

Documentación Proyecto DAM



Adrián Rodríguez Rodríguez

Contenido

1.	Introducción al Juego	3
2.	Requisitos del Sistema.....	3
3.	Instalación y Primer Arranque.....	4
4.	Stack Tecnológico.....	4
5.	Acceso a la Aplicación (Login / Registro)	5
6.	Roles y Permisos	6
7.	Pantallas y Funciones Principales.....	7
7.1.	Pantalla de Login/Registro	7
7.2.	Menú principal (Usuario)	8
7.3.	Panel de Administración (Admin)	8
7.4.	HUD en partida	9
7.5.	Menú de Pausa.....	9
7.6.	Game Over.....	10
7.7.	Victoria	10
8.	Controles de Juego	11
9.	Diseño de la Base de Datos (Firestore)	12
9.1.	Colección usuarios	12
9.2.	Colección partidasGuardadas	12
10.	Script SQL (aunque nuestra BD es NoSQL).....	13
11.	Reglas de Seguridad Firebase	14
12.	Flujo de Navegación del Usuario	15
13.	Diseño del videojuego	16
13.1.	Jugador	16
14.	Guardado y Carga de Partida.....	18
15.	Preguntas Frecuentes	18
16.	Solución de Problemas.....	19
17.	Retos Encontrados	19
18.	Posibles Ampliaciones.....	20
19.	Créditos y soporte.....	20
Anexo A.	Casos de Prueba de Usuario.....	21
A.1.	Inicio Sesión (Usuario Existente).....	21
A.2.	Registrarse (Nuevo Usuario).....	21
A.3.	Nueva Partida.....	21
B.4.	Guardar y Cargar.....	22
B.5.	Derrota y Game Over	22
B.6.	Admin: Listar y Eliminar partida	22

1. Introducción al Juego

Dungeon Fighter es un videojuego 2D de acción y exploración con vista cenital (top-down). El jugador recorre salas de una mazmorra, combate enemigos, consigue ítems y progresa hasta derrotar al jefe final.

Dispone de **autenticación de usuarios y guardado/carga en la nube** para continuar la partida desde cualquier equipo.

Objetivos para el usuario:

- Iniciar sesión o registrarse con gmail/contraseña.
- Comenzar una **Nueva Partida** o **Cargar** la existente.
- Jugar, pausar, guardar y continuar más tarde.
- En caso de derrota, usar la pantalla **Game Over** para retomar el progreso.

2. Requisitos del Sistema

Software

- Windows 10/11 (64 bits).
- Conexión a Internet para login y guardado/carga.

Hardware Recomendado

- CPU de 2 núcleos, 4 GB RAM.
- GPU integrada o dedicada compatible con gráficos 2D.
- 1 GB de espacio libre para juego + datos de usuario.

Red

- Conexión a internet para autenticación y base de datos.

3. Instalación y Primer Arranque

1. Descarga el ejecutable o instala el proyecto desde el repositorio del curso.
2. Extrae en una carpeta sin permisos especiales (por ejemplo, C:\Juegos\DungeonFighter).
3. Ejecuta DungeonFighter.exe.
4. La aplicación abrirá la **pantalla de Login/Registro**.

4. Stack Tecnológico

- **Motor:** Unity 6.0 (2D)
- **Lenguaje:** C#
- **Sistemas Unity usados:**
 - **2D:** Tilemap, Rigidbody2D/Colliders, Animator.
 - **UI:** Canvas, Botones, Textos, Paneles (Login, Menú, Pausa, Admin...).
 - **SceneManager** para cambio de escenas (Menú <-> Juego <-> GameOver/Win).
- **Persistencia en la nube:** Firebase
 - **Firebase Auth:** usando la API REST de Identity Toolkit (accounts:singUp, accounts:singInWithPassword).
 - **Firestore** usando la API REST (firestore.googleapis.com/v1/.../documents).
- **Comunicación HTTP en Unity:** UnityWebRequest.
- **Serialización/JSON:** Newtonsoft.Json (JsonConvert) + parseo con JObject/Jtoken.
- **Control de versiones:** Git + GitHub.
- **Plataforma:** PC (Windows con ejecutable .exe).

5. Acceso a la Aplicación (Login / Registro)

Al iniciar, verás dos modos:

- **Login:** introduce Correo y Contraseña y pulsa Aceptar.
- **Registro:** pulsa Cambiar para desbloquear el panel de registro. Introduce Correo, Contraseña y Nombre. Confirma con Aceptar (si quieres iniciar como admin, deberás poner la contraseña de admin proporcionada por el creador de la app).

Mensajes habituales:

- *Error de logueo.*



6. Roles y Permisos

- **Usuario:** puede iniciar/cargar partida, pausar, guardar y cerrar sesión.
- **Admin:** además puede listar y eliminar partidas guardadas de usuarios de pruebas (panel específico). Este rol se consigue al iniciar como admin, o crear una cuenta conociendo la contraseña de los administradores.



7. Pantallas y Funciones Principales

7.1. Pantalla de Login/Registro

- **Campos:** Correo, Contraseña. En registro también Nombre y check de Admin.
- **Botones:** Aceptar / Cambiar (para alternar Login/Registro).
- **Resultado:** Tras iniciar sesión correctamente, pasarás al Menú principal (Usuario) o al Panel Admin, en función del rol.



7.2. Menú principal (Usuario)

- **Nueva Partida:** crea una partida nueva y entra en el juego.
- **Cargar Partida:** recupera tu progreso en la nube y entra en el juego.
- **Cerrar Sesión:** vuelve a la pantalla de Login/Registro.



7.3. Panel de Administración (Admin)

- **Listar Partidas:** muestra partidas guardadas.
- **Eliminar (por UID):** borra la partida asociada a un usuario de pruebas.
- **Cerrar Sesión:** vuelve a Login/Registro.



7.4. HUD en partida

- **Vida del jugador:** mostrada de forma constante.
- **Cámara:** sigue al jugador suavemente mientras se desplaza.
- **ESC:** Para salir al menú de pausa.



7.5. Menú de Pausa

- **Continuar:** reanuda la partida.
- **Guardar:** envía el nuevo estado a la nube.
- **Login:** regresa al menú principal (no cierra sesión).



7.6. Game Over

Aparece cuando la vida del jugador llega a 0. Presionando cualquier tecla, te llevará al menú principal.



7.7. Victoria

Aparece cuando la vida del jugador llega a 0. Presionando cualquier tecla, te llevará al menú principal.



8. Controles de Juego

- **Movimiento:** W A S D
- **Atacar:** J o click izquierdo del ratón
- **Pausa:** Esc

Consejos:

- Al atacar, tu personaje orienta la animación hacia la dirección en la que está mirando.
- Los enemigos tienen un rango de detección y pueden realizar ataques cuerpo a cuerpo.

9. Diseño de la Base de Datos (Firestore)

9.1. Colección usuarios

Documento por usuario: usuarios/{uid}

Campos que guardamos/actualizamos:

- email (string)
- nombreUsuario (string)
- fechaCreacion (string)
- esAdmin (boolean)
- activo (boolean)

Uso: Al registrarnos/loguearnos, se crea/actualiza este perfil.

9.2. Colección partidasGuardadas

Documento por usuario: partidasGuardadas/{uid}

Campos que guardamos/actualizamos:

- nombrePartida (string)
- datosJugador (map)
 - vida (int)
 - vidaMaxima (int)
 - posX (float)
 - posY (float)
 - nombreEscena (string)
- datosInventario (map) (no implementado por ahora)
 - monedas (int)
 - llaves (int)
 - pociones (int)
- estadoMundo (map)
 - enemigosEliminados (array<string>)
 - cofresAbiertos (array<string>)
 -

Uso:

Guardar -> se hace PATCH al documento del UID.

Cargar -> se hace GET y se reconstruye el objeto de partida parseando el JSON tipado de Firestore.

10. Script SQL (aunque nuestra BD es NoSQL)

```
CREATE TABLE usuarios (  
  uid          VARCHAR(128) PRIMARY KEY,  
  email        VARCHAR(255) NOT NULL UNIQUE,  
  nombre_usuario VARCHAR(100) NOT NULL,  
  es_admin     BOOLEAN NOT NULL DEFAULT FALSE,  
  activo       BOOLEAN NOT NULL DEFAULT TRUE,  
  fecha_creacion TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP  
);
```

```
CREATE TABLE partidas_guardadas (  
  uid          VARCHAR(128) PRIMARY KEY,  
  nombre_partida VARCHAR(100) NOT NULL,  
  vida         INT NOT NULL,  
  vida_maxima  INT NOT NULL,  
  pos_x        REAL NOT NULL,  
  pos_y        REAL NOT NULL,  
  nombre_escena VARCHAR(100) NOT NULL,  
  monedas      INT NOT NULL DEFAULT 0,  
  llaves       INT NOT NULL DEFAULT 0,  
  pociones     INT NOT NULL DEFAULT 0,  
  -- Arrays/Map de Firestore representados como JSON  
  enemigos_eliminados JSON NOT NULL,  
  cofres_abiertos  JSON NOT NULL,  
  ultima_actualizacion TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  CONSTRAINT fk_partida_usuario  
    FOREIGN KEY (uid) REFERENCES usuarios(uid) ON DELETE CASCADE  
);
```

```
CREATE INDEX idx_partidas_escena ON partidas_guardadas(nombre_escena);
```

11. Reglas de Seguridad Firebase

```
rules_version = '2';

service cloud.firestore {

  match /databases/{database}/documents {

    function signedIn() {

      return request.auth != null;

    }

    function isOwner(uid) {

      return signedIn() && request.auth.uid == uid;

    }

    function isAdmin() {

      return signedIn() &&

        get(/databases/{database}/documents/usuarios/{request.auth.uid}).data.esAdmin == true;

    }

    match /usuarios/{uid} {

      // El usuario ve su perfil, entonces el admin leer para gestionar los datos
      allow read: if isOwner(uid) || isAdmin();

      // Crea/actualiza su perfil el usuario
      allow create, update: if isOwner(uid);

      // Solo admin puede borrar perfiles
      allow delete: if isAdmin();

    }

    match /partidasGuardadas/{uid} {

      // Gestión de la partida por el usuario
      allow read, create, update, delete: if isOwner(uid);

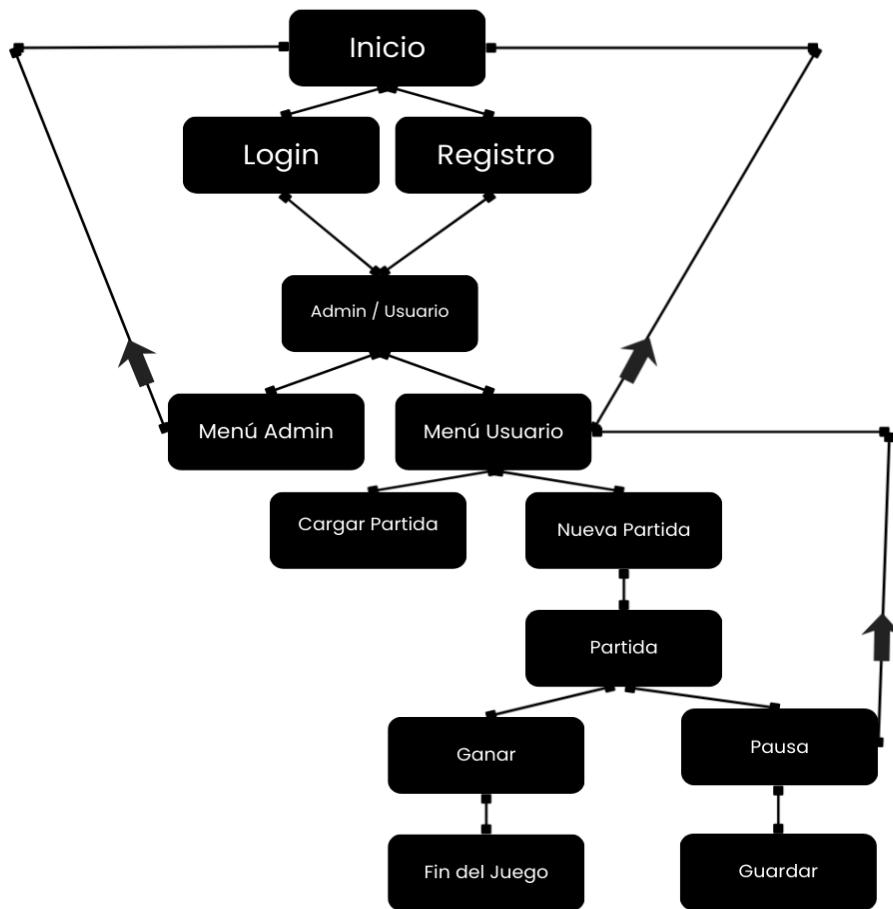
      // El Admin puede borrar partidas
      allow delete: if isAdmin();

    }

  }

}
```

12. Flujo de Navegación del Usuario



13. Diseño del videojuego

13.1. Jugador



Vida: 40 hp

Daño de espada: 1 hp

Cooldown: 0.4 s

13.2. Enemigos

Goblin Spearman		Vida: 5 hp Daño de lanza: 2 hp Cooldown: 1.5 s
Orc Grunt		Vida: 10 hp Daño de espada: 5 hp Cooldown: 1.9 s
Goblin Maceman		Vida: 5 hp Daño des maza: 3 hp Cooldown: 1.5 s

13.3. Mecánicas

Pinchos de Suelo		Daño: 1 hp Cooldown: 0.8 s
Puerta		Se activa al derrotar a todos los enemigos de una sala.
Cofre		Restaura la vida al máximo.

14. Guardado y Carga de Partida

- **Al Guardar desde la Pausa, el juego almacena: vida, posición, escena y elementos básicos del estado del mundo.**
- **Al Cargar, se recupera el último estado disponible para el usuario autenticado.**
- **El progreso queda asociado a tu cuenta; si inicias sesión en otro PC, podrás continuar.**

Recomendación: guarda tras limpiar una sala o antes de explorar áreas nuevas.

15. Preguntas Frecuentes

¿Puedo jugar sin Internet?

Puedes jugar, pero el login y el guardado/carga requieren conexión.

He olvidado la contraseña.

Contacta con el administrador del proyecto para el reseteo.

No veo mis partidas guardadas.

Comprueba que has iniciado sesión con el mismo correo y que tienes conexión a internet.

16. Solución de Problemas

“Error de autenticación”: credenciales incorrectas o cuenta inexistente. Revisa el correo y la contraseña.

“No se puede conectar”: verifica tu red o.

El juego no responde al atacar: confirma que el cursor está dentro de la ventana del juego y que no mantienes otra tecla presionada.

La cámara no sigue al jugador: reinicia la escena desde el menú; si persiste, reinstala el juego.

17. Retos Encontrados

- **Asincronía en Unity**
 - Login/guardar/cargar son async task.
 - Reto: no bloquear el main thread y manejar errores de red.
 - Solución: UnityWebRequest + TaskCompletionSource para await limpio.
- **Persistencia de sesión**
 - Guardar IdToken y Uid en un servicio común para poder guardar/cargar desde cualquier escena.
 - Solución: GameSaveServicio + objeto persistente con DontDestroyOnLoad.
- **Balance y cooldowns de combate**
 - Evitar daño múltiple por frames/colisiones repetidas.
 - Solución: cooldowns por ataque y por hit (jugador/enemigos/trampas).

18. Posibles Ampliaciones

- Slot de guardado múltiples
(partidasGuardadas/{uid}/slots/{slotId})
- Inventario real (lista de ítems con Ids, rarezas, equipamiento...).
- Progresión (más niveles de dificultad avanzada).
- Más IA: Ataques a distancia...
- Optimización: Gestión de mejora en las animaciones para más fluidez visual.
- Mejora de seguridad:
 - Validación estricta en rules (keys permitidas, rangos de vida, etc.).
 - Admin gestionado desde backend/console, no por contraseña en la UI.

19. Créditos y soporte

- **Desarrollo y arte:** Adrián Rodríguez Rodríguez (Dungeon Fighter).
- **Motor:** Unity.
- **Soporte:** Escribe al administrador (adrianrodry55@gmail.com).

Anexo A. Casos de Prueba de Usuario

A.1. Inicio Sesión (Usuario Existente)

Precondición: usuario ya registrado.

Pasos:

1. Abrir el juego → aparece Login.
2. Escribir correo/contraseña.
3. Pulsar Aceptar.

Resultado esperado: accede al Menú principal.

A.2. Registrarse (Nuevo Usuario)

Pasos:

1. En Login, pulsar Cambiar.
2. Rellenar Correo/Contraseña/Nombre.
3. Pulsar Aceptar.

Resultado esperado: crea cuenta y muestra Menú (Usuario).

A.3. Nueva Partida

Pasos: Menú principal → Nueva Partida.

Resultado esperado: entra al HUD de juego con el jugador logueado en la escena inicial de su nueva partida.

B.4. Guardar y Cargar

Pasos:

1. En juego, abre **Pausa** → **Guardar Partida**.
2. Vuelve al **Menú** → **Cargar Partida**.
Resultado esperado: reaparece el jugador con el último estado guardado de su partida.

B.5. Derrota y Game Over

Pasos:

1. Recibir **daño** hasta 0 de vida.
2. En **Game Over**, pulsar cualquier tecla.
Resultado esperado: vuelve al menú principal de tu usuario logueado.

B.6. Admin: Listar y Eliminar partida

Precondición: cuenta con rol Admin.

Pasos:

1. **Iniciar sesión como Admin** → **Panel Admin**.
2. **Listar partidas**.
3. Introducir **UID de prueba** → **Eliminar**.
Resultado esperado: la partida de ese usuario deja de estar disponible.