

**CIFP Juan de Colonia**

**Informática y Comunicaciones**

**G.S. Desarrollo de Aplicaciones Web**



**RacingTracker**

Desarrollado por:

**Víctor García Murillo**

19 / 05 / 2025

# Índice de contenidos

[Índice de contenidos 2](#_Toc197033072)

[Índice de ilustraciones 4](#_Toc197033073)

[Índice de tablas 5](#_Toc197033074)

[Planificación 6](#_Toc197033075)

[Presentación del problema planteado 6](#_Toc197033076)

[Objetivos 6](#_Toc197033077)

[Requisitos 7](#_Toc197033078)

[Requisitos funcionales 7](#_Toc197033079)

[Requisitos no funcionales 7](#_Toc197033080)

[Recursos 7](#_Toc197033081)

[Recursos de Hardware 7](#_Toc197033082)

[Recursos de Software 7](#_Toc197033083)

[Tecnologías 8](#_Toc197033084)

[Planificación del trabajo 9](#_Toc197033085)

[Planificación económica 9](#_Toc197033086)

[Análisis 10](#_Toc197033087)

[Público objetivo 10](#_Toc197033088)

[Nombre de la aplicación 10](#_Toc197033089)

[Diagrama Entidad-Relación 11](#_Toc197033090)

[Diagrama de la aplicación 12](#_Toc197033091)

[Diagrama de casos de uso 14](#_Toc197033092)

[Diseño 15](#_Toc197033093)

[Prototipo de la interfaz 15](#_Toc197033094)

[Primeros prototipos 15](#_Toc197033095)

[Prototipos finales 17](#_Toc197033096)

[Diagrama relacional 19](#_Toc197033097)

[Paleta de colores 20](#_Toc197033098)

[Tipografía 21](#_Toc197033099)

[Pruebas 22](#_Toc197033100)

[Diario de trabajo 23](#_Toc197033101)

[Conclusiones 29](#_Toc197033102)

[Objetivos cumplidos 29](#_Toc197033103)

[Propuestas de mejora y ampliaciones futuras 29](#_Toc197033104)

[Reflexiones sobre el proyecto 29](#_Toc197033105)

[Guía 30](#_Toc197033106)

[Despliegue 30](#_Toc197033107)

[Guía de administración 30](#_Toc197033108)

[Guía de usuario 30](#_Toc197033109)

[Bibliografía 31](#_Toc197033110)

# Índice de ilustraciones

[Ilustración 1: Diagrama de Gantt. 9](#_Toc197033058)

[Ilustración 2: Diagrama Entidad-Relación de la base de datos. 11](#_Toc197033059)

[Ilustración 3: Diagrama de aplicación 12](#_Toc197033060)

[Ilustración 4: Diagrama del repositorio. 13](#_Toc197033061)

[Ilustración 5: Diagrama de casos de uso. 14](#_Toc197033062)

[Ilustración 6: Boceto de la página de pilotos y su perfil. 15](#_Toc197033063)

[Ilustración 7: Boceto de la página de escuderías y su perfil. 16](#_Toc197033064)

[Ilustración 8: Boceto del funcionamiento del menú de navegación. 17](#_Toc197033065)

[Ilustración 9: Boceto de la página de inicio. 18](#_Toc197033066)

[Ilustración 10: Boceto de la página de pilotos. 19](#_Toc197033067)

[Ilustración 11: Diagrama relacional de la base de datos. 19](#_Toc197033068)

[Ilustración 12: Paleta de colores. 20](#_Toc197033069)

[Ilustración 13: Contraste de la paleta de colores. 20](file:///C:\Users\Victor\Documents\dev\RacingTracker\documents\memoria.docx#_Toc197033070)

[Ilustración 14: Ejemplo gráfico de las tipografías. 21](file:///C:\Users\Victor\Documents\dev\RacingTracker\documents\memoria.docx#_Toc197033071)

# Índice de tablas

[Tabla 1: Planificación económica. 9](#_Toc197033014)

[Tabla 2: Pruebas realizadas. 22](#_Toc197033015)

# Planificación

## Presentación del problema planteado

En las competiciones de motor, el sistema de puntuación predominante se basa exclusivamente en la posición final de los participantes. Este enfoque tradicional asigna puntuaciones fijas a cada puesto sin considerar factores clave como el rendimiento relativo, las expectativas previas o las condiciones particulares de cada piloto o equipo. Como consecuencia, el sistema actual presenta dos deficiencias principales:

1. **Falta de reconocimiento al desempeño relativo**: Un piloto o equipo con recursos limitados que logra un resultado modesto podría estar rindiendo a un nivel excepcional en función de sus capacidades, pero el sistema no refleja este mérito.
2. **Poca valoración de la superación de expectativas**: Un piloto de un equipo líder que finaliza en una posición alta simplemente cumple con lo esperado, sin recibir reconocimiento adicional si su desempeño supera las proyecciones iniciales.

Estas limitaciones plantean una pregunta fundamental: ¿es realmente equitativo un sistema de puntuación que solo considera la posición final sin evaluar el contexto ni el rendimiento relativo? La respuesta, desde mi perspectiva, es no. Por ello, propongo el desarrollo de un sistema de clasificación más justo y dinámico, capaz de reflejar con mayor precisión el verdadero mérito de cada piloto y equipo.

## Objetivos

1. **Desarrollar un sistema eficiente para la gestión de carreras**: Diseñar una aplicación web intuitiva y funcional que facilite la organización y administración de competiciones de motor, optimizando la gestión de participantes, clasificaciones y resultados.
2. **Implementar un sistema de puntuación equitativo**: Integrar un modelo basado en TrueSkill (sistema de clasificación desarrollado por Microsoft Research para el servicio de clasificación y matchmaking), que no solo evalúe la posición final de los participantes, sino que también considere factores como el rendimiento esperado y la evolución del desempeño a lo largo del tiempo. Este enfoque permitirá reconocer tanto a los ganadores como a quienes superan sus propios límites y expectativas.
3. **Optimizar la eficiencia y experiencia de usuario**: Garantizar el máximo rendimiento de la plataforma mediante una interfaz intuitiva, tiempos de carga reducidos y una navegación fluida. El objetivo es ofrecer una experiencia excepcional tanto para aficionados y competidores como para las organizaciones encargadas de gestionar las carreras.

## Requisitos

### Requisitos funcionales

* **Gestión de competiciones**: Los usuarios con permisos de administrador podrán gestionar pilotos, escuderías y carreras.
* **Visualización de estadísticas**: Todos los usuarios podrán acceder a:
  + Estadísticas detalladas de cada temporada, incluyendo puntos obtenidos por pilotos y equipos en cada carrera.
  + Historial de clasificaciones y las mejores estadísticas de toda la historia tanto de escuderías como de pilotos.
  + Listado de carreras disputadas en cada temporada.
  + Perfiles de pilotos y escuderías con sus respectivas estadísticas.

### Requisitos no funcionales

* Accesible desde dispositivos de escritorio y móviles.
* Disponible en inglés.
* Probada en navegadores basados en Firefox y Chrome.

## Recursos

### Recursos de Hardware

* Entorno de escritorio Windows 11.
* Dispositivo móvil.

### Recursos de Software

* **Visual Studio Code**: Editor de código desarrollado por Microsoft. Incluye características como autocompletado de código, control de versiones con Git integrado, y una gran cantidad de extensiones que mejoran su funcionalidad, como soporte para frameworks y herramientas de desarrollo.
* **Firefox y Chrome**: Navegadores web.
* **AWS**: Plataforma de servicios en la nube que ofrece herramientas para gestionar aplicaciones sin necesidad de infraestructura física.

### Tecnologías

* **NodeJS**: Entorno de ejecución de JavaScript que permite ejecutar código desde el servidor.
* **SQLite**: Sistema de gestión de bases de datos relacional ligero y autónomo, que se utiliza para almacenar datos de manera estructurada en aplicaciones sin la necesidad de requierir un servidor separado para funcionar; los datos se guardan directamente en un archivo en el sistema de archivos del dispositivo.
* **Laravel + React**: Combinación de Laravel, un framework PHP (backend), y React, una biblioteca de JavaScript (frontend). Laravel se encarga de la lógica del servidor, la gestión de bases de datos y la autenticación, mientras que React maneja la interfaz de usuario. Proporciona herramientas y configuraciones prediseñadas para facilitar el desarrollo del frontend, permitiendo una integración eficiente entre ambos.
* **Eloquent**: Sistema ORM de Laravel que permite interactuar con bases de datos usando PHP. Convierte las filas de la base de datos en objetos sin necesidad de escribir SQL.
* **Tailwind CSS**: Framework de CSS de bajo nivel que permite crear diseños mediante clases predefinidas. A diferencia de otros frameworks, Tailwind no ofrece componentes listos para usar, sino que proporciona un conjunto de clases que se pueden combinar.
* **PHP**: Lenguaje de programación diseñado para el desarrollo web. Se ejecuta en el servidor.
* **TypeScript**: Superset de JavaScript que añade tipado estático y características de programación orientada a objetos.
* **TSX**: Extensión de sintaxis que se usa en TypeScript para trabajar con JSX, un formato de sintaxis similar a HTML utilizado en React. TSX permite escribir componentes React dentro de archivos TypeScript.
* **Nginx**: Servidor web que aloja aplicaciones, gestionando las solicitudes HTTP, sirviendo contenido estático y enviando las solicitudes PHP a PHP-FPM para procesar el código de Laravel.

## Planificación del trabajo

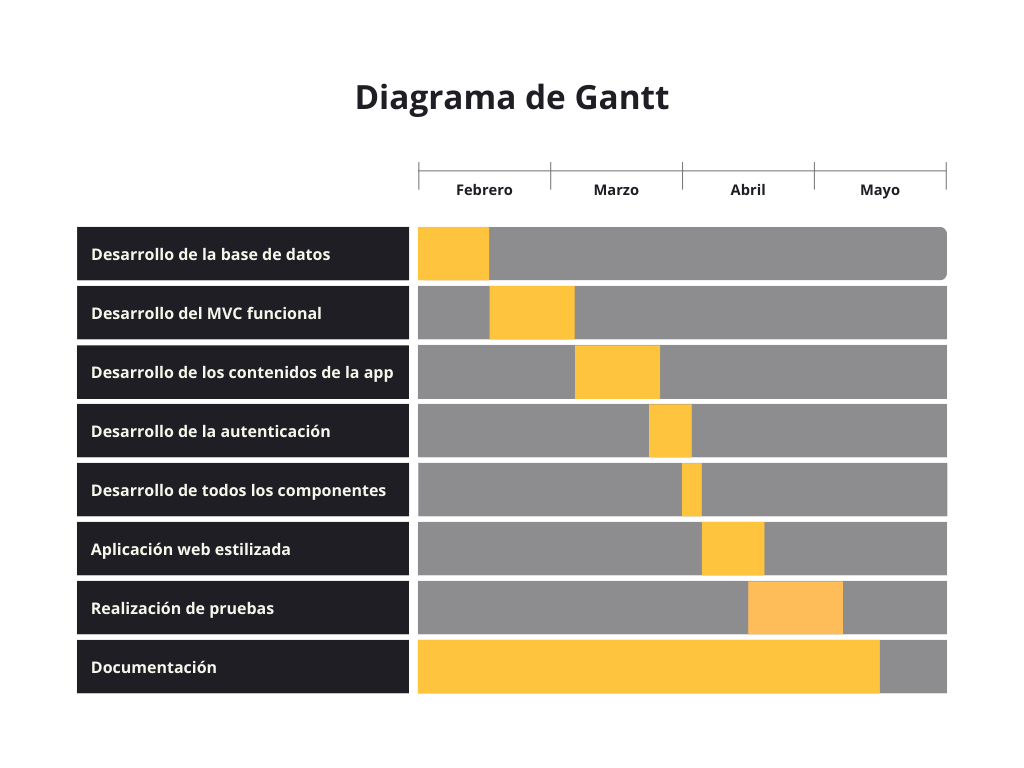


Ilustración 1: Diagrama de Gantt.

## Planificación económica

|  |  |  |
| --- | --- | --- |
| Concepto | Detalle / Supuesto | Coste estimado |
| **Ordenador personal** | PC de gama media | 1000,00 € |
| **Programador 30h** | Freelance a 20 €/h | 600,00 € |
| **Gasto eléctrico** 30h | Consumo 100W a 0,25 €/kWh | 0,75 € |
| **Programas** | Licencias básicas | 30,00 € |
| **Dominio** | Dominio común anual | 15,00 € |
| AWS EC2 + VPC | EC2 (tráfico básico 720h/mes) | 35,00 € |
| Hosting DNS | 0,50 €/mes | 0,50 € |
| ****Total**** | | **1.681,25 €** |

Tabla 1: Planificación económica.

# Análisis

## Público objetivo

Esta aplicación web está dirigida principalmente a empresas que organizan y gestionan eventos de motor, ya que la plataforma está diseñada para ser implementada en sus sitios web.

En segundo lugar, está orientada a aficionados que desean seguir el desempeño de sus pilotos o equipos favoritos, así como a pilotos y equipos que usan la plataforma para ver su puntuación, analizar su rendimiento y compararse con otros competidores.

## Nombre de la aplicación

El nombre de la aplicación, RacingTracker, es simple y clara, reflejando las principales funcionalidades de la aplicación, la gestión y consulta de información relacionada con carreras.

## Diagrama Entidad-Relación

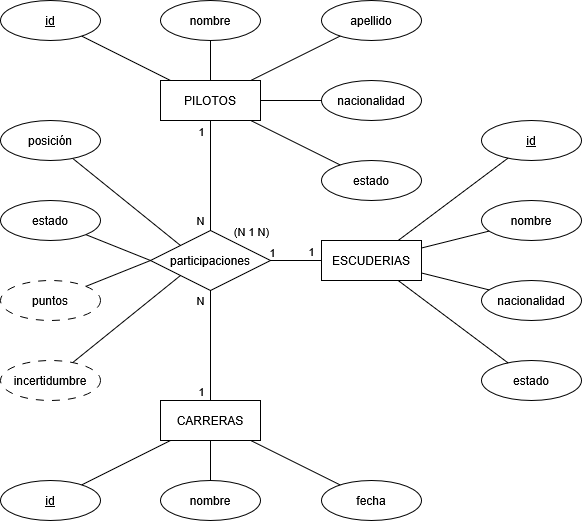


Ilustración 2: Diagrama Entidad-Relación de la base de datos.

## Diagrama de la aplicación

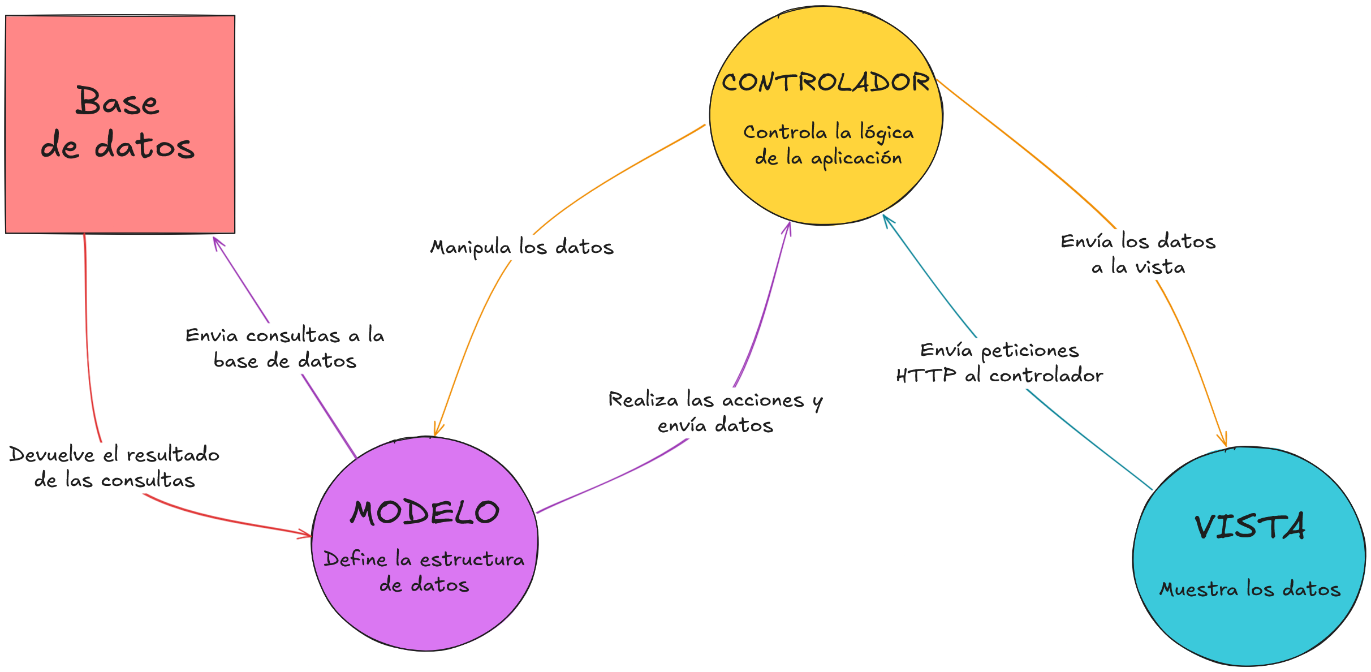


Ilustración 3: Diagrama de aplicación

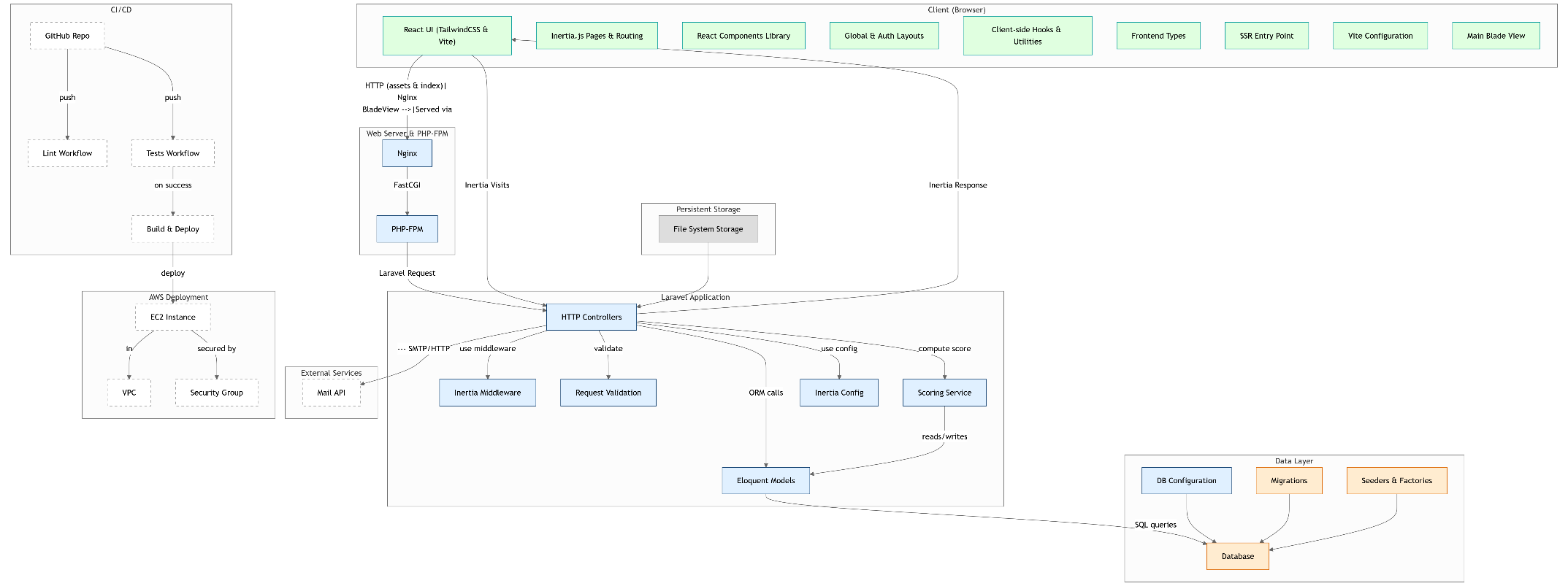


Ilustración 4: Diagrama del repositorio.

## Diagrama de casos de uso

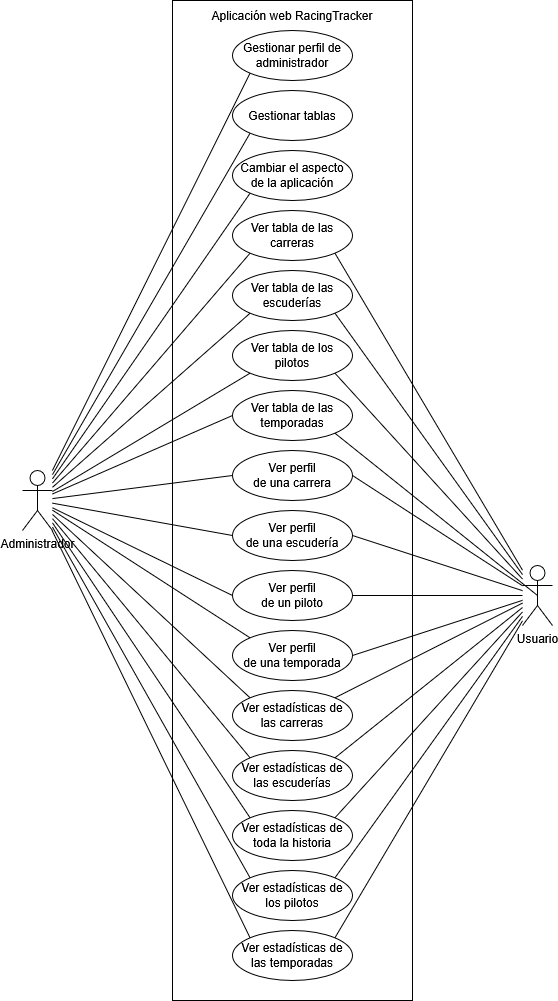


Ilustración 5: Diagrama de casos de uso.

# Diseño

## Prototipo de la interfaz

### Primeros prototipos

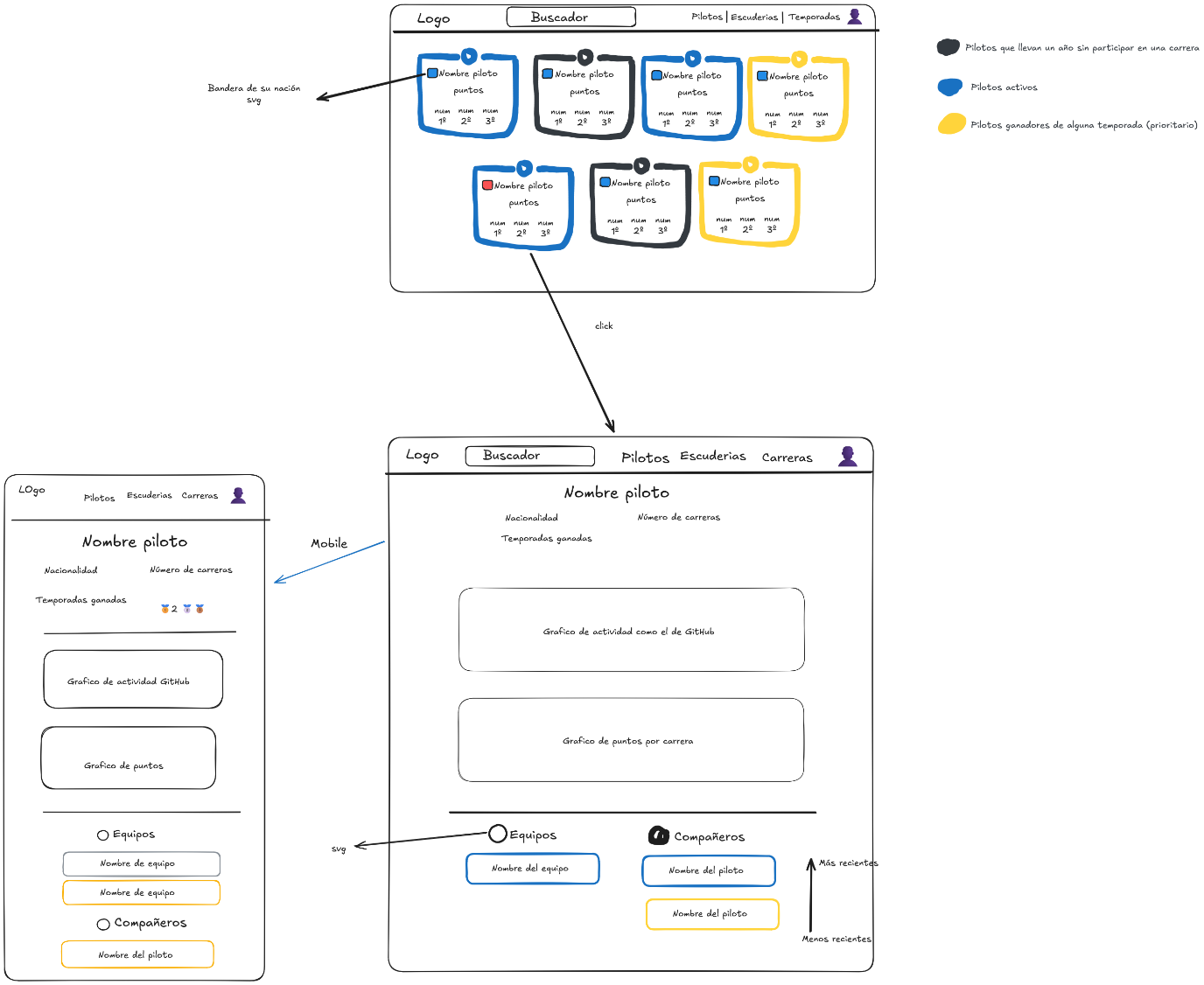


Ilustración 6: Boceto de la página de pilotos y su perfil.

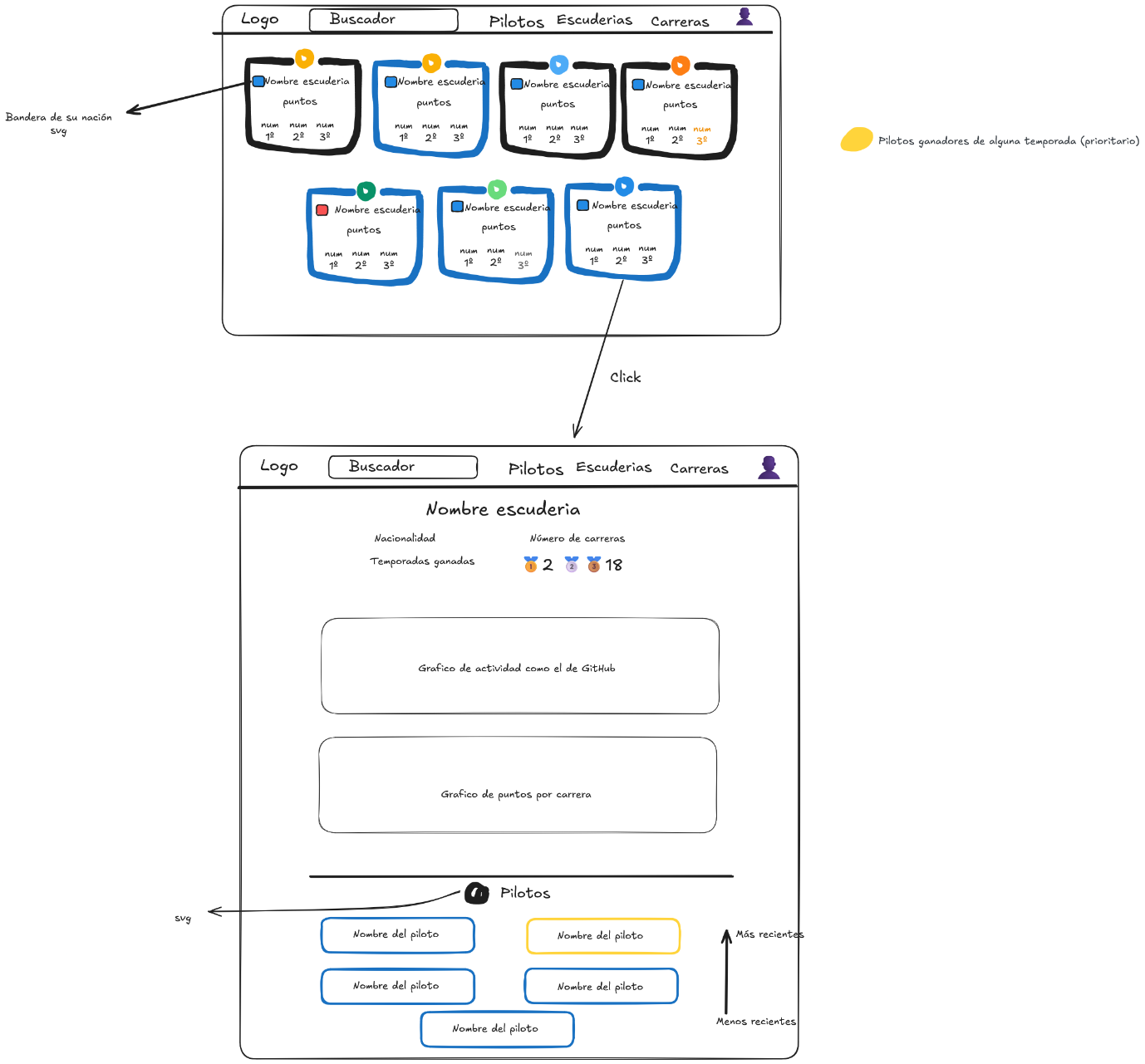


Ilustración 7: Boceto de la página de escuderías y su perfil.

### Prototipos finales



Ilustración 8: Boceto del funcionamiento del menú de navegación.

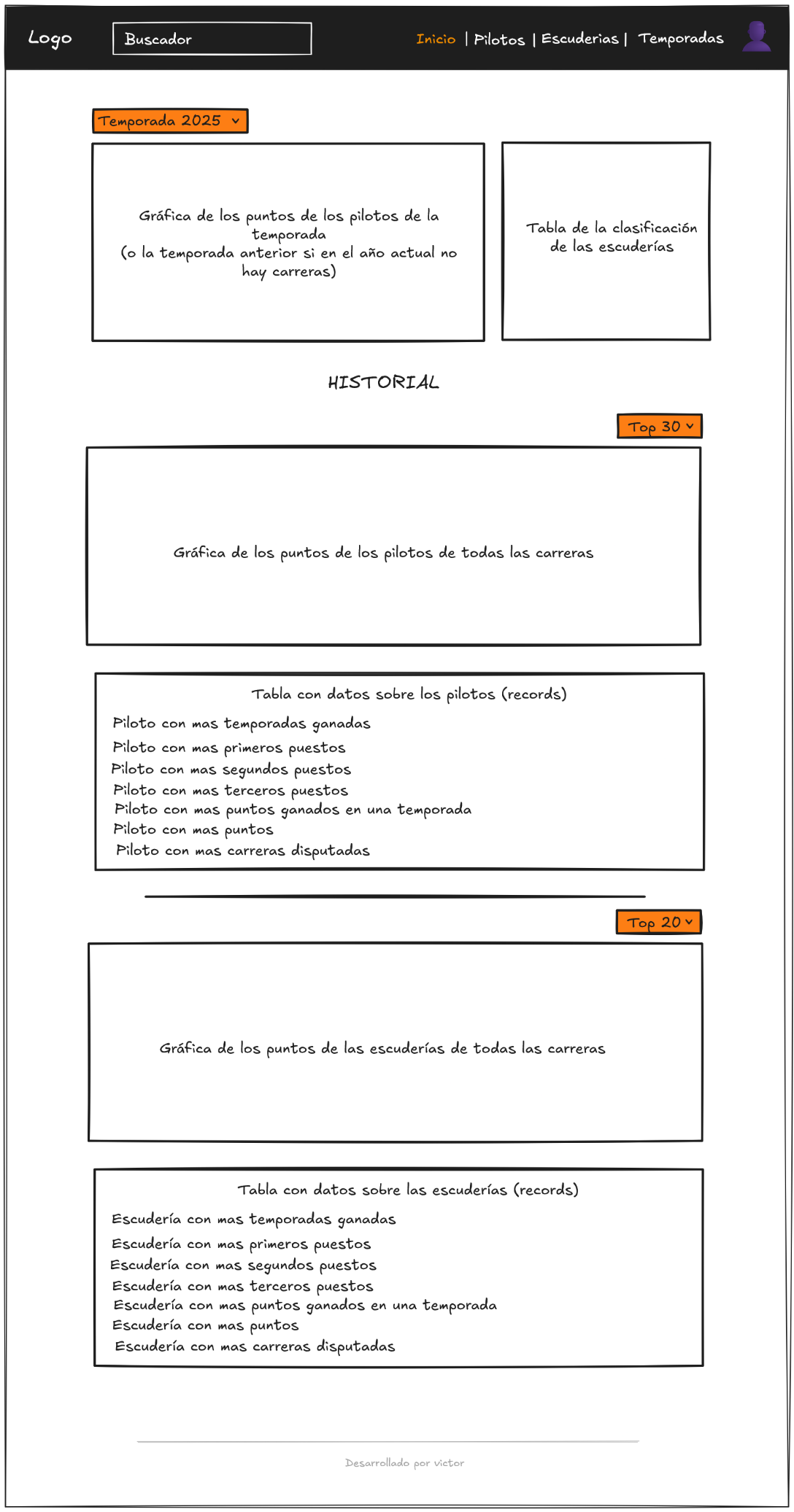


Ilustración 9: Boceto de la página de inicio.

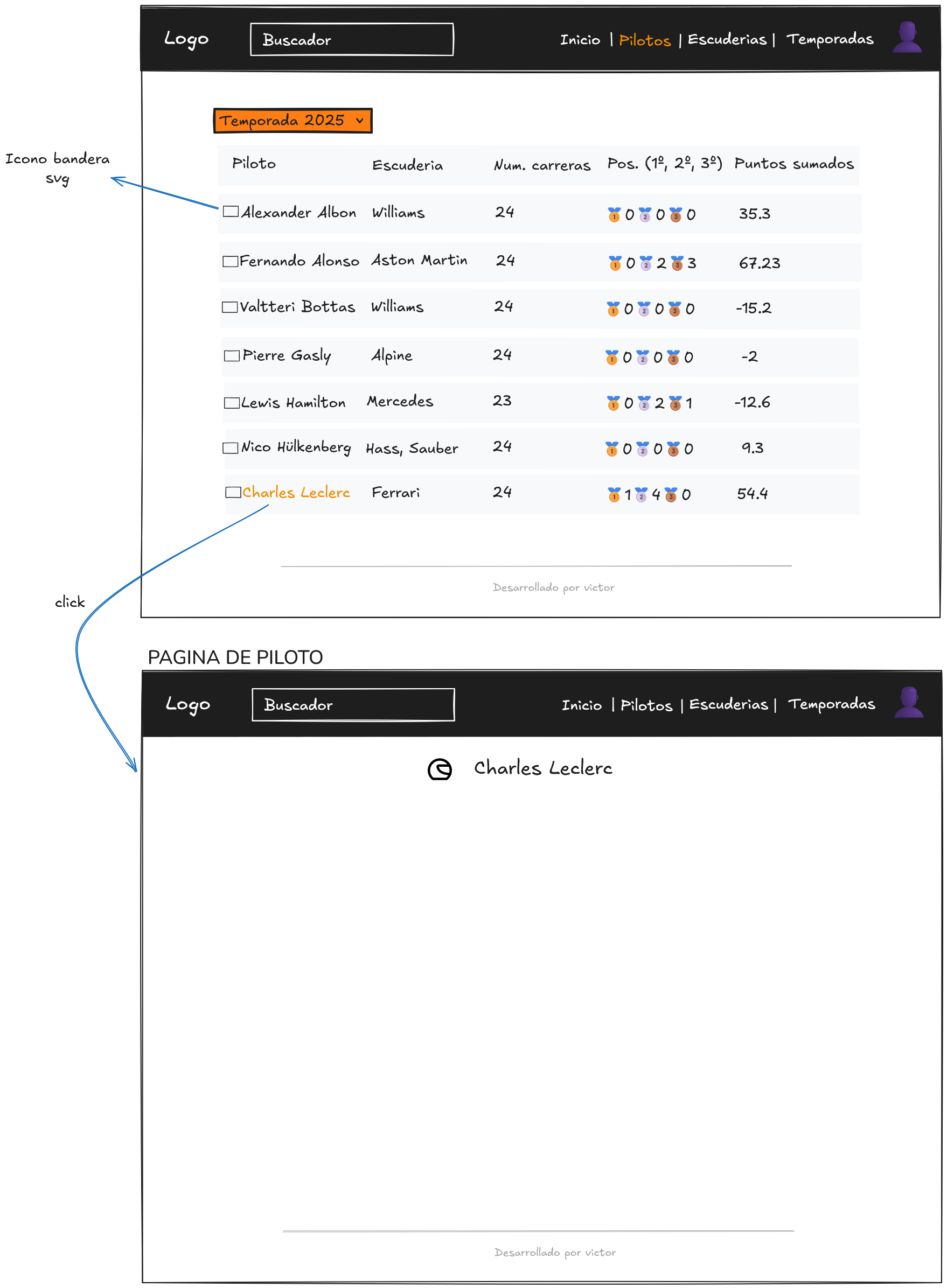


Ilustración 10: Boceto de la página de pilotos.

## Diagrama relacional

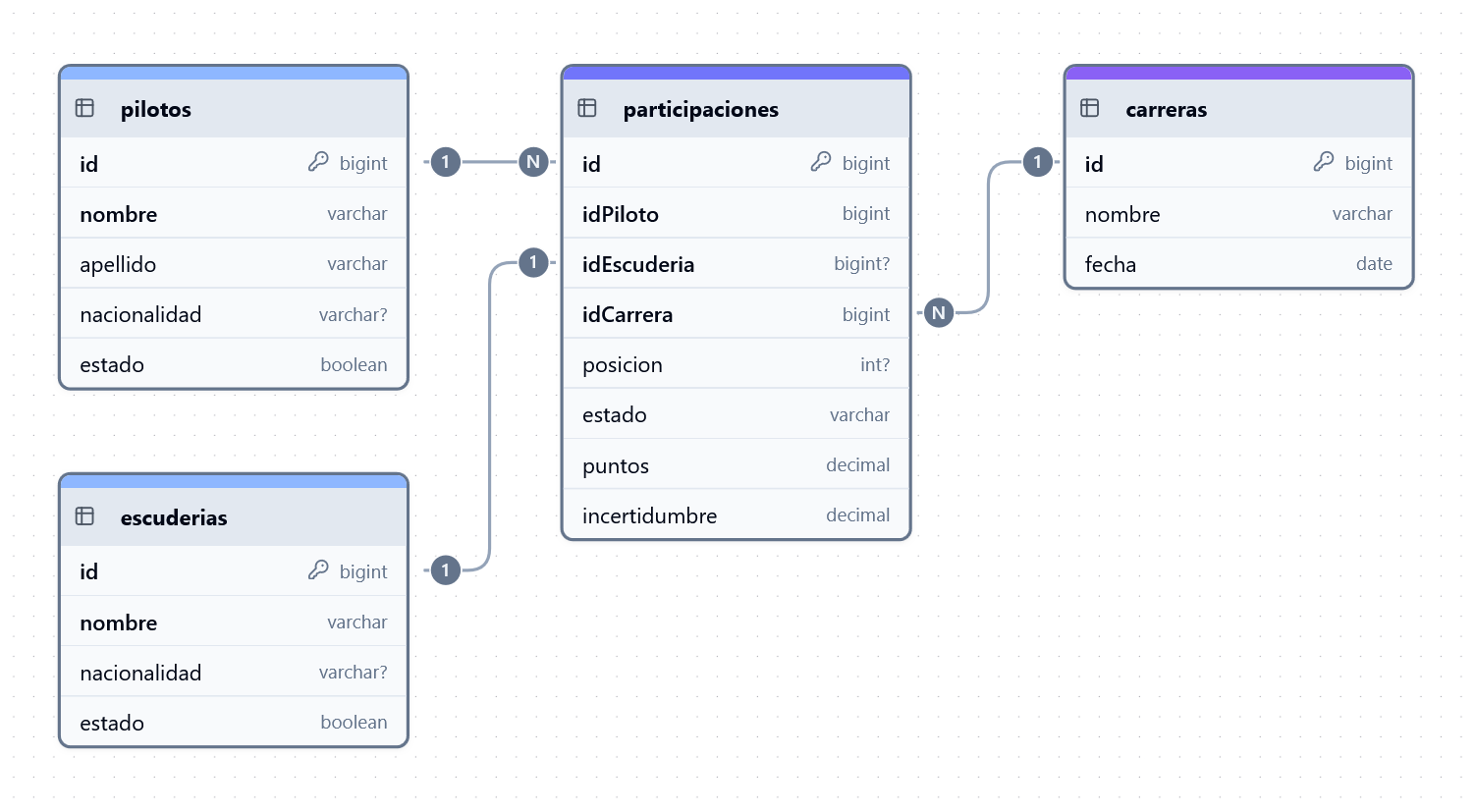


Ilustración 11: Diagrama relacional de la base de datos.

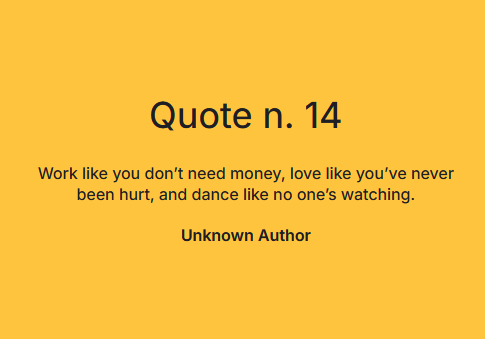
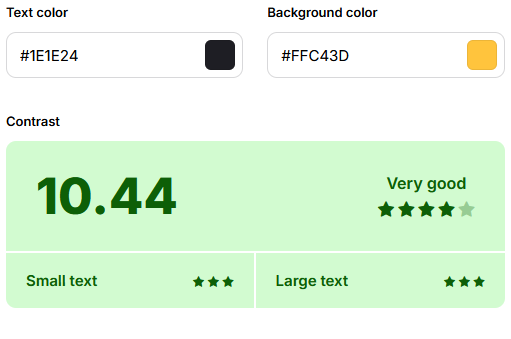
## Paleta de colores

He elegido esta paleta de colores porque transmite elegancia y simplicidad, y se adapta bien tanto a navegadores con temas claros como oscuros.



Ilustración 12: Paleta de colores.

Ilustración 13: Contraste de la paleta de colores.



## Tipografía

He elegido las tipografías Roboto y Open Sans porque son simples, fáciles de leer y visualmente similares entre sí. Además, opté por incluir ambas para asegurar compatibilidad en caso de que algún navegador no soporte una de ellas.

Ilustración 14: Ejemplo gráfico de las tipografías.



# Pruebas

|  |  |  |
| --- | --- | --- |
| ****Nombre de la Prueba**** | ****Descripción**** | ****Fecha de Realización**** |
| Pruebas unitarias (Backend) | Verificar funciones individuales o métodos de Laravel (controladores y modelos). | Desde el inicio del desarrollo (continuamente) |
| Pruebas unitarias (Frontend) | Validar componentes de React. | Al desarrollar un nuevo componente |
| Pruebas de integración | Evalúar cómo interactúan distintos módulos entre sí (controladores y modelos). | 19/04/2025 |
| Pruebas E2E (End-to-End) | Simular el comportamiento real del usuario desde el frontend hasta la base de datos. | Cuando hay flujo funcional completo |
| Pruebas de validación | Asegurar que los formularios tengan validaciones adecuadas tanto en React como en Laravel. | 18/04/2025 |
| Pruebas de seguridad | Verificar vulnerabilidades comunes como CSRF, XSS, autenticación y autorización. |  |
| Pruebas de regresión | Asegurar que nuevas funcionalidades no rompen las existentes. | Cada vez que se agrega una nueva funcionalidad (continuamente) |
| Pruebas de usabilidad | Evalúar la experiencia del usuario con la interfaz de React. | 30/04/2025 |
| Pruebas de rendimiento | Evalúar el tiempo de respuesta del backend Laravel y la carga del frontend. |  |
| Pruebas de accesibilidad | Garantizar que la app pueda ser usada por personas con discapacidades. |  |
| Pruebas de aceptación (UAT) | Probar que la app cumple con los requisitos. |  |

Tabla 2: Pruebas realizadas.

# Diario de trabajo

09/10/2024 - 12/01/2025

Desarrollo de varias propuestas para el proyecto, incluyendo una breve descripción y un resumen de las tablas, atributos y relaciones de la base de datos.

13/01/2025

Selección definitiva del tema del proyecto.

14/01/2025

Investigación sobre las tecnologías que se utilizarán en el proyecto.

18/01/2025

Investigación sobre las tecnologías que se utilizarán en el proyecto.

Diseño de la portada y definición de los puntos principales de la memoria del proyecto.

19/01/2025 al 22/01/2025

Desarrollo del diseño de la base de datos.

23/01/2025

Corrección del diseño de la base de datos con la profesora del módulo de Desarrollo Web en Entorno Servidor.

24/01/2025

Desarrollo del diagrama entidad relación y relacional.

26/01/2025

Investigación de la fórmula para calcular la puntuación.

Diseño del logotipo de la aplicación.

27/01/2025

Desarrollo del problema planteado y los objetivos.

Desarrollo del boceto de la página de pilotos.

28/01/2025

Documentado el público objetivo del proyecto.

05/02/2025

Investigación sobre servidores de hosting gratuitos.

20/02/2025

Rediseño de la base de datos.

21/02/2025

Corrección del diseño de la base de datos con la profesora del módulo de Desarrollo Web en Entorno Servidor.

Investigación de las tecnologías a utilizar en el proyecto.

22/02/2025

Desarrollo del diagrama entidad-relación y el diagrama relacional.

Creación de la base de datos.

Desarrollo del diseño del logo de la aplicación.

23/02/2025

Desarrollo del backend de la aplicación; configuración de Next.js, el ORM de Prisma y un modelo y controlador de prueba.

24/02/2025

Anteproyecto desarrollado.

Corrección de la implementación de la base de datos en el backend.

Probado que el MVC funciona correctamente.

26/02/2025

Corrección del tipo de dato de la posición en la base de datos.

Investigación de la paleta de colores de la aplicación.

Documentación de las tecnologías a implementar.

27/02/2025

Desarrollando funciones sobre las nacionalidades de equipos y pilotos.

28/02/2025

Inserción de datos de prueba en la base de datos.

Corrección de errores en las vistas para poder visualizar el contenido de la base de datos.

Investigación sobre la codificación de la base de datos y prisma para que los caracteres extraños se mostrasen correctamente.

01/03/2025

Desarrollado un componente para mostrar el icono de la bandera dependiendo de la nacionalidad e implementado.

Modificación de la base de datos eliminando el campo del color de la tabla equipo y la modificación de los diagramas, entidad relación y relacional.

02/03/2025

Investigación sobre el uso de APIs en la aplicación.

03/03/2025

Desarrollo del diagrama de Gantt.

04/03/2025

Desarrollo de la página de gestión de los pilotos, tabla, botón de borrar, botón de actualizar.

05/03/2025

Implementada la funcionalidad de borrar un piloto en su página de gestión.

Desarrollado un buscador para filtrar por el nombre o apellido de los pilotos.

06/03/2025

Desarrollado la funcionalidad para crear un piloto en la página de gestión.

Desarrollada la interfaz de modificación de pilotos y de la funcionalidad para editar los campos.

07/03/2025

Optimizando el CRUD de los pilotos.

Editando el logo de la aplicación

08/03/2025

Abstracción del código del CRUD de los pilotos.

09/03/2025

Documentada la planificación en la memoria del proyecto.

Abstracción del código del CRUD de los pilotos.

10/03/2025

Desarrollado la página de gestión de escuderías.

11/03/2025

Desarrollo de la página de gestión de carreras.

12/03/2025

Desarrollo de algunas funciones de los controladores.

13/03/2025

Optimización de las funciones de los controladores

Redesarrollo de la página de gestión de carreras debido a una mala estructuración del código.

Desarrollo de las propuestas de mejora en la memoria.

14/03/2025

Abstracción del código de la página de gestión de pilotos.

Optimización del componente de la bandera.

15/03/2025

Abstracción del código de gestión de pilotos, creando componentes para los modales, el formulario y la tabla de gestión.

16/03/2025

Desarrollo del hook para la página de gestión de pilotos

Desarrollo de un middleware para la validación de los campos introducidos en el formulario de pilotos.

Ampliación de la API de los pilotos, incluyendo el método GET.

Fusión de los componentes abstraídos en la vista de gestión de pilotos

Pruebas con la codificación de la base de datos.

17/03/2025

Corrección de la respuesta de la API y optimización de su código.

Investigación y pruebas sobre la codificación de la base de datos sin resultados.

Intento de desarrollo de mensajes de alerta después de la respuesta de la API.

Finalizada la abstracción de la página de gestión de pilotos.

18/03/2025

Desarrollo del logotipo de la aplicación

19/03/2025

Desarrollo del logotipo de la aplicación

20/03/2025

Investigación de Supabase y creación de la base de datos.

21/03/2025

Importación de la base de datos de Supabase en el archivo de Prisma.

22/03/2025

Mejora del menú de navegación

Implementación de metadatos y un componente head personalizable para cada página.

Implementado el favicon.

Desarrollada la página para error 404.

23/03/2025

Implementación de la API de Supabase junto a las políticas de select.

Probando la API enviando peticiones HTTP con RapidAPI.

24/03/2025

Inserción de datos usando RapidAPI con la validación de un usuario.

25/03/2025

Actualización de datos usando RapidAPI con la validación de un usuario.

26/03/2025

Reinicio del proyecto, instalación de Laravel con React

Pruebas de instalación de la conexión con la base de datos con Turso.

27/03/2025

Instalación de Laravel + React

Pruebas con componentes en la página de dashboard.

28/03/2025

Editado el diseño del header

Modificados los colores de los temas de la aplicación

29/03/2025

Desarrollado el header

Definida la página de inicio

30/03/2025

Desarrolladas las migraciones de la base de datos

Implementación de las rutas restantes

Empezado el desarrollo de la página de index del modelo de Pilotos.

31/03/2025

Implementación de los fillables de los modelos.

Desarrollo de atributos del modelo de Pilotos.

01/04/2025

Implementación de columnas y celdas en la tabla de pilotos.

02/04/2025

Desarrollo de la tabla de pilotos y sus columnas, separándolas por las que pueden ver todos los usuarios y las que solo los usuarios autenticados.

Desarrollo del desplegable para seleccionar los pilotos de una determinada temporada.

Desarrollo de la responsividad de la tabla de pilotos.

03/04/2025

Cambio de la redirección al cerrar sesión para que vaya a la página anterior en vez de redirigir a la página de inicio.

Abstracción del código de la tabla de Pilotos.

05/04/2025

Configuradas las columnas de la tabla para que se muestren o no dependiendo de la temporada seleccionada.

Desarrollado el botón de crear y finalizada la estructura de la página de pilotos.

06/04/2025

Optimizado el código que redirige a los metodos del controlador.

Investigación sobre la implementación del formulario de actualización.

07/04/2025

Desarrollo del formulario de creación y edición de pilotos.

Corrección de los estilos del radio del borde de los elementos de la aplicación.

10/04/2025

Desarrollado el CRUD de los pilotos.

11/04/2025

Desarrollado el CRUD de las escuderias.

12/04/2025

Desarrollada la vista de las carreras y el perfil de las carreras.

13/04/2025

Implementado en formulario de la clasificacion la opcion de moverlos con dnd-kit en la vista de crear carreras.

14/04/2025

Documentado el apartado de los colores de la aplicación.

15/04/2025

Configuración de la tipografia.

Corrección del logo del header.

16/04/2025

Añadidos atributos de estado para las tablas de pilotos y escuderías.

Configurados los CRUD de pilotos y escuderías para configurar el nuevo atributo.

17/04/2025

Desarrollo del formulario para introducir los datos de la clasificación de una carrera y validación de los datos introducidos.

18/04/2025

Desarrollado el sistema de puntuación de los pilotos e implementado al crear una carrera.

19/04/2025

Implementada la función de actualizar y borrar carreras.

Añadidos los datos a la página de index de las carreras.

20/04/2025

Desarrollado un nuevo icono con fondo para la documentación.

Añadidos datos a la página de index de los pilotos.

21/04/2025

Corregidos los datos de la página de index de los pilotos.

22/04/2025

Añadidos los datos de la página de index de las escuderías.

Corregidos los estilos de las tablas de las páginas de index.

Corregido el enlace de la tipografía para añadir todos los tamaños de las fuentes.

23/04/2025

Añadia la columna de mejor piloto en la página de index de las carreras.

Optimizadas algunas consultas.

24/04/2025

Implementado el perfil de los pilotos.

Añadidos gráficos al perfil de los pilotos.

25/04/2025

Probando a desplegar la aplicación en diferentes plataformas como Laravel Cloud, Railway, Render, sin éxito.

Optimizado el código del modelo de pilotos.

26/04/2025

Desarrollado el readme.

Desarrollada la documentación para el despliegue en local.

Desarrollada la documentación para el despliegue en AWS.

27/04/2025

Corrección de la fórmula para calcular los puntos de las participaciones.

28/04/2025

Diseñado el diagrama de aplicación.

30/04/2025

Desarrollado el diagrama de casos de uso.

01/05/2025

Corregida la redirección a otras rutas desde el controlador.

Eliminada la opción de registrarse.

Corregidos los datos que se envían a la vista de perfil de pilotos.

Implementado el perfil de las escuderías.

Refactorizado el código.

Corregidos algunos estilos de los perfiles.

Corregidos los diagramas de la base de datos.

Desarrolladas las propuestas de la documentación.

# Conclusiones

## Objetivos cumplidos

## Propuestas de mejora y ampliaciones futuras

* Incorporar nuevos idiomas además del inglés, incluyendo español, portugués, francés e italiano. Esto permitirá mejorar la experiencia de los usuarios al ofrecerles una interfaz en su idioma. La implementación incluirá la traducción de la interfaz y un sistema de selección de idioma en la configuración, asegurando una navegación fluida y adaptada a cada usuario.
* Automatizar completamente el proceso de creación y configuración de la aplicación para facilitar su despliegue por terceros, mediante scripts, permitiendo que cualquier usuario externo pueda clonar el proyecto, configurar su entorno y desplegar la aplicación de forma rápida, estandarizada y sin necesidad de conocimientos técnicos avanzados.
* Incorporar en la página de inicio una funcionalidad que permita comparar el rendimiento de hasta cuatro pilotos o escuderías a lo largo del tiempo, mediante gráficos. Esta funcionalidad facilitará a los usuarios el análisis de estadísticas y la comparación de desempeños en distintas temporadas, mejorando la experiencia de navegación.

## Reflexiones sobre el proyecto

# Guía

## Despliegue

## Guía de administración

## Guía de usuario

# Bibliografía

* Editor de diagramas de base de datos → app.chartdb.io
* Editor de diagramas → drawio.com
* Diagrama de repositorio → gitdiagram.com
* Inteligencia artificial → chatgpt.com y perplexity.ai
* Editor de prototipos y diagrama de aplicación → excalidraw.com
* Iconos de banderas → svgrepo.com
* Iconos SVG → lucide.dev
* Logotipo y diagrama de Gantt → canva.com
* Componentes UI → ui.shadcn.com
* Colores y paleta → coolors.co
* Tipografía → fonts.google.com
* Proyecto que utiliza el sistema de puntuación TrueSkill → trueskill.org
* Documentación del sistema de puntuación → en.wikipedia.org/wiki/TrueSkill
* Ensayo del sistema bayesiano de calificación de habilidades TrueSkill → proceedings.neurips.cc/paper\_files/paper/2006/file/f44ee263952e65b3610b8ba51229d1f9-Paper.pdf
* Tutorial de despliegue de Laravel en AWS → www.youtube.com/watch?v=IgJWK4aQ2jQ&list=PL1tt6av2E5dZyI15I658u\_lVhKg3jbtjS