ConnectToPlay

***Desarrollo de una aplicación web** orientada a conectar usuarios con **intereses comunes**, permitiendo la organización de **partidas de videojuegos en línea** o **encuentros presenciales** para la práctica de **deportes físicos**.*

Alcalá de Henares, Madrid, España
IES Isidra de Guzmán

***Carlos Manglano** – cmf672@educa.madrid.org*
***Andrés Inciarte Casas** – aic443@educa.madrid.org*
***Álvaro Hermosilla Alameda** – alvaro.hermosilla@educa.madrid.org*

Abstract

*The present project outlines the **design** and **development** of **ConnectToPlay**, a **web application** with a **social network structure**, aimed at facilitating **user connections** for the organisation of **online video game sessions** or **in-person meetups** for the practice of **physical sports**. The application enables users to **create** and **join rooms** categorised by **interests**, **location**, and **level of experience**. Its implementation makes use of **modern technologies** such as **NextJS**, **NestJS**, and **PostgreSQL**, following a **fullstack architecture** and an **agile methodology** based on **SCRUM**.*

Keywords**: **NextJS**, **NestJS**, **React**, **PostgreSQL**, **social network**, **SCRUM**, **fullstack app

I. Introducción

*El crecimiento exponencial de la **industria de los videojuegos** y el interés sostenido por la **actividad física organizada** han incrementado la demanda de **plataformas digitales** orientadas a la **conexión social**. Por un lado, el ámbito del **gaming** ha generado **comunidades globales activas**; por otro, el **deporte** continúa siendo una práctica extendida entre diversos perfiles demográficos.*

*Actualmente, el **ecosistema digital** carece de una **solución centralizada** que permita la **interacción eficiente** entre personas con **intereses comunes**, tanto en **videojuegos** como en **deportes físicos**. Ante esta carencia, se plantea **ConnectToPlay** como **propuesta de valor**: una **plataforma web***

funcional *que permite* ***segmentar usuarios*** *por* ***tipo de actividad***, ***nivel*** *y* ***localización***, *ofreciendo* ***mecanismos estructurados*** *de* ***emparejamiento*** *y* ***comunicación***.

La solución integra ***funcionalidades propias de redes sociales***, ***motores de búsqueda temáticos*** *y* ***mensajería en tiempo real***. *Permite la* ***creación y gestión de salas*** *asociadas a videojuegos como League of Legends, Valorant o Counter-Strike 2, así como a deportes como fútbol, baloncesto o running. La lógica de diseño se apoya en la* ***segmentación inteligente***, *la* ***geolocalización*** *y la* ***afinidad de intereses***, *asegurando una* ***experiencia centrada en el usuario*** *y orientada a la* ***interacción activa***.

II. Motivación

La fase inicial del proyecto incluyó un ***estudio de mercado*** *orientado a identificar* ***soluciones existentes*** *con objetivos similares. Las plataformas analizadas fueron:*

- ***Discord****: utilizada en el entorno gamer, ofrece* ***comunicación eficiente*** *por voz y texto, pero no dispone de un* ***sistema para descubrir nuevos jugadores*** *ni de* ***integrar actividades presenciales***.
- ***Meetup****: especializada en* ***encuentros físicos***, *muestra* ***poca orientación hacia la comunidad gamer*** *y presenta una* ***interfaz genérica***, *sin opciones de* ***emparejamiento automatizado***.
- ***LFG apps*** *como GamerLink o GameTree: centradas en videojuegos, carecen de* ***soporte para actividades físicas*** *y no ofrecen* ***funcionalidades híbridas***.
- ***Redes sociales generalistas*** *(Facebook, Telegram): aunque permiten la* ***organización de grupos***, *su* ***falta de especialización*** *limita su efectividad como herramienta de* ***emparejamiento estructurado***.

A partir de dicho análisis, se identifica una ***brecha significativa en el mercado****: la* ***ausencia de una aplicación híbrida*** *que combine* ***criterios de emparejamiento*** *por* ***intereses***, ***localización*** *y* ***tipo de actividad*** *en un* ***entorno unificado***. ***ConnectToPlay*** *surge como* ***respuesta directa a esta necesidad***, *ofreciendo una plataforma que optimiza tanto el* ***descubrimiento*** *como la* ***interacción entre usuarios***.

III. Requisitos

A. Requisitos de Hardware

*Para garantizar un **desarrollo fluido** y compatible con las herramientas modernas, se establece el siguiente entorno mínimo de trabajo:*

- ***Procesador**: Intel i5 o equivalente*
- ***Memoria RAM**: 16 GB*
- ***Almacenamiento**: SSD de 256 GB o superior*
- ***Sistema operativo**: Windows 10/11, macOS o Linux compatible con **Docker***
- ***Software esencial**: **Node.js**, **Docker**, **Git**, **Visual Studio Code***

*El acceso a la aplicación por parte de los usuarios finales requiere únicamente un **navegador web actualizado** y **conexión a Internet**. La interfaz ha sido diseñada para ofrecer una **experiencia responsive**, compatible con **ordenadores de escritorio** y **dispositivos móviles**.*

B. Requisitos de Software

*Durante el desarrollo del proyecto se han utilizado herramientas seleccionadas por su **eficiencia**, **escalabilidad** y **adopción profesional**:*

- ***Frontend**: implementado con **React**, **NextJS** y **TailwindCSS***
- ***Backend**: desarrollado en **NestJS**, framework modular basado en **Node.js***
- ***Base de datos**: gestionada mediante **PostgreSQL***
- ***Control de versiones**: gestionado con **Git** en **GitHub***
- ***Diseño UI y prototipado**: realizado con **Figma***
- ***Despliegue y hosting**: realizados directamente desde **Render***

IV. Metodología

*Adoptar la **metodología SCRUM** como marco ágil. Establecer **sprints semanales** con **reuniones de planificación**, **revisión** y **retrospectiva**.*

*Gestionar tareas con **Trello**, dividiendo el desarrollo en módulos como **autenticación**, **salas** y **chat**.*

*Implementar un flujo de trabajo en **GitHub** con **issues** y **ramas específicas** por funcionalidad.*

V. Análisis

A. Casos de uso principales

Identificar los flujos de interacción esenciales para el funcionamiento de la aplicación:

- *Registro e inicio de sesión.*
- *Creación o unión a una sala de juego o deporte.*
- *Comunicación mediante chat entre los miembros.*
- *Búsqueda y filtrado por ubicación y nivel de experiencia.*

B. Requisitos funcionales y técnicos

Establecer las características básicas que debe cumplir el sistema para garantizar una experiencia adecuada al usuario:

- *Interfaz de usuario sencilla, con diseño **mobile-first**.*
- *Escalabilidad del sistema para adaptarse a un número creciente de usuarios.*
- *Funcionalidad de **chat en tiempo real** para facilitar la comunicación inmediata.*
- *Gestión de sesiones mediante **JWT** (JSON Web Tokens) para garantizar autenticación y seguridad.*

VI. Diseño

Diseño de la interfaz de usuario (UI)

Definir la apariencia y estructura visual de la aplicación:

- *Creación de **wireframes** mediante **Capturas**, representando pantallas clave y navegación.*
- *Implementación de un diseño **responsive**, compatible con dispositivos móviles y escritorio.*
- *Aplicación de una temática visual centrada en el contexto **deportivo y gamer**, acorde a la identidad del proyecto.*

VII. Desarrollo

A. Frontend

*Implementar la interfaz de usuario como una **Single Page Application (SPA)** utilizando **React**, con el fin de ofrecer una experiencia fluida y dinámica sin recargas de página.*

*Configurar el sistema de navegación mediante **React Router**, permitiendo el enrutamiento entre vistas internas.*

*Gestionar el estado de la aplicación mediante **Context API**, favoreciendo la compartición eficiente de datos entre componentes sin depender de bibliotecas externas adicionales.*

B. Backend

*Construir la lógica del servidor empleando **NestJS**, estructurado de forma **modular** para facilitar la escalabilidad y la separación de responsabilidades.*

*Exponer los recursos de la aplicación a través de una **API REST**, organizada por controladores y servicios, siguiendo los principios de diseño limpio.*

*Proteger los endpoints mediante **autenticación JWT**, permitiendo la validación de identidad y el manejo seguro de sesiones.*

*Utilizar **pipes** para validación de datos y **DTOs (Data Transfer Objects)** para estandarizar las estructuras de entrada y salida entre el cliente y el servidor.*

C. Base de datos

*Configurar la persistencia de datos mediante **PostgreSQL**, utilizando **TypeORM** como herramienta de mapeo objeto-relacional.*

*Definir las siguientes **entidades** principales:*

- ***Usuarios**: datos de acceso, perfil e historial de actividad.*
- ***Salas**: información sobre tipo de actividad, visibilidad, participantes y horarios.*
- ***Mensajes**: contenidos del chat asociados a cada sala y remitente.*

D. Funcionalidades principales

Incorporar las características básicas necesarias para el uso completo de la aplicación:

- *Permitir **crear salas**, configurables por tipo de actividad e invitación.*
- *Habilitar un **filtro por categoría** para facilitar la búsqueda por tipo de juego o deporte.*
- *Desarrollar un sistema de **chat en tiempo real**, utilizando tecnologías de mensajería instantánea compatibles con WebSockets.*

VIII. Pruebas e Implementación

A. Plan de pruebas

Para validar el correcto funcionamiento de la aplicación se establecen los siguientes tipos de pruebas:

- ***Pruebas unitarias**: realizadas con **Jest** para verificar el comportamiento individual de funciones y componentes.*
- ***Pruebas funcionales**: llevadas a cabo con **Cypress**, enfocadas en comprobar los flujos principales desde la perspectiva del usuario.*
- ***Pruebas de carga**: ejecutadas mediante **Artillery**, con el objetivo de medir el rendimiento y la respuesta del sistema bajo demanda simultánea de usuarios.*

B. Hosting

Para el despliegue del proyecto se utilizan plataformas que permiten una integración ágil y continua:

- ***Frontend**: desplegado en **Render**, optimizado para aplicaciones desarrolladas con React.*

- ***Backend**: desplegado en **Render**, con soporte para proyectos Node.js y NestJS.*

C. Publicidad

Se crean materiales gráficos y audiovisuales para presentar el proyecto:

- ***Cartel A3** diseñado con **Canva**, incluyendo nombre, logo y descripción visual de la aplicación.*
- ***Video tutorial** explicativo, de una duración máxima de cinco minutos.*
- ***Manual en PDF** que recoge las instrucciones básicas de uso y funcionalidades principales.*

IX. Mantenimiento y Futuras Mejoras

Se define un plan de mantenimiento con revisiones mensuales para comprobar el rendimiento de la aplicación, corregir errores detectados y aplicar actualizaciones necesarias.

Mejoras previstas:

- *Incorporación de **notificaciones en tiempo real** para mensajes, invitaciones.*
- *Integración de un **calendario de eventos deportivos** para organizar partidas y encuentros centrados en deportes.*
- *Desarrollo de una **aplicación móvil** utilizando **React Native**, adaptando la interfaz a dispositivos móviles.*
- *Implementación de un **sistema de matchmaking** para emparejar usuarios por nivel, preferencias e historial de actividad.*

X. Referencias

- *https://docs.nestjs.com* – ***NestJS Documentation***
- *https://react.dev/learn* – ***React Documentation***
- *https://www.postgresql.org/docs/* – ***PostgreSQL Documentation***
- *https://socket.io/docs/* – ***Socket.IO Documentation***
- *https://tailwindcss.com/docs/installation* – ***Tailwind***
- *https://www.figma.com* – ***Figma***
- *https://github.com* – ***GitHub***
- *https://docs.github.com/en/actions* – ***GitHub Actions***
- *https://pages.github.com* – ***GitHub Pages***
- *https://www.docker.com/products/docker-desktop* –***Docker***
- *https://code.visualstudio.com* – ***Visual Studio Code***
- *https://scrumguides.org* – ***SCRUM Guide***
- *https://stackoverflow.com* – ***Stack Overflow***