



# DOCUMENTO FUNCIONAL DE AZERIS

ALUMNO/GRUPO: Dylan Castro & Mario Taipe

## 1. Introducción y contexto

El proyecto consiste en un videojuego con el nombre **Azeris** del género Roguelike (subgénero de los videojuegos que se caracterizan por la exploración de mazmorras/niveles de forma procedural) en el que se incluye el guardado del progreso, una sección donde puedas comparar con otras personas la máxima puntuación posible en un tiempo limitado y ser accesible para todo tipo de plataformas. La meta de este proyecto es llevar a Azeris a su límite en torno a diversión y entretenimiento.

Asimismo, contempla la creación de una base de datos para almacenar registros de usuario y partidas guardadas, y una página web informativa que ofrece al jugador una mejor experiencia a la hora de encontrar más información en el juego.

Azeris será exclusivo para computadoras, con la capacidad de descargarlo en sistemas operativo Windows y Linux, y programada en Java utilizando el framework de LibGDX. El objetivo final es obtener conocimientos adicionales sobre el desarrollo de videojuegos y consolidar nuestros conocimientos que hemos adquirido durante el ciclo.



Icono principal de Azeris

## 2. Análisis de requisitos

### 2.1. Requisitos funcionales (RF)

Para la página web:

Código	Descripción del requisito funcional
RF1	La página web permitirá registrar nuevos usuarios.
RF2	Los usuarios podrán iniciar sesión, gestionar su perfil y juegos guardados.
RF3	Los jugadores podrán visualizar y comparar sus puntuaciones con las de otros usuarios en una tabla de clasificación.
RF4	El usuario podrá cerrar sesión de forma segura y eliminar su cuenta si lo desea.
RF5	Cualquiera que visite la página web oficial podrá descargar fácilmente Azeris en su sistema Windows o Linux.

Los requisitos funcionales para Azeris serán:

RF1	Azeris generará niveles de forma procedural para ofrecer niveles únicos en cada ejecución.
RF2	Azeris debe poder guardar el progreso de varias partidas y conservar sus estados originales.
RF3	Azeris permitirá a los jugadores subir de nivel mediante experiencia que se obtiene derrotando a enemigos.
RF4	Los jugadores podrán obtener diversos ítems, armaduras y armas a lo largo del juego derrotando enemigos o explorando los niveles.
RF5	Después de cada muerte/pérdida los jugadores disfrutarán de distintas mejoras y beneficios incrementales al empezar futuras partidas.
RF6	Azeris permitirá escoger entre tres tipos de personajes al comenzar la partida.
RF7	Mientras se va progresando en Azeris, se irá otorgando puntos al jugador conforme haga ciertas acciones en la partida. Por ejemplo: Matar enemigos, adquiriendo monedas de oro y ítems, ganando experiencia, subiendo de nivel, pasando de una etapa de la mazmorra a otra (teniendo en cuenta el tiempo que se tarda en hacerlo), etc. La puntuación total al final de la partida quedará guardada.

## 2.2. Requisitos no funcionales (RNF)

Código	Descripción del requisito no funcional
RNF1	Azeris tendrá un número mínimo de errores de programación que causen actitudes o resultado inesperados, como mínimo que no afecte a la habilidad del jugador para disfrutar del juego o interferir con la puntuación y progreso guardado.
RNF2	Azeris estará disponible para dispositivos Windows y Linux.
RNF3	La página oficial de Azeris debe poder ser fácilmente accesible en ordenadores para poder ver estadísticas y descargar el juego en sí.
RNF4	Azeris mantendrá un nivel aceptable de FPS.

## 2.3. Restricciones

- Lenguajes o tecnologías obligatorias:
  - **Java:** Lenguaje de programación para el desarrollo del juego.
  - **IntelliJ:** Entorno de Desarrollo para programar la lógica de Azeris.
  - **LibGDX:** Framework basado en Java que se utilizara junto con IntelliJ.
  - **Postgres:** Lenguaje de base de datos, Supabase proporcionará el backend para la gestión de usuarios y progresos.
  - **Vercel:** Plataforma para desplegar la página web del proyecto.
  - **Firestore Authentication:** para el sistema de login seguro.
  - **Krita:** Editor gráfico para crear personajes, mapas y animaciones en 2D.
- Recursos disponibles (tiempo, equipo, materiales):
  - **Tiempo:** Duración del proyecto limitada al calendario estudiantil.
  - **Equipo:** 2 personas participando en el proyecto.
  - **Materiales:** Ordenadores con sistemas operativos Windows, Linux, con acceso a Internet y software de desarrollo.
- Dependencias o limitaciones técnicas:
  - El juego debe ser compatible con PC Windows, Linux.
  - La página web depende de servicios externos (Vercel, Supabase) para gestión de datos y autenticación.
  - La generación procedural de niveles estará limitada por la capacidad de procesamiento del hardware de los jugadores.
  - Se requiere conexión a Internet para sincronización de datos y rankings online.

## 3. Análisis de usuarios y roles

<b>Rol</b>	<b>Descripción</b>	<b>Permisos principales</b>
Administrador	Gestionar cuentas, estadísticas y progreso guardado de usuarios en la página web oficial de Azeris.	Administración de cuentas (Borrar y creación) y edición de estadísticas.
Usuario	<p>Con una cuenta de Azeris creada, puede acceder a las estadísticas de otros jugadores actualizadas en tiempo real.</p> <p>Pueden descargar y jugar a Azeris.</p>	Creación de cuenta, descarga del juego, acceso a las estadísticas.
Visitante	<p>Sin cuenta de Azeris creada se podrá visualizar la página web oficial de Azeris, no se puede acceder a funciones exclusivas a usuarios.</p> <p>Pueden descargar y jugar a Azeris.</p>	Creación de cuenta, descarga del juego.

## 4. Casos de uso / Escenarios de uso

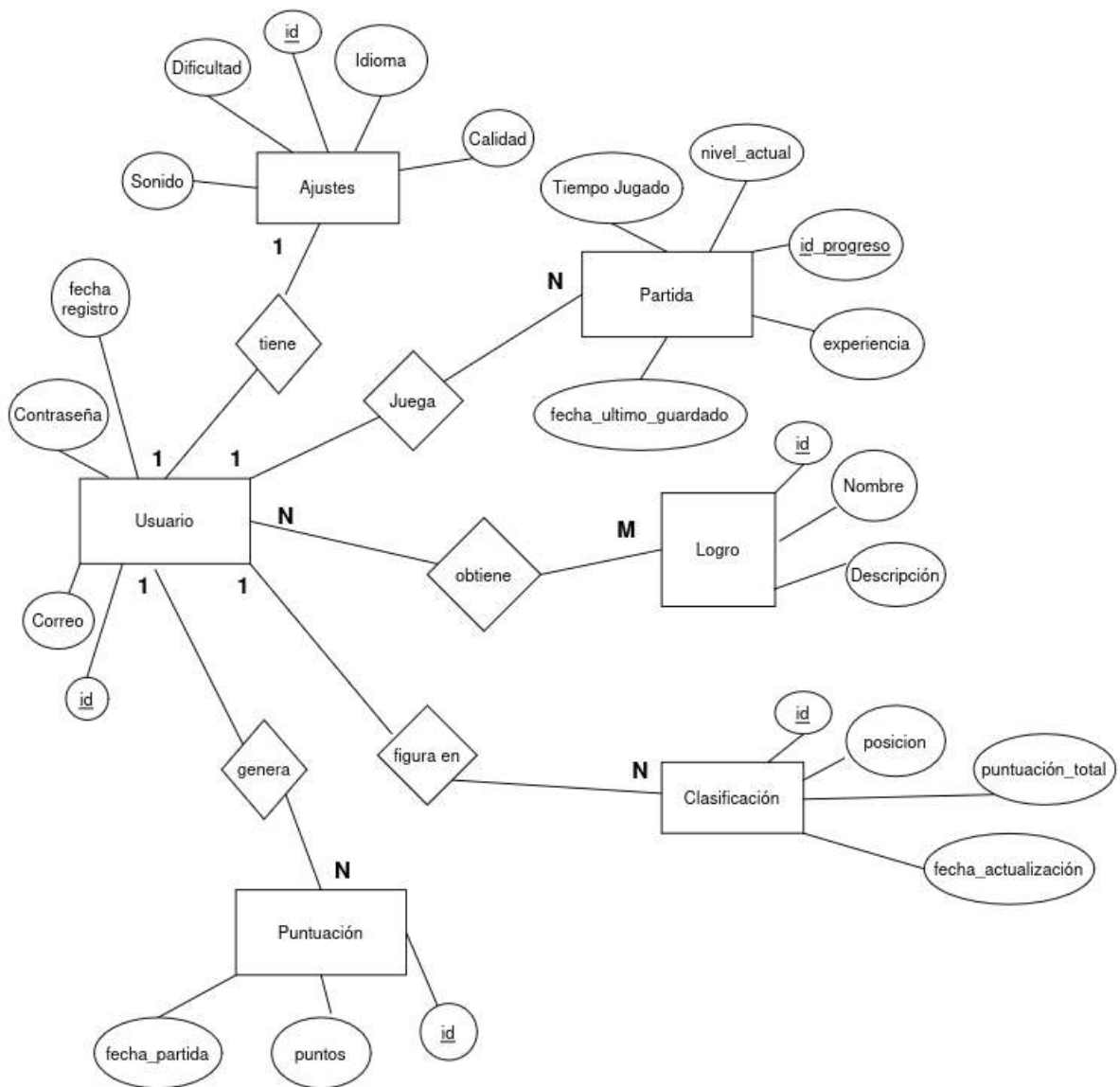
### *Página Web Oficial Azeris:*

<b>Código</b>	<b>Nombre del caso de uso</b>	<b>Actor principal</b>	<b>Descripción</b>	<b>Resultado esperado</b>
CU1	Registrar usuario	Visitante	El usuario introduce su correo electrónico y contraseña para crear una nueva cuenta en el sistema.	Usuario registrado correctamente.
CU2	Iniciar sesión	Usuario	El jugador introduce sus credenciales para acceder a su cuenta.	Acceso concedido.
CU3	Consultar clasificación	Usuario	El jugador accede a la sección de rankings para ver su posición y la de otros jugadores.	Mostrar las puntuaciones y posiciones actualizadas.
CU4	Cerrar sesión	Usuario	El jugador selecciona la opción de cerrar sesión en la página web.	Sesión cerrada
CU5	Descarga de Azeris	Visitante, Usuario	El usuario/visitante descarga Azeris mediante un botón en la página web.	Descarga del software Azeris (ver. Windows/Linux)

**Azeris:**

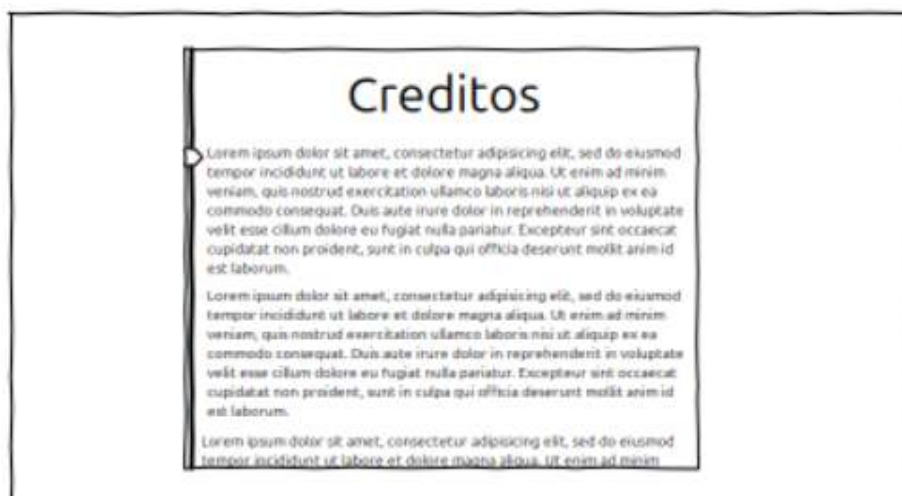
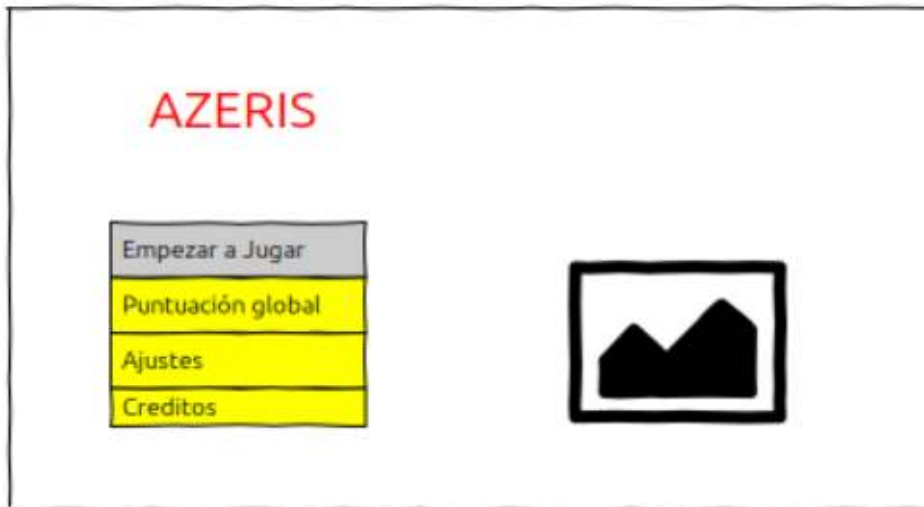
CU1	Acceso a sesiones de juego guardadas	Usuario	Un usuario accede a sus partidas guardadas Azeris mediante el uso de su cuenta y contraseña.
CU2	Cierre del Azeris con partida sin cuenta	Visitante	Un visitante jugando Azeris cierra la aplicación sin terminar la partida actual nueva.
CU3	Cierre del Azeris con partida con cuenta	Usuario	Un usuario jugando Azeris cierra la aplicación sin terminar la partida actual nueva.
CU4	Acabar la partida derrotando al jefe final	Usuario, Visitante	El jugador derrota al jefe final de Azeris y gana el juego.

## 5. Modelo de datos o estructura de la información




## 6. Diseño de la interfaz

Juego:



Página web vinculada al juego:



#	Jugador	Puntuación		Tiempo	Fecha	Verificado
1	Peter	41356	<a href="#">ir</a>	1h 02m 33s	hace 4 meses	<input checked="" type="checkbox"/>
2	Henni	40911		35m 21s	hace 1 año	<input type="checkbox"/>
3	Olga	39910	<a href="#">ir</a>	52m 49s	hace 11 meses	<input checked="" type="checkbox"/>
4	Velma	39890		51m 12s	hace 2 semanas	<input type="checkbox"/>
5	Super	38884		52m 45s	hace 5 meses	<input checked="" type="checkbox"/>

La pantalla principal consiste en mostrar el nombre del juego acompañado de diversas opciones que el jugador podrá seleccionar, tales como: iniciar una nueva partida, consultar las puntuaciones obtenidas por los jugadores en el “modo libre”, acceder a la configuración de controles, sonidos y más ajustes y los debidos créditos. Además, se incluirá una imagen representativa del personaje principal en dicha pantalla.

## 7. Planificación técnica

*Objetivo:* planificar el desarrollo del proyecto.\*

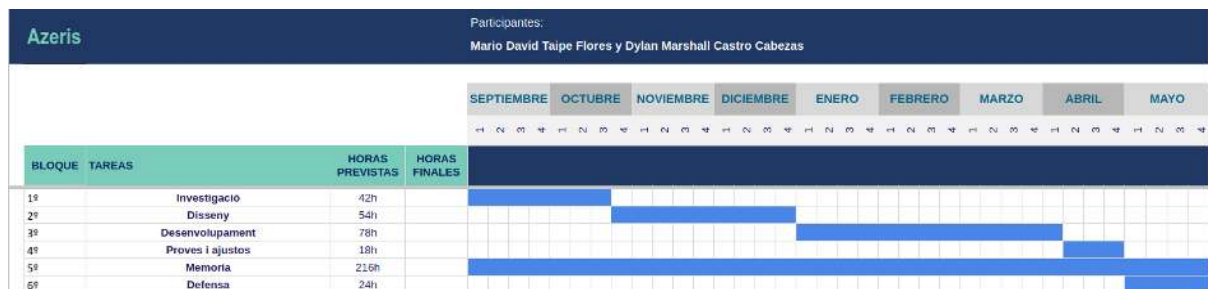
Indica las tecnologías y herramientas que se utilizarán, y cómo se organizará el trabajo.

- **Lenguajes:** Java, PostgreSQL, HTML y XML
- **Servicios:** SupaBase (administracion de BBDD), Vercel (administracion de paginas web)
- **IDEs y Frameworks:** IntelliJ (IDE) y LibGDX (Java Framework)
- **Herramientas de diseño o edición:** Krita (Diseño de Pantallas y UIs), Modelio (Diseño de esquemas)
- **Reparto de tareas:**

**Mario Taipe** - Diseño de interfaces, recursos visuales, página web e implementación de BBDD.

**Dylan Castro** - Implementación de logica y codigo fuente de Azeris.

**Cronograma:**



## 8. Análisis de riesgos

### 8.1. Identificación de riesgos

- **Falta de experiencia técnica:** Al ser principiantes, pueden encontrarse con dificultades técnicas como errores de programación, mal uso de herramientas o elección inadecuada del motor de juego.
- **Mala gestión del tiempo:** Subestimar el tiempo necesario para cada fase del proyecto (diseño, programación, arte, pruebas, etc.).
- **Problemas de comunicación o coordinación:** Falta de claridad en la distribución de tareas o en la visión del juego.
- **Desmotivación o abandono del proyecto:** La carga de trabajo, frustraciones técnicas o falta de progreso visible pueden afectar la motivación.
- **Alcance del proyecto demasiado ambicioso:** Intentar hacer un juego demasiado complejo para el nivel y los recursos disponibles.

### 8.2. Valoración y respuesta

Clasifica cada riesgo según su probabilidad e impacto, e indica cómo se mitigará.

Riesgo	Probabilidad	Impacto	Estrategia de mitigación
<b>Falta de experiencia técnica</b>	Alta	Medio	Aprender con tutoriales básicos al inicio acerca de LibGDX, y hacer un prototipo pequeño primero.
<b>Mala gestión del tiempo</b>	Media	Alta	Crear un plan de trabajo con metas mensuales y semanales, y usar herramientas como Trello o Taiga.io para organizar tareas.
<b>Problemas de comunicación</b>	Media	Media	Establecer reuniones regulares (por ejemplo, semanales) para revisar el progreso y mantener una comunicación clara y constante.
<b>Desmotivación o abandono</b>	Media	Alta	Dividir el proyecto en partes pequeñas y alcanzables, celebrar cada logro, y mantener la motivación con objetivos claros y visibles.
<b>Alcance demasiado ambicioso</b>	Alta	Media	Definir claramente el alcance desde el principio (lo que sí y lo que no tendrá el

			juego) y hacer una versión mínima viable antes de agregar funciones adicionales.
--	--	--	----------------------------------------------------------------------------------

## 9. Validación y criterios de éxito

El proyecto de Azeris se dará por finalizado cuando:

1. Se acabe el desarrollo base de Azeris, es decir: Implementación de pantallas (Menú, Configuración, Pausa, Inventario), generación procedural de niveles que permite crear mapas y escenarios dinámicos de manera automática, implementación de enemigos, ítems básicos, mecanismos de salud y mejoras para el jugador, creación de base de datos para guardar información importante (progreso de partidas y puntuación). Además, que Azeris sea disponible para usuarios de Windows/Linux.
2. Se finalice el desarrollo de la página web oficial de Azeris, que incluirá: Área para registrarse y iniciar sesión con una cuenta de Azeris mediante un correo electrónico, botón para descargar Azeris, área para comparar y visualizar estadísticas de las puntuaciones de otros usuarios (solo accesible con cuenta).
3. Se compruebe que las funciones básicas y mínimas del proyecto acabado funcione correctamente en todos los casos. Para esto, se planea hacer tests extensivos al finalizar la lógica o implementación de funciones antes de dar lo por acabado.

## 10. Conclusión

Durante el análisis funcional del proyecto Azeris se han tomado decisiones clave que permiten definir el alcance y las bases para la siguiente fase de desarrollo:

- **Funciones que tendrá el sistema:**

1. Registro e inicio de sesión de usuarios.
2. Gestión y almacenamiento del progreso del jugador.
3. Generación procedural de niveles y escenarios dinámicos.
4. Consulta de clasificaciones y estadísticas de jugadores.
5. Página web informativa y funcional para descarga del juego y acceso a rankings.

- **Tecnologías seleccionadas:**

1. **Java y LibGDX** para el desarrollo del videojuego.
2. **Supabase** para la base de datos y gestión de usuarios.
3. **Vercel** para el despliegue de la página web.
4. **Firebase Authentication** para el sistema de login seguro.
5. **Krita** para la creación de recursos gráficos y animaciones 2D.

- **Valor aportado por el proyecto:**

Este proyecto mezcla entretenimiento y aprendizaje, ofreciendo una experiencia de juego única con sus niveles generados proceduralmente, promoviendo la competición entre jugadores y proporcionando conocimientos en programación, bases de datos y desarrollo web.

- **Próximos pasos:**

1. Preparar el entorno de desarrollo y framework completo para el equipo.
2. Crear la estructura inicial del repositorio y de la base de datos.
3. Comenzar la implementación de los casos de uso prioritarios (registro, login y guardado de progreso).
4. Empezar con los sprites del personaje.

### **Reflexión final:**

Gracias a este análisis, contamos con una guía estructurada que facilitará la implementación eficiente de Azeris.