempla la posibilidad de poder apostar una cantidad nula (es decir, 0)?
2. Una vez iniciada la sesión, ¿tiene que comprobarse que un determinado usuario tenga equipo activo?
3. Con respecto al ranking global, ¿se tiene que hacer mediante puntuación o mediante el oro obtenido en los distintos desafíos?
4. Para utilizar los dones de un determinado tipo de personaje, se utiliza un valor de coste mínimo. Se puede asumir que, si no tienes ese mínimo, no se puede utilizar ese don. ¿Puede ser dicho coste 0?
5. ¿Cómo se añaden los esbirros al personaje?
6. ¿Cómo se añaden los esbirros a los esbirros?

7. ¿Las distintas formas de lealtad en los esbirros, qué función cumplen?
8. ¿A qué se refiere al hablar sobre la “forma bestia” de los licántropos? ¿Cómo afecta al personaje o al juego?
9. ¿Las debilidades, fortalezas, y habilidades, vienen preestablecidas para cada personaje, o se pueden elegir?
10. ¿Debe haber un mínimo de habilidades por personaje?
11. Los objetos o entidades seleccionables, ¿vienen predefinidos en ficheros?
12. En el documento facilitado, se da información descriptiva sobre los personajes. ¿Debería aparecer esta información en el juego en forma de una descripción de cada personaje?

## REQUISITOS

En este apartado se enumerarán y describirán los requisitos (funcionales y no funcionales), de manera general, que debe cumplir el producto software. Los requisitos específicos relacionados con cada entidad se mencionarán y definirán en sus correspondientes apartados (en este mismo documento).

### Requisitos funcionales

1. Guardar la información referente a los personajes de ejecución en ejecución.
2. Darse de alta en el sistema.
3. Iniciar sesión en el sistema.
4. Cerrar sesión en el sistema.
5. Tener un registro de los usuarios dados de alta en la aplicación.
6. Darse de baja en el sistema.
7. Se debe poder crear personajes.
8. Se debe poder eliminar personajes.
9. Elegir el equipo activo (arma(s) y armadura) de los personajes.
10. Se debe de poder consultar el oro disponible.
11. Deberá existir un ranking global.
12. Se podrá consultar el ranking global.
13. Se debe poder consultar el registro de personajes.
14. Se podrá editar los personajes.
15. Consultar los desafíos pendientes de validación.
16. Se debe poder validar los desafíos que lancen los usuarios.
17. Se debe poder banear a los usuarios que no cumplan las normas.
18. Se debe poder desbanear a los usuarios.
19. Consultar el historial de oro ganado/perdido.
20. Desafiar a otros usuarios por el *Nick*.
21. Definir cantidad de oro apostada por combate.
22. Enviar notificaciones a usuarios desafiados.
23. Aceptar o rechazar desafíos al iniciar sesión.
24. Penalizar a usuarios que rechacen desafíos.
25. Permitir cambiar equipo activo antes de un desafío.

26. Hacer un combate entre dos personajes.
27. Mostrar a los usuarios el resultado del desafío.
28. Se deberá guardar la información de los combates.

Explicación en detalle de cada requisito:

1. Guardar la información referente a los personajes de ejecución en ejecución  
**Tipo:** obligatorio  
**Descripción:** El sistema debe guardar la información de los personajes, para ser consultada en sucesivas ejecuciones.  
**Riesgo:** medio  
**Observaciones:** Requiere acceso a los archivos de almacenamiento de información.
2. Darse de alta en el sistema  
**Tipo:** obligatorio  
**Descripción:** El sistema debe permitir introducir un nuevo usuario, a través de un nombre de usuario, un *Nick* único, y una contraseña (obligatorias para darse de alta).  
**Riesgo:** alto  
**Observaciones:** Si no funciona, no se podrá acceder al sistema. Requiere acceso a los archivos de almacenamiento de información.
3. Iniciar sesión  
**Tipo:** obligatorio  
**Descripción:** el sistema debe permitir a un usuario iniciar sesión mediante su Nick y contraseña.  
**Riesgo:** Alto  
**Observaciones:** Se debe acceder a los archivos de almacenamiento de información. Si un usuario no pudiera iniciar sesión, no se podría jugar.
4. Cerrar sesión  
**Tipo:** obligatorio  
**Descripción:** Una vez iniciada sesión, jugadores y administradores pueden cerrar sesión para salir del juego a la pantalla inicial.  
**Riesgo:** Medio  
**Observaciones:** Si no se incluyera, se podría cerrar sesión cerrando el programa y volviéndolo a ejecutar, pero esto sería incómodo para el usuario.
5. Tener un registro de los usuarios dados de alta en la aplicación  
**Tipo:** obligatorio  
**Descripción:** El sistema debe de almacenar la información de los usuarios que se den de alta en el juego.  
**Riesgo:** Medio  
**Observaciones:** Requiere acceso a los archivos de almacenamiento de información.

6. Darse de baja en el sistema  
**Tipo:** obligatorio  
**Descripción:** El sistema debe permitir a cualquier usuario darse de baja, borrando así toda su información del sistema.  
**Riesgo:** Medio  
**Observaciones:** Requiere acceso a los archivos de almacenamiento de información.
  
7. Se debe poder crear personajes  
**Tipo:** obligatorio  
**Descripción:** El sistema debe permitir al usuario crear un personaje.  
**Riesgo:** Medio  
**Observaciones:** Si el usuario no cuenta con personajes, no podrá iniciar ningún desafío, así que es importante permitirle crear un personaje para evitar esta situación. Además, si el usuario ya cuenta con un personaje, no podrá crear otro.
  
8. Se debe poder eliminar personajes  
**Tipo:** obligatorio  
**Descripción:** El sistema debe permitir al usuario eliminar su personaje.  
**Riesgo:** Medio  
**Observaciones:** El usuario no podrá borrar un personaje a no ser que lo posea.
  
9. Elegir el equipo activo (arma(s) y armadura) de los personajes.  
**Tipo:** obligatorio  
**Descripción:** El sistema debe permitir elegir el equipo activo para cada personaje (este equipo será elegido dentro del rango accesible de armas/armaduras para cada personaje).  
**Riesgo:** Medio  
**Observaciones:** Si el usuario no tiene equipo activo, no podrá participar en ningún combate.
  
10. Se debe mostrar el oro disponible  
**Tipo:** obligatorio  
**Descripción:** el software debe de permitir al usuario ver el oro disponible que tiene su personaje. Este dato aparecerá en el menú principal, junto a la opción de “Desafiar a un usuario”. Adicionalmente, el usuario podrá acceder a un historial que mostrará más en detalle su oro, ganancias y pérdidas. Esto está recogido en el requisito 19.  
**Riesgo:** Medio  
**Observaciones:** Se debe de acceder a los archivos de almacenamiento de información.

11. Deberá existir un ranking global  
**Tipo:** obligatorio  
**Descripción:** el programa hará de manera automática un ranking de los mejores usuarios (por número de victorias).  
**Riesgo:** Medio  
**Observaciones:** Se debe de acceder a los archivos de almacenamiento de información.
12. Se podrá consultar el ranking global  
**Tipo:** obligatorio  
**Descripción:** Los jugadores podrán acceder al ranking global para ver su posición en él.  
**Riesgo:** Medio  
**Observaciones:** Se debe de acceder a los archivos de almacenamiento de información. Si el ranking fuera demasiado grande, se pondría primero la posición del usuario (siempre y cuando no esté entre los primeros), seguida por los primeros puestos del ranking (para evitar que el usuario tenga que buscar su Nick entre cientos).
13. Se debe poder consultar el registro de personajes  
**Tipo:** obligatorio  
**Descripción:** Los administradores deben poder consultar un registro de personajes, que contendrá toda la información de los mismos.  
**Riesgo:** Medio  
**Observaciones:** Se debe acceder a los archivos de almacenamiento de información.
14. Se podrá editar los personajes.  
**Tipo:** obligatorio  
**Descripción:** el programa debe permitir a los administradores editar todas las propiedades de los personajes. Estos cambios quedarán guardados para próximas partidas.  
**Riesgo:** Medio  
**Observaciones:** Se debe de acceder a los archivos de almacenamiento de información.
15. Consultar los desafíos pendientes de validación  
**Tipo:** obligatorio  
**Descripción:** los administradores deben poder ver el registro de los desafíos que aún no han sido validados, con todas las características de los mismos  
**Riesgo:** Alto  
**Observaciones:** Sin esta funcionalidad, no se podría validar ningún desafío, interfiriendo con el objetivo del software.
16. Se debe de poder validar los desafíos que lancen los usuarios.  
**Tipo:** obligatorio



**Descripción:** el programa tiene que validar los desafíos a través del usuario que inicie sesión como administrador. Es decir, el desafío debe de ser aprobado por un administrador del sistema antes de poder realizarse.

**Riesgo:** Alto

**Observaciones:** En caso de que el software no ejecute dicha validación correctamente, no se podrá retar a alguien, y, por lo tanto, no se podrán realizar combates.

17. Se debe poder banear a los usuarios que no cumplan las normas.

**Tipo:** obligatorio

**Descripción:** un administrador debe poder banear a los usuarios que no cumplan las normas, como forma de “reprimenda” para que así se respeten las reglas del juego.

**Riesgo:** Medio

**Observaciones:** si esta función fallara, no se respetarán las reglas del juego. La regla más importante es que no se debe retar a un usuario que haya perdido un combate en las últimas 24 horas.

18. Se debe poder desbanear a los usuarios.

**Tipo:** obligatorio

**Descripción:** un administrador debe tener la opción de desbanear a los usuarios baneados.

**Riesgo:** Medio

**Observaciones:** Debe haber usuarios baneados para poderlos desbanear. Esta opción se añade para darle la oportunidad a los usuarios de aprender de sus errores.

19. Consultar el Historial de oro ganado/perdido

**Tipo:** obligatorio

**Descripción:** Los jugadores deberán de ser capaces de ver el oro ganado/perdido en los combates en los que hayan participado.

**Riesgo:** Medio

**Observaciones:** Se debe de acceder a los archivos de almacenamiento de información.

20. Desafiar a otros usuarios por el Nick

**Tipo:** Obligatorio.

**Descripción:** El sistema debe permitir que un jugador proponga un desafío a otro, identificándolo mediante su Nick o alias. Una vez enviado y validado por el administrador, el desafío se mostrará en la pantalla principal del usuario desafiado esperando a ser aceptado o rechazado.

**Riesgo:** Alto.

**Observaciones:** En caso de no funcionar el sistema de desafíos, los usuarios no podrían interactuar entre ellos, por lo que no se estaría cumpliendo el objetivo del software.

21. Definir cantidad de oro apostada por combate

**Tipo:** Obligatorio.

**Descripción:** A la hora de que un usuario desafíe a otros, este podrá elegir la cantidad de oro que quiere apostar en ese desafío. La cantidad de oro apostada por el usuario siempre debe de ser menor o igual a la cantidad de oro que posea su personaje en ese momento, debiendo impedir el sistema que se definan cantidades superiores o negativas.

**Riesgo:** Alto.

**Observaciones:** En el caso de no funcionar correctamente, el sistema de recompensas no funcionaría correctamente.

22. Enviar notificaciones a usuarios desafiados

**Tipo:** Obligatorio.

**Descripción:** Si un jugador desafía a otro, una vez este desafío haya sido validado, el sistema deberá enviar una notificación de nuevo desafío al jugador desafiado cuando inicie sesión en el sistema, no dejando al jugador realizar ninguna acción hasta que acepte o rechace el desafío. Dentro de esta notificación el sistema deberá mostrar algunos datos del jugador desafiante para ayudar al usuario desafiado a tomar una decisión sobre el desafío propuesto.

**Riesgo:** Alto.

**Observaciones:** Si esta funcionalidad no estuviese o no funcionase correctamente, el sistema de desafíos no notificaría a los usuarios desafiados, por lo que es posible que el desafío tarde mucho tiempo en gestionarse o no se gestione nunca.

23. Aceptar o rechazar desafíos al iniciar sesión

**Tipo:** Obligatorio.

**Descripción:** Cuando el usuario inicie sesión en el sistema y reciba una notificación de desafío por parte de otro usuario, el usuario desafiado deberá poder aceptar o rechazar el desafío propuesto, impidiéndole que use ninguna otra función del sistema hasta que tome la decisión.

**Riesgo:** Alto.

**Observaciones:** Sin esta funcionalidad, los usuarios que hayan sido desafiados no podrán responder a los desafíos, por lo que no se podrán llevar a cabo combates, que son una parte muy importante del sistema.

24. Penalizar a usuarios que rechacen desafíos

**Tipo:** Obligatorio.

**Descripción:** Si un jugador rechaza un desafío deberá pagar un 10% de lo que su oponente hubiera apostado.

**Riesgo:** Medio.

**Observación:** si el usuario no tiene suficiente oro para pagar la penalización, pagará todo lo que pueda, pero nunca incurrirá deuda (el mínimo oro de personaje es 0).

25. Permitir cambiar equipo activo antes de un desafío

**Tipo:** Obligatorio.

**Descripción:** Al aceptarse un desafío, el sistema deberá permitir que el jugador desafiado cambie el equipo activo de su personaje antes de iniciar el combate.

**Riesgo:** Alto.

**Observaciones:** En el caso de que esta funcionalidad estuviese ausente o no funcionase correctamente, los usuarios no podrían planear su estrategia, y los combates serían mucho más ordinarios.

26. Hacer un combate entre dos personajes

**Tipo:** Obligatorio

**Descripción:** Al aceptarse un desafío por parte del desafiado, el sistema debe realizar un combate por turnos entre los personajes del desafiante y el desafiado. Este combate puede tener un vencedor o no (caso de empate).

**Riesgo:** Alto.

**Observaciones:** Sin esta funcionalidad se pierde el objetivo principal del software.

27. Mostrar a los usuarios el resultado del desafío

**Tipo:** Obligatorio.

**Descripción:** En el momento en el que se finalice un combate entre dos usuarios, el sistema deberá mostrarles los resultados finales del desafío, mostrando a cada uno de ellos el resultado del combate y los sucesos acontecidos en cada una de las rondas, así como la cantidad de oro ganada o perdida por cada uno de los personajes de los usuarios.

**Riesgo:** Medio.

**Observaciones:** esta información se le mostrará inmediatamente al usuario desafiado, mientras que al usuario desafiante le aparecerá en forma de notificación al iniciar sesión.

28. Se deberá guardar la información de los combates

**Tipo:** Obligatorio.

**Descripción:** El software guardará por cada combate, quien fue el vencedor, (en caso de empate, se guardará como empate), el oro ganado/perdido, la duración del combate, la fecha del combate, el usuario que haya quedado con algún esbirro vivo (si lo hay), y un registro de lo ocurrido cada ronda.

**Riesgo:** Medio.

**Observaciones:** Aunque por ahora sólo se pueda acceder a parte de esta información, se guardará por si en un futuro se quisiera utilizar dichos datos en la aplicación (dar más información, uso estadístico, etc.).

## Requisitos no funcionales

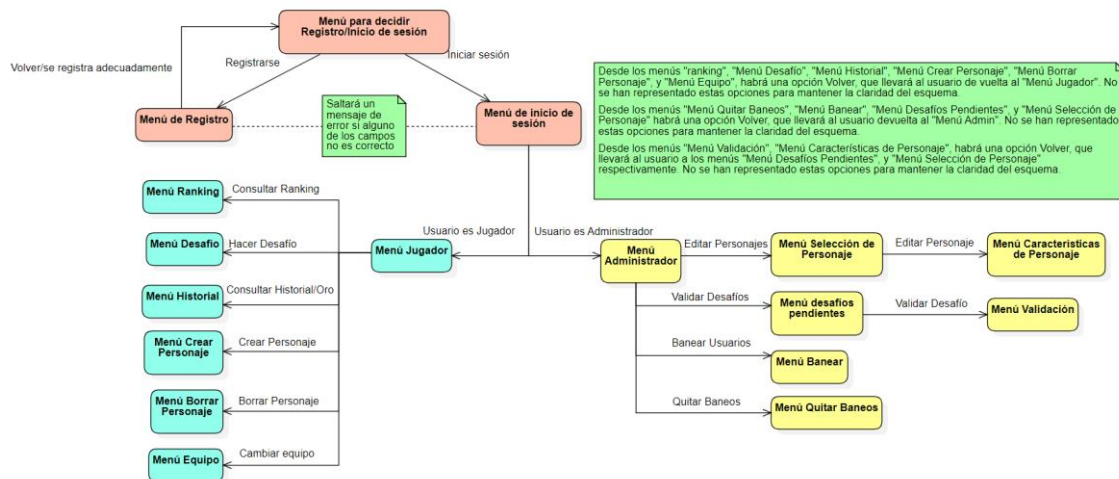
### Interfaz del sistema

La interfaz del sistema consistirá en una serie de menús mostrados por consola. Cada menú tendrá una serie de opciones. Si hubiera algún error en alguno de los menús, se mostraría un

mensaje indicándolo, y cuando el usuario decida continuar, se le devolverá al menú en el que se causara el error (salvo excepciones).

Las opciones de cada menú y su funcionamiento se explican en detalle en el anexo 1.

A continuación, se muestra un diagrama sencillo que detalla la navegación entre los distintos menús:



### Mensajes de error

Se producirá un mensaje de error cuando:

- No se pueda efectuar un desafío porque los personajes implicados no tengan equipo activo.
- Al registrarse, el Nick introducido por el usuario ya haya sido escogido por otro.
- No se encuentre un usuario con el nombre y contraseña especificados al iniciar sesión.
- Si el usuario intenta llevar a cabo cualquier acción, pero no tiene un personaje.
- La cantidad de oro a apostar en un desafío no sea válida.
- El usuario intenta crear un personaje y ya tiene uno creado
- Si el administrador introduce valores no válidos al editar un personaje
- En cualquier otro caso en el que el input del usuario no sea válido.

### Requisitos operacionales

#### Modos de operación

El programa contará con varios modos de operación, dependiendo del rol que tome el usuario (administrador/jugador). Una vez determinado este rol, el modo asociado al mismo se mantendrá activo hasta que el usuario cierre sesión.

### Back-ups

La gestión de la información irá ligada a una serie de archivos de almacenamiento, los cuales contendrán todos los datos asociados a la aplicación. Se usarán ficheros de tipo binario, que

emularán bases de datos, y documentos de texto plano que ofrecerán información por defecto para la aplicación.

## Requisitos de documentación

### *Idiomas*

Este programa sólo estará disponible en castellano.

### *Ayuda*

La aplicación no proporcionará ayuda en la web (online). Toda la ayuda se pondrá a disposición del usuario en la aplicación misma.

## Requisitos de seguridad

### *Nivel de acceso*

Habrán dos niveles de acceso, uno de jugador y uno de administrador. Los jugadores podrán consultar una serie de datos, y modificar una selección muy limitada de ellos (crear/eliminar su personaje asociado, y crear nuevos desafíos). Los administradores podrán consultar una mayor cantidad de información, y serán capaces de editarla libremente (cumpliendo con algunos requisitos base).

Existen 3 tipos de usuarios que pueden acceder a distintas funcionalidades del sistema: el usuario anónimo, el jugador, y el administrador. Las funcionalidades a las que puede acceder cada uno de estos usuarios se especificarán en detalle en el apartado “Roles de usuario”

## Requisitos de portabilidad y mantenimiento

Una vez finalizada la fase de implementación y pruebas, este videojuego no tendrá ningún tipo de mantenimiento (activo o pasivo).

## Requisitos de recursos

### *Memoria*

La aplicación se guardará en memoria local, pero tendrá persistencia. Dicho software será ligero, y no consumirá muchos recursos del sistema.

### *Almacenamiento*

Todos los datos (tanto introducidos manualmente como recabados por el programa) se almacenarán en archivos.

Además, habrá “información por defecto” que se creará previamente a ejecutar el sistema por primera vez y se almacenará en una serie de ficheros.

## Requisitos de rendimiento

### *Tiempo de respuesta*

Los tiempos de respuesta serán lo más bajos posible.

### *Número de usuarios*

Se tendrán varios usuarios dados de alta, pero sólo uno podrá tener sesión iniciada a la vez (login con usuario ya registrado en el sistema).

## ENTIDADES DE LA APLICACIÓN

### **Entidad Personaje**

#### Propiedades

- Nombre: cadena de caracteres.
- Descripción: cadena de caracteres
- Oro: valor numérico.
- Salud: valor numérico.
- Poder: valor numérico.
- Armas activas: colección de la entidad Equipo.
- Armas disponibles: colección de la entidad Equipo.
- Armadura activa: elemento de la entidad Equipo.
- Armaduras disponibles: colección de la entidad Equipo.
- Habilidad especial: elemento de la entidad Habilidad especial.
- Tipo de recurso: elemento del tipo Recurso
- Cantidad de recurso: valor numérico.
- Esbirros: colección de la entidad Esbirro.
- Fortalezas: colección de la entidad Modificador.
- Debilidades: colección de la entidad Modificador.

#### Restricciones de la entidad

- La cantidad de oro nunca puede ser negativa, pero sí 0.
- El valor de salud debe estar comprendido entre 0 y 5.
- El valor de poder debe estar comprendido entre 1 y 5.
- Las armas activas y las disponibles deben ser del tipo de Equipo: "Arma".
- La armadura activa y las armaduras disponibles deben ser del tipo de Equipo: "Armadura".
- La cantidad de esbirros, fortalezas y debilidades es indefinida.
- Las armas activas pueden ser como máximo 2 (si ambas son a 1 mano) o 1 (si el arma es a 2 manos)

#### Tipos de personaje

- I. Vampiro:
  - o Edad: valor numérico.
- II. Licántropo

### III. Cazador

#### Restricciones del tipo de personaje

- Vampiro:
  - Los vampiros no pueden tener esbirros humanos.
  - El tipo de recurso debe tomar como valor "Sangre".
  - La propiedad "Cantidad de recurso" toma valores entre 0 y 10 (empezando en 0).
  - El recurso aumenta en 4 al hacer un ataque efectivo.
  - La habilidad especial debe ser del tipo: "Disciplina".
- Licántropo:
  - El tipo de recurso debe tomar como valor "Rabia".
  - El recurso toma valores entre 0 y 3 (empezando en 0).
  - El recurso aumenta en 1 cuando el personaje recibe daño.
  - La habilidad especial debe ser del tipo: "Don".
- Cazador:
  - El tipo de recurso debe tomar como valor: "Voluntad".
  - El recurso toma valores entre 0 y 3 (empezando en 3).
  - El recurso disminuye en 1 cuando el personaje recibe daño.
  - La habilidad especial debe ser del tipo: "Talento".

#### **Propiedad Recurso**

##### Valores posibles:

- Sangre
- Rabia
- Voluntad

#### **Entidad Usuario**

##### Propiedades

- Nombre: cadena de caracteres.
- Nick: cadena de caracteres.
- Contraseña: cadena de caracteres.

##### Tipos de usuario

- Administrador
  - Desafíos pendientes de validación: colección de la entidad Desafío
- Jugador
  - Número de registro: elemento de la entidad Número de Registro.
  - Personaje: elemento de la entidad Personaje.
  - Registro de combates: colección de la entidad Combate
  - Desafíos recibidos: colección de la entidad Desafío
  - Combates ganados: valor numérico

#### Restricciones del tipo de usuario

- Jugador
  - o Combates ganados no puede tomar valores negativos, pero sí 0

#### **Entidad Número de Registro**

##### Propiedades

- Carácter inicial: carácter
- Número uno: valor numérico
- Número dos: valor numérico
- Coletilla: cadena de caracteres

##### Restricciones de la entidad

- Las propiedades “número uno” y “número dos” son enteros entre 0 y 9
- La coletilla tiene longitud 2

#### **Entidad Desafío**

La entidad Desafío representa un enfrentamiento entre dos usuarios. Cada usuario podrá tener tantos desafíos pendientes como haya propuesto, y al finalizar el combate asociado a ese desafío, sus resultados deberán de tenerse en cuenta para el conteo total de victorias y la cantidad de oro total de cada uno de los personajes involucrados.

##### Propiedades:

- Usuario desafiante: cadena de caracteres
- Usuario desafiado: cadena de caracteres
- Cantidad de oro en juego: Valor numérico
- Estado del combate: Cadena de caracteres.
- Combate: elemento de la entidad Combate.

##### Restricciones de la entidad:

- El estado del combate debe estar comprendido entre los valores “Pendiente de validación”, “Pendiente de Respuesta” y “Completado”.
- La cantidad de oro apostada en cada desafío debe ser positiva, pudiendo ser 0.
- La propiedad combate no tomará un valor hasta que el estado del combate sea “Completado”

#### **Entidad Combate**

##### Propiedades

- Usuario desafiante: cadena de caracteres
- Usuario desafiado: cadena de caracteres
- Rondas empleadas: valor numérico
- Fecha: cadena de caracteres
- Vencedor: cadena de caracteres
- Usuario con esbirros sin derrotar: cadena de caracteres



- Oro ganado: valor numérico
- Registro de rondas: colección de la entidad Ronda

#### Restricciones de la entidad

- La propiedad "Vencedor" sólo puede tomar el valor de "Empate", o ser igual al usuario desafiante o desafiado
- La fecha tendrá el siguiente formato "Día de la semana dd/mm/aaaa"
- La propiedad "Usuario con esbirros sin derrotar" sólo puede tomar el valor de "Ninguno", o ser igual al usuario desafiante o desafiado
- Oro ganado no puede ser menor o igual a 0

#### **Entidad Ronda**

##### Propiedades

- Vida inicial del personaje A: valor numérico
- Vida inicial del personaje B: valor numérico
- Vida final del personaje A: valor numérico
- Vida final del personaje B: valor numérico
- El personaje A ha usado su habilidad especial: valor binario
- El personaje B ha usado su habilidad especial: valor binario
- El valor de ataque final del personaje A: valor numérico
- El valor de ataque final del personaje B: valor numérico
- El valor de defensa final del personaje A: valor numérico
- El valor de defensa final del personaje B: valor numérico

#### Restricciones de la entidad

- Todos los valores de vida están comprendidos entre 0 y 5

#### **Entidad Esbirro**

##### Propiedades:

- Nombre: cadena de caracteres
- Salud: valor numérico

#### Restricciones de la entidad

- La salud toma valores entre 1 y 3

#### Tipos de Esbirros

- I. Humano
  - Lealtad: cadena de caracteres
- II. Ghoul
  - Dependencia: valor numérico
- III. Demonio
  - Pacto: cadena de caracteres
  - Esbirros: colección de la entidad Esbirro

#### Restricciones del tipo de esbirro

- Ghoul
  - o La dependencia toma valores entre 1 y 5

#### **Entidad Modificador**

Afectarán de forma positiva (en el caso de las fortalezas) o negativa (en el caso de las debilidades) a las estadísticas del personaje en caso de que estén activos. Estos modificadores tendrán un nombre y un valor de intensidad entre 1 y 5. Cuanto más alto sea el valor asociado al modificador, mayor será su efecto sobre el personaje.

#### Propiedades:

- Nombre: cadena de caracteres
- Valor de ataque: valor numérico
- Valor de defensa: valor numérico

#### Restricciones de la entidad:

- El valor de ataque/defensa de un modificador, ya sea una debilidad o una fortaleza, deberá tener un valor entre 1 y 5.

#### Tipos de modificadores:

- Debilidad
- Fortaleza

#### Restricciones del tipo de modificador

- Debilidad
  - o Los valores de ataque y defensa serán negativos (en el mismo rango indicado).

#### **Entidad Equipo**

#### Propiedades

- Nombre: cadena de caracteres
- Ataque: valor numérico
- Defensa: valor numérico

#### Tipos de equipo

- Arma
  - o Variante: elemento del tipo Variante
- Armadura

#### Restricciones del tipo de equipo

- Arma

- El valor de ataque tomará valores entre 1 y 3
- El valor de defensa tomará valores entre 0 y 3
- Armadura
  - El valor de ataque tomará valores entre 0 y 3
  - El valor de defensa tomará valores entre 1 y 3

### **Propiedad Variante**

#### **Valores posibles**

- Una mano
- Dos manos

### **Entidad Habilidad Especial**

#### **Propiedades**

- Nombre: cadena de caracteres
- Ataque: valor numérico
- Defensa: valor numérico
- Coste: valor numérico

#### **Restricciones de la entidad**

- Las propiedades de Ataque y Defensa toman valores entre 1 y 3

#### **Tipos de Habilidad Especial**

- Disciplina
- Don
- Talento

#### **Restricciones del tipo de Habilidad Especial**

- Disciplina
  - El coste tomará valores entre 0 y 3
- Don
  - El coste tomará valores entre 0 y 3
- Talento
  - El coste estará fijado en 0

#### **Aclaraciones**

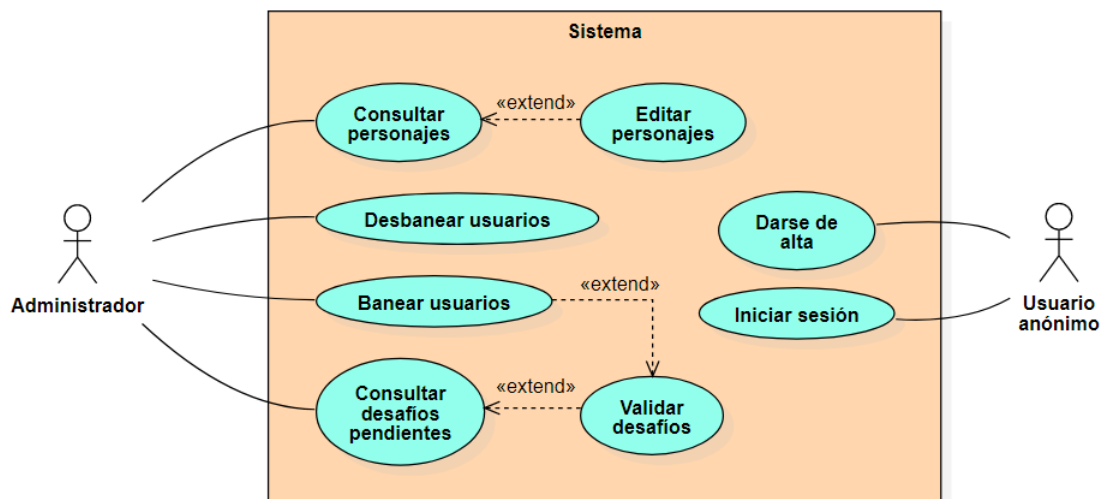
- El coste de los dones representa un mínimo necesario para ser usado. Al usar un don no se resta su “coste” al recurso acumulado del personaje.

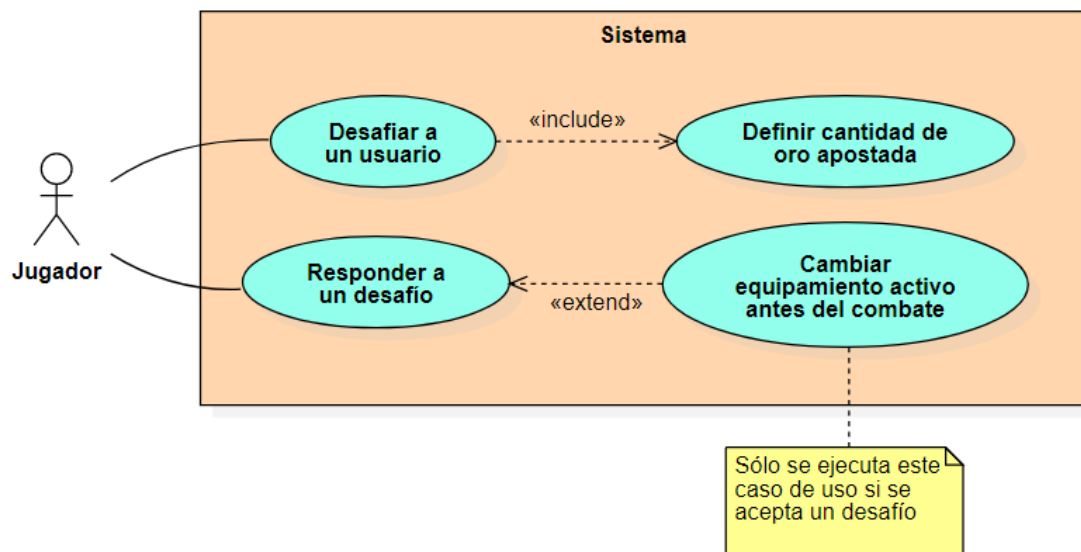
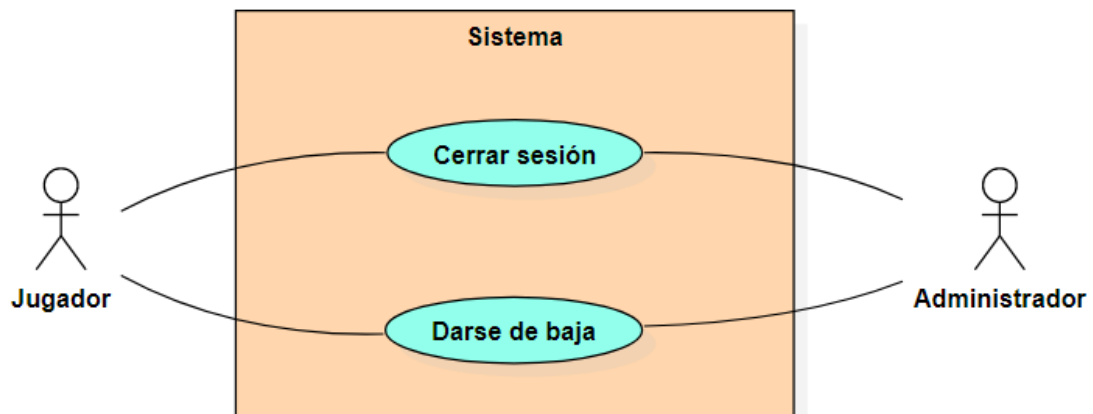
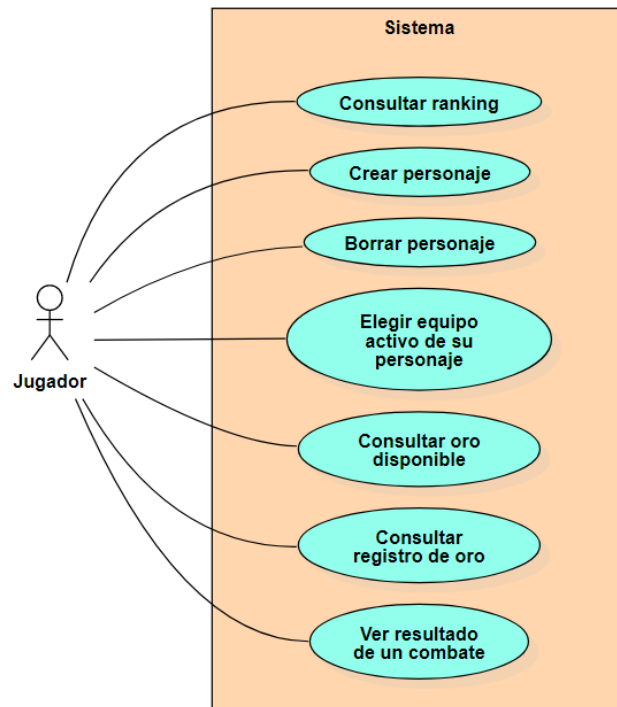
## ROLES DE USUARIO

La aplicación tendrá varios roles de usuario en función de los permisos que tiene el usuario a la hora de usar el software. Los tipos de usuario son:

- Administrador: tiene como permisos banear/desbanear usuarios, consultar todos los personajes, editar todos los personajes, consultar todos los desafíos pendientes de validación, validar desafíos, darse de baja, y cerrar sesión.
- Jugador: Tiene como permisos consultar el ranking global, crear y borrar un personaje, elegir el equipo activo de este mismo, consultar el oro disponible, consultar su registro de oro ganado/perdido, ver el resultado de un combate, desafiar a otro usuario por su Nick, definir la cantidad de oro apostada en ese desafío, responder a un desafío (aceptar/rechazar), cambiar el equipo activo de su personaje antes de un combate, cerrar sesión y darse de baja.
- Usuario anónimo: Tiene como permisos registrar un usuario e iniciar sesión.

A continuación, se muestran una serie de diagramas que representan los casos de uso del software y los actores que los ejecutan:





## PROCESOS DEL SOFTWARE

### Combate

En este apartado se definirá y ampliará lo que el usuario puede hacer en el videojuego. Para profundizar en dicha definición, utilizaremos las relaciones que se deben de dar entre las entidades definidas previamente, así como la secuencia que sigue la ejecución en cada uno de los **combates** que se realizan. Habrá que tener en cuenta que este combate se acabará cuando uno de los dos usuarios agote sus puntos de salud, lo que quiere decir que pueden darse varias rondas de ataque y defensa.

El combate se realizará a través de un desafío. Ambos usuarios deberán de tener un personaje. De este personaje se deberá de tener en cuenta:

- *Atributos*: se deberá tener en cuenta la vida y el poder, y, en función del tipo de personaje que se haya escogido, deberemos de tener en cuenta la cantidad de su recurso especial.
- *Habilidades especiales*: se deberán de tener en cuenta para los valores de ataque y defensa.
- *Equipo activo*: se deberá de tener en cuenta para los valores de ataque y defensa.
- *Fortalezas y debilidades*: se deberá de tener en cuenta para los valores de ataque y defensa.

Al principio del combate se calculará la vida de cada personaje. Esto se hará sumando los puntos de vida del personaje con los puntos de vida de cada uno de sus esbirros.

En cada ronda habrá que decidir si cada personaje recibe daño. Para esto, calcularemos su potencial de ataque y de defensa. Se usarán las siguientes fórmulas para calcular estos potenciales:

#### *Potencial de Ataque*

$$= \text{Poder del personaje} + \text{Ataque del arma} + \text{Ataque de la armadura} + \text{Ataque de la habilidad} + (\text{fortalezas} - \text{debilidades})$$

#### *Potencial de Defensa*

$$= \text{Defensa del arma} + \text{Defensa de la armadura} + \text{Defensa de la habilidad} + (\text{fortalezas} - \text{debilidades})$$

Los valores “Ataque de la habilidad” y “Defensa de la habilidad” se considerarán como 0 si el personaje no tiene la cantidad suficiente de recurso para usarla.

A las fórmulas se tiene que sumar un valor referente a la cantidad de recurso del personaje, que cambia para cada tipo de personaje:

- Vampiro: se suma 2 si la cantidad de sangre es mayor o igual a 5.
- Licántropo: se suma la cantidad de rabia.
- Cazador: se suma la cantidad de voluntad (recordar también que las habilidades del Cazador suman 0).

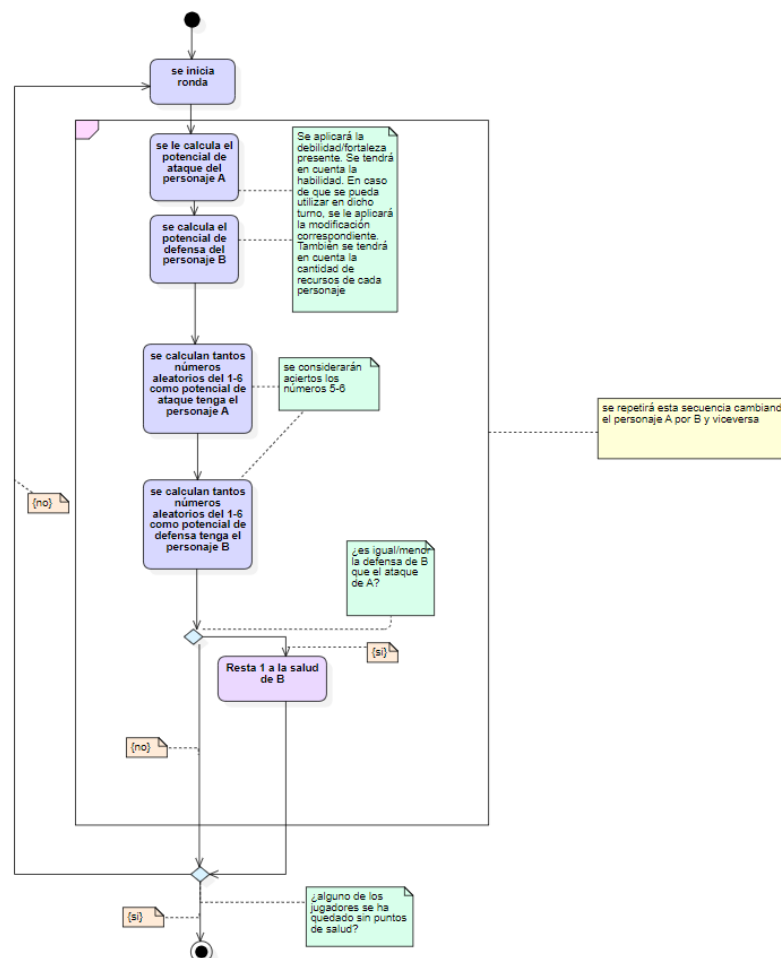
Luego, se calculará el ataque y defensa de cada personaje. Para ello, se tirarán  $n$  dados de 6 caras, siendo  $n$  el valor del potencial de ataque/defensa de cada personaje. El valor de ataque/defensa de un personaje será el número de dados que hayan sacado un 5 o un 6.

Si la defensa de un personaje es menor o igual al ataque del otro personaje, el defensor recibirá un punto de daño.

Al acabar el combate, deberemos indicar el vencedor (o que ha habido un empate). El vencedor del combate recuperará su apuesta, y recibirá el oro apostado por su oponente. Además de eso, el ganador recibirá una cantidad fija de oro.

En caso de empate, se indicará que ha habido empate, y cada jugador recuperará la cantidad de oro apostada.

El proceso de realización de un combate se hará de la siguiente manera (indicado mediante un diagrama de actividad):



Para cada combate debe quedar registrado lo que ocurre en cada ronda, así como los usuarios que participan, el número de rondas empleadas, la fecha del combate, el usuario vencedor, y el oro ganado (por cada personaje).

## Desafío

Un jugador puede desafiar a otro, apostando una cantidad de oro en el combate. Un jugador no podrá desafiar a nadie si su personaje no tiene equipo activo, y de igual manera, no se puede desafiar a un jugador si su personaje no tiene equipo activo.

Una vez lanzado el desafío, quedará pendiente de validación, siendo necesario que un administrador lo valide. Mediante este proceso, el administrador puede activar fortalezas o debilidades de los personajes.

Una vez validado el desafío, se le envía una notificación al jugador desafiado. El jugador desafiado deberá aceptar o rechazar el combate. Si acepta, puede cambiar el equipo activo de su personaje antes de que empiece el combate. Si rechaza el combate, deberá pagar un 10% de la cantidad apostada como penalización.

Una vez realizado el combate, se muestra a cada jugador una notificación con el resultado del combate y lo ocurrido en cada ronda. El jugador desafiado lo verá inmediatamente, mientras que el desafiante lo verá en forma de notificación al iniciar sesión.

### Persistencia

Las entidades usuario, personaje, desafío, y combate deberán ser almacenadas en ficheros para poder acceder a su información en sucesivas ejecuciones del software.

Aunque no toda la información de los combates será accesible actualmente, será necesario guardarla, ya que posibles cambios en el software pueden necesitar de esta información (cambios en la información mostrada a los jugadores), y también facilita posibles estudios del software (estudios estadísticos generales o sobre datos específicos del programa).



## Anexo 1: Opciones y funcionamiento de los menús

Este anexo detalla los distintos menús que existen en el sistema, explicando brevemente sus funciones y opciones. Los menús están divididos en función de qué tipo de usuario tendrá acceso a ellos.

Muchos de los menús tendrán una opción “Volver”. Esta opción devolverá al usuario al menú anterior. La opción “Salir”, presente únicamente en el menú de inicio, saca al usuario del sistema, cerrando la aplicación. La opción “Cerrar Sesión” devuelve al usuario al menú de inicio.

Siempre que un menú muestre información paginada (denotado generalmente por un paréntesis indicándolo), se mostrarán los resultados de  $n$  en  $n$ , teniendo la opción el usuario de acceder a los  $n$  elementos siguientes o anteriores.

Habrà una serie de "pantallas de error" que ocurrirán cuando el usuario intente llevar a cabo una opción imposible, o si ocurre algún tipo de error. Estas pantallas mostrarán un mensaje sencillo (por ejemplo "El *Nick* ya está cogido"), y pedirán introducir cualquier input para continuar, devolviendo al usuario al menú en el que ocurriera el error (salvo excepciones).

### Usuario Anónimo

#### Menú de Inicio

- Permite al usuario decidir si se quiere registrar o iniciar sesión
- Opciones:
  - Registrarse
  - Iniciar sesión
  - Salir

#### Menú de Registro

- Permitirá al usuario registrarse como administrador o jugador
- Pregunta si quieres registrar un administrador o un jugador, y luego pide la información necesaria para cada caso.
- Una vez la información ha sido recogida correctamente, se registrará la cuenta creada, y se devolverá al usuario al “Menú de Inicio”

#### Menú de Inicio de Sesión

- Permite al usuario iniciar sesión en una cuenta registrada
- Pide introducir primero el Nick, y luego la contraseña
- Informa al usuario si ha habido un error al iniciar sesión, y si no lo ha habido le lleva al menú pertinente

### Jugador

#### Menú Jugador

- Es el menú principal del jugador, que muestra todas sus opciones.
- Opciones:
  - Consultar ranking
  - Cambiar equipo del personaje

- Hacer desafío
- Consultar historial/oro
- Crear personaje
- Borrar personaje
- Cerrar Sesión
- Junto a la opción de “hacer desafío” aparecerá el oro disponible del personaje (p. ej. Hacer desafío [600 de oro disponible]).

#### Menú Ranking

- Muestra el ranking, y te da la opción de volver al menú jugador.

#### Menú Equipo

- Muestra el arma(s) activa(s) (o "Ninguna" si no la hay)
- Muestra la armadura activa (o "Ninguna" si no la hay)
- Opciones:
  - Cambiar arma
  - Cambiar Armadura
  - Volver

#### Menú Desafío

- Permite al jugador desafiar a otros, apostando una cantidad de oro
- Pasos:
  1. Pregunta cantidad de oro a apostar (mostrando oro disponible)
  2. Pregunta *Nick* del usuario a desafiar (proporcionando una lista paginada de usuarios)
  3. Confirma que quieres enviar el desafío (mostrando un resumen del mismo)
  4. Muestra confirmación de envío
- Opción de volver disponible en cualquier momento

#### Menú Historial

- Muestra el historial de oro del usuario
- Pasos:
  1. Muestra n combates (por ejemplo, n = 5), mostrando el oro ganado o perdido en cada uno (los combates se muestran de manera paginada).
- Opciones:
  1. Siguiendo página
  2. Anterior página
  3. Volver al menú principal

#### Menú Crear Personaje

- Permitirá al usuario crear un personaje nuevo siempre y cuando no tenga ya un personaje creado
- Pasos:
  1. Pide nombre del personaje
  2. Pide el tipo de personaje (especie/raza)
  3. El sistema genera valores para las demás propiedades
- Opción de cancelar/volver disponible en todo momento

### Menú Borrar Personaje

- Permite al usuario borrar su personaje
- Pasos:
  1. Pregunta si el usuario está seguro de querer borrar su personaje, y en caso afirmativo, lo borra y devuelve al usuario al “Menú Jugador”.
- Opción cancelar/volver disponible en todo momento

### Administrador

#### Menú Administrador

- Es el menú principal del administrador que muestra todas sus opciones
- Opciones:
  - Editar personajes
  - Validar desafíos
  - Banear usuarios
  - Desbanear usuarios
  - Cerrar Sesión

#### Menú Selección Personaje

- Muestra todos los personajes de todos los jugadores, y permite seleccionar uno para ser editado
- Pasos:
  1. Se muestran n personajes (de forma paginada)
  2. Una vez el Administrador decide editar un personaje, se muestra el “Menú Características de Personaje” de ese personaje
- Opciones:
  - Editar un personaje según su posición
  - Siguiente página
  - Anterior página
  - Volver

#### Menú Características de Personaje

- Se mostrarán todas las características asociadas al personaje. Por ejemplo,
  - Nombre: Drácula
  - Habilidad especial: chupar sangre
  - Conjunto de armas activas: espada corta a una mano
  - ...
- Pasos:
  1. Pregunta qué campo desea modificar
  2. Si decide modificar un campo, se le pedirá introducir el nuevo valor de ese campo.
    - a. Si el campo toma valores de un rango concreto (por ejemplo, el arma activa tiene que ser un objeto contenido en la lista de armas del personaje, que a su vez tienen que ser objetos contenidos en el archivo de armas), entonces se le mostrará al usuario los valores posibles (de manera paginada)
  3. Una vez realizado el cambio, se volverá a mostrar el menú de características del personaje, pero actualizado

#### Menú Desafíos Pendientes

- Permite al administrador seleccionar un desafío para validarlo.
- Muestra una lista de los desafíos no validados (paginada)
- Opciones:
  - Validar desafío según su posición
  - Siguiente página
  - Anterior página
  - Volver

#### Menú Validación

- Permite al administrador validar un desafío concreto, y activar fortalezas/debilidades de los personajes. Primero se mostrarán las debilidades/fortalezas de un personaje, y a continuación las del otro.
- Muestra una lista de debilidades y fortalezas de los personajes
- Opciones:
  - Activar una debilidad/fortaleza según su posición
  - Volver

#### Menú Baneos

- Permite al administrador banear uno o varios usuarios no baneados
- Muestra una lista de los usuarios no baneados (paginada)
- Opciones:
  - Banear a un usuario según su posición
  - Siguiente página
  - Anterior página
  - Volver

#### Menú Quitar Baneos

- Permite al administrador quitar el ban de uno o varios usuarios baneados
- Muestra una lista de los usuarios baneados (paginada)
- Opciones:
  - Desbanear a un usuario según su posición
  - Siguiente página
  - Anterior página
  - Volver