



# Tecnológico de Monterrey

## Construcción de software y toma de decisiones

Gilberto Echeverría Furió  
Octavio Navarro Hinojosa  
Dr. Esteban Castillo Juarez

**Equipo: Legends**  
Gabriel Rodríguez De Los Reyes  
Iker García Germán  
Rodrigo Núñez Magallanes  
Enrique Cabrera Aguilar  
Alejandro Arouesty Galván  
Pablo Banzo Prida

27/03/2023

# Índice

<b>User Stories</b>	<b>3</b>
<b>Product Backlog</b>	<b>3</b>
<b>Descripciones de casos de uso y requerimientos</b>	<b>6</b>
Categorización y descripción de requerimientos (WEB, DB, Unity)	6
Web (WEB)	6
Unity/Video Games (U)	6
Database (DB)	7
Web/DB/Unity (G)	7
Web/DB (WDB)	7
Unity/DB (UDB)	7
Casos de Uso	7
<b>Sprint 1 Backlog</b>	<b>13</b>
Sprint Information (Trello)	14
<b>Tablas Descriptivas de Requerimientos</b>	<b>14</b>
W1. Create a webpage to embed the project (Functional, WEB)	15
W2. Embed project on the main page (Functional, WEB)	16
W3. Add attractive images of the game to the website (Non-functional, WEB)	17
W4. Define the color palette of the website (Non-functional, WEB)	18
W5. Create a secondary page within the website with the game manual (Non-functional, WEB)	19
DB1. Create the entity-relationship diagram of the database (Non-functional, DB)	20
DB2. Implement the database and its tables (Functional, DB)	21
DB3. Create a stats table and link it with the user (Functional, DB)	22
DB4. Implement CRUD in the database (Functional, DB)	23
U1. Create a project in Unity (Functional, UNITY/Video games)	24
U2. Define and integrate Sound Effects (Non-functional, UNITY/Video games)	25
U3. Compose and implement Music (Non-functional, UNITY/Video games)	26
U4. Develop Prefabs (Functional, UNITY/Video games)	27
U5. Create Scenery (Non-functional, UNITY/Video games)	28
U6. Implement game mechanics (Functional, UNITY/Video games)	29
G1. UML General of the entire system (Non-functional, DB, UNITY/Video games, WEB)	30
G2. Create use case tables (Non-functional, WEB, UNITY/Video games, DB)	31
WDB1. Create authentication system logic (Functional, WEB, DB)	32
WDB2. Create secondary webpage with stats linked to the database (Functional, WEB, DB)	33
WDB3. Define host for the webpage and database (Functional, WEB, DB)	34
UDB1. Connect database with the game (Functional, UNITY/Video games, DB)	35
<b>Anexo: Diagramas UML adicionales</b>	<b>36</b>

# User Stories

1. Como usuario quiero poder entrar a cada dungeon.
2. Como usuario quiero ver un instructivo del juego.
3. Como usuario quiero ver las estadísticas de otros jugadores y que se guarden las mías.
4. Como usuario quiero poder dialogar con los NPCs.
5. Como usuario quiero ver a los NPCs interactuar entre ellos.
6. Como usuario quiero poder usar mi pico-hacha que tengo por default.
7. Como usuario quiero poder recoger el cuchillo y usarlo.
8. Como usuario quiero poder usar lanzas si las tengo.
9. Como usuario quiero usar la serbatana si es que la tengo.
10. Como usuario quiero escuchar mis pasos en el overworld y en los dungeons.
11. Como usuario quiero poder golpear madera (árboles) y piedra, escuchar su sonido y recoger lo que obtenga por eso.
12. Como usuario quiero escuchar música en el main, en el dungeon y en el boss fight.
13. Como usuario quiero que el BOSS me ataque y poder defenderme y atacar de regreso.
14. Como usuario quiero iniciar sesión para poder guardar mi progreso en el juego.
15. Como usuario quiero guardar mi progreso en el juego.
16. Como usuario quiero poder insertar, eliminar y modificar información de la base de datos.
17. Como usuario quiero ver estadísticas del juego.
18. Como usuario quiero ver un modelo conceptual de la base de datos.

# Product Backlog

- W1. Crear página web para embeber el proyecto (Req Funcional, WEB)
- W2. Empotrar proyecto en main page (Req Funcional, WEB)
- W3. Agregar imágenes atractivas del juego en el sitio web (Req no Funcional, WEB)
- W4. Definir la gama de colores de la página web (Req no Funcional, WEB)
- W5. Página secundaria dentro de la página web con el manual de juego (Req no Funcional, WEB)
- DB1. Crear el diagrama entidad relación de la base de datos (Req no Funcional, DB)
- DB2. Implementar la base de datos y sus tablas (Req Funcional, DB)
- DB3. Crear tabla de stats y ligarlas con el usuario (Req Funcional, DB)
- DB4. Implementar CRUD en la base de datos (Req Funcional, DB)
- U1. Crear proyecto en Unity (Req Funcional, UNITY/Videojuegos)
- U2. Definir e integrar Sound Effects (Req no Funcional, UNITY/Videojuegos)
- U3. Componer e implementar Música (Req no Funcional, UNITY/Videojuegos)
- U4. Desarrollar Prefabs (Req Funcional, UNITY/Videojuegos)
- U5. Elaboración de fondos (scenery) (Req no Funcional, UNITY/Videojuegos)

- U6. Implementación de Mecánicas de Juego (Req Funcional, UNITY/Videojuegos)
- G1. UML General de todo el sistema (Req no Funcional, DB, UNITY/Videojuegos, WEB)
- G2. Hacer las tablas de casos de uso (Req no Funcional, WEB, UNITY/Videojuegos, DB)
- WDB1. Crear lógica de sistema de identificación (auth) (Req Funcional, WEB, DB)
- WDB2. Página secundaria dentro de la página web para stats con la tabla ligada a la base de datos (Req Funcional, WEB, DB)
- WDB3. Definir host para la página y la base de datos (Req Funcional, WEB, DB)
- UDB1. Conectar database con juego (Req Funcional, UNITY/Videojuegos, DB)

The image shows a Trello board with two main columns:

- Historia de usuario (15)**: Contains 18 cards. Each card has a title and a description. Below each title are three colored boxes indicating requirements: WEB (orange), UNITY/Videojuegos (red), and DB (blue).
- Product backlog (21)**: Contains 21 cards. Each card has a title and a description. Below each title are three colored boxes indicating requirements: Req no Funcional (purple), Req Funcional (green), and DB (blue).

Some examples from the lists:

- Historia de usuario (15) Cards:**
  - Como usuario quiero iniciar sesión para poder guardar mi progreso en el juego.
  - Como usuario quiero guardar mi progreso en el juego.
  - Como usuario quiero ver las estadísticas de otros jugadores.
  - Como usuario quiero poder dialogar con los NPCs
  - Como usuario quiero ver cómo interactúan entre ellos los NPCs
  - Como usuario quiero ver estadísticas del juego.
  - Como usuario quiero poder insertar.
- Product backlog (21) Cards:**
  - DB1. Crear el diagrama entidad relación de la base de datos
  - DB2. Implementar la base de datos y sus tablas
  - DB3. Crear tabla de stats y ligarla con el usuario
  - UDB1. Conectar database con juego
  - G1. UML General de todo el sistema
  - WDB1. Crear lógica de sistema de identificación (auth)
  - WDB2. Página secundaria dentro de la página web para stats con la tabla ligada a la base de datos

Imagen 1: Implementación de Backlog e Historias en Trello

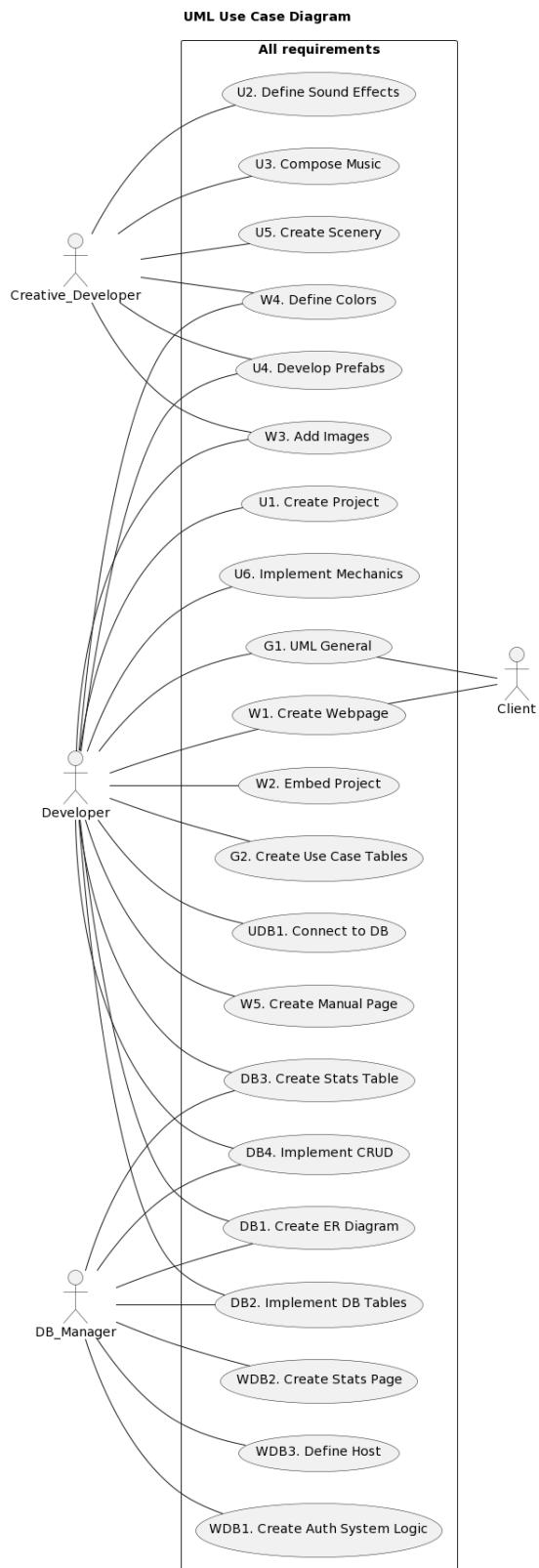


Figura 1. UML de todos los requerimientos

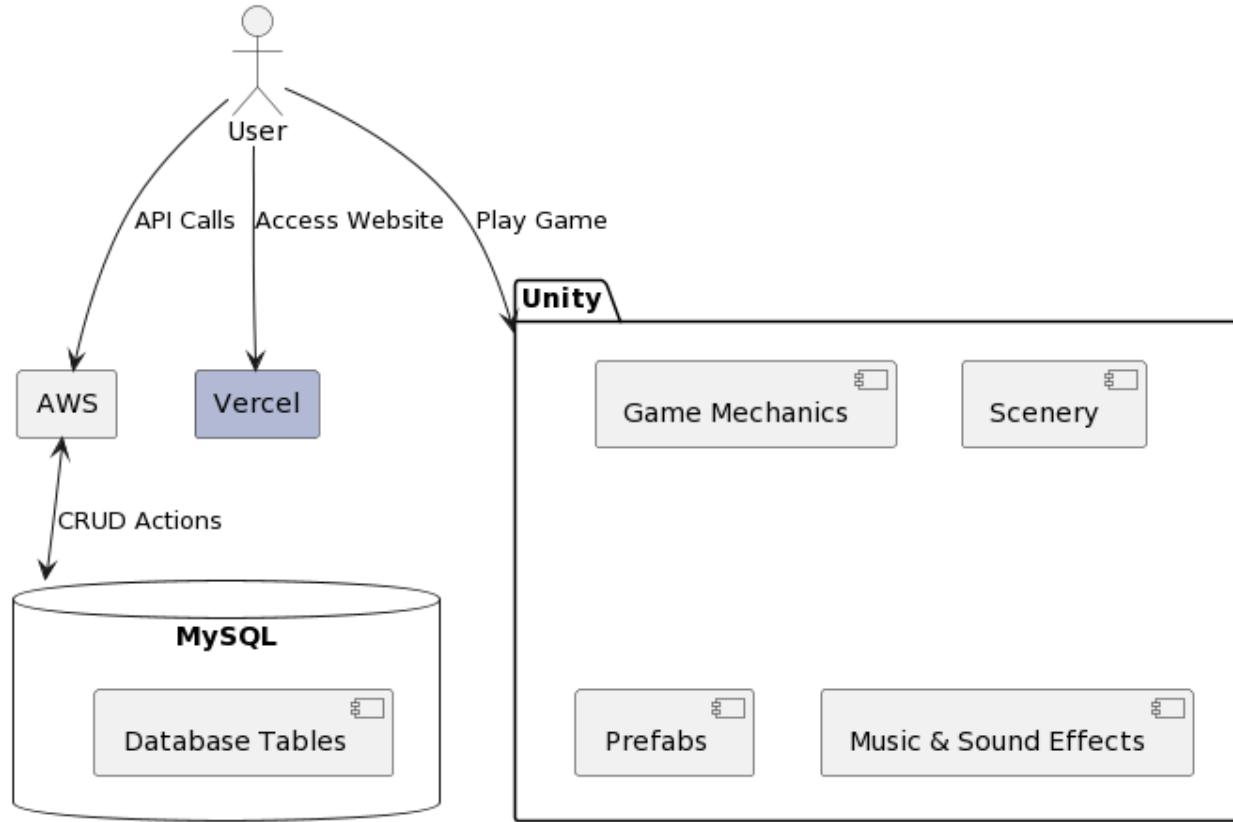


Figura 2. UML de Infraestructura

## Descripciones de casos de uso y requerimientos

Categorización y descripción de requerimientos (WEB, DB, Unity)

### Web (WEB)

- Create Webpage: Create a webpage to showcase the game.
- Embed Project: Embed the game on the main page of the website.
- Add Images: Add attractive images of the game to the website.
- Define Colors: Define the color palette of the website.
- Create Manual Page: Create a secondary page within the website with the game manual.

### Unity/Video Games (U)

- Create Project: Create a project in Unity for the game.
- Define Sound Effects: Define and integrate sound effects into the game.
- Compose Music: Compose and implement music into the game.

- Develop Prefabs: Develop prefabs for game objects.
- Create Scenery: Create the game scenery.
- Implement Mechanics: Implement the game mechanics.

## Database (DB)

- Create ER Diagram: Create the entity-relationship diagram of the database.
- Implement DB Tables: Implement the database and its tables.
- Create Stats Table: Create a stats table and link it with the user.
- Implement CRUD: Implement CRUD (Create, Read, Update, Delete) in the database.

## Web/DB/Unity (G)

- UML General: Create a UML general of the entire system.
- Create Use Case Tables: Create use case tables for the web, database, and Unity.

## Web/DB (WDB)

- Create Auth System Logic: Create authentication system logic.
- Create Stats Page: Create a secondary webpage with the user's stats linked to the database.
- Define Host: Define the host for the web page and database.

## Unity/DB (UDB)

- Connect to DB: Connect the game to the database.

## Casos de Uso

1. **Registrarse:** Este caso de uso describe el proceso que un usuario debe seguir para crear una cuenta en el sistema. Incluye el ingreso de información personal, como nombre de usuario, correo electrónico y contraseña, y la validación de los datos ingresados.
  - a. Requerimientos asociados: WDB1, W1, W2

### UML Use Case Diagram

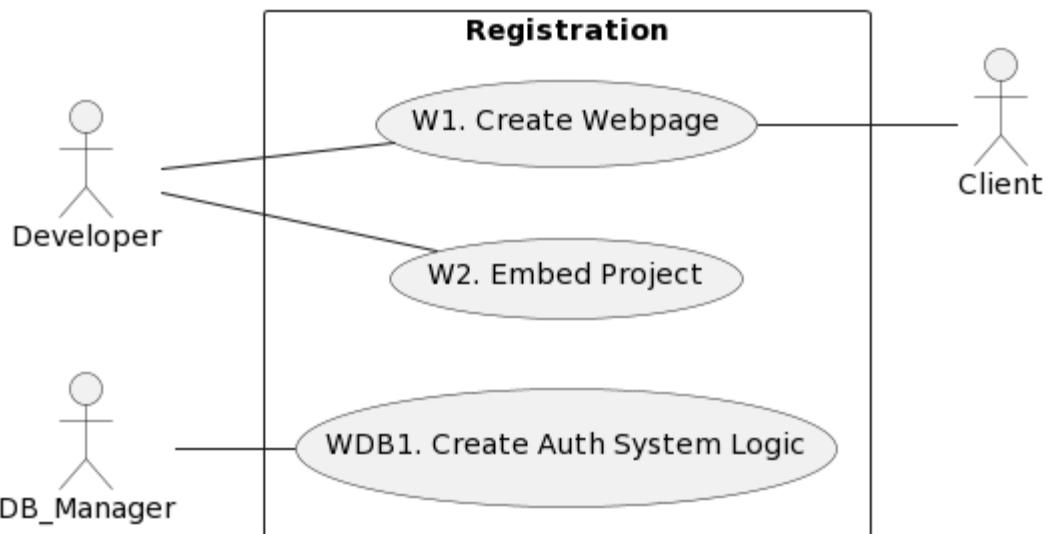


Figura 3. caso de uso UML - Registro

2. **Iniciar sesión:** Este caso de uso describe el proceso que un usuario debe seguir para acceder al sistema con su cuenta previamente creada. Incluye el ingreso de su nombre de usuario y contraseña, y la validación de la información para permitir el acceso al sistema.
  - a. Requerimientos asociados: WDB1, W1, W2

### UML Use Case Diagram

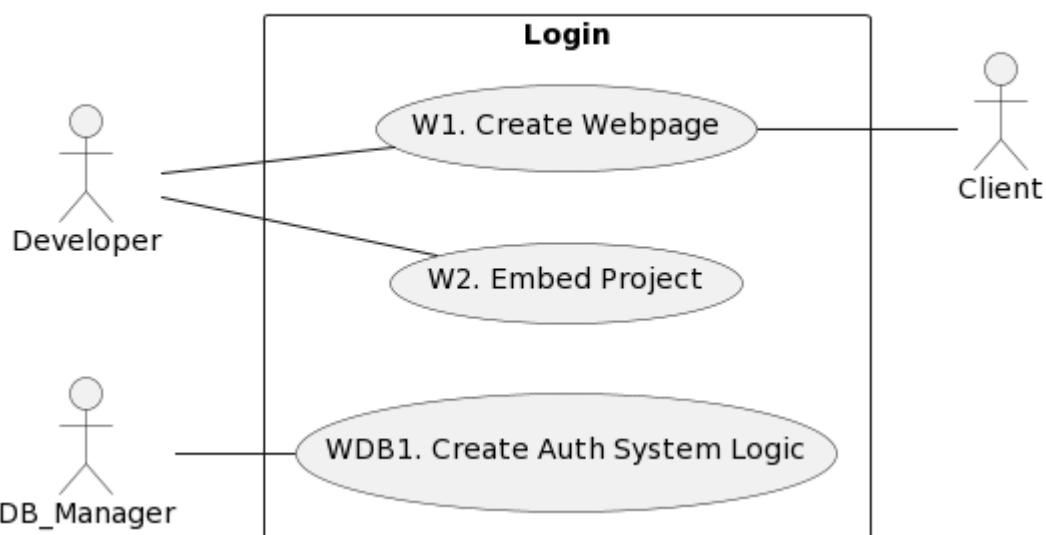


Figura 4. caso de uso UML - Login

3. **Consultar las estadísticas del juego:** Este caso de uso describe la funcionalidad que permite a un usuario ver sus estadísticas de juego, como el número de partidas jugadas, victorias, derrotas y puntaje total.

a. Requerimientos asociados: WDB1, WDB2, W1, W2

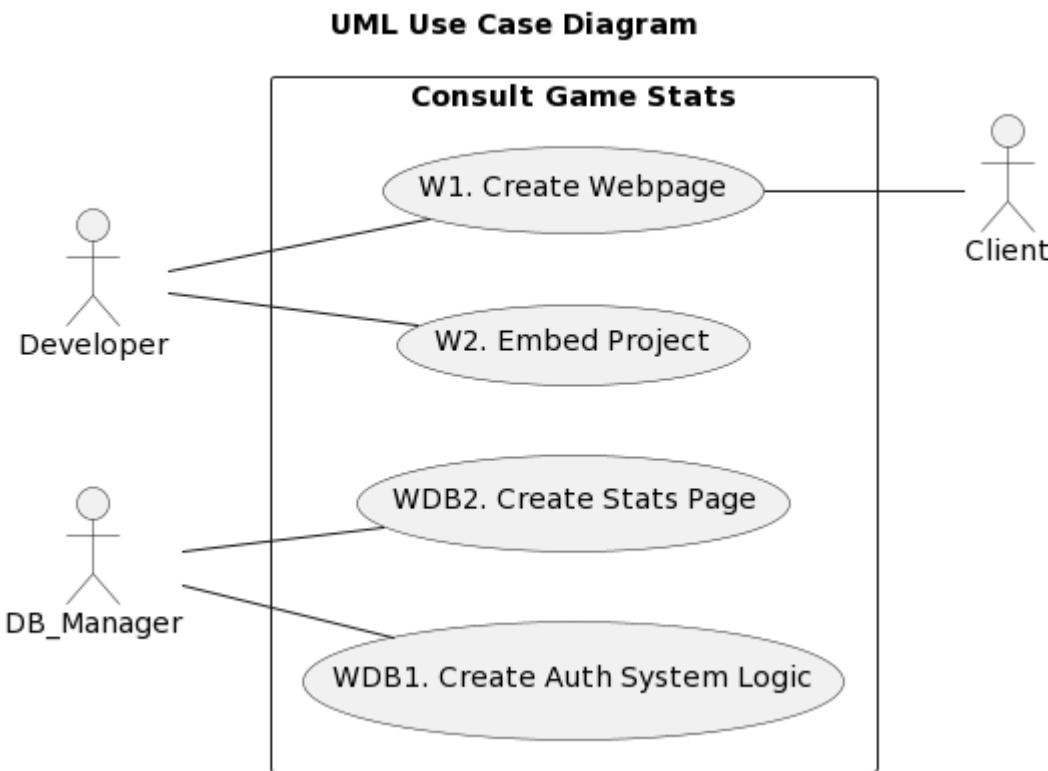


Figura 5. Caso de uso UML - Consultar estadísticas de Juego

4. **Consultar instructivo del juego:** Este caso de uso describe la funcionalidad que permite a un usuario acceder al manual de juego para conocer las reglas y mecánicas del mismo.

a. Requerimientos asociados: W5, WDB1, W1, W2

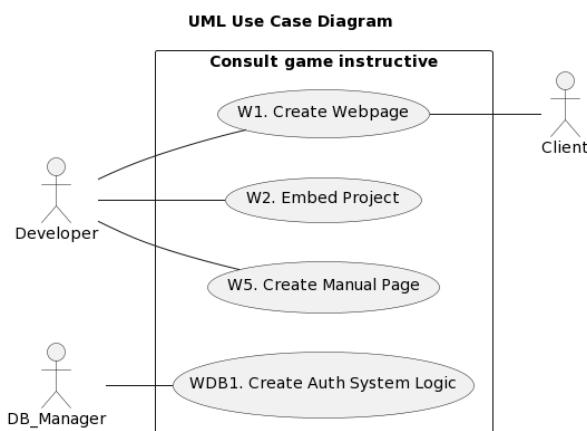


Figura 6. Caso de uso UML - Consultar instructivo de Juego

5. **Jugar el juego:** Este caso de uso describe el proceso que un usuario sigue para jugar el juego en sí. Incluye la interacción del usuario con el juego para completar los objetivos del mismo.
  - a. Requerimientos asociados: U1, U2, U3, U4, U5, U6, UDB1, W1, W2

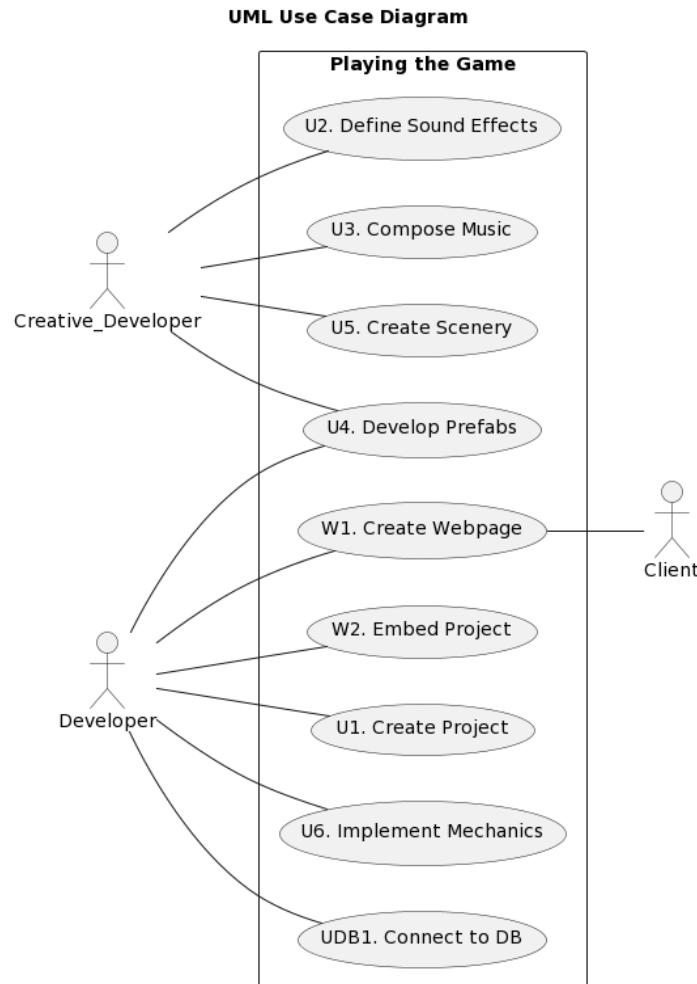


Figura 5. Caso de uso UML - Jugar el Juego

6. **Guardar progreso del jugador en el juego:** Este caso de uso describe la funcionalidad que permite a un usuario guardar su progreso en el juego, para poder continuar desde donde lo dejó en una próxima sesión.
  - a. Requerimientos asociados: UDB1, DB2, DB3, DB4, U6

### UML Use Case Diagram

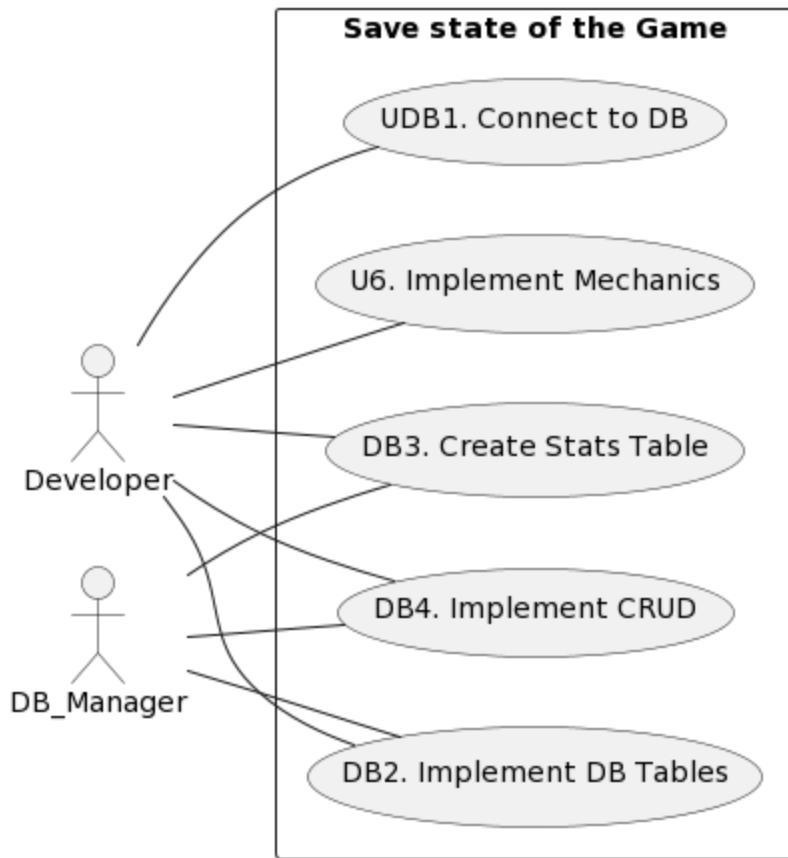


Figura 6. Caso de uso UML - Guardar progreso

7. **Cargar el progreso del jugador en el juego:** Este caso de uso describe la funcionalidad que permite a un usuario cargar su progreso previamente guardado en el juego.
  - a. Requerimientos asociados: UDB1, DB2, DB3, DB4, U6

### UML Use Case Diagram

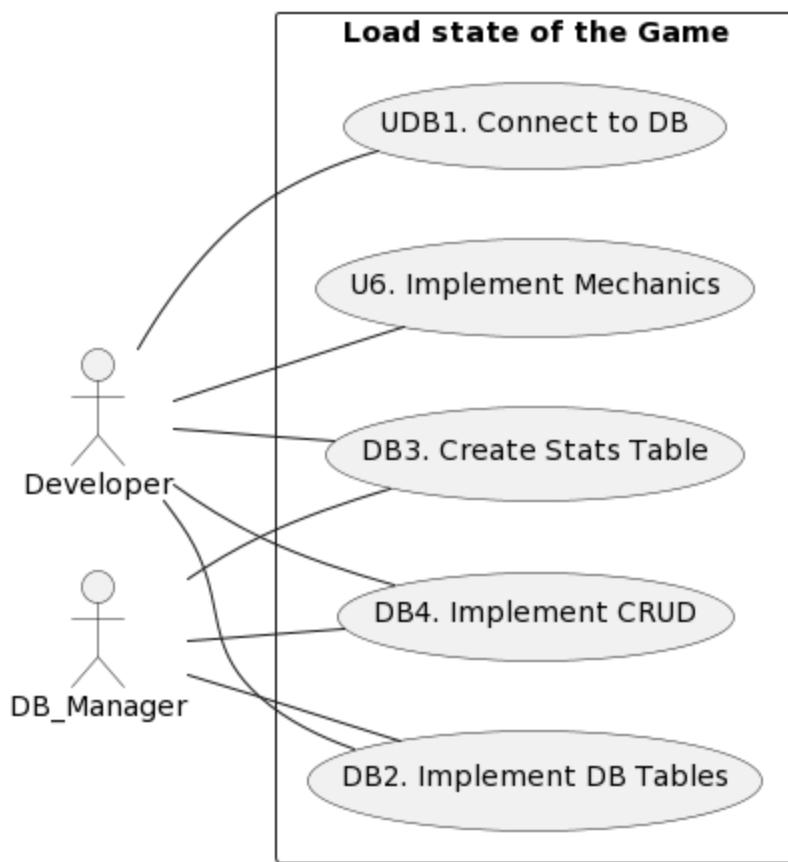


Figura 7. Caso de uso UML - Cargar progreso

8. **Cerrar sesión:** Este caso de uso describe el proceso que un usuario sigue para cerrar su sesión y salir del sistema. Incluye la eliminación de la sesión activa del usuario y la redirección a la página de inicio de sesión.
  - a. Requerimientos asociados: UDB1, DB2, DB3, DB4, W1, W2

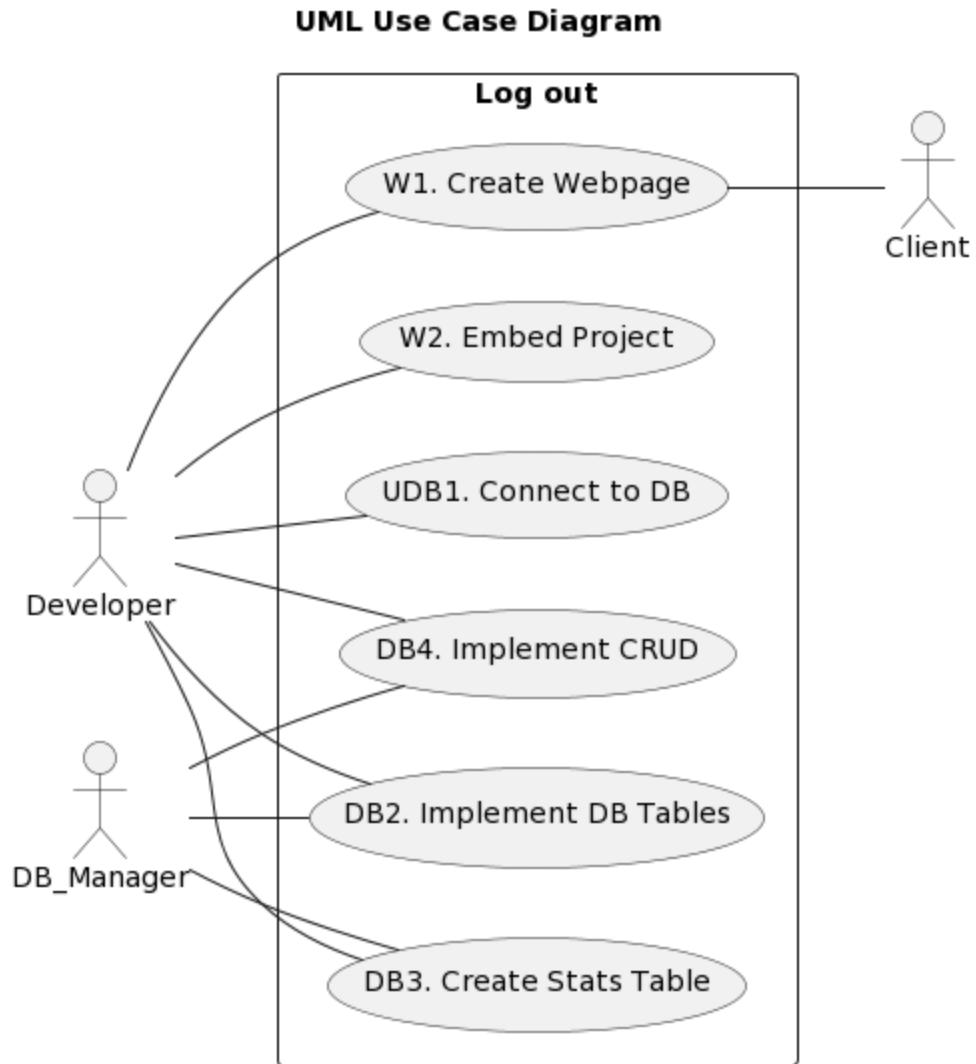


Figura 8. Caso de uso UML - Cerrar sesión.

## Sprint 1 Backlog

- W1. Crear página web para embedear el proyecto (Iker García Germán)
- G1. UML General de todo el sistema (Rodrigo Núñez Magallanes)
- DB1. Crear el diagrama entidad relación de la base de datos (Alejandro Arouesty Galván)
- WDB1. Crear lógica de sistema de identificación (auth) (E)
- G2. Hacer las tablas de casos de uso (Pablo Banzo Prida)
- U1. Crear proyecto en Unity (gabordlr)

## Sprint Information ([Trello](#))

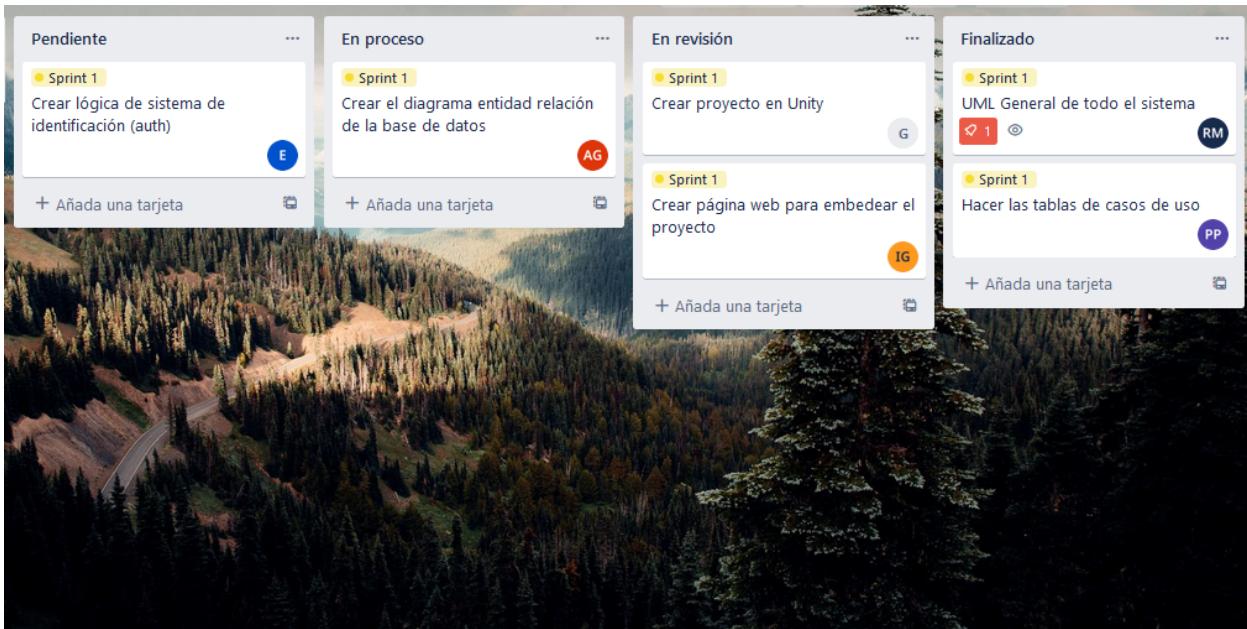


Imagen 2: Captura de Pantalla del Sprint 1

## Tablas Descriptivas de Requerimientos

## W1. Create a webpage to embed the project (Functional, WEB)

<b>Use case name</b>	<b>W1. Embed the project in a webpage</b>	
Related Requirements	W2, W3, WDB1	
Goal In Context	Design and implement a webpage to host the project and allow users to access the game	
Preconditions	Web development tools and web server configured	
Successful End Condition	Webpage created and project embedded successfully	
Failed End Condition	Webpage not created or project not embedded successfully	
Primary Actors	Web developers	
Secondary Actors	None	
Triggers	The decision to deploy the project on a website	
Main Flow	Step	Action
	1.	Web developer selects a suitable web development framework and creates a new webpage
	2.	Web developer integrates the project into the webpage using appropriate embedding techniques
	3.	Web developer tests the embedded project to ensure it functions as expected
	4.	Web developer deploys the webpage to the web server and configures necessary settings
Extensions	Step	Branching Action
	1a.	If the chosen web development framework is incompatible, the web developer selects a different framework and repeats steps 1-4
	3a.	If the embedded project does not function as expected, the web developer troubleshoots and resolves the issue, then repeats steps 3-4

## W2. Embed project on the main page (Functional, WEB)

<b>Use case name</b>	<b>W2. Embed the project on the main page</b>	
Related Requirements	W1, W3, WDB1	
Goal In Context	Ensure the project is accessible and visible on the main page of the website	
Preconditions	Webpage created	
Successful End Condition	Project embedded and visible on the main page	
Failed End Condition	Project not embedded or not visible on the main page	
Primary Actors	Web developers	
Secondary Actors	None	
Triggers	The decision to make the project accessible on the main page	
Main Flow	Step	Action
	1.	Web developer identifies the desired location for the project on the main page
	2.	Web developer embeds the project using appropriate embedding techniques
	3.	Web developer tests the embedded project on the main page to ensure it is visible and accessible
Extensions	Step	Branching Action
	2a.	If the project does not display correctly on the main page, the web developer troubleshoots and resolves the issue, then repeats steps 2-3

## W3. Add attractive images of the game to the website (Non-functional, WEB)

Use case name	W3. Add attractive images of the game on the website	
Related Requirements	W1, W2	
Goal In Context	Enhance the website's visual appeal with attractive images of the game	
Preconditions	Webpage created	
Successful End Condition	Attractive images of the game added to the website	
Failed End Condition	Images not added or not visually appealing	
Primary Actors	Web developers	
Secondary Actors	None	
Triggers	The decision to improve the website's visual appeal	
Main Flow	Step	Action
	1.	Web developer sources or creates attractive images of the game
	2.	Web developer adds the images to the website in appropriate locations
	3.	Web developer tests the website to ensure images are displayed correctly and are visually appealing
Extensions	Step	Branching Action

## W4. Define the color palette of the website (Non-functional, WEB)

<b>Use case name</b>	<b>W4. Define the color scheme of the webpage</b>	
Related Requirements	W1, W3	
Goal In Context	Establish a visually appealing and consistent color scheme for the website	
Preconditions	Webpage created	
Successful End Condition	Color scheme defined and applied to the website	
Failed End Condition	Color scheme not defined or not applied	
Primary Actors	Web developers	
Secondary Actors	None	
Triggers	The decision to create a visually appealing and consistent website	
Main Flow	Step	Action
	1.	Web developer researches and selects a suitable color scheme
	2.	Web developer applies the color scheme to the website's design elements
	3.	Web developer tests the website to ensure the color scheme is applied consistently and is visually appealing
Extensions	Step	Branching Action
	1a.	If the chosen color scheme is not visually appealing, the web developer selects a different color scheme and repeats steps 1-3

## W5. Create a secondary page within the website with the game manual (Non-functional, WEB)

<b>Use case name</b>	<b>W5. Secondary page within the website with the game manual</b>	
Related Requirements	W1, W4	
Goal In Context	Provide a dedicated page on the website with instructions on how to play the game	
Preconditions	Webpage created	
Successful End Condition	Game manual page created and accessible from the main page	
Failed End Condition	Game manual page not created or not accessible	
Primary Actors	Web developers	
Secondary Actors	None	
Triggers	The decision to provide instructions for the game on the website	
Main Flow	Step	Action
	1.	Web developer creates a new secondary page for the game manual
	2.	Web developer adds the game manual content to the secondary page
	3.	Web developer links the game manual page to the main page

## DB1. Create the entity-relationship diagram of the database (Non-functional, DB)

<b>Use case name</b>	<b>DB1. Create the entity-relationship diagram of the database</b>	
Related Requirements	DB2, DB3, DB4, WDB1	
Goal In Context	Design the database structure by creating an entity-relationship diagram	
Preconditions	Knowledge of the project's data requirements	
Successful End Condition	Entity-relationship diagram created and approved	
Failed End Condition	Entity-relationship diagram not created or not approved	
Primary Actors	Database developers	
Secondary Actors	Stakeholders	
Triggers	The decision to design the database structure	
Main Flow	Step	Action
	1.	Database developer identifies the entities, attributes, and relationships required for the project
	2.	Database developer creates the entity-relationship diagram
	3.	Project stakeholders review and approve the entity-relationship diagram
Extensions	Step	Branching Action
	3a.	If the entity-relationship diagram is not approved, the database developer revises the diagram and repeats steps 2-3

## DB2. Implement the database and its tables (Functional, DB)

Use case name	DB2. Implement the database and its tables	
Main Flow	Step	Action
Related Requirements	DB1, DB3, DB4, WDB1	
Goal In Context	Create the database and its tables based on the approved entity-relationship diagram	
Preconditions	Entity-relationship diagram approved	
Successful End Condition	Database and tables implemented	
Failed End Condition	Database and tables not implemented	
Primary Actors	Database developers	
Secondary Actors	None	
Triggers	The decision to build the storage and sorter of game data	
	1.	Database developer creates the database and tables according to the entity-relationship diagram
	2.	Database developer tests the database and tables to ensure they function as expected

## DB3. Create a stats table and link it with the user (Functional, DB)

Use case name	DB3. Create a stats table and link it with the user	
Related Requirements	DB1, DB2, DB4, WDB1	
Goal In Context	Implement a stats table and associate it with the user entity in the database	
Preconditions	Database and tables implemented	
Successful End Condition	Stats table created and linked with the user	
Failed End Condition	Stats table not created or not linked with the user	
Primary Actors	Database developers	
Secondary Actors	None	
Triggers	The decision to track user statistics	
Main Flow	Step	Action
	1.	Database developer creates the stats table and defines its structure
	2.	Database developer links the stats table with the user entity
	3.	Database developer tests the functionality of the stats table and its association with the user

## DB4. Implement CRUD in the database (Functional, DB)

Use case name	DB4. Implement CRUD in the database	
Related Requirements	DB1, DB2, DB3, WDB1, WDB2, WDB3, UDB1	
Goal In Context	Implement Create, Read, Update, and Delete (CRUD) operations in the database	
Preconditions	Database and its tables created and configured	
Successful End Condition	CRUD operations functioning correctly in the database	
Failed End Condition	CRUD operations not functioning correctly in the database	
Primary Actors	Database administrators	
Secondary Actors	None	
Triggers	The need for managing data in the database	
Main Flow	Step	Action
	1.	Database administrator defines CRUD operations for each table in the database
	2.	Database administrator implements the CRUD operations using appropriate programming techniques and tools
	3.	Database administrator tests the CRUD operations to ensure they function as expected

## U1. Create a project in Unity (Functional, UNITY/Video games)

Use case name	U1. Create a Unity project	
Related Requirements	U2, U3, U4, U5, U6	
Goal In Context	Set up a new Unity project for the game development	
Preconditions	Unity installed, knowledge of Unity game development	
Successful End Condition	Unity project created and configured	
Primary Actors	Game developers	
Secondary Actors	None	
Triggers	The decision to develop the game using Unity	
Main Flow	Step	Action
	1.	Game developer opens Unity and creates a new project
	2.	Game developer configures project settings and organizes the project structure
	3.	Game developer imports necessary assets and plugins

## U2. Define and integrate Sound Effects (Non-functional, UNITY/Video games)

Use case name	U2. Define and integrate Sound Effects	
Related Requirements	U1, U3, U4, U5, U6	
Goal In Context	Design and implement sound effects for the game	
Preconditions	Unity project created	
Successful End Condition	Sound effects defined and integrated into the game	
Failed End Condition	Sound effects not defined or not integrated	
Primary Actors	Game developers, creative developers	
Secondary Actors	None	
Triggers	The decision to include sound effects in the game	
Main Flow	Step	Action
	1.	Creative developer creates or sources suitable sound effects
	2.	Game developer imports the sound effects into the Unity project
	3.	Game developer implements the sound effects in the game logic
	4.	Game developer tests the sound effects in the game to ensure they function as intended

## U3. Compose and implement Music (Non-functional, UNITY/Video games)

Use case name	U3. Compose and implement Music	
Related Requirements	U1, U2, U4, U5, U6	
Goal In Context	Design and implement background music for the game	
Preconditions	Unity project created	
Successful End Condition	Background music composed and integrated into the game	
Failed End Condition	Background music not composed or not integrated	
Primary Actors	Game developers, creative developers	
Secondary Actors	None	
Triggers	The decision to include background music in the game	
Main Flow	Step	Action
	1.	Creative developer creates or sources suitable background music for the game
	2.	Game developer imports the background music into the Unity project
	3.	Game developer implements the background music in the game logic
	4.	Game developer tests the background music in the game to ensure it functions as intended

## U4. Develop Prefabs (Functional, UNITY/Video games)

Use case name	U4. Develop Prefabs	
Related Requirements	U1, U2, U3, U5, U6	
Goal In Context	Create reusable prefabs for efficient game development	
Preconditions	Unity project created	
Successful End Condition	Prefabs created and functioning as intended	
Failed End Condition	Prefabs not created or not functioning as intended	
Primary Actors	Game developers	
Secondary Actors	None	
Triggers	The need for reusable game objects in the game	
Main Flow	Step	Action
	1.	Game developer identifies necessary prefabs for the game
	2.	Game developer creates prefabs in Unity
	3.	Game developer tests the prefabs to ensure they function as intended

## U5. Create Scenery (Non-functional, UNITY/Video games)

Use case name	U5. Create backgrounds (scenery)	
Related Requirements	U1, U2, U3, U4, U6	
Goal In Context	Design and implement visually appealing backgrounds for the game	
Preconditions	Unity project created	
Successful End Condition	Backgrounds created and integrated into the game	
Failed End Condition	Backgrounds not created or not integrated	
Primary Actors	Creative developers	
Secondary Actors	Game developers	
Triggers	The need for visually appealing backgrounds in the game	
Main Flow	Step	Action
	1.	Creative developer designs backgrounds for the game
	2.	Game developer imports the backgrounds into the Unity project
	3.	Game developer implements the backgrounds in the game logic
	4.	Game developer tests the backgrounds in the game to ensure they function as intended

## U6. Implement game mechanics (Functional, UNITY/Video games)

Use case name	U6. Implement Game Mechanics	
Related Requirements	U1, U2, U3, U4, U5	
Goal In Context	Develop and integrate core game mechanics	
Preconditions	Unity project created	
Successful End Condition	Game mechanics implemented and functioning as intended	
Failed End Condition	Game mechanics not implemented or not functioning as intended	
Primary Actors	Game developers	
Secondary Actors	None	
Triggers	The need for core game mechanics in the game	
Main Flow	Step	Action
	1.	Game developer designs and plans the core game mechanics
	2.	Game developer implements the game mechanics in Unity
	3.	Game developer tests the game mechanics to ensure they function as intended

## G1. UML General of the entire system (Non-functional, DB, UNITY/Video games, WEB)

Use case name	G1. General UML of the entire system	
Related Requirements	G2, DB1, DB2, DB3, DB4, U1, U2, U3, U4, U5, U6, W1, W2, W3, W4, W5, WDB1, WDB2, WDB3	
Goal In Context	Create a comprehensive UML diagram for the system	
Preconditions	All components designed and implemented	
Successful End Condition	UML diagram accurately represents the system and its components	
Failed End Condition	UML diagram does not accurately represent the system and its components	
Primary Actors	System architects	
Secondary Actors	Game developers	
Triggers	The need for a visual representation of the system	
Main Flow	Step	Action
	1.	System architect identifies all components and their relationships
	2.	System architect creates the UML diagram
	3.	System architect and developers review the UML diagram for accuracy and consistency

## G2. Create use case tables (Non-functional, WEB, UNITY/Video games, DB)

Use case name	G2. Create use case tables	
Related Requirements	G1, DB1, DB2, DB3, DB4, U1, U2, U3, U4, U5, U6, W1, W2, W3, W4, W5, WDB1, WDB2, WDB3	
Goal In Context	Develop use case tables for all components and requirements	
Preconditions	All components and requirements defined	
Successful End Condition	Use case tables created for all components and requirements	
Failed End Condition	Use case tables not created or incomplete	
Primary Actors	System architects	
Secondary Actors	Game developers	
Triggers	The need for detailed documentation of the system's use cases	
Main Flow	Step	Action
	1.	System architect identifies all components and their related requirements
	2.	System architect and developers create use case tables for all components
	3.	System architect and developers review the use case tables for accuracy and consistency

## WDB1. Create authentication system logic (Functional, WEB, DB)

<b>Use case name</b>	<b>WDB1. Create identification system (auth) logic</b>	
Related Requirements	W1, W2, DB1, DB2, DB3, DB4, WDB2, WDB3, UDB1	
Goal In Context	Implement an authentication system for users to access restricted parts of the website	
Preconditions	Web development tools, web server, and database configured	
Successful End Condition	Authentication system functions correctly and securely	
Failed End Condition	Authentication system does not function correctly or securely	
Primary Actors	Web developers, database administrators	
Secondary Actors	None	
Triggers	The need for secure user access to restricted parts of the website	
Main Flow	Step	Action
	1.	Web developer and database administrator design the authentication system and its requirements
	2.	Web developer implements the authentication logic using appropriate programming techniques and tools
	3.	Database administrator creates and configures necessary database tables and structures to support the authentication system
	4.	Web developer and database administrator test the authentication system to ensure it functions as expected and securely

## WDB2. Create secondary webpage with stats linked to the database (Functional, WEB, DB)

<b>Use case name</b>	<b>WDB2. Secondary page within the website for stats with the table linked to the database</b>	
<b>Related Requirements</b>	W1, W5, WDB1, WDB3, DB1, DB2, DB3, DB4	
<b>Goal In Context</b>	Display user statistics on a secondary page within the website	
<b>Preconditions</b>	Web server, database server, and development tools configured	
<b>Successful End Condition</b>	Secondary page created and displaying user statistics	
<b>Failed End Condition</b>	Secondary page not created or not displaying user statistics	
<b>Primary Actors</b>	Web developers	
<b>Secondary Actors</b>	None	
<b>Triggers</b>	The need for user statistics to be displayed on the website	
<b>Main Flow</b>	<b>Step</b>	<b>Action</b>
	1.	Web developer designs the secondary page layout and user statistics table
	2.	Web developer implements the secondary page on the web server
	3.	Web developer integrates the general communication with the database
	4.	Web developer tests the secondary page to ensure it displays user statistics correctly

## WDB3. Define host for the webpage and database (Functional, WEB, DB)

Use case name	WDB3. Define host for the website and the database	
Related Requirements	W1, W2, W3, W4, W5, WDB1, WDB2, DB1, DB2, DB3, DB4	
Goal In Context	Set up hosting for the website and database	
Preconditions	Web server, database server, and development tools configured	
Successful End Condition	Website and database hosted and accessible	
Failed End Condition	Website and database not hosted or not accessible	
Primary Actors	Web developers, database administrators	
Secondary Actors	None	
Triggers	The need to host the website and database	
Main Flow	Step	Action
	1.	Web developer and database administrator select a suitable hosting provider
	2.	Web developer and database administrator configure hosting settings and deploy the website and database
	3.	Web developer and database administrator test the hosted website and database for accessibility and performance

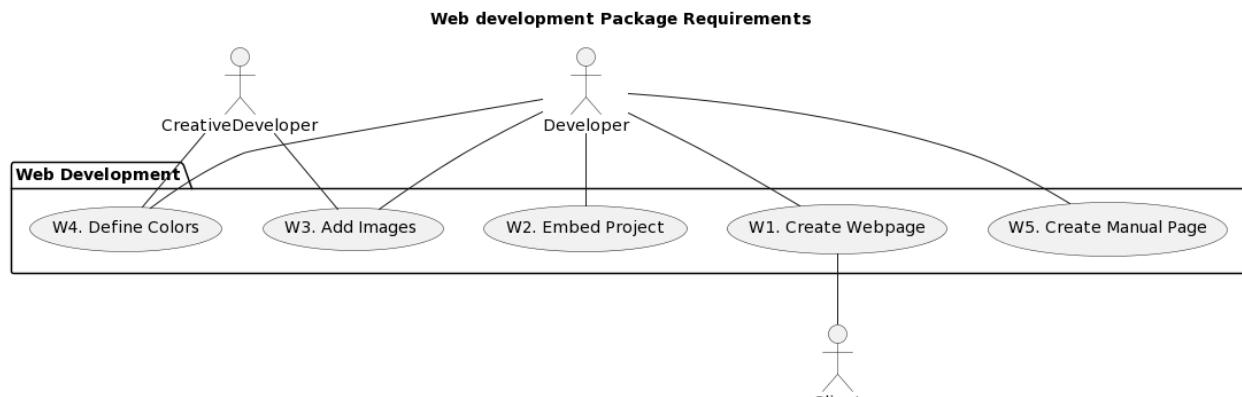
## UDB1. Connect database with the game (Functional, UNITY/Video games, DB)

<b>Use case name</b>	<b>UDB1. Connect the database with the game</b>	
Related Requirements	U1, U2, U3, U4, U5, U6, DB1, DB2, DB3, DB4	
Goal In Context	Integrate the game with the database for storing and retrieving data	
Preconditions	Unity project created, database implemented	
Successful End Condition	Game connected to the database and data exchange functioning correctly	
Failed End Condition	Game not connected to the database or data exchange not functioning correctly	
Primary Actors	Game developers, database administrators	
Secondary Actors	None	
Triggers	The need for data exchange between the game and the database	
Main Flow	Step	Action
	1.	Game developer identifies data exchange requirements between the game and the database
	2.	Game developer implements the connection between the game and the database

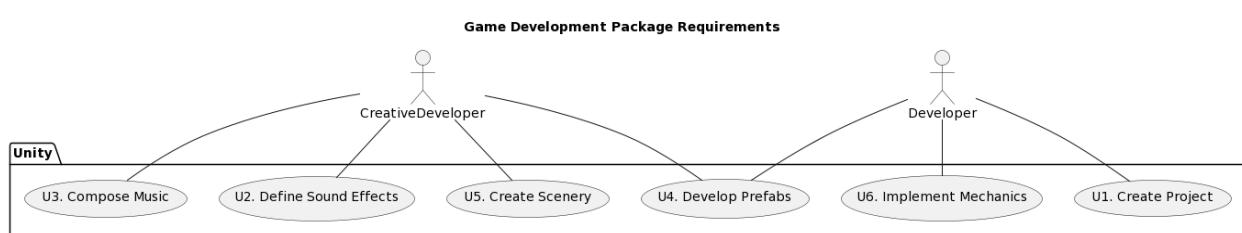
## Anexo: Diagramas UML adicionales



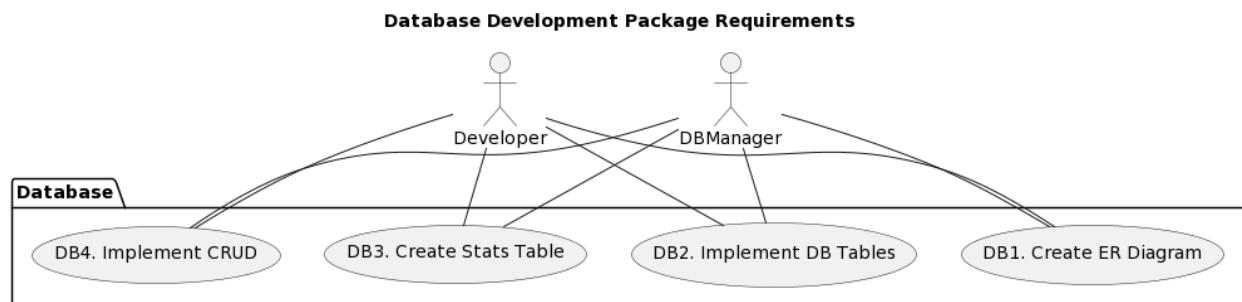
Anexo 1. Flujo de Juego



Anexo 2. Requerimientos WEB

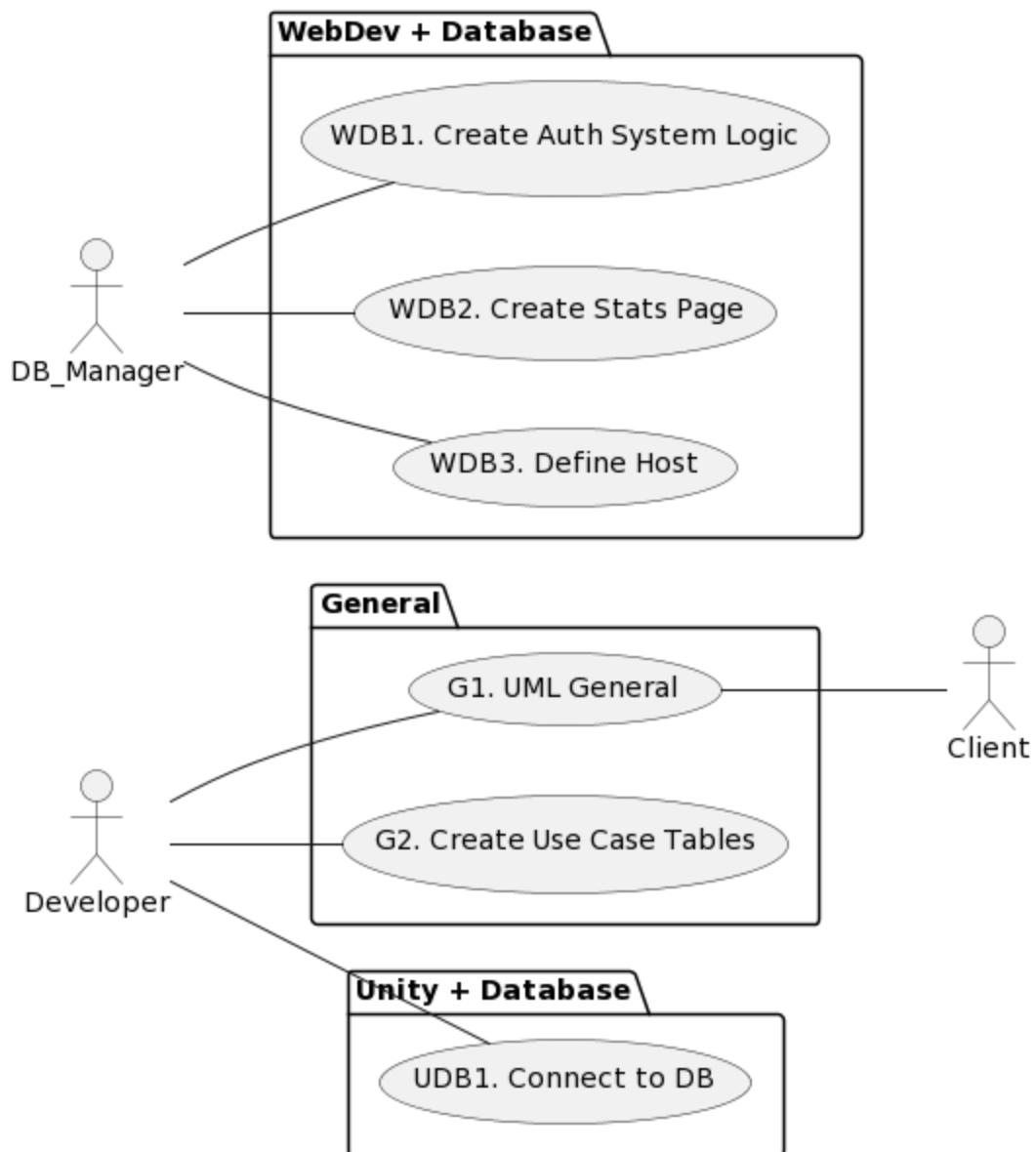


Anexo 3. Requerimientos de desarrollo de Videojuegos



Anexo 4. Requerimientos de Base de Datos

### **General Package Requirements**



Anexo 5. Requerimientos Mixtos (Generales)