

# **Documento de Diseño de Software**

**Equipo:** EvoLogic

**1. Introducción** Este documento proporciona el diseño del software para la versión final del producto, siguiendo el estándar ISO 29110 5-1-2:2024. Incluye aspectos tanto de diseño de alto nivel como de diseño detallado del sistema.

## **2. Diseño de Software de Alto Nivel**

**2.1 Visión General del Sistema** El sistema es una aplicación web diseñada para gestionar pacientes y juegos interactivos. Incluye múltiples interfaces para autenticación de usuarios, gestión de datos de pacientes e interacción con juegos.

**2.2 Componentes y Estructura del Software** El sistema consta de los siguientes componentes principales:

- **Frontend (HTML, JavaScript):** Maneja las interfaces de usuario e interacciones.
- **Backend (Configuración de Firebase):** Gestiona la autenticación y almacenamiento de datos.
- **Base de Datos (JSON y Firebase Firestore):** Almacena datos de pacientes e información relacionada con los juegos.

## **2.3 Relación entre Componentes**

- El frontend interactúa con Firebase para la autenticación y recuperación de datos.
- El sistema de gestión de pacientes recupera y actualiza registros en la base de datos.

- Los módulos de juego obtienen los datos necesarios y brindan una experiencia interactiva.

## **2.4 Consideraciones**

- **Rendimiento:** Optimizado para interacciones fluidas.
- **Seguridad:** Autenticación de usuarios mediante Firebase.
- **Manejo de Errores:** Validación básica y mensajes de error para la entrada del usuario.
- **Diseño de Base de Datos:** Uso de JSON y Firestore para almacenamiento estructurado.

## **3. Diseño de Software de Bajo Nivel**

**3.1 Diseño Detallado de Componentes** Cada módulo está diseñado de la siguiente manera:

- **Autenticación (inicioSesion.html, loginPaciente.html, firebase-config.js)**
  - Inicio de sesión de usuarios a través de Firebase.
  - Manejo de sesiones con tokens de autenticación.
- **Gestión de Pacientes (paciente.html, pacienteDatos.js, listadoPacientes.html)**
  - Muestra la información de los pacientes.
  - Operaciones CRUD en los registros de pacientes.

- **Módulos de Juego (juego.html, juego1Asistido.html, juego2.html, juego3.html, juego3.js, palabra.json)**
  - Juegos interactivos para la participación del usuario.
  - Obtiene datos de juego desde archivos JSON.

### **3.2 Formato y Almacenamiento de Datos**

- JSON para almacenar datos de juegos predefinidos.
- Firestore para registros de pacientes y datos de autenticación.

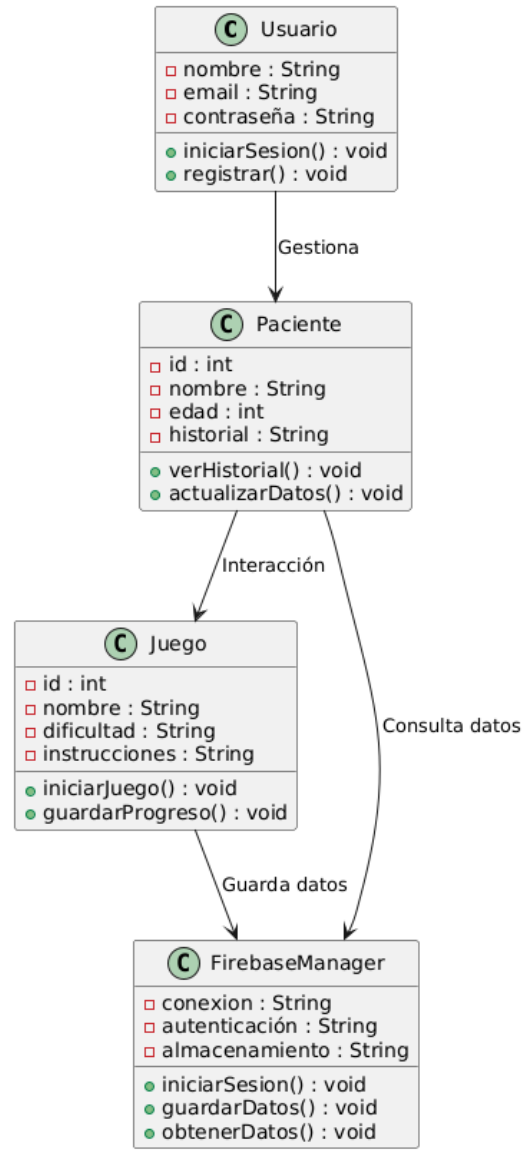
### **3.3 Convenciones de Nombres**

- Archivos HTML representan componentes de la interfaz de usuario.
- Archivos JavaScript manejan la lógica e interacciones.
- Archivos JSON almacenan datos estructurados.

## **4. Estructura del Programa** El sistema sigue un enfoque modular donde:

- Los archivos HTML definen la interfaz de usuario.
- Los archivos JavaScript manejan comportamientos dinámicos.
- Firebase actúa como servicio backend.

## **5. Diagrama de diseño:**



Este diseño garantiza escalabilidad, mantenibilidad y rendimiento para la versión final del producto.