

# MEMORIA DEL PROYECTO

TIRAKA

Autor/s:

Isabel Yangxin Moreno Basurto

## 1.1 MEMORIA

# INDICE DE CONTENIDO

<b>1. ANTECEDENTES</b>	<b>3</b>
<b>2. OBJETO</b>	<b>4</b>
<b>3. SITUACIÓN ACTUAL – TO BE</b>	<b>5</b>
3.1. Situación actual (AS IS)	5
3.2. Situación esperada (TO BE)	5
<b>4. JUSTIFICACIÓN DE LA SOLUCIÓN ADOPTADA – AS IS</b>	<b>6</b>
4.1 Tecnologías escogidas	6
4.2 Motor de bases de datos	6
<b>5. DESCRIPCIÓN DEL PROYECTO</b>	<b>7</b>
5.1 Funcionalidades principales	7
5.2 Mapa del sitio	7
5.3 Modelo de la base de datos	8
5.4 Organización de archivos	9
5.4.1 Backend (/back)	9
5.4.2 Frontend (/front)	9
5.4.3 Dependencias y configuración	9
5.5 Seguridad y privacidad	10
<b>6. PRESUPUESTOS</b>	<b>11</b>
6.1. Hardware y equipo	11
6.2. Desarrollo tecnológico	11
6.3. Infraestructura y servidores	11
6.4. Diseño y software adicional	11
<b>7. PLAZO DE EJECUCIÓN</b>	<b>12</b>
7.1. Etapas del desarrollo	12
7.1.1. Fase 1: Planificación y organización (Marzo 2025)	12
7.1.2. Fase 2: Desarrollo del frontend y backend (Abril 2025)	13
7.1.3. Fase 3: Pruebas y ajustes (Mayo 2025)	13
7.1.4. Fase 3: Continuidad (Junio 2025)	13
7.2 Metodología del trabajo	14
<b>8. DOCUMENTOS INTEGRANTES DEL PROYECTO</b>	<b>15</b>

## Memoria del Proyecto

---

8.1 Código fuente	15
8.2 Script SQL de la base de datos	15
8.3 Manual de instalación (en local)	16
8.3.1 Requisitos previos	16
8.3.2 Clonar el proyecto desde GitHub	16
8.3.3 Instalar las dependencias	16
8.3.4 Crear el archivo de configuración (.env)	16
8.3.5 Iniciar servidor y abrir la web desde el navegador	17
8.5. Enlace al proyecto	17
<b>9. CONCLUSIÓN FINAL</b>	<b>18</b>

## 1. ANTECEDENTES

La esgrima es un deporte con una larga tradición, tanto a nivel competitivo como formativo. Desde sus orígenes como disciplina militar en Europa hasta su evolución como deporte olímpico, ha pasado por numerosos cambios. Las armas, los trajes, los sistemas de puntuación e incluso los formatos de las competiciones han sido modernizados.

Sin embargo, mientras que la tecnología ha influido de forma notable en los aspectos materiales del deporte (por ejemplo, la electrónica en el arbitraje), no ha tenido el mismo impacto en la parte organizativa y de seguimiento. Hoy en día, muchos clubes y tiradores siguen registrando sus entrenamientos y competiciones de forma manual, en hojas de papel o simplemente de memoria.

Este método tradicional presenta varias limitaciones: se pierde información valiosa, no se permite un análisis a largo plazo del rendimiento, y se hace difícil llevar una planificación coherente, especialmente si hay varios entrenadores o si el tirador participa en muchas competiciones.

Por ejemplo, es común que un tirador no tenga constancia clara de cuántas competiciones ha hecho en una temporada, cuántos asaltos ha ganado o cuál ha sido su evolución técnica en los entrenamientos. Todo esto dificulta la toma de decisiones, tanto por parte del propio deportista como por parte del entrenador.

Además, mientras otros deportes han comenzado a aplicar herramientas de análisis y registro (como apps o software de entrenamiento), la esgrima aún no cuenta con muchas soluciones especializadas, y las que existen suelen estar orientadas al alto rendimiento o son demasiado genéricas.

Esta falta de digitalización no solo afecta al seguimiento deportivo, sino también a la organización interna de los clubes, que a menudo manejan los datos de sus miembros y actividades sin una estructura unificada.

Por tanto, existe una necesidad real y actual de modernizar el seguimiento deportivo en esgrima, adaptándolo a las herramientas digitales que ya se usan en otros ámbitos.

## 2. OBJETO

El objetivo principal de este proyecto es desarrollar una plataforma web específica para la comunidad de esgrima, centrada en mejorar el seguimiento del progreso de los tiradores y optimizar la comunicación con los entrenadores. Esta plataforma pretende cubrir una necesidad concreta del ámbito deportivo: la falta de herramientas digitales adaptadas a los deportes individuales como la esgrima.

La aplicación está dirigida tanto a tiradores como a entrenadores, permitiendo a los primeros registrar de forma organizada sus entrenamientos, competiciones, sensaciones y resultados, mientras que los entrenadores podrán consultar esta información y planificar sesiones más personalizadas y eficaces. Gracias a esta herramienta, será posible llevar un control más detallado del rendimiento de cada tirador a lo largo del tiempo.

Con una interfaz accesible y orientada al usuario, la plataforma ofrecerá funcionalidades como: Registro de sesiones de entrenamiento con fecha, tipo de ejercicio y observaciones.

Historial de competiciones y resultados obtenidos.

Estadísticas de progreso por período (mensual, trimestral, anual).

Sistema de comentarios entre tiradores y entrenadores.

Posibilidad de planificación a corto, medio y largo plazo.

Este enfoque busca mejorar no solo la organización y el análisis del rendimiento, sino también fortalecer la relación entre entrenador y deportista, creando un espacio compartido donde ambos puedan trabajar de manera más coordinada.

A nivel técnico, la plataforma estará desarrollada con tecnologías web modernas como Node.js, PostgreSQL y Express, lo que permitirá su fácil implementación y mantenimiento. Además, su diseño modular permitirá futuras mejoras, como la inclusión de gráficos, notificaciones o incluso funciones sociales entre miembros del club.

En definitiva, el objeto de este proyecto es crear una solución útil, práctica y adaptable que contribuya a modernizar el entrenamiento en esgrima, permitiendo aprovechar los beneficios de la tecnología para un mejor rendimiento deportivo.

### **3. SITUACIÓN ACTUAL – TO BE**

#### **3.1. SITUACIÓN ACTUAL (AS IS)**

Actualmente, el seguimiento del progreso en esgrima se realiza de manera manual y poco estructurada. Los tiradores y entrenadores suelen depender de libretas, hojas de cálculo o simplemente de la memoria para registrar entrenamientos y competiciones. Esto genera varios problemas:

- Falta de organización: La información no siempre está bien estructurada, lo que dificulta el análisis del rendimiento y la planificación de entrenamientos.
- Dificultad para compartir datos: Los entrenadores no tienen un canal eficiente para enviar rutinas personalizadas o hacer ajustes en tiempo real.
- Poca accesibilidad: Si un tirador quiere revisar su progreso o recibir indicaciones fuera del entrenamiento, muchas veces no tiene una herramienta que le permita hacerlo fácilmente.

#### **3.2. SITUACIÓN ESPERADA (TO BE)**

Con la implementación de la plataforma web, la gestión del entrenamiento en esgrima dará un salto hacia la digitalización y la eficiencia. La herramienta permitirá:

- Registro estructurado y accesible: Los tiradores podrán almacenar sus entrenamientos, competiciones y estadísticas en un solo lugar, con datos organizados y fáciles de consultar.
- Comunicación fluida: Los entrenadores podrán enviar rutinas personalizadas, hacer ajustes en tiempo real y dar seguimiento al progreso de cada deportista sin necesidad de esperar a la próxima sesión presencial.
- Análisis de rendimiento: La plataforma ofrecerá gráficos y métricas que ayudarán a visualizar la evolución de cada tirador, facilitando la toma de decisiones en el entrenamiento.
- Accesibilidad desde cualquier lugar: Tanto entrenadores como tiradores podrán acceder a la información desde cualquier dispositivo, asegurando que el seguimiento del progreso no dependa de registros físicos o notas dispersas.

## **4. JUSTIFICACIÓN DE LA SOLUCIÓN ADOPTADA – AS IS**

Para solucionar estos problemas, se ha optado por desarrollar una plataforma web que centralice la información, facilite el análisis del rendimiento y mejore la interacción entre entrenadores y deportistas. Con esta herramienta, los tiradores podrán registrar sus entrenamientos y competiciones de manera estructurada, mientras que los entrenadores podrán diseñar rutinas adaptadas y hacer ajustes en tiempo real.

La digitalización del proceso permitirá una gestión más clara y accesible, optimizando el entrenamiento y asegurando que cada tirador pueda seguir su evolución de forma precisa.

### **4.1 TECNOLOGÍAS ESCOGIDAS**

La aplicación se desarrolla principalmente con tecnologías del lado del cliente como HTML, CSS y JavaScript, junto con Node.js para la parte del servidor. Esta combinación permite una experiencia web moderna, interactiva y rápida.

El uso de Node.js como entorno de ejecución para JavaScript en el servidor permite aprovechar un mismo lenguaje tanto en el cliente como en el servidor, facilitando el desarrollo y mantenimiento del proyecto. Además, Node.js destaca por su rendimiento y su modelo de operación asíncrono, ideal para aplicaciones en tiempo real o con alta concurrencia.

Todo el software utilizado es de código abierto, lo que representa un ahorro importante en licencias para la empresa. Más allá de su gratuidad, estas tecnologías están ampliamente respaldadas por la comunidad y se encuentran entre las más populares y fiables del desarrollo web actual.

### **4.2 MOTOR DE BASES DE DATOS**

El motor de bases de datos utilizado es PostgreSQL. Las razones de su elección son las siguientes:

- Es de código abierto y gratuito, lo que permite una reducción significativa de costes para el cliente.
- Es multiplataforma, funcionando en Windows, Linux y Mac, lo que asegura compatibilidad con la mayoría de sistemas operativos actuales.
- Goza de gran popularidad en la comunidad de desarrolladores, lo que facilita encontrar documentación, tutoriales y soporte técnico.

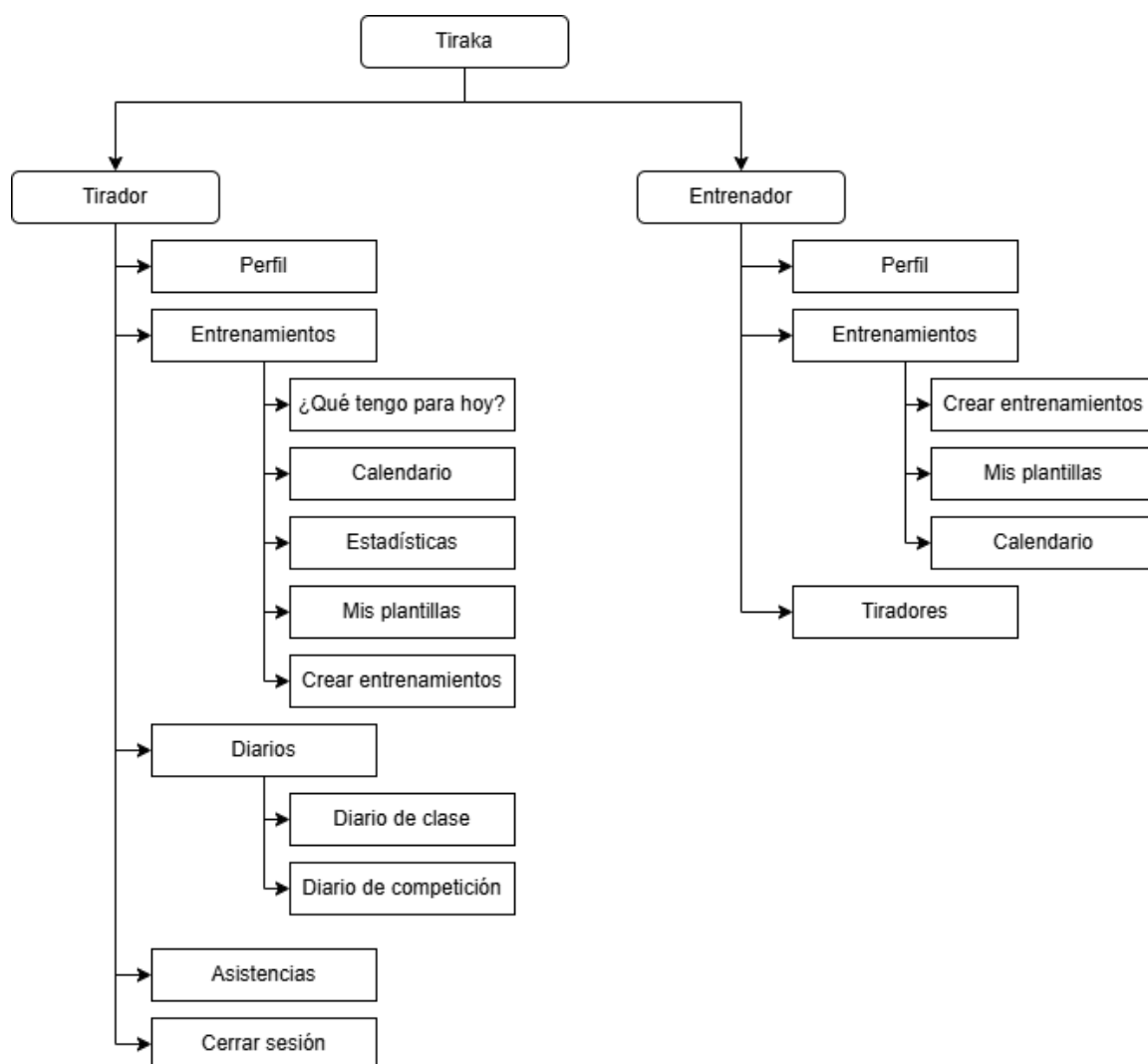


## 5. DESCRIPCIÓN DEL PROYECTO

### 5.1 FUNCIONALIDADES PRINCIPALES

- Registro de entrenamientos, competiciones y clases, con detalles como fecha, duración, tipo de ejercicio y resultados.
- Análisis de rendimiento, mostrando estadísticas y gráficos para evaluar la evolución de cada tirador.
- Gestión de rutinas personalizadas, permitiendo a los entrenadores asignar ejercicios y ajustar entrenamientos según el progreso.
- Acceso desde cualquier dispositivo, asegurando disponibilidad en todo momento.

### 5.2 MAPA DEL SITIO



[illegible]

- **Usuarios:** Se clasifican en esgrimistas y entrenadores, almacenando sus datos personales y credenciales.
- **Entrenamientos:** Registros detallados de sesiones, incluyendo duración, tipo de ejercicio y plantillas de entrenamiento.
- **Competiciones:** Seguimiento de torneos, posiciones obtenidas y detalles de cada etapa.
- **Relación tirador-entrenador:** Registro de entrenamientos personalizados y sesiones específicas según el progreso del deportista.
- **Sistema de asistencia:** Control de asistencia cada día.
- **Plantillas:** Las plantillas de entrenamiento agilizará el proceso de planificación y garantizará una metodología de trabajo eficiente
- **Diarios:** Permitirán a los tiradores registrar su experiencia personal

## 5.4 ORGANIZACIÓN DE ARCHIVOS

### 5.4.1 Backend (/back)

Contiene toda la lógica del servidor, incluyendo:

- /functions: Funciones específicas del backend.
- /routes: Definición de rutas y endpoints.

### 5.4.2 Frontend (/front)

El lado del cliente está estructurado con:

- /css: Archivos de estilos.
- /font: Fuentes utilizadas.
- /img: Imágenes del proyecto.
- /js: Código JavaScript separado por módulos:
  - /coach: Scripts específicos para la vista de entrenadores.
  - /fencer: Scripts para los tiradores.
  - archivos.js: Archivo general para la gestión de archivos.
- archivos.html: Página relacionada con la gestión de archivos.

### 5.4.3 Dependencias y configuración

- /node\_modules: Paquete de dependencias instaladas con Node.js.
- db.js: Configuración de la base de datos.
- index.js: Archivo principal del servidor.
- package.json: Gestión de dependencias y scripts del proyecto.

```
tiraka
|
| /back
|   | /functions
|   | /routes
|
| /front
|   | /css
|   | /font
|   | /img
|   | /js
|   |   | /coach
|   |   | /fencer
|   |   | archivos.js
|   |   |
|   |   | archivos.html
|
| /node_modules
| db.js
| index.js
| package.json
```

## **5.5 SEGURIDAD Y PRIVACIDAD**

En primer lugar, las contraseñas de los usuarios se almacenan utilizando técnicas de cifrado robustas mediante la librería bcryptjs, que aplica un hash seguro para evitar el acceso no autorizado incluso en caso de vulneración de la base de datos.

Además, el sistema de autenticación y gestión de sesiones se ha desarrollado usando la librería express-session, que establece sesiones seguras para cada usuario, evitando accesos indebidos y garantizando que solo los usuarios autorizados puedan acceder a sus datos.

Las comunicaciones entre el cliente y el servidor están diseñadas para ser compatibles con conexiones seguras (HTTPS), especialmente en entornos de producción, para evitar interceptaciones y ataques de tipo "man-in-the-middle".

## 6. PRESUPUESTOS

Este proyecto se está desarrollando de manera independiente, lo que reduce costos en personal, pero aún requiere inversión en software, hardware y alojamiento.

### 6.1. HARDWARE Y EQUIPO

- Ordenador portátil o PC → 1.200€ - 1.500€ (según especificaciones)
- Monitor adicional → 150€ - 300€
- Teclado y ratón → 50€ - 100€
- Auriculares/micrófono → 50€ - 150€

### 6.2. DESARROLLO TECNOLÓGICO

- Visual Studio Code (IDE) → 0€ (gratuito)
- Node.js y librerías → 0€ (open source)
- PostgreSQL (Base de datos) → 0€ (gratis en su versión estándar)
- Railway (Cloud hosting para BD y backend) → 5-20€/mes, dependiendo del tráfico

### 6.3. INFRAESTRUCTURA Y SERVIDORES

- Alojamiento en Railway → 10-50€/mes
- Dominio web → 12-15€/año
- Certificado SSL → 0€ (let's encrypt)

### 6.4. DISEÑO Y SOFTWARE ADICIONAL

- Diseño UI/UX → 0€
- Elementos gráficos (iconos, imágenes) → 0€

## 7. PLAZO DE EJECUCIÓN

El proyecto Tiraka se inició en marzo de 2025, marcando el comienzo de un proceso de desarrollo que ha seguido un ritmo constante y estructurado. Desde el inicio, se ha trabajado en la implementación de funcionalidades clave, la optimización de la plataforma y la corrección de errores, con el objetivo de ofrecer una herramienta robusta y útil para la comunidad de esgrima.

La planificación inicial estableció un plazo para que la plataforma estuviera lista en junio de 2025, coincidiendo con el inicio de la temporada 2025/2026. Sin embargo, se contempla la posibilidad de continuar mejorando y puliendo detalles más allá de esta fecha para asegurar que la experiencia de usuario sea óptima y que la aplicación cumpla con las expectativas de deportistas y entrenadores.

A continuación, se describen las etapas principales en las que se ha dividido el desarrollo, detallando las actividades y objetivos de cada una.

### 7.1. ETAPAS DEL DESARROLLO

#### 7.1.1. Fase 1: Planificación y organización (Marzo 2025)

Esta etapa inicial fue fundamental para sentar las bases del proyecto. Se llevaron a cabo las siguientes tareas:

- Definición de las funcionalidades principales que debía incluir la plataforma, considerando las necesidades reales de los tiradores y entrenadores.
- Diseño preliminar del esquema de la base de datos, asegurando que la estructura permitiera almacenar y relacionar adecuadamente la información sobre usuarios, entrenamientos, competiciones y estadísticas.
- Selección de las tecnologías más adecuadas para el desarrollo, optando por Node.js para el backend, PostgreSQL como sistema gestor de bases de datos y Railway para el despliegue en la nube.
- Creación de un plan de trabajo con objetivos mensuales y entregables, para garantizar el avance controlado del proyecto.

Esta fase sentó las bases para un desarrollo organizado y eficiente.

### **7.1.2. Fase 2: Desarrollo del frontend y backend (Abril 2025)**

En esta fase se comenzó con la construcción del sistema propiamente dicho, con las siguientes actividades:

- Desarrollo del backend utilizando Node.js y PostgreSQL, implementando la lógica de negocio, las rutas, la conexión a la base de datos y la gestión de sesiones y autenticación de usuarios.
- Diseño y programación del frontend, enfocándose en una interfaz intuitiva, accesible y visualmente atractiva, que facilitara la interacción del usuario con la plataforma.
- Implementación del sistema de registro y login para usuarios, incluyendo medidas básicas de seguridad como el cifrado de contraseñas.
- Integración entre frontend y backend para asegurar la comunicación efectiva y la correcta gestión de datos.

Durante esta etapa, se realizaron pruebas unitarias iniciales para verificar que cada módulo funcionaba correctamente.

### **7.1.3. Fase 3: Pruebas y ajustes (Mayo 2025)**

Con la mayoría de funcionalidades desarrolladas, esta fase se centró en garantizar la calidad y estabilidad del sistema:

- Ejecución de pruebas exhaustivas para detectar errores o comportamientos inesperados, abarcando tanto la lógica backend como la experiencia de usuario en el frontend.
- Optimización del rendimiento, reduciendo tiempos de carga y mejorando la eficiencia de consultas a la base de datos.
- Fortalecimiento de la seguridad, corrigiendo posibles vulnerabilidades detectadas durante las pruebas.
- Ajustes visuales y de usabilidad para hacer la plataforma más intuitiva y agradable para el usuario.
- Despliegue de la aplicación en Railway, verificando la correcta configuración del entorno de producción y estabilidad del servidor.
- Pruebas finales con usuarios reales para recoger feedback y detectar mejoras antes del lanzamiento oficial.

Publicación de la plataforma y apertura para pruebas públicas.

### **7.1.4. Fase 3: Continuidad (Junio 2025)**

- Seguimiento constante de la plataforma para identificar posibles errores o fallos.
- Corrección rápida de incidencias reportadas por usuarios o detectadas por el equipo.
- Optimización adicional del rendimiento y refuerzo de la seguridad.
- Mejoras en la interfaz de usuario basadas en la retroalimentación recibida.
- Planificación de nuevas funcionalidades para futuras versiones.

Esta fase garantiza que Tiraka se mantenga actualizada, estable y útil para su comunidad a largo plazo.

## **7.2 METODOLOGÍA DEL TRABAJO**

El desarrollo del proyecto Tiraka se organizó y ejecutó trabajando de forma semanal, dividiendo las tareas en objetivos claros para cada semana. Este enfoque permitió mantener un ritmo constante, facilitar el seguimiento del progreso y ajustar las prioridades según las necesidades que iban surgiendo.

Cada semana se definían metas concretas, como implementar funcionalidades específicas, corregir errores detectados o mejorar la interfaz de usuario. Al finalizar cada semana, se revisaban los avances y se planificaban las tareas para la siguiente, favoreciendo una mejora continua y un desarrollo iterativo.

Para el control y seguimiento del código, se utilizó Git y GitHub, que además permitieron mantener un historial detallado de los cambios realizados y facilitar la organización del trabajo. Esta práctica también hizo posible volver a versiones anteriores si era necesario y gestionar adecuadamente las distintas ramas de desarrollo.



## 8. DOCUMENTOS INTEGRANTES DEL PROYECTO

### 8.1 CÓDIGO FUENTE

El código fuente del proyecto Tiraka se encuentra disponible en un repositorio público de GitHub, al que se puede acceder desde el siguiente enlace:

<https://github.com/isabelamorena/Tiraka>

Este repositorio contiene todos los archivos necesarios para la ejecución del proyecto, organizados en carpetas que separan claramente las distintas partes del sistema:

- Frontend: Archivos HTML, CSS y JavaScript que conforman la interfaz de usuario.
- Backend: Código en Node.js que gestiona la lógica de negocio, rutas, controladores y conexión con la base de datos.
- Configuraciones: Archivos .env.example, package.json y otros ficheros de configuración necesarios para la instalación y despliegue local o en producción.

Además, se han seguido buenas prácticas de desarrollo como la organización modular del código, uso de control de versiones y documentación mínima dentro del mismo repositorio para facilitar su mantenimiento y comprensión por parte de otros desarrolladores.

### 8.2 SCRIPT SQL DE LA BASE DE DATOS

El proyecto incluye un script SQL con el esquema de la base de datos utilizado en la aplicación. Este script define la estructura básica necesaria para el funcionamiento del sistema, incluyendo:

- La creación de las tablas principales (usuarios, entrenamientos, competiciones, etc.).
- Las relaciones entre tablas mediante claves foráneas.
- Algunos valores por defecto o pruebas para facilitar la puesta en marcha inicial.

El script se encuentra disponible en el mismo repositorio de GitHub, en la ruta:

[https://github.com/isabelamorena/Tiraka/blob/main/tiraka\\_db.sql](https://github.com/isabelamorena/Tiraka/blob/main/tiraka_db.sql)

Este archivo permite crear la base de datos local en cualquier sistema compatible con PostgreSQL, facilitando así la reproducción del entorno por parte de otros desarrolladores o revisores del proyecto.

## 8.3 MANUAL DE INSTALACIÓN (EN LOCAL)

### 8.3.1 Requisitos previos

Antes de empezar, necesitas tener instalados en tu computadora:

- Node.js: Descargar desde la web oficial (elige la versión LTS)
- PostgreSQL: Descargar desde la web oficial
- Un editor de código como Visual Studio Code
- Una terminal o consola

### 8.3.2 Clonar el proyecto desde GitHub

1. Abre la terminal.
2. Escribe este comando y pulsa Enter: `git clone https://github.com/tu-usuario/nombre-del-repo.git`
3. Entra a la carpeta del proyecto: `cd nombre-del-repo`

### 8.3.3 Instalar las dependencias

En la misma terminal, escribe: `npm install`

Este comando descargará todo lo necesario para que el proyecto funcione correctamente.

- Express: para manejar rutas del servidor web
- Pg: para conectarse a la base de datos PostgreSQL
- dotenv: para leer variables desde un archivo `.env`.
- bcryptjs: para encriptar contraseñas.
- cookie-parser: para leer cookies del navegador.
- cors: para permitir peticiones entre distintas direcciones.
- express-session: para manejar sesiones de usuarios.

### 8.3.4 Crear el archivo de configuración (.env)

Este archivo guarda la información de conexión a la base de datos y otros datos sensibles.

Crea un archivo llamado `.env` en la carpeta principal del proyecto (donde está `index.js`) y escribe lo siguiente:

```
# URL de conexión a la base de datos PostgreSQL
```

```
DATABASE_URL=postgresql://USUARIO:CONTRASEÑA@localhost:5432/NOMBRE_DE_LA_BD
```

# Clave secreta para las sesiones

SESSION\_SECRET=miclaveultrasecreta

# Puerto en el que se ejecutará el servidor

PORT=3000

### **8.3.5 Iniciar servidor y abrir la web desde el navegador**

- En la terminal del proyecto escribe: `node index.js`
- Debería salir un mensaje así: Servidor corriendo en el puerto 3000
- Abre tu navegador y visita: <http://localhost:3000/home.html>

## **8.5. ENLACE AL PROYECTO**

El proyecto Tiraka está disponible online y puede ser accedido públicamente a través del siguiente enlace:

<https://tiraka.up.railway.app/>

La plataforma está alojada en un servidor en la nube, lo que garantiza su disponibilidad y acceso desde cualquier dispositivo con conexión a internet. Se recomienda a los usuarios acceder mediante navegadores actualizados para una experiencia óptima.

## 9. CONCLUSIÓN FINAL

Después de cuatro meses de desarrollo, Tiraka ha pasado por distintas etapas, desde la planificación inicial hasta su implementación. A lo largo del proceso, se han definido las funcionalidades clave, estructurado la base de datos y diseñado una interfaz intuitiva para mejorar el seguimiento de los entrenamientos en esgrima.

Con esta plataforma, los tiradores y entrenadores podrán gestionar entrenamientos, competencias y análisis de rendimiento de manera organizada y accesible. La digitalización facilita la recopilación de datos y optimiza la comunicación dentro de la comunidad deportiva.

Aunque el desarrollo principal ha concluido, siempre habrá oportunidad para ajustes y mejoras. Tiraka tiene una base sólida y puede evolucionar con nuevas funcionalidades según las necesidades de los usuarios.

En Madrid, Alcorcón el día 09 de Junio de 2025

*Fda.: Isabel Yangxin Moreno Basurto*



## 1.2 ANEJOS A LA MEMORIA

No se incluyen anejos, ya que toda la información relevante está integrada dentro del cuerpo de la memoria.