

**MEMORIA: SWIMAPP**

**SISTEMAS DE INFORMACIÓN DE GESTIÓN Y BUSSINESS INTELLIGENCE**

JORGE GRAJAL HERRERO

Índice

[1. Problema que resuelve 3](#_Toc187497924)

[a. Tecnologías utilizadas 3](#_Toc187497925)

[b. Implementación de Inteligencia Artificial 3](#_Toc187497926)

[i. Flujo de la integración 4](#_Toc187497927)

[2. Lecciones aprendidas y análisis DAFO 4](#_Toc187497928)

[3. Visión de futuro 5](#_Toc187497929)

[4. Bibliografía 5](#_Toc187497930)

# Problema que resuelve

La aplicación aborda la necesidad de centralizar la gestión de equipos de natación, un desafío común para entrenadores y nadadores. Tradicionalmente, estas actividades han dependido de herramientas aisladas como hojas de cálculo, calendarios y grupos de mensajería, lo que genera ineficiencias y dificultades en el seguimiento del progreso. Esta plataforma ofrece una solución digital integral, permitiendo la organización de equipos, la planificación de entrenamientos y el análisis del rendimiento deportivo.

Entre sus principales funcionalidades se destacan la creación y gestión de equipos por parte de los entrenadores, la asignación de roles a los nadadores, el seguimiento del progreso individual mediante métricas detalladas y gráficas, así como la comunicación interna a través de notificaciones y mensajería. Con este enfoque, se busca no solo optimizar la gestión diaria, sino también ofrecer a los usuarios herramientas avanzadas para mejorar su desempeño deportivo.

## Tecnologías utilizadas

El desarrollo de la aplicación ha sido posible gracias a un ecosistema tecnológico moderno y bien integrado que garantiza tanto rendimiento como escalabilidad. En el backend, Prisma ORM ha sido fundamental para gestionar la base de datos PostgreSQL, permitiendo operaciones complejas con datos de usuarios, entrenamientos y estadísticas de rendimiento. En el frontend, Next.js ha ofrecido una experiencia de usuario optimizada mediante la combinación de páginas estáticas y dinámicas, mientras que React y TailwindCSS han facilitado el diseño de interfaces accesibles e intuitivas.

Para la validación de datos y formularios, se ha utilizado Zod junto con React Hook Form, lo que ha garantizado una experiencia de usuario fluida y segura. La autenticación, por su parte, ha sido gestionada con NextAuth, que permite iniciar sesión mediante OAuth o credenciales, reforzado con bcryptjs para el cifrado de contraseñas.

En términos de visualización de datos, Recharts ha sido la librería principal para mostrar gráficas interactivas sobre el rendimiento de los nadadores. Adicionalmente, herramientas como Embla Carousel y Date-fns han enriquecido el diseño y la planificación, mientras que Zustand ha sido clave para manejar el estado global de la aplicación de manera eficiente.

## Implementación de Inteligencia Artificial

La implementación de inteligencia artificial en la aplicación se centra en una funcionalidad de chat que utiliza OpenAI GPT-4 para ofrecer respuestas personalizadas y relevantes basadas en las consultas de los usuarios. Esta integración transforma el chat en un sistema de recomendaciones diseñado específicamente para entrenadores de natación, permitiendo optimizar sus decisiones diarias a través de datos procesados y contextualizados.

### Flujo de la integración

#### Recepción de la consulta del usuario

El proceso comienza con la entrada del usuario en el chat, donde se captura su pregunta o solicitud. Este mensaje inicial es gestionado por la función handleSendMessage, que organiza el flujo de datos necesario para procesar la consulta.

#### Construcción del prompt dinámico

La función createQuery genera un prompt altamente contextualizado que incluye:

• El esquema de la base de datos relevante para el sistema, obtenido dinámicamente.

• El mensaje proporcionado por el usuario.

• Instrucciones específicas para guiar a GPT-4, como generar respuestas claras, eliminar datos técnicos innecesarios y proporcionar recomendaciones directamente aplicables.

Este prompt se envía a GPT-4 como base para generar una consulta SQL o una respuesta textual avanzada.

#### Ejecución de consultas SQL generadas por IA

Si la respuesta de GPT-4 incluye una consulta SQL, esta se ejecuta mediante la función executeQuery. Esta función interactúa con la base de datos PostgreSQL para recuperar datos específicos relacionados con el entrenamiento, el rendimiento de los nadadores o cualquier información solicitada. Los resultados se transforman y se preparan para su posterior procesamiento.

#### Generación de recomendaciones finales

Una vez obtenidos los datos relevantes, se envían nuevamente a GPT-4 junto con el contexto general e instrucciones específicas. Este paso permite a GPT-4 generar respuestas finales que:

• Contextualicen los datos para su fácil interpretación.

• Ofrezcan recomendaciones claras y prácticas.

• Organicen la información en puntos clave si es necesario.

#### Presentación en tiempo real

La respuesta generada por GPT-4 es transmitida al usuario en tiempo real. El sistema utiliza un flujo de datos continuo, decodificando los fragmentos recibidos desde la API de OpenAI para mostrar el contenido de manera fluida en la interfaz del chat.

# Lecciones aprendidas y análisis DAFO

Durante el desarrollo del proyecto, se han obtenido múltiples aprendizajes clave. Una de las lecciones más importantes ha sido la capacidad de combinar tecnologías modernas con herramientas de IA, lo que ha elevado significativamente el potencial del producto. La importancia de planificar la arquitectura desde el inicio quedó clara, ya que permitió integrar la inteligencia artificial de manera natural con el resto de las funcionalidades.

En el análisis DAFO, las fortalezas principales incluyen la incorporación de tecnologías innovadoras, la integración de IA y la escalabilidad de la solución. Sin embargo, también se identificaron algunas debilidades, como la complejidad inicial en la configuración de los modelos de IA y la necesidad de recursos computacionales significativos. En términos de oportunidades, la IA ofrece un campo vasto para expandir las funcionalidades, como análisis en tiempo real y recomendaciones más precisas. No obstante, las amenazas incluyen la dependencia de infraestructura avanzada y la rápida evolución del sector tecnológico.

# Visión de futuro

El proyecto tiene una visión ambiciosa que busca convertirlo en una referencia dentro del ámbito deportivo. En el corto plazo, se prevé mejorar los modelos de inteligencia artificial, optimizando la precisión de las predicciones y recomendaciones. Además, se planea expandir la integración con dispositivos IoT para capturar métricas más detalladas y en tiempo real.

A largo plazo, la aplicación podría evolucionar hacia una plataforma multipropósito, adaptándose a otros deportes y expandiendo su mercado. La implementación de un sistema de gamificación, que incluya rankings y recompensas, está en los planes para mejorar la participación de los usuarios. Asimismo, se contempla la creación de un marketplace donde los deportistas puedan acceder a productos y servicios relacionados con su actividad. Estas iniciativas, combinadas con el uso continuo de inteligencia artificial, aseguran que la aplicación permanezca a la vanguardia del sector.

# Bibliografía

1. Documentación oficial de Next.js: <https://nextjs.org/docs>

2. Guía de Prisma ORM: <https://www.prisma.io/docs>

3. TailwindCSS: <https://tailwindcss.com/docs>

4. Documentación de Recharts: <https://recharts.org/en-US>

5. API de OpenAI: <https://platform.openai.com/>

6. RAG2SQL: <https://medium.com/@marvin_thompson/text2sql-is-out-rag2sql-is-in-5fd160a004f0>