



UNIVERSIDAD DE MÁLAGA



Graduado en Ingeniería del Software

Aplicación web para registros de partidas de jugadores de
golf

Web application for records of golf players' games

Realizado por
Alejandro Ramet Espinosa

Tutorizado por
Eduardo Guzmán de los Riscos

Departamento
LENGUAJES Y CIENCIAS DE LA COMPUTACIÓN

MÁLAGA, Septiembre 2024

Abstract

TechGreen is a web application designed to optimize golf activity management for both players and golf courses through modern technology. Its main goal is to digitize the golfing experience.

For players, TechGreen serves as a social network, allowing them to connect with friends or meet new players for matches across various courses. The platform tracks match results, helping players monitor their progress and improve their performance at a professional level.

Golf courses can also join the platform and utilize it as a CRM ¹, providing features that boost player engagement, such as posting news, organizing and managing tournaments, and tracking all matches played throughout the day.

The application has been structured into two distinct components: the Frontend and the Backend. For the Backend, ASP.NET Core [1] was utilized, with MySQL [2] chosen to handle database management. On the other hand, the frontend was developed with Angular [3].

Keywords: Golf courses, .NET, Angular, MySQL

¹Customer Relationship Management

Resumen

TechGreen es una aplicación web que surge de la necesidad de poder administrar de una forma más eficiente diversos aspectos relacionados con el golf, tanto a nivel de jugador, como de gestionar un campo de golf, haciendo uso de las nuevas tecnologías de las que se dispone en la actualidad. En resumen, su finalidad consiste en intentar digitalizar el mundo del golf.

Los jugadores tendrán la opción de conectarse como en una red social, donde podrán encontrar a sus amigos o conocer a nuevas personas, con las que disputarán partidos entre ellos en los distintos campos disponibles, quedando registrados los resultados y permitiendo así, ver la evolución a nivel profesional como golfista.

Además, los campos de golf también podrán registrarse en la plataforma y darle el uso de un CRM ², con distintas funcionalidades con las que fomentar las interacciones de los jugadores en la aplicación: desde escribir noticias de interés, crear y dirigir torneos y gestionar todas las partidas que se disputan a lo largo del día en sus propios clubs.

La aplicación se ha estructurado en dos componentes distintos: el Frontend y el Backend. Para el Backend, se ha utilizado ASP.NET Core [1], y se eligió MySQL [2] para gestionar la base de datos. Por otro lado, el Frontend se ha desarrollado con Angular [3].

Palabras clave: Campos de golf, .Net, Angular, MySQL

²Customer Relationship Management

Índice

1. Introducción	5
1.1. Motivación	5
1.2. Objetivos	5
1.3. Estructura del Documento	6
2. Tecnologías y herramientas utilizadas	9
2.1. Tecnologías Backend	9
2.2. Tecnologías Frontend	10
2.3. Otros	11
3. Especificación y Análisis	13
3.1. Roles de usuario	13
3.2. Requisitos Funcionales	15
3.3. Requisitos No Funcionales	27
4. Diseño del Sistema	29
4.1. Base de Datos	31
4.2. Backend	32
4.3. Frontend	33
4.4. Diagrama de Secuencia	36
5. Implementación y Pruebas	39
5.1. Primera Iteración	40
5.2. Segunda Iteración	42
5.3. Tercera Iteración	42
5.4. Cuarta Iteración	42
5.5. Pruebas	43
6. Conclusiones y Líneas Futuras	45
6.1. Conclusiones	45

6.1.1.	Conclusiones sobre el Proceso de Desarrollo	45
6.1.2.	Conclusiones sobre la Aplicación	46
6.2.	Líneas Futuras	46
Apéndice A. Manual de Instalación		51
A.I.	Prerrequisitos	51
A.II.	Clonación del Repositorio	52
A.III.	Configuración de la Base de Datos con MySQL	52
A.IV.	Ejecución de la Base de Datos con Docker Compose	53
A.V.	Configuración del Frontend	54
A.VI.	Configuración del Backend	54
A.VII.	Ejecución Completa del Proyecto	54
Apéndice B. Manual de Usuario		57

1

Introducción

1.1. Motivación

A todos los golfistas les agrada ver su progreso a medida que entrenan y juegan partidas. Sin embargo, para llevar un registro detallado de todos los golpes y puntos realizados, actualmente se requieren métodos que resultan complejos. Esto se debe a que el golf, como deporte, ha estado históricamente desvinculado de las tecnologías modernas. Si se lograra una integración efectiva entre ambos, se podría alcanzar un crecimiento potencial significativo. Tal avance no solo incentivaría a nuevos jugadores a iniciarse en el golf, gracias a las facilidades ofrecidas, sino que también fomentaría una participación más activa entre los golfistas, quienes podrían monitorear continuamente su progreso, tener un mayor control sobre su hándicap y poder conocer a otras personas de su mismo nivel.

1.2. Objetivos

Hay varios objetivos que se intentan abarcar con el desarrollo de esta aplicación y dependen de los roles que se han definido en ella:

▪ ADMINISTRADOR

- **Control de cuentas de usuario:** Los administradores podrán visualizar los perfiles de todos los usuarios que hay en la plataforma y tendrán a su disposición la capacidad de desactivar o activar cuentas según vean conveniente para el bien de la aplicación.

▪ JUGADOR

- **Aumentar la interacción de los jugadores:** Los usuarios podrán organizar partidas entre ellos desde la aplicación en un campo específico. También se podrá

invitar a otros jugadores mientras exista una relación de amistad entre ellos. Con esto se espera además la integración de más personas en el mundo del golf.

- **Registrar partidas:** Las partidas se quedarán almacenadas para poder ser visualizadas en cualquier momento. Cuando se finalicen, los resultados se registrarán para todos los usuarios por igual, añadiendo estos datos en la base de datos del jugador. Además, podrás analizar los resultados de otros usuarios.
- **Mostrar evolución del jugador:** Habrá un apartado estadísticos donde se podrá monitorear el progreso de un jugador a través de distintos KPIs y gráficas que permiten el análisis de las partidas.

■ CAMPO

- **Informar:** Los moderadores de los campos podrán publicar noticias de interés que serán visibles para los usuarios, con tal de darse a conocer y captar la atención de más personas.
- **Organizar torneos:** Los jugadores podrán unirse a los torneos que organicen los campos permitiendo ver los resultados en directo y ver en que posición van en todo momento. Al finalizarlos se mostrará el ranking con los puntos obtenidos
- **Gestionar calendario:** Se facilitará un calendario donde se podrá ver el número de partidas que hay al día, en que intervalo se encuentran y los participantes de ellas, permitiendo agregar o eliminar usuarios según sea conveniente.

1.3. Estructura del Documento

Esta memoria está estructurada en diferentes capítulos cuyo contenido se explica a continuación:

- **Tecnologías y Herramientas Utilizadas:** El primer punto a tratar se dedicará en introducir las diferentes tecnologías y herramientas que se han usado a lo largo del desarrollo de la aplicación con una breve descripción.
- **Especificación y análisis:** Posteriormente se detallarán los requisitos tanto funcionales como no funcionales de la aplicación dependiendo de cada rol.

- **Diseño del Sistema:** En esta sección se explicará detalladamente la arquitectura que se ha seguido en el desarrollo del proyecto tanto de la Base de Datos, como del Backend y el Frontend, explicando a su vez las distintas interacciones que se encuentran entre ellas.
- **Implementación y Pruebas:** En este punto se encontrará la documentación relacionada a la implementación de la aplicación, abordando las decisiones tomadas, la metodología aplicada junto al flujo de trabajo y se contemplarán algunos algoritmos que son de mayor complejidad que el resto. Por otro lado, se podrá contemplar las pruebas realizadas en el proyecto.
- **Conclusiones y Líneas Futuras:** Esta sección recoge las conclusiones finales del proyecto una vez implementado. Se evaluará si se han cumplido los objetivos satisfactoriamente y a su vez, se hablará de posibles mejoras y próximos desarrollos con la finalidad de potenciar la aplicación para versiones futuras.
- **Apéndices:** Finalmente se dispondrá de dos manuales de interés para el usuario, uno de cómo usar la aplicación y otro de instalación.

2

Tecnologías y herramientas utilizadas

En este apartado se van a dividir las tecnologías, herramientas y librerías utilizadas en el proyecto en Backend y Frontend.

2.1. Tecnologías Backend

- **MySQL:** [2] Sistema de gestión de bases de datos relacional creada por Oracle que se basa en un modelo cliente y servidor. (Versión 8.0.33)
- **Docker:** [4] Software utilizado para desplegar aplicaciones virtuales en contenedores. Para este proyecto su finalidad ha sido desplegar la base de datos en un contenedor como si se tratara de un servidor. (Versión 24.0.6)
- **MySQL Workbench:** [5] Herramienta que permite la visualización de bases de datos MySQL. Destaca por las características de permitir diseñar y modelar bases de datos relacionales directamente a través de una interfaz gráfica. (Versión 8.0)
- **C#:** [6] Lenguaje de programación de código abierto orientado a objetos fuertemente tipado, utilizado en este caso para el desarrollo Backend de la aplicación.
- **.NET:** [7] Framework multiplataforma desarrollado por Microsoft que usa C# como lenguaje. Permite la programación asíncrona y dispone de una gran cantidad de librerías que son de gran utilidad a la hora del desarrollo. Se caracteriza por su confiabilidad, rendimiento y seguridad. Para este proyecto se ha utilizado ASP NET Core [1] para facilitar la creación de una API Rest [8]. (Versión 6.0)

- **Visual Studio 2022:** [9] Editor de código compatible con el desarrollo de aplicaciones .NET, y permite la edición, depuración, compilación y despliegue de código desde la propia interfaz sin necesidad de terceros.
- **Cloudinary:** [10] API que permite el almacenamiento y gestión en la nube de distintos archivos media.
- **Postman:** [11] Herramienta que permite enviar solicitudes HTTP de manera sencilla. Se ha empleado en este caso para realizar pruebas de integración de la API con la base de datos, verificando que los diferentes endpoints funcionen correctamente y las respuestas sean las deseadas.
- **Bibliotecas:** .NET permite gestionar librerías externas a través de los paquetes NuGet. Para el desarrollo back de tTechGreen se han utilizado las siguientes bibliotecas:
 - **CloudinaryDotNet:** Permite la integración de Cloudinary en .NET.
 - **MailKit:** Permite el envío de correos electrónicos a través de unos credenciales preestablecidos.
 - **MySQLData:** Permite la conexión entre .NET y una base de datos MySQL.
 - **JwtBearer:** [12] Permite gestionar la autorización de los endpoints a través de tokens.
 - **Newtonsoft.Json:** Permite trabajar con JSON³ para enviar objetos entre Backend y Frontend con este formato.

2.2. Tecnologías Frontend

- **Node.js:** [13] Entorno de tiempo de ejecución de JavaScript controlado por eventos. Permite la conexión HTTP con el back. (Versión 18.18.2)
- **TypeScript:** [14] Lenguaje de programación fuertemente tipado basado en JavaScript, agregando sintaxis adicional.

³JavaScript Object Notation

- **HTML:** ⁴ Lenguaje de etiquetas de hipertexto cuya finalidad es dar la estructura a una página web.
- **SASS:** [15] Lenguaje de hojas de estilos compilada en CSS ⁵ que aporta una sintaxis más extensa. Su funcionalidad es dar estilos a una página web.
- **Angular:** [3] Framework web gestionado por Google que dispone de un gran repertorio de bibliotecas y APIs, cuya finalidad consiste en el desarrollo de aplicaciones sólidas y escalables. Se basa en la arquitectura Modelo-Vista-Controlador y su trabajo se basa en el uso de módulos y componentes. (Versión 16.2.7)
- **PrimeNG:** [16] Biblioteca de componentes de interfaz de usuario de Angular. Dispone de componentes propios con estilos ya predefinidos pero que pueden ser alterados a gusto del desarrollador. Es de gran ayuda para agilizar el tramo de diseño web. (Versión 14.2.3)
- **Visual Studio Code:** [17] Editor de código multiplataforma que cuenta con una gran variedad de extensiones para los lenguajes de programación y entornos de ejecución.
- **Bibliotecas:** Node.js permite agregar bibliotecas externas de manera sencilla a nuestro proyecto. En ese caso se han utilizado las siguientes:
 - **Leaflet:** Permite la integración de mapas en el proyecto.
 - **Chart.js:** Permite la visualización de datos en gráficas.
 - **Ckeditor5:** Permite trabajar con un editor de texto avanzado.
 - **Xlsx:** Permite interactuar con hojas Excels.
 - **Fullcalendar:** Permite la gestión de eventos en calendarios.
 - **ngx-cookie-service:** Permite controlar las cookies de un navegador

2.3. Otros

- **Latex:** Sistema de composición de textos orientado a la creación de documentos.

⁴HyperText Markup Language

⁵Cascading Style Sheets

- **Overleaf:** [18] Plataforma web que permite la visualización directa de documentos escritos en Latex.
- **Git:** [19] Software que sirve como sistema de control de versiones en el desarrollo de una aplicación.
- **GitHub:** [20] Servicio en la nube donde se aloja Git. Permite que varios desarrolladores colaboren en un mismo proyecto y facilita un seguimiento de la evolución de este.

3

Especificación y Análisis

3.1. Roles de usuario

En primer lugar, se van a describir todos los roles que componen la aplicación junto a un breve explicación de las distintas funcionalidades generales de las que dispone cada uno de ellos (Cuadros 1, 2, 3, 4 y 5).

Rol	Administrador
Descripción	Este usuario es único, en este caso corresponde al encargado de gestionar a los moderadores. No obstante, tiene las mismas funcionalidades que un moderador
Funcionalidades	Gestión de moderadores, Gestión de Campos de Golf y Gestión de Jugadores

Cuadro 1: Rol Administrador

Rol	Moderador
Descripción	Un moderador tendría visibilidad de todos los usuarios de la aplicación, a su vez, será en el encargado de poder desactivar y activar cuentas según vea adecuado
Funcionalidades	Gestión de Campos de Golf y Gestión de Jugadores

Cuadro 2: Rol Moderador

Rol	Administrador de Campo de Golf
Descripción	Este usuario es con el que se registra por primera vez un campo de golf y tiene las mismas funcionalidades que un moderador de campo de golf con la diferencia de que este se encarga de dar de altas nuevos moderadores o ascenderlos a administradores.
Funcionalidades	Gestión de Moderadores del Campo, Gestión del Perfil, Gestión de Noticias, Gestión de Torneos, Gestión de Partidas y Gestión de Hoyos

Cuadro 3: Rol Administrador Campo

Rol	Moderador de Campo de Golf
Descripción	Un moderador de un campo de golf es el encargado de gestionar los distintos eventos que ofrece la aplicación relacionados al club al que pertenecen.
Funcionalidades	Gestión del Perfil, Gestión de Noticias, Gestión de Torneos, Gestión de Partidas y Gestión de Hoyos

Cuadro 4: Rol Moderador Campo

Rol	Jugador
Descripción	Un jugador podrá inscribirse en torneos, y a su vez organizar partidas con otros jugadores con los que tenga amistad. Asimismo tendrá una visualización de sus partidas pasadas acompañadas de gráficas con estadísticas que muestren cómo va evolucionando.
Funcionalidades	Perfil de Jugador, Amistades, Visualización de Campos, Inscripción de Torneos y Partidas

Cuadro 5: Rol Jugador

3.2. Requisitos Funcionales

En este punto se van a tratar las diversas funcionalidades de las que disponen los usuarios de la app junto a una breve descripción de cada una de ellas. Igual que en el punto anterior, se van a desglosar por los distintos roles que hay. (Cuadros 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21 y 22)

ID	Nombre	Descripción
R-00.1	Iniciar Sesión	Un usuario podrá iniciar sesión en la aplicación con su correo electrónico y su contraseña.
R-00.2	Cerrar Sesión	Un usuario podrá cerrar su sesión de la aplicación, si previamente tiene una sesión iniciada en ella.

Cuadro 6: R-00 Sesión

- *Administrador*

ID	Nombre	Descripción
R-01.1	Listar Moderadores	Un administrador podrá ver todos los usuarios con el rol moderador existente en la aplicación.
R-01.2	Registrar Moderador	Un administrador podrá registrar a un moderador con un correo electrónico.
R-01.3	Desactivar Moderador	Un administrador podrá desactivar la cuenta de un moderador activo, siendo notificado por su correo electrónico.
R-01.4	Activar Moderador	Un administrador podrá activar la cuenta de un moderador inactivo, siendo notificado por su correo electrónico.

Cuadro 7: R-01 Gestión Moderadores

- *Administrador y Moderador*

ID	Nombre	Descripción
R-02.1	Listar Campos	Un administrador o moderador podrá ver todos los campos existentes en la aplicación con sus correspondientes datos.
R-02.2	Listar Representantes	Un administrador o moderador podrá ver todos los usuarios vinculados a un campo de golf.
R-02.3	Desactivar Campo	Un administrador o moderador podrá desactivar un campo de golf activo y a todos los usuarios vinculados a él, siendo notificados por sus correos electrónicos.
R-02.4	Activar Campo	Un administrador o moderador podrá activar un campo de golf inactivo y a todos los usuarios vinculados a él, siendo notificados por sus correos electrónicos.
R-02.5	Desactivar Representante	Un administrador o moderador podrá desactivar a un usuario activo vinculado a un campo en específico y ser notificado por su correo electrónico.
R-02.6	Activar Representante	Un administrador o moderador podrá activar a un usuario inactivo vinculado a un campo en específico y ser notificado por su correo electrónico.

Cuadro 8: R-02 Gestión de Campos

ID	Nombre	Descripción
R-03.1	Listar Jugadores	Un administrador o moderador podrá ver todos los jugadores existentes en la aplicación con sus correspondientes datos.
R-03.2	Ver Gráfica	Un administrador o moderador podrá ver una gráfica lineal donde se muestran los registros por días en la app.
R-03.3	Ver Perfil	Un administrador o moderador podrá ver el perfil de un jugador.
R-03.4	Desactivar Jugador	Un administrador o moderador podrá desactivar a un jugador activo y notificarlo por su correo electrónico.
R-03.5	Activar Jugador	Un administrador o moderador podrá activar a un jugador inactivo y notificarlo por su correo electrónico.

Cuadro 9: R-03 Gestión de Jugadores

- *Administrador de Campo*

ID	Nombre	Descripción
R-04.1	Registrar Campo	Un administrador de campo podrá registrar a un campo en la aplicación, proporcionando todos los datos necesarios. (El administrador de campo no podrá acceder a la app hasta que un moderador active su cuenta)

Cuadro 10: R-04 Registro de Campo

ID	Nombre	Descripción
R-05.1	Listar Moderadores de Campo	Un administrador de campo podrá ver todos los usuarios con el rol moderador de campo vinculados a su mismo campo.
R-05.2	Registrar Moderador de Campo	Un administrador de campo podrá registrar a un moderador con un correo electrónico y una contraseña y vincularlo al campo al que pertenecen.
R-05.3	Desactivar Moderador de Campo	Un administrador de campo podrá desactivar la cuenta de un moderador de campo activo siendo notificado por su correo electrónico.
R-05.4	Activar Moderador de Campo	Un administrador de campo podrá activar la cuenta de un moderador inactivo siendo notificado por su correo electrónico.
R-05.5	Ascender Moderador de Campo	Un administrador de campo podrá ascender la cuenta de un moderador de campo a administrador de campo.

Cuadro 11: R-05 Gestión de Moderadores de Campo

- *Administrador y Moderador de Campo*

ID	Nombre	Descripción
R-06.1	Ver Perfil	Un administrador o moderador de campo podrá acceder al perfil del campo al que pertenece y ver todos sus datos.
R-06.2	Editar Perfil	Un administrador o moderador de campo podrá editar el perfil del campo al que pertenece, pudiendo modificar cualquiera de los datos.

Cuadro 12: R-06 Gestión de Perfil de Campo

ID	Nombre	Descripción
R-07.1	Ver Noticias	Un administrador o moderador de campo podrá ver todas las noticias creadas para el campo al que pertenecen.
R-07.2	Crear Noticia	Un administrador o moderador de campo podrá crear una noticia (activa o inactiva) para el campo al que pertenece y quedará registrado como el responsable de esa noticia.
R-07.3	Editar Noticia	Un administrador o moderador de campo podrá editar una noticia seleccionada y pasar a ser el nuevo responsable de esta.
R-07.4	Eliminar Noticia	Un administrador o moderador de campo podrá eliminar una noticia.
R-07.5	Ver Likes	Un administrador o moderador de campo podrá ver que jugadores de la plataforma han dado like a una noticia.

Cuadro 13: R-07 Gestión de Noticias

ID	Nombre	Descripción
R-08.1	Ver Torneos	Un administrador o moderador de campo podrá ver todos los torneos creados para el campo al que pertenecen independientemente del estado en el que se encuentren.
R-08.2	Crear Torneo	Un administrador o moderador de campo podrá crear un torneo que se disputará en el campo al que pertenecen.
R-08.3	Editar Torneo	Un administrador o moderador de campo podrá editar un torneo seleccionado, si este aún no ha sido finalizado.

R-08.4	Subir Imágenes	Un administrador o moderador de campo podrá subir imágenes relacionadas a un torneo en específico.
R-08.5	Eliminar Imágenes	Un administrador o moderador de campo podrá eliminar cualquier imagen de un torneo.
R-08.6	Ver Inscripciones	Un administrador o moderador de campo podrá ver todos los jugadores inscritos en un torneo.
R-08.7	Descargar Excel Inscripciones	Un administrador o moderador de campo podrá descargar un excel con todos los jugadores inscrito en un torneo.
R-08.8	Cancelar Torneo	Un administrador o moderador de campo podrá cancelar un torneo antes de ser iniciado.
R-08.9	Iniciar Torneo	Un administrador o moderador de campo podrá iniciar un torneo para que los jugadores puedan empezar a jugar. El sistema realiza el matchmaking automáticamente y agrupa a los jugadores en partidas.
R-08.10	Finalizar Torneo	Un administrador o moderador de campo podrá finalizar un torneo. Automáticamente se genera un ranking a partir de las puntuaciones conseguidas por los jugadores del torneo.
R-08.11	Ver Resultados	Un administrador o moderador de campo podrá ver los resultados de los jugadores, tanto en directo, cómo una vez finalizados.

Cuadro 14: R-08 Gestión de Torneos

ID	Nombre	Descripción
R-09.1	Ver Calendario	Un administrador o moderador de campo podrá ver un calendario con todas las partidas que hay en el campo al que pertenece en los días de un mes seleccionado.
R-09.2	Ver Día Concreto	Un administrador o moderador de campo podrá ver las partidas que hay en el campo al que pertenecen en un día seleccionado en franjas de 15 minutos.
R-09.3	Crear Partida	Un administrador o moderador de campo podrá crear una partida en una franja horaria que se encuentre libre.
R-09.4	Ver Partida	Un administrador o moderador de campo podrá ver una partida dada una franja horaria seleccionada junto a todos los jugadores que la disputan.
R-09.5	Agregar Jugador	Un administrador o moderador de campo podrá agregar un jugador a una partida antes de la hora de salida. Si no hay jugadores previamente, pasaría a ser el marcador de la partida.
R-09.6	Eliminar Jugador	Un administrador o moderador de campo podrá eliminar a un jugador que participara en una partida en específico antes de la hora de salida
R-09.7	Eliminar Partida	Un administrador o moderador de campo podrá eliminar una partida y dejar esa franja horaria libre antes de la hora de salida.

Cuadro 15: R-09 Gestión de Partidas

ID	Nombre	Descripción
R-10.1	Ver Hoyos	Un administrador o moderador de campo podrá ver todos los hoyos del campo al que pertenecen.

R-10.2	Descargar Plantilla	Un administrador o moderador de campo podrá descargar una plantilla para generar un excel con la información de los hoyos del campo al que pertenece.
R-10.3	Crear Hoyos (Excel)	Un administrador o moderador de campo podrá agregar los 18 hoyos del campo al que pertenecen en caso de que no hayan sido previamente creados subiendo un excel con la información necesaria.
R-10.4	Crear Hoyos (Formulario)	Un administrador o moderador de campo podrá agregar los 18 hoyos del campo al que pertenecen en caso de que no hayan sido previamente creados rellenando un formulario.
R-10.5	Editar Hoyo	Un administrador o moderador de campo podrá editar la información de un hoyo seleccionado.

Cuadro 16: R-10 Gestión de Hoyos

▪ *Jugador*

ID	Nombre	Descripción
R-11.1	Registrar Jugador	Un jugador podrá registrarse en la aplicación proporcionando los datos requeridos.

Cuadro 17: R-11 Registro de Jugador

ID	Nombre	Descripción
R-12.1	Ver Perfil	Un jugador podrá acceder a su perfil y ver todos sus datos.
R-12.2	Editar Perfil	Un jugador podrá editar su perfil y modificar los datos que desee.
R-12.3	Ver Estadísticas	Un jugador podrá ver datos estadísticos y gráficas relacionadas a sus partidas, golpes y puntos obtenidos.

Cuadro 18: R-12 Perfil de Jugador

ID	Nombre	Descripción
R-13.1	Ver Usuarios	Un jugador podrá ver a los demás usuarios de la plataforma con los que no tenga una relación de amistad o una solicitud pendiente por parte del usuario.
R-13.2	Enviar Solicitud	Un jugador podrá enviar una solicitud de amistad a otro usuario si no lo ha hecho ya previamente.
R-13.3	Ver Solicitudes	Un jugador podrá ver las solicitudes de amistad que tenga pendientes.
R-13.4	Aceptar Solicitud	Un jugador podrá aceptar una solicitud de amistad.
R-13.5	Rechazar Solicitud	Un jugador podrá rechazar una solicitud de amistad.
R-13.6	Ver Amigos	Un jugador podrá ver a todos los usuarios con los que tenga una relación de amistad en la aplicación.
R-13.7	Eliminar Amistad	Un jugador podrá eliminar una relación de amistad con otro usuario.
R-13.8	Filtrar Nombre Usuarios	Un jugador podrá filtrar por el nombre de otro usuario en cualquiera de las vistas que se encuentre.

Cuadro 19: R-13 Amistad

ID	Nombre	Descripción
R-14.1	Ver Campos	Un jugador podrá ver los distintos campos de golf activos en la aplicación filtrando por nombre o la provincia en la que se encuentran.
R-14.2	Ver Ubicación	Un jugador podrá ver la ubicación exacta de un campo de golf en un mapa.
R-14.3	Ver Hoyos	Un jugador podrá ver los hoyos que componen un recorrido de un campo de golf seleccionado.
R-14.4	Ver Noticias	Un jugador podrá ver todas las noticias activas publicadas por un campo.
R-14.5	Dar Like Noticia	Un jugador podrá dar like a una noticia si aún no lo ha hecho.
R-14.6	Quitar Like Noticia	Un jugador podrá quitar un like de una noticia después de haberle dado like previamente.

Cuadro 20: R-14 Visualización de Campos

ID	Nombre	Descripción
R-15.1	Ver Torneos Pendientes	Un jugador podrá ver los torneos en los que está inscrito y aún no han finalizado.
R-15.2	Desinscribir	Un jugador podrá desinscribirse de un torneo pendiente si aún no se ha iniciado.
R-15.3	Jugar	Un jugador podrá acceder a su tarjeta en un torneo que ha sido iniciado.
R-15.4	Ver Próximos Torneos	Un jugador podrá ver los próximos torneos que se disputan en los distintos campos activos de la aplicación.
R-15.5	Inscribir	Un jugador podrá inscribirse a un torneo si aún no se ha iniciado o si no estaba inscrito previamente.
R-15.6	Ver Torneos Pasados	Un jugador podrá ver los torneos que ha disputado.

R-15.7	Ver Imágenes	Un jugador podrá ver las imágenes de un torneo una vez que haya finalizado.
R-15.8	Ver Inscripciones	Un jugador podrá ver todos los jugadores que participan en un torneo.
R-15.9	Ver Resultados	Un jugador podrá ver el resultado de los jugadores de un torneo en vivo o una vez haya finalizado.
R-15.10	Filtrar Torneos	Un jugador podrá filtrar torneos por un campo seleccionado.
R-15.11	Ver Campos	Un jugador podrá ir a la vista de visualización de campo seleccionando el campo de un torneo en específico.

Cuadro 21: R-15 Inscripción de Torneos

ID	Nombre	Descripción
R-16.1	Ver Partidas Pendientes	Un jugador podrá ver las partidas en las que participa y aún no han finalizado.
R-16.2	Abandonar	Un jugador podrá abandonar una partida en la que participa si aún no ha sido iniciada.
R-16.3	Modificar Datos	Un jugador podrá cambiar su handicap de juego y el color de salida de una partida en la que pertenece antes de ser iniciada seleccionandose a sí mismo en la ventana de la partida.
R-16.4	Expulsar Jugador	Un jugador podrá expulsar a otros jugadores seleccionados en la ventana de la partida siempre que él sea el creador de esta.
R-16.5	Invitar Jugador	Un jugador podrá invitar a otros jugadores mientras haya un hueco libre (menos de 4 jugadores) y tengan una relación de amistad.

R-16.6	Jugar	Un jugador podrá acceder a su tarjeta en una partida que ha sido iniciado.
R-16.7	Ver Partidas Amigos	Un jugador podrá ver todas las partidas pendientes de sus amigos en las que no participa.
R-16.8	Unirse Amistad	Un jugador podrá unirse a una partida de un amigo siempre que haya un hueco libre y no haya sido iniciada.
R-16.9	Ver Partidas Pasadas	Un jugador podrá ver todas las partidas que ha disputado y están finalizadas.
R-16.10	Ver Resultados	Un jugador podrá ver su tarjeta y la de los demás jugadores de aquellas partidas que han jugado juntos y han sido finalizadas.
R-16.11	Filtrar Partidas	Un jugador podrá filtrar por campo, y ver las partidas que se juegan en ese campo, o por amigos, viendo aquellas en las que forma parte el jugador seleccionado.
R-16.12	Ver Campos	Un jugador podrá ir a la vista de visualización de campo seleccionando el campo de una partida en específico.
R-16.13	Crear Partida	Un jugador podrá acceder al calendario de un campo en específico para crear una partida en una franja horaria que esté libre.
R-16.14	Unirse	Un jugador podrá unirse a una partida ya creada en una franja horaria desde el calendario de un campo en concreto.
R-16.15	Marcar Golpes	Un jugador podrá acceder a una tarjeta de una partida iniciada y marcar cuantos golpes ha hecho él y el resto de jugadores en una partida siempre que él sea el creador de esta.

3.3. Requisitos No Funcionales

1. El sistema cifrará las contraseñas de los usuarios en la base de datos y las almacenará de tal forma que no sean visibles para nadie.
2. El sistema deberá implementar un mecanismo de autenticación utilizando JWT (JSON Web Tokens) [12] en ASP.NET Core [1], asegurando una gestión de autorización basada en roles y políticas de seguridad robustas.
3. El sistema deberá ser capaz de gestionar eficientemente el volumen previsto de usuarios sin experimentar ninguna degradación en el rendimiento.
4. El sistema debe ser intuitivo y comprensible, proporcionando una experiencia de usuario accesible y fluida.
5. El sistema deberá estar diseñado para facilitar su mantenimiento y actualización, permitiendo que estas tareas se realicen con eficacia y mínimo impacto en la operación.

4

Diseño del Sistema

Tanto el Backend como el FrontEnd siguen una arquitectura *MVC*⁶ basada en módulos. Gracias a este modelo se consigue diferenciar de manera concisa la lógica que hay que aplicar en cada lugar y facilita tanto la comunicación de ambas partes a través de las rutas especificadas con urls y peticiones http, como la comodidad a la hora del desarrollo.

Ambas partes de la aplicación se encuentran divididas en módulos. En el front se separan debido a las vistas con las que el usuario puede interactuar y en el back por el conjunto de tablas relacionadas entre sí en la base de datos.

El modelo de conexión que se usa entre Frontend y Backend es un modelo cliente-servidor, donde el usuario realiza una acción y espera a obtener una respuesta por parte del sistema. (Figuras 1 y 2)

⁶Model-View-Controller

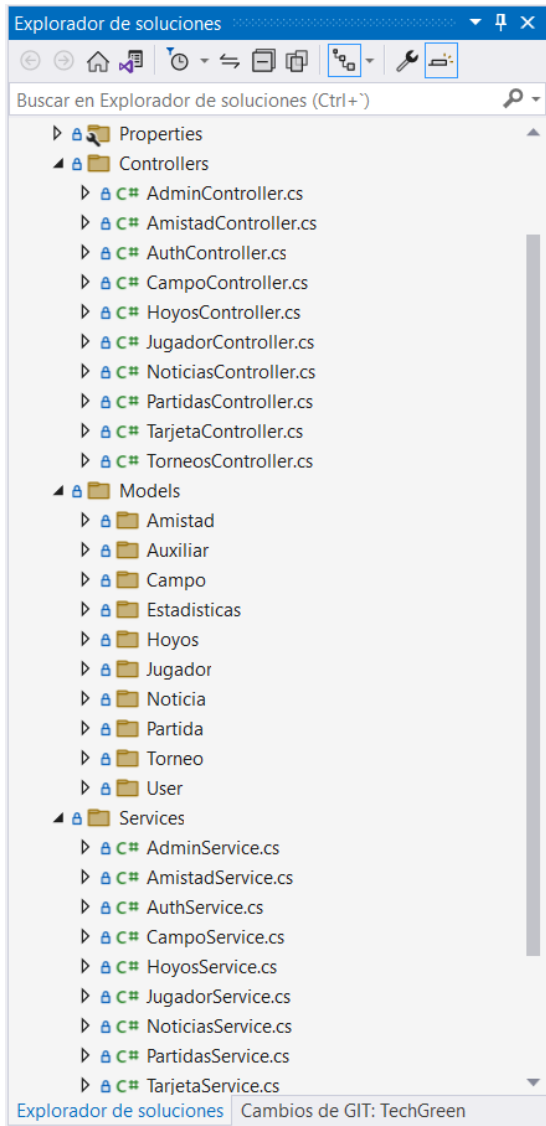


Figura 1: Modelo Backend

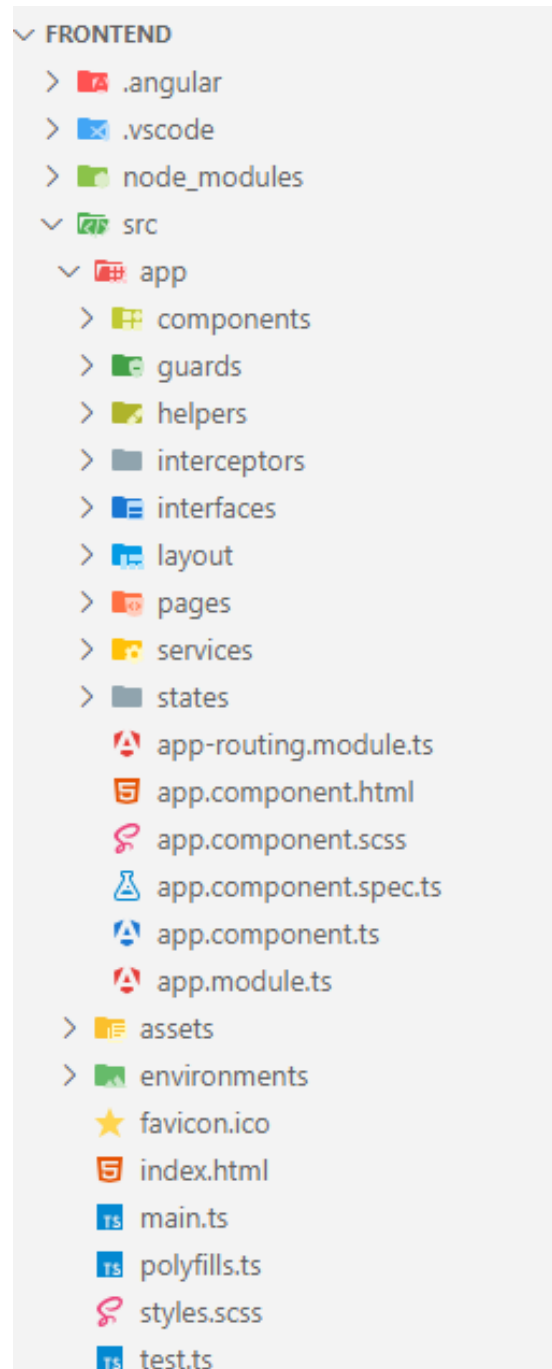


Figura 2: Modelo Frontend

4.1. Base de Datos

Para el desarrollo de esta aplicación se ha utilizado una base de datos relacional MySQL [2]. Estas bases de datos se caracterizan por almacenar los datos en tablas, donde cada fila representa un objeto y cada columna un atributo de este. Además, las tablas se relacionan entre ellas a través de las claves primarias y claves foráneas, permitiendo identificar características de un registro en otra tabla de la que se encuentra. (Figura 3)

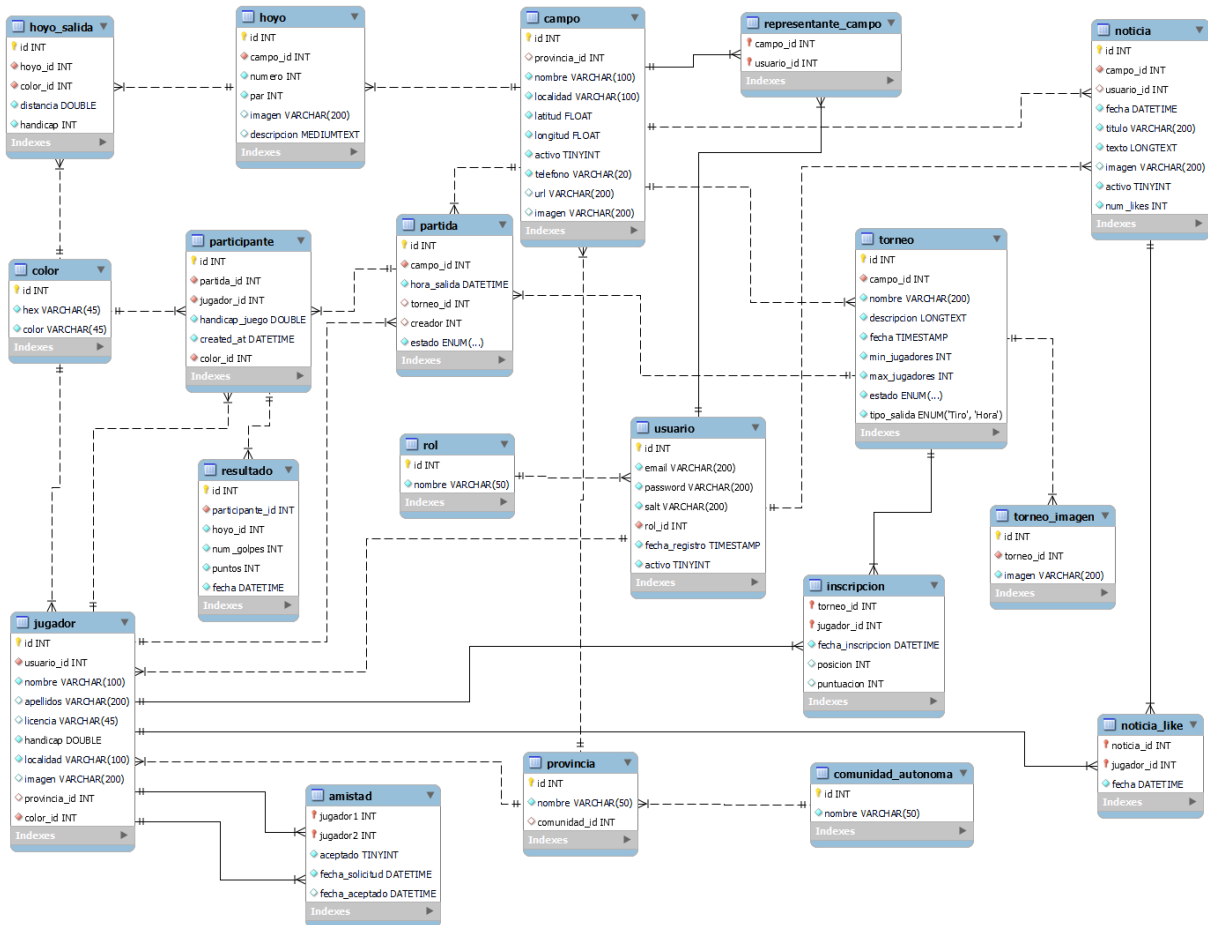


Figura 3: Diagrama Entidad Relación

En el diagrama se pueden contemplar todas las tablas que se han generado para hacer posible el desarrollo del proyecto con sus correspondiente relaciones. Por ejemplo, la tabla “Usuarios” guarda todos los datos referente a la cuenta de los usuarios y se indica qué rol tiene cada uno.

Así mismo, se puede observar cuáles son las dos principales entidades de la app: “Jugador” y “Campo”. Estas se relacionan a su vez con otras entidades secundarias como lo son “Torneo”, “Noticia” y “Partida”. La relación entre “Campo” y estas entidades es de cardinalidad **1:N**, ya que cada una de estas entidades tan solo pueden pertenecer a un mismo campo. Mientras que por otro lado, con “Jugador” la relación es de **N:N**, de esta última aparecen tablas intermedias como lo son “Participante” o “Inscripcion” debido a que varios jugadores pueden interactuar a la vez con varias entidades.

La base de datos ha sido desplegada en un contenedor en Docker [4]. Para poder acceder a ella es necesario estar en la misma red en la que se encuentra el contenedor corriendo y usar el puerto 3307, que es el que se ha usado en este caso. (Figura 4)

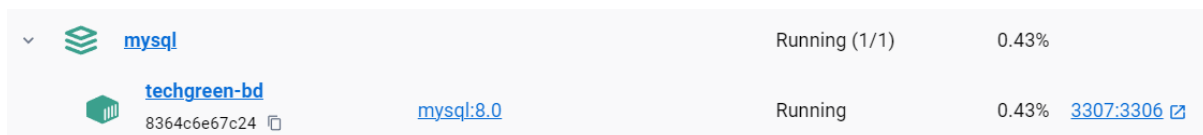


Figura 4: DockerDesktop Running

4.2. Backend

Para este caso ASP NET Core [1] se le ha dado el uso de una API REST [8], a la que le front está llamando cada vez que un usuario realiza una acción a través de peticiones http. A través de Swagger [21], se pueden mostrar los posibles *endpoints* de las que se disponen en el Backend. Además, esta herramienta failita la opción de simular las peticiones introduciendo los parámetros manualmente para obtener las posibles respuestas, así como si simulara el comportamiento del sistema. (Figura 5)

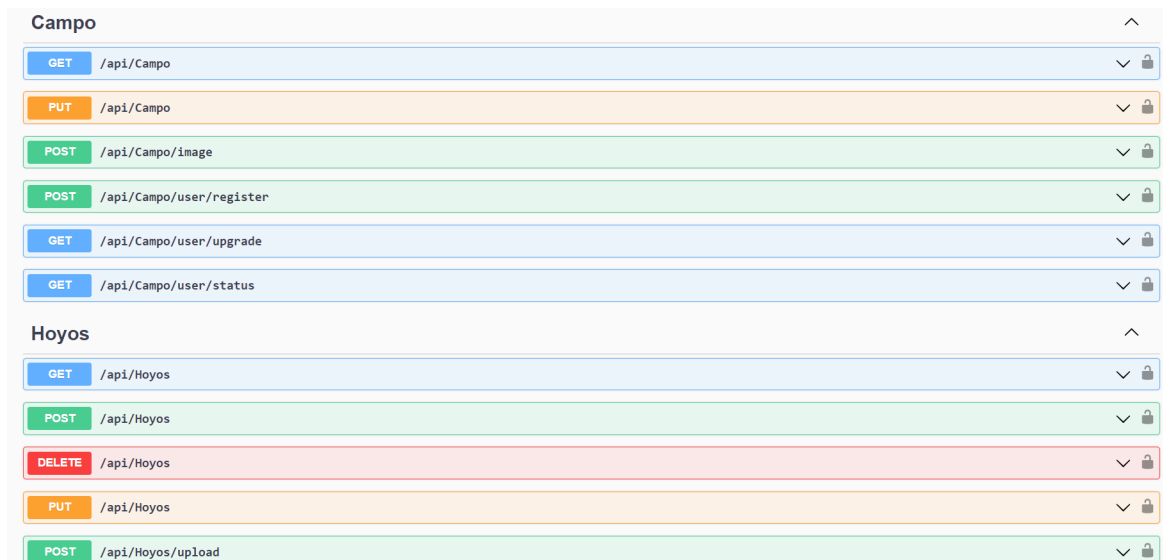


Figura 5: Swagger

■ Models

Los modelos son las representaciones de los objetos que se guardan en la base de datos con sus atributos tipados. Sirven para indicar que datos se quieren extraer exactamente a la hora de hacer una consulta.

■ Services

En este apartado se encuentran los repositorios, que tienen la función de realizar las consultas *SQL*⁷ a la base de datos para obtener los datos necesarios que pide el usuario. Aquí se tipan los datos obtenidos con el modelo deseado.

■ Controllers

Son los puentes entre el Backend y el Frontend. Cada función tiene asociada un método http junto a una url para poder indentificarlos y saber a cuál llamar en cada caso. La mayoría de la lógica del Back se encuentra en esta parte.

4.3. Frontend

Un proyecto en Angular [3] parte de un módulo raíz donde se inicia todo el proyecto y se van generando todos los distintos componentes que forman la estructura de la app. En este

⁷Structured Query Language

módulo raíz se genera el enrutamiento de la aplicación, que será lo que permita al usuario navegar entre las distintas vistas implementadas.

- **Services**

En este apartado se hacen las llamadas a la API de ASP NET Core a través de peticiones http. En cada servicio se accede a una url distinta dependiendo del módulo al que haga referencia. Cada petición puede recibir tipos simples (int, boolean, ...), o bien, un JSON que hace referencia a un modelo preestablecido en la sección "Interface". Además permite pasarle objetos a través de parámetros o por el cuerpo de la petición, con el mismo formato que el back necesite.

- **Interfaces**

Las interfaces sirven para replicar los modelos predefinidos en el Backen para poder manipular los datos desde el Frontend. De este forma se consigue conectar los datos entre el back y el front para ser visualizados.

- **Components**

Se encuentran dentro de la sección "Pages" en el proyecto, y se dividen en varios archivos (html, scss, ...), pero comparte la misma funcionalidad: tratar los datos de tal forma que permita su visualización como se requiere en cada página. Se pueden mostrar los datos directamente en los componentes visuales que conforman la página en la que el usuario se encuentre, o se pueden aplicar algoritmos para modificar u obtener nuevos datos.

Cabe de destacar de la estructura del proyecto, que se han creado módulos independiente para cada componente. Esto se ha realizado con el fin de que tan solo se generen los paquetes que se necesita en cada página haciendo que sea más eficiente y se consuman los menos recursos posibles. En el global tan solo se han llamado a los paquetes que van a utilizar todos los módulos.

- **Extras**

- **Guards**

Sirven para restringir el acceso a las distintas páginas de la app dependiendo de

las condiciones que se requieran para cada una, como estar autenticado o que el rol correspondiente tenga permitida dicha funcionalidad.

- **State**

En el “State” se almacenan los datos que se obtienen del usuario cuando hace login o se registra para poder acceder a ellos desde cualquier módulo. Estos datos son los que hay almacenados en el token cifrado.

- **Layout**

Corresponde a la barra de navegación que se encuentra en la aplicación. Dependiendo del rol del usuario aparecerán unas secciones u otras.

- **Components**

En esta sección aparecen componentes Standalone [22] que se usan en distintas partes del proyecto. Estos componentes son completamente independientes y no requieren de un módulo asociado para su uso.

4.4. Diagrama de Secuencia

Cuando un usuario interactúa con la aplicación, ya sea cargando una vista para obtener una serie de datos y visualizarlos en diferentes componente, o al rellenar un formulario para generar un nuevo registro en la base de datos, todas las capas del proyecto están involucradas siguiendo un trayecto de ida y vuelta. (Figura 6)

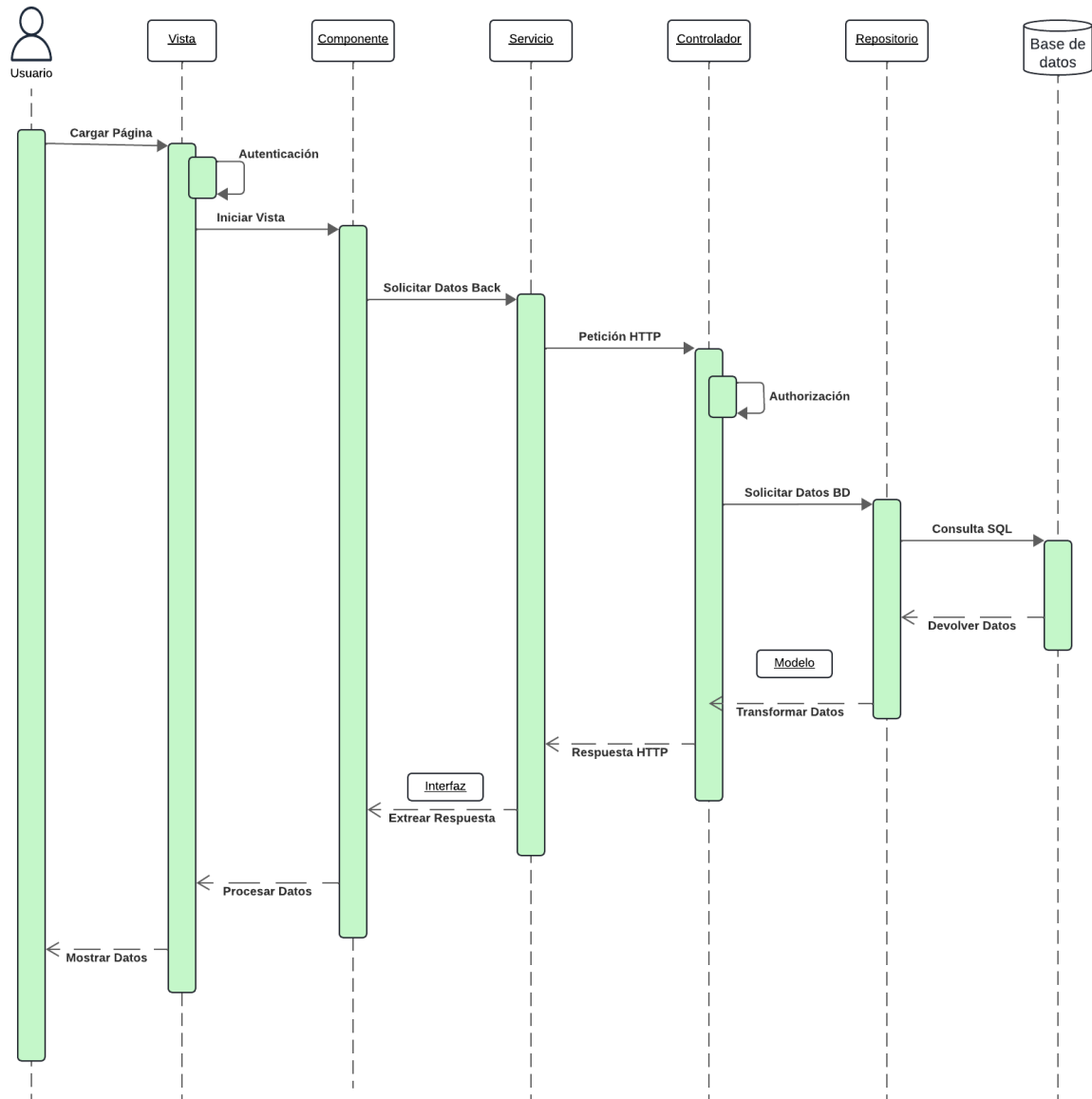


Figura 6: Diagrama de Secuencia

Para este caso se ha usado de ejemplo el acceso a una vista donde se mostrarán los datos

correspondiente a ella. Si el rol del usuario es válido, el componente llamará al servicio para realizar una petición http a una ruta específica del Backend pasándole los parámetros que sean necesarios.

El controlador recibirá la petición y validará tanto si el usuario está autorizado cómo si los parámetros obtenidos son correctos. En caso contrario, éste lanzará una excepción al servicio. Si todo está correcto, se le solicitará al repositorio que realice una consulta a la base de datos para obtener los datos en el formato concreto definido por el modelo que se utilice.

Finalmente, se devolverán los datos obtenidos como respuesta al servicio para que éste lo pueda procesar en la interfaz de la petición y el componente se encargue de cargarlos en la vista para mostrarlos.

5

Implementación y Pruebas

La metodología a seguir para el desarrollo del proyecto está basada en iteraciones. Todos los requisitos definidos previamente se han desglosado en subgrupos y cada uno de estos corresponde a una iteración. Siguiendo esta línea de trabajo se consigue que la aplicación vaya siendo funcional poco a poco, permitiendo que se puedan integrar y probar las partes que se van desarrollando e ir evaluandola de forma progresiva. Con esto se consigue una mayor flexibilidad a los cambios, debido a que no hay que esperar a que la app esté completamente funcional para ver si la aplicación desarrollada va cumpliendo con las expectativas.

Como se ha podido ver a lo largo de este documento, las distintas características de la aplicación se agrupan en los roles que la compone. Por lo tanto, cada iteración hace referencia al conjunto de funcionalidades de un rol.

El flujo de trabajo siempre va a ser el mismo. Del todas de vistas que dispone un rol, se elige una y se implementan todos los requisitos necesarios. Se empieza por el Backend con la generación de modelos, consultas SQL y endpoints. Y se pasa al Frontend con la creación de interfaces, peticiones https y algoritmos necesarios. Finalmente se muestran los datos obtenidos en el formato necesario con los estilos definidos.(Figura 7)

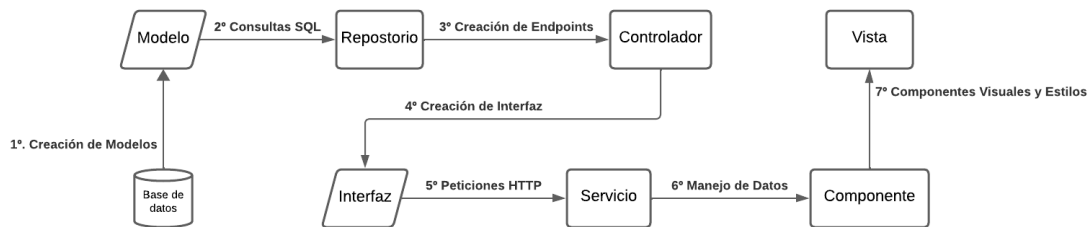


Figura 7: Flujo de trabajo

Para empezar a implementar otro conjunto de requisitos hay que validar previamente que todos los que se acaban de finalizar siguen un flujo correcto, que tengan un comportamiento esperado y, en caso de error, que permita seguir usando la aplicación sin quedarse congelado.

5.1. Primera Iteración

La primera fase del proyecto es la que menos funcionalidades ha implementado, pero más carga de trabajo ha llevado. Esta fase se puede dividir en tres partes:

1. **Base de Datos:** La primera línea que ha seguido el desarrollo del proyecto se ha enfocado en la creación de la base de datos. Gracias a MySQL Workbench [5], se puede crear un diagrama UML ⁸ con la interfaz que tiene integrada, que además, permite generar automáticamente la base de datos a partir de ella. En esta fase se definieron las distintas entidades con sus relaciones correspondientes de tal forma que se permitiera la posibilidad de generar tablas intermedias o nuevos atributos en caso de que fuera necesario.

Con la base de datos ya definida, se desplegó en Docker. Todo esto se realizó gracias al archivo DockerCompose [23], donde se crea un contenedor con una imagen de MySQL [2], se le otorgan unos credenciales y se guarda el script en un volumen. (Figura ??).

⁸UML: Unified Modeling Language



Figura 8: DockerCompose

2. **Arquitectura:** En este punto, se continuó con la creación del proyecto Backend. Para ello se generaron las distintas secciones que lo forman (Services, Models y Controllers) y se instalaron todos los paquetes necesarios para realizar la conexión con la base de datos y para el uso de JWT [12] con la finalidad de generar tokens y restringir las peticiones de la API.

Posteriormente se creó un proyecto Frontend de Angular [3] a partir de Node.js [13], y se integró desde un inicio con PrimeNG [16]. Después, se configuró el archivo *app.module.ts*, que corresponde al módulo raíz del proyecto, con todos los componentes que comparten el resto de submódulos. Y finalmente, se crearon todas las partes mencionadas anteriormente que conforman el front de la aplicación (Services, Pages, Guards...).

3. **Inicio de Sesión:** Una vez con la base de datos y el esqueleto del proyecto ya creados, lo primero que se implementó fue la posibilidad de que un usuario iniciara sesión con una cuenta registrada en la app. Para ello se creó un usuario inicial con el correo electrónico *techgreen.golf@gmail.com* que corresponde al único administrador del sistema. Esta funcionalidad además genera los tokens desde el back y los guarda en el tanto en el "State" como en las cookies del navegador en el front.

5.2. Segunda Iteración

A partir de este punto se configuraron las bibliotecas restantes del proyecto tanto del front (Leaflet, Ngx-Cookies ...), como del back (CloudinaryNet, Mailkit ...) y se empezó a aplicar el flujo de trabajo descrito previamente de forma iterativa. Para este punto, se definieron todas las rutas que se usarán en el navegador para referenciar a los módulos.

En esta iteración los roles seleccionados a implementar fueron los de Administrador y de Moderador. Todas las vistas tienen un estilo muy similar, listar datos en una tabla, ver perfiles y activar/desactivar cuentas de usuarios.

5.3. Tercera Iteración

Para esta fase del proyecto se terminó de pulir la barra de navegación. Según el rol que esté usando la aplicación en el momento se mostrarán unas pestañas u otras. Además, se agregaron los “Guards” e “Interceptors” en las rutas del proyecto, para no permitir que usuarios no deseados, ya puedan ser por tener rol inválido o no disponer de una cuenta en la aplicación, puedan acceder a ciertos apartados.

Esta iteración se centra sobre todo en el rol de Campos. Al ser la que más funcionalidades y módulos dispone, es la en que más tiempo de implementación se tuvo que invertir. En esta etapa del proyecto se han implementado diversas funciones que han servido como plantilla para futuras implementaciones, como por ejemplo la subida de fotos a Cloudinary [10] o la integración del mapa y del calendario. Al ser un CRM tiene varios aspectos muy similares a los del administrador, permitiendo integrar CRUDs ⁹ en la mayoría de los módulos.

5.4. Cuarta Iteración

La última iteración consta de todos los requisitos del rol Jugador. Es la etapa final del proyecto en lo referente al desarrollo. En este caso las vistas se han ajustado con la idea de que un jugador utilice la aplicación desde su dispositivo móvil.

⁹Create-Read-Update-Delete

Cabe destacar en este punto, que se han implementado dos algoritmos dentro del sistema para mejorar la experiencia del usuario. El primero se puede encontrar a la hora de registrar a un jugador. Se puede ver que al rellenar los campos nombre y apellidos, se hace una petición al Backend para obtener una lista de posible licencias federativas de golf que coincidan con los parámetros ya mencionados. Esto se consigue realizando “Web Scrapping” [24] a la página web de la federación española de golf, donde hay un servicio que te facilita obtener la licencia y el handicap de un jugador de golf. Si el usuario seleccionara una de las posibles licencias, se completaría el campo handicap automáticamente con el valor asociado a la opción seleccionada. Además, si se escribe directamente la licencia, se puede obtener el handicap siguiendo los mismos pasos.

El otro algoritmo que se ha implementado aprovecha las coordenadas geográficas de los campos de golf para ordenarlos de más cercano a más lejano dependiendo de la posición actual del jugador en la que se encuentra cuando accede a la vista de los posibles campos de la aplicación, Esto ha sido posibles gracias a la implementación de la fórmula de Haversine, que permite calcular la distancia entre dos puntos en una esfera.

Finalmente se han realizado varias pruebas de validación a nivel de usabilidad. Se han registrado varios usuarios con distintos roles y características, y se ha simulado su comportamiento en diferentes entornos para comprobar que las respuestas dadas por el sistema son las esperadas.

5.5. Pruebas

La etapa final en lo relacionado a la implementación, se ha centrado en realizar pruebas al sistema y obtener las respuestas esparadas. Para este proyecto, se ha decidido priorizar la realización de pruebas de integración, ya que son las que mayor importancia tienen dentro del funcionamiento global del sistema. Como se ha comentado en varios puntos de este documento, la mayoría de la lógica se encuentra en realizar peticiones al Backend y guardar u obtener diferentes datos en MySQL. Por lo tanto, las pruebas se han centrado exclusivamente en esta parte, cubriendo diferentes aspectos para simular situaciones excepcionales que po-

drían ocurrir en escenarios reales como inicios de sesión inválidos, peticiones sin autorización y obtención de datos esperados entre otros.

Para facilitar la ejecución de estas pruebas, se ha utilizado Postman [11], donde se han importado todos los JSONs generados a partir de Swagger [21] en ASP.NET Core [1], permitiendo una gestión eficiente de las pruebas. Asimismo, se ha realizado una copia de la base de datos real del sistema para garantizar que las pruebas se ejecuten en un entorno lo más cercano posible al que tendría un usuario al usar la aplicación, pero sin modificar los registros reales. En el repositorio de GitHub [20], se encuentra un json con todas las pruebas de integración que se han realizado.

Las pruebas realizadas incluyen múltiples iteraciones para observar principalmente los tiempos de esperas obtenidos para cada caso y ver si realmente son eficientes. Se han validado el comportamiento del sistema ante diferentes situaciones, como errores inesperados o entradas que tengan una salida esperada. Finalmente, se ha capturado el registro de todas las pruebas ejecutadas, incluyendo el tiempo de ejecución de cada una, lo que ha permitido analizar el rendimiento y la robustez del sistema de manera precisa (Figura 9).

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	Valid	5	24s 810ms	195	61 ms

All Tests Passed (195) Failed (0) Skipped (0)

Figura 9: Registro de Pruebas

6

Conclusiones y Líneas Futuras

6.1. Conclusiones

Una vez finalizada la implementación del proyecto las conclusiones a destacar pueden llegar a dividirse en dos bloques.

6.1.1. Conclusiones sobre el Proceso de Desarrollo

Gracias a la estructuración que se ha seguido en la creación del proyecto, el desarrollo no ha sido muy tedioso, debido a que se había especificado en detalle qué función debía realizar cada parte.

En lo referente al desarrollo Backend, trabajar con ASP NET Core [1] ha resultado ser muy satisfactorio. La mayoría de desarrolladores suelen trabajar en .NET [7] con Entity Framework [25], un *ORM*¹⁰ que genera directamente los modelos acorde a la base de datos que se seleccione incluyendo una serie de funciones que equivalen a consultas SQL básica sobre las diferentes tablas. No obstante, en este proyecto se han implementado los modelos y las consultas a la base de datos manualmente, permitiendo así una mayor flexibilidad a la hora de definir los objetos y al hacer las peticiones, pudiendo ajustarlos a las necesidades que se requerían en cada caso.

Por otro lado, para el Frontend, Angular [3] ha resultado ser muy completo. En algunos aspectos, sobre todo para aquellos componentes que eran más simples, puede resultar algo tediosos debido a lo desglosado que se encuentra un solo componente. No obstante, esto se ha

¹⁰ORM: Object Relational Mapping

podido solventar con los ya mencionados componentes Standalone [22] permitiendo así una mayor flexibilidad.

6.1.2. Conclusiones sobre la Aplicación

Con todos los requisitos implementados y las pruebas de la app a nivel de usuario, se puede concluir que los resultados obtenidos son los inicialmente establecidos. El objetivo principal de unir el mundo del golf con las nuevas tecnologías de la sociedad se cumple con creces.

Se ha obtenido una interfaz de usuario simple, útil y fácil de comprender, tanto a nivel de jugador como de administrador de campo. La interacción entre roles permite que la aplicación vaya creciendo constantemente a medida que los usuarios la emplean, consiguiendo así que el número de registros pueda llegar a aumentar significativamente con el paso del tiempo.

6.2. Líneas Futuras

El principal objetivo con visión de futuro, sería conseguir desplegar la aplicación en la nube con algún servicio de alojamiento en esta (AWS [26], Azure [27], Google Cloud [28]...). Gracias a esto, los usuarios podrían acceder desde cualquier dispositivo con acceso a Internet y no tener que instalar todas las tecnologías necesarias para ejecutar el proyecto a nivel local.

Tras el uso de la aplicación, una vez finalizado su desarrollo, se han observado diversas mejoras que podrían añadirse a la aplicación.

1. **Guía de Estilos.** *TechGreen* sigue una línea de estilos muy simplificada. Para su primera versión la principal importancia ha sido crear un flujo sencillo, eficiente y fácil de comprender por parte del usuario. Se podría llegar a crear una paleta de colores, o una guía de estilos para enriquecer todas las vistas.
2. **Plantillas para Emails.** Ahora mismo los emails se están enviando solo con textos planos. La posibilidad de generar unas plantillas HTML con estilos conseguiría hacer más atractivos los correos. Aunque también, se podría contratar un servicio de terceros como podría ser “Brevo” [29], permitiendo así la creación de plantillas y su posible envío a través de la propia API.

3. **Notificaciones vía Emails.** Relacionado también a los emails, sería interesante la integración de un sistema de notificaciones para formentar la interacción del usuario en la app. Ejemplos de esto, podríamos ser si un campo crea un torneo, o si un amigo invita a un partida a un jugador, este reciba un email con los datos relacionado a la acción realizada.
4. **Evolución Automática de Handicap.** Ahora mismo el handicap tan solo varía dependiendo de sí un jugador lo cambia antes de empezar una partida. Se ha pensado en desarrollar un algoritmo que dado los golpes y los puntos obtenidos te actualice el handicap de manera automática, para así no tener que modificarlo de manera manual y así se asemeje un poco más a la realidad.
5. **Seguimiento de Campos.** Permitir a los jugadores seguir a los campos que deseen para no perderse las novedades de las noticias que publiquen o ser los primeros en enterarse de los torneos que se creen.
6. **Modalidades de Juego.** El algoritmo que ahora mismo genera los puntos a la hora de insertar el número de golpes, simula la modalidad de golf “Stableford” [30]. Debido a que existen más modalidades, se podrían llegar a integrar diferentes algoritmos para simularlos y que así el jugador pueda seleccionar que modalidad desea jugar antes de crear una partida.
7. **Visualización de Perfiles.** Otra mejora planteada sería que los jugadores tuvieran la opción de poder compartir sus resultados o sus datos estadísticos con otros jugadores de la app. A su vez, que puedan poner privadas sus cuentas para que otros usuarios no tengan acceso a dicha información.
8. **Tareas Programables.** Finalmente, tras realizar varias pruebas se han dado casos donde el sistema espera una acción por parte del usuario pero este no lo realiza, con lo que el flujo queda alterado. Ejemplos de esto sería una partida que ya ha terminado no finalizarla nunca, o un torneo que nunca se inicie. Por lo que sería interesante integrar un sistema de tareas programables para evitar estos posibles escenarios y que se llegaran a realizar automáticamente.

Referencias

- [1] Documentación de asp net core. <https://dotnet.microsoft.com/es-es/learn/aspnet/what-is-aspnet-core>.
- [2] Documentación de mysql. <https://dev.mysql.com/doc/>.
- [3] Documentación de angular. <https://angular.dev/overview>.
- [4] Documentación de docker. <https://docs.docker.com/>.
- [5] Documentación de mysqlworkbench. <https://www.mysql.com/products/workbench/>.
- [6] Documentación de c#. https://learn.microsoft.com/es-es/dotnet/csharp/?WT.mc_id=dotnet-35129-website.
- [7] Portal de .net. <https://dotnet.microsoft.com/es-es/>.
- [8] Significado de api rest. <https://aws.amazon.com/es/what-is/restful-api/>.
- [9] Documentación de visual studio 2022. <https://visualstudio.microsoft.com/es/#vs-section>.
- [10] Documentación de cloudinary. <https://cloudinary.com/>.
- [11] Documentación de postman. <https://learning.postman.com/docs/introduction/overview/>.
- [12] Documentación de jwt. <https://jwt.io/>.
- [13] Documentación de node.js. <https://nodejs.org/docs/latest/api/>.
- [14] Documentación de typescript. <https://www.typescriptlang.org/docs/>.
- [15] Documentación de sass. <https://sass-lang.com/documentation/>.
- [16] Documentación de primeng. <https://www.primefaces.org/primeng-v14-lts/>.
- [17] Documentación de vs code. <https://code.visualstudio.com/docs>.

- [18] Página web de overleaf. <https://es.overleaf.com/>.
- [19] Documentación de git. <https://git-scm.com/doc>.
- [20] Página web de github. <https://github.com/>.
- [21] Documentación de swagger. <https://swagger.io/>.
- [22] Documentación de standalone. <https://v17.angular.io/guide/standalone-components>.
- [23] Documentación de dockercompose. <https://docs.docker.com/compose/compose-application-model/>.
- [24] ¿qué es el web scraping? <https://kinsta.com/es/base-de-conocimiento/que-es-web-scraping/>.
- [25] Documentación de entityframework. <https://learn.microsoft.com/es-es/ef/>.
- [26] Página web de amazon web services. <https://aws.amazon.com/es/>.
- [27] Página web de azure. <https://azure.microsoft.com/es-es>.
- [28] Página web de google cloud. <https://cloud.google.com/?hl=es>.
- [29] Api de brevo. https://www.brevo.com/es/features/email-api/?utm_source=adwords_brand&utm_medium=lastclick&utm_content=SendinBlue&utm_extension=sitelinks&utm_term=brevo&utm_matchtype=p&utm_campaign=20027374472&utm_network=g&utm_adid=656177208155&utm_adposition=&utm_device=c&utm_adgroupid=148424525676&utm_gad_source=1&utm_gclid=Cj0KCQjw_sq2BhCUARIsAIVqmQtJlvkSPWYXJ483XbCMoUc2gviwmynIte6bdHMKOfZT2W0sEQIYYTka_wcB.
- [30] Explicación de stableford. <https://www.golfibiza.com/modalidad-de-juego-stableford/>.

Apéndice A

Manual de Instalación

En este apéndice se describen los pasos necesarios a seguir para la instalación y configuración de las tecnologías requeridas. El proyecto se puede encontrar alojado en el siguiente repositorio de GitHub [20].: <https://github.com/AlexRamet4/TechGreen>

Como se ha mencionado anteriormente, una visión futura del proyecto sería poder desplegarlo en la nube, con lo que estos pasos tan solo estarían dirigidos para futuros desarrolladores que se incorporaran en el desarrollo de nuevas funcionalidades de la aplicación. Para los usuarios finales bastaría con abrir la app desde la web en cualquier dispositivo.

A.I. Prerrequisitos

Antes de comenzar, hay que tener instaladas las siguientes herramientas:

- **Git** [19] - Para clonar el repositorio. Se puede descargar desde <https://git-scm.com/>.
- **Node.js** [13] - Necesario para ejecutar Angular [3]. Accediendo a <https://nodejs.org/> se puede obtener.
- **Angular CLI** - Instalado globalmente con el comando:

```
npm install -g @angular/cli
```

- **.NET Core SDK** - Necesario para ejecutar el Backend. Ir a <https://dotnet.microsoft.com/download> para descargar.
- *(Opcional, si se quiere ejecutar la base de datos en local)* **MySQL** - Hay que acceder a <https://dev.mysql.com/downloads/installer/> para obtener el instalador.

- (Opcional, si se quiere ejecutar la base de datos en un conetendor) **Docker** [4] y **DockerDesktop** [?] - Se instalan desde <https://www.docker.com/get-started/> y <https://www.docker.com/products/docker-desktop/>.
- **Un editor de texto o IDE** - Recomendados: Visual Studio Code [17] y Visual Studio 2022 [9]

A.II. Clonación del Repositorio

Para obtener una copia local del proyecto, hay que seguir una serie de pasos:

1. Abrir una terminal en la ubicación donde se desea clonar el repositorio.
2. Ejecutar el siguiente comando:

```
git clone https://github.com/AlexRamet4/TechGreen
```

3. Navegar al directorio del proyecto:

```
cd proyecto
```

A.III. Configuración de la Base de Datos con MySQL

En caso de querer tener la base de datos corriendo localmente en el ordenador hay que seguir estos pasos:

1. Iniciar el servicio de MySQL.
2. Crear una base de datos para el proyecto:

```
CREATE DATABASE nombre_de_tu_base_de_datos;
```

3. Importar la base de datos desde el archivo SQL que se encuentra en el repositorio:
 - Si se hace uso de MySQL Workbench:

- a) Abrir MySQL Workbench y conectarse a tu servidor MySQL.
 - b) Ir a *File > Open SQL Script* y seleccionar el archivo *ScriptBD.sql* que se encuentra en el repositorio.
 - c) Ejecutar el script para crear las tablas y poblar la base de datos.
- Desde la terminal se podría llegar a realizar los mismo gracias a este comando:

```
mysql -u usuario -p
nombre_de_tu_base_de_datos < ruta/al/fichero/
base_de_datos.sql
```

A.IV. Ejecución de la Base de Datos con Docker Compose

En caso de no instalar MySQL localmente, se puede dar uso de Docker Compose [23] para levantar la base de datos.

1. En la raíz del proyecto, se encuentra un archivo *docker-compose.yml*.
2. Ejecutar el siguiente comando desde el directorio donde se encuentre el archivo para iniciar los servicios:

```
docker -compose up -d
```

3. Esto levantará un contenedor de MySQL configurado según las especificaciones del archivo *docker-compose.yml*.
4. Una vez el contenedor esté en ejecución, hay que importar la base de datos usando el siguiente comando:

```
docker exec -i techgreen mysql -u usuario -p
aplicacion_golf < ruta/al/fichero/ScriptBD.sql
```

A.V. Configuración del Frontend

1. Abrir una terminal en el directorio *Frontend* dentro del proyecto.
2. Instalar las dependencias necesarias con:

```
npm install
```

3. Para levantar el servidor de desarrollo de Angular, hay que ejecutar el siguiente comando:

```
ng serve
```

El proyecto estará disponible en *http://localhost:4200* desde cualquier navegador.

A.VI. Configuración del Backend

1. Abrir una terminal en el directorio *Backend* del proyecto.
2. Restaurar las dependencias y paquetes NuGet con:

```
dotnet restore
```

3. Configurar la cadena de conexión a la base de datos MySQL en el archivo *appsettings.json*.
4. Para ejecutar el servidor backend, se usa el siguiente comando:

```
dotnet run
```

El backend estará disponible en *http://localhost:5000* o el puerto que esté configurado.

A.VII. Ejecución Completa del Proyecto

1. Comprobar que el backend (ASP.NET Core [1]) y el frontend (Angular) estén ejecutándose simultáneamente.

2. Abrir un navegador y navegar a *http://localhost:4200* para acceder al Frontend.
3. Ambas deberían poder comunicarse sin ningún problema.

Siguiendo estos pasos, se debería ser capaz de iniciar el proyecto localmente, ya sea utilizando MySQL Workbench, Docker Compose o cualquier otro método preferido para gestionar la base de datos. En caso de existir algún tipo de problema, hay que confirmar que todas las dependencias estén correctamente instaladas.

Apéndice B

Manual de Usuario

Finalmente se va a desarrollar un manual de usuario donde se demostrarán las distintas funcionalidades que componen la aplicación, acompañadas con capturas de pantallas de las mismas.

1. Inicio de Sesión

Los usuarios podrán rellenar un formulario con su correo electrónico y la contraseña correspondiente. Si el sistema no encontrara el correo o la contraseña no es la asociada lanzaría un error. Dependiendo del rol del usuario se le redireccionaría a una vista diferente. (Figura 10)

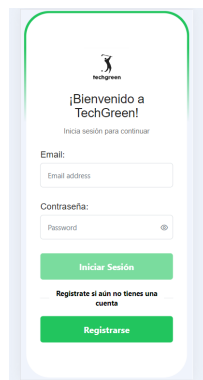


Figura 10: Inicio de Sesión

2. Administración

Si un usuario accede como moderador o administrador, dispondrá de varias tablas donde pueden ver la información de los jugadores y de los campos de golf. Podrán activar y desactivar cuentas con los botones que se encuentran en la vista. Si el que inicia sesión es un administrador, además podrá ver los moderadores que hay, interactuar con ellos y tener la posibilidad de dar de alta a nuevos usuarios con este rol. (Figuras 11, 12 y 13)

techgreen.golf@gmail.com Jugadores Campos Moderadores Cerrar Sesión

Lista de Campos

ID	Nombre	Telefono	Página Web	Localidad	Provincia	Estado	Ubicación
<input type="text"/>	<input type="text"/>				<input type="text"/>	<input type="checkbox"/>	
5	La Reserva	956 785 252		Sotogrande	Cádiz		
6	Guadalhorce Golf	952 179 378		Málaga	Málaga		

Página 1 de 1 << < 1 > >>

Total de campos 2.

Figura 11: Campos de Golf

techgreen.golf@gmail.com Jugadores Campos Moderadores Cerrar Sesión

Lista de Jugadores

ID	Email	Licencia	Estado	Provincia
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="text"/>
6	alexrametespino@uma.es	AM1221234		Cádiz
7	jugador1@gmail.com	AM1221345		Cádiz
8	mesmdrd@gmail.com	AM0900012		Madrid
9	alejandroramet4@gmail.com	AM12233431		Cádiz

Página 1 de 1 << < 1 > >>

Total de jugadores 4 jugadores. [Gráfica](#)

Figura 12: Jugadores

techgreen.golf@gmail.com Jugadores Campos Moderadores Cerrar Sesión

Lista de Moderadores

ID	Email	Fecha Registro	Estado	Agregar Moderador
<input type="text"/>	<input type="text"/>		<input type="checkbox"/>	
36	alexramet4@gmail.com	11:37 31/08/24		Desactivar

Página 1 de 1 << < 1 > >>

Total de moderadores 1.

Figura 13: Moderadores

3. Registro Campo de Golf

Para un usuario poder crear una cuenta como un campo de golf, tendrá que rellenar todos los datos que se solicitan en el formulario del registro y esperar a que un moderador verifique los datos y active su cuenta. (Figuras 14 y 15)

Nuevo Campo

Rellene el formulario con sus datos

Email:

Contraseña:

Repetir Contraseña:

Nombre del Campo:

Figura 14: Registro de Campo

Provincia:

Localidad del Campo:


Registrar

Figura 15: Registro de Campo

4. Perfil de Campo de Golf

Desde esta vista cualquier usuario relacionado con un campo de golf podrá acceder a su perfil y editar cualquier dato que desee. Adicionalmente, si es administrador de ese campo podrá dar de alta nuevos moderadores y ascenderlos a su mismo rol. (Figura 16)

La Cañada




Ubicación

Provincia: Cádiz
Localidad: Guadaro

Contacto

Teléfono: 956 794 100
Página web: <https://lacanadagolf.com/>




[Editar Perfil](#)

Moderadores

Email	Fecha Registro	Estado	Rol
lacanada.golf@techgreen.com	19:36 03/09/24	✔	Admin Campo
moderador.lacanada@techgreen.com	19:41 03/09/24	✔	Moderador Campo

Registro de Partidas



Total de partidas: 0

Figura 16: Perfil de Campo

5. Gestión de Noticias

Para este apartado se dispone de una tabla ordenada por las fechas de las noticias que se han publicado por el campo. Cada usuario podrá crear nuevas noticias, editarlas, eliminarlas y ver los likes que tiene cada publicación. (Figuras 17 y 18)

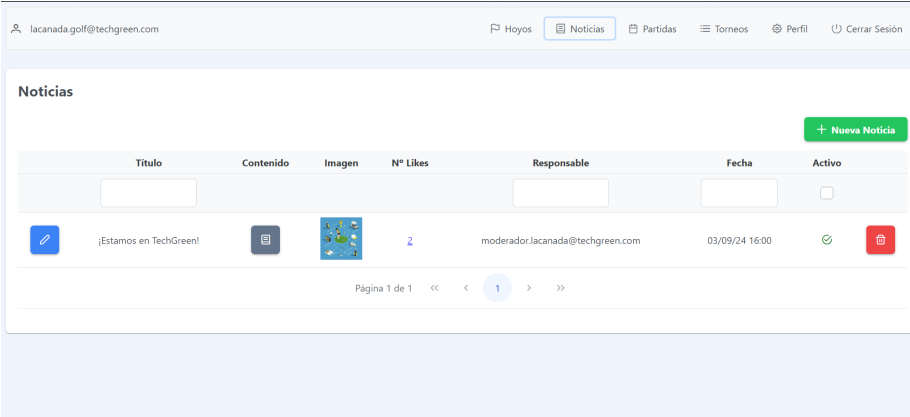


Figura 17: Lista de Noticias



Figura 18: Formulario de Noticias

6. Gestión de Torneos

Si un administrador o moderador del campo desea crear un torneo podrá hacerlo desde esta vista. Además de la creación, dispondrá de varias funcionalidades extras para el torneo que deseen. Desde editarlo si aún no se ha iniciado, publicar imágenes, o acceder a sus inscripciones. Desde esta última vista, se podrán realizar una serie de acciones gracias a los botones que nos encontramos: descargar un excel con los datos de todos los jugadores, iniciar el torneo, finalizarlo, cancelarlo o ver las partidas de los jugadores tanto en directo o una vez finalizadas. (Figuras 19, y 20)

Torneos								
Nombre	Descripción	Fecha	Estado	Imágenes	Máx. Jugadores	Min. Jugadores	Inscripciones	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
	Torneo StableFord		27-08-2024 15:00	Finalizado		12	0	
	Primer Torneo		31-07-2024 12:21	Finalizado		32	0	

Página 1 de 1 << < 1 > >> 10

Figura 19: Lista de Torneos

Inscripciones

Torneo **StableFord**

Total de Plazas: 4 Estado: Finalizado Plazas libres: 8





Jugador	Email	Licencia	Handicap	Color Salida	Fecha Inscripción	Resultado	Puntuación	Posición
 Pedro Luis Ramet Sánchez	alejandro.ramet4@gmail.com	AM12233431	28	●	27-08-2024 09:18	🏆	64	1º
 Alex Ramet Espinosa	alexrametespinosa@uma.es	AM1221234	24	●	23-08-2024 04:18	🏆	60	2º
 Antonio Rodriguez Pérez	jugador1@gmail.com	AM1221345	20	●	27-08-2024 11:12	🏆	56	3º
 Mario Ramet Mesequer	mesmdrd@gmail.com	AM0900012	15	●	27-08-2024 11:12	🏆	51	4º

Figura 20: Lista de Inscripciones

7. Gestión de Campos

Gracias a esta vista se podrán gestionar todas las partidas que se disputan en el día a día en un campo de golf. Si se selecciona una fecha pueden llegar a pasar dos cosas. Si está libre se podrá crear una partida nueva en esa franja horaria y agregar a jugadores. En caso contrario, se accederá a la partida seleccionada incluyendo varias opciones como la de eliminarla o la de expulsar a jugadores de ella. (Figuras 21 y 22)



Figura 21: Calendario Partidas



Figura 22: Datos de Partida

8. Gestión de Hoyos

Inicialmente un campo de golf se registra en la aplicación sin hoyos y para poder crearlos tendrá que hacerlo a través de un formulario o subiendo un excel cuya plantilla se facilita desde la opción de descarga. En ambos casos el usuario tendrá que aportar la información de cada hoyo. (Figuras 23 y 24)

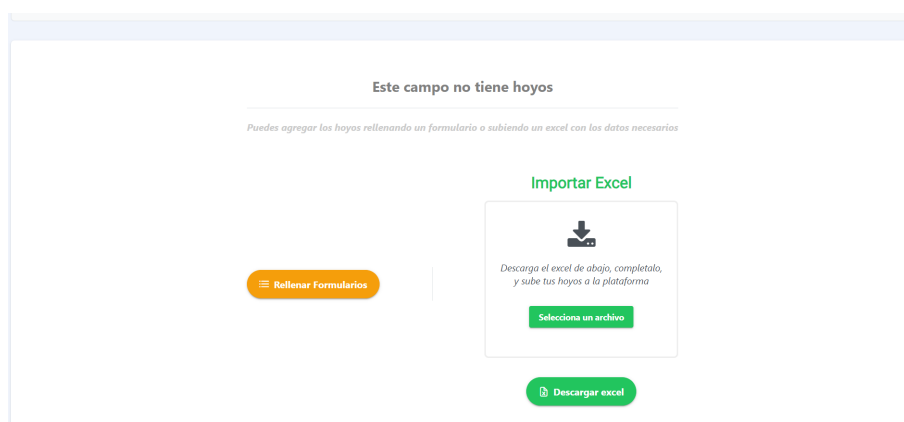


Figura 23: Creación de Hoyos

Creación de Hoyos

Número	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Par																		
Handicap																		
Distancia																		
Handicap																		
Distancia																		
Handicap																		
Distancia																		

[< Volver](#)

En caso de que el handicap sea el mismo para todos los casos, seleccione este modo

Figura 24: Formulario de Hoyos

Una vez creados, no se podrá volver a la vista anterior. En su lugar, se podrán visualizar todos los hoyos agregados y editar la información correspondiente a cada uno. (Figura 25)

Hoyo 5 Par 3

Este hoyo no tiene descripción

Salidas

Color	Handicap	Distancia
	14	160 m / 175 yd
	14	147 m / 161 yd
	14	115 m / 126 yd
	14	130 m / 142 yd



Figura 25: Vista de Hoyo

9. Registro Jugador

Para un usuario poder registrarse como un jugador, tendrá que rellenar todos los datos que se solicitan. En este caso no tiene que esperar a que le activen la cuenta, podrá acceder inmediatamente una vez complete el formulario de registro. (Figuras 26 y 27)

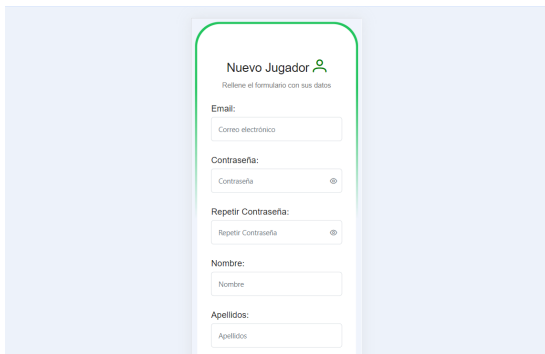


Figura 26: Registro de Jugador

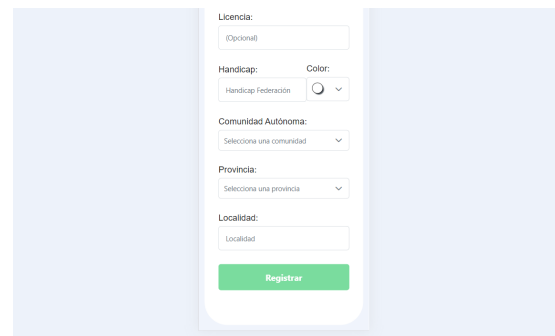


Figura 27: Registro de Jugador

10. Perfil Jugador

Al entrar en la aplicación lo primero que ve el usuario es su perfil junto a la opción de editarlo y de ver las estadísticas dadas por las partidas jugadas. (Figuras 28, 29 y 30)

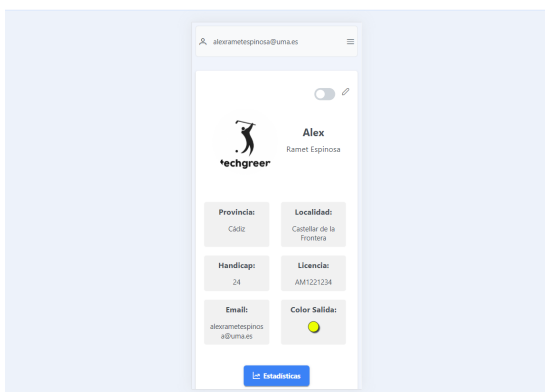


Figura 28: Perfil de Jugador

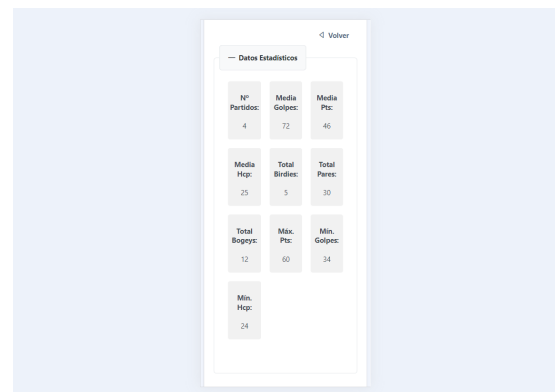


Figura 29: Datos Estadísticos



Figura 30: Gráficas Estadísticas

11. Amistades

Desde esta sección, los usuarios podrán buscar a otras personas y enviar solicitudes de amistad, aceptar o rechazar las solicitudes que les envíen y ver todos los jugadores con los que mantiene una relación de amistad, con la posibilidad de eliminarla. (Figuras 31, 32 y 33)

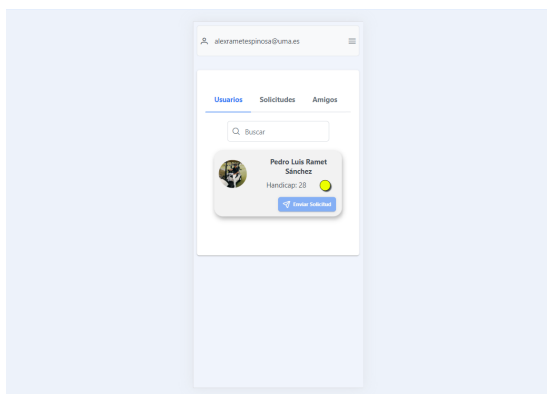


Figura 31: Jugadores sin Relación

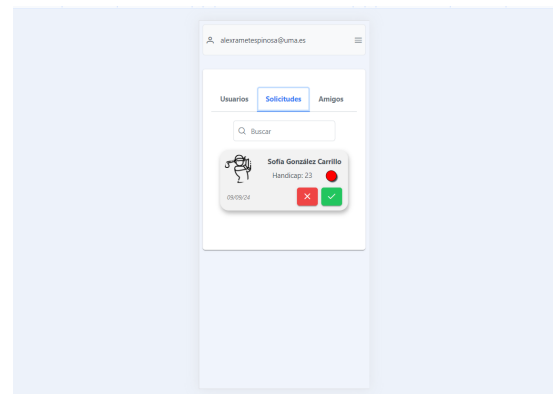


Figura 32: Solicitudes de Amistad

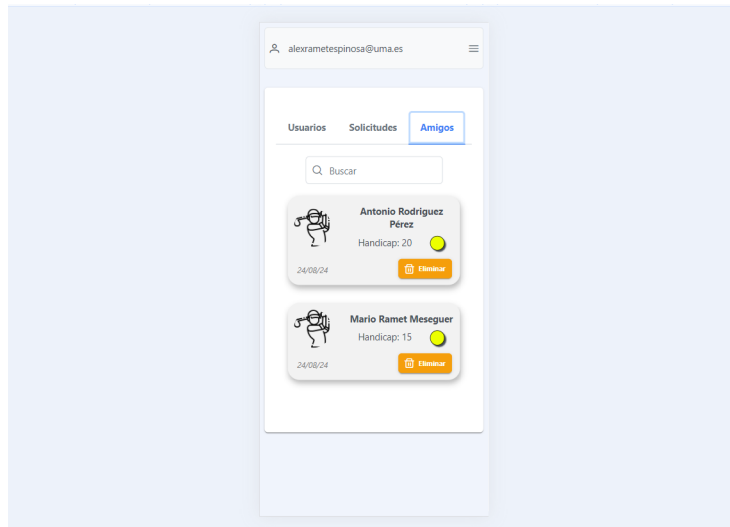


Figura 33: Amigos

12. Campos

En esta vista se muestran todos los campos activos dentro de la aplicación. Los jugadores podrán buscarlos por el nombre del club o la provincia en la que se encuentren gracias al filtro de búsqueda. Además, tendrá a su disposición la información referente a la ubicación exacta, los hoyos que puede jugar y las noticias referente al campo que deseen. (Figuras 34 y 35)

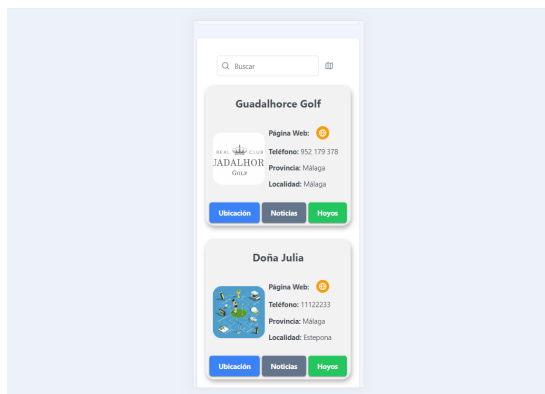


Figura 34: Vista de Campos

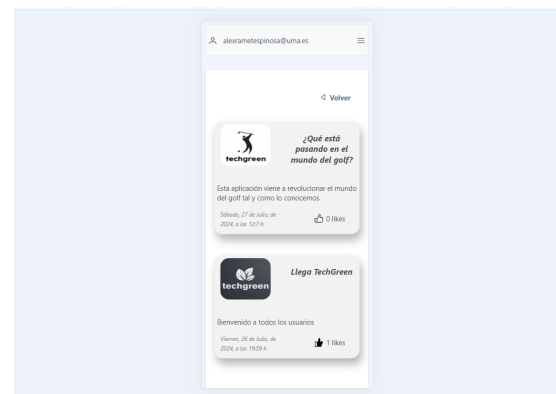


Figura 35: Vista de Noticias

13. Torneos

Los jugadores podrán ver los torneos que se disputan en la aplicación desde esta sección,

que se divide en tres bloques. En cada uno de ellos, podrá realizar una búsqueda por un campo en específico y ver los jugadores que hay inscritos en cada torneo junto a una funcionalidad única de cada bloque. En caso de no existir ningún torneo se mostraría un mensaje indicándolo.

- Torneos Pendientes: Torneos en los que el usuario está inscrito y aún no ha iniciado. Podrá eliminar su inscripción antes de que inicie el torneo o redirigirse a la tarjeta de juego una vez se inicie. (Figura 36)
- Torneos Futuros: Torneos en los que el usuario no está inscrito. Podrá inscribirse si lo desea. (Figura 37)
- Torneos Pasados: Torneos en los que ha participado el usuario. Podrá ver las imágenes que se publiquen. (Figura 38)

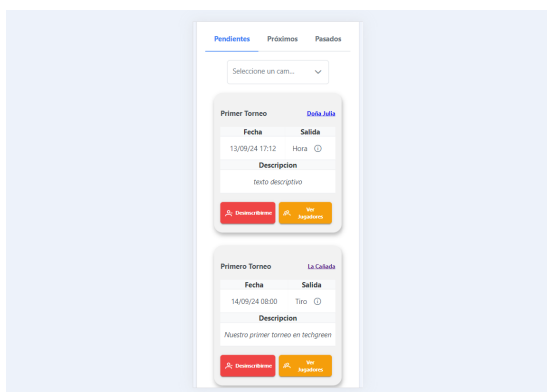


Figura 36: Torneos Pendientes

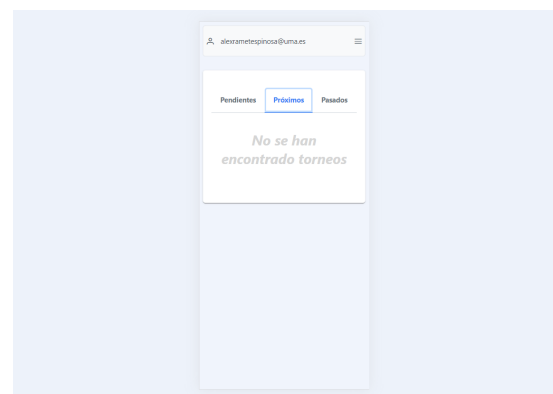


Figura 37: Torneos Futuros

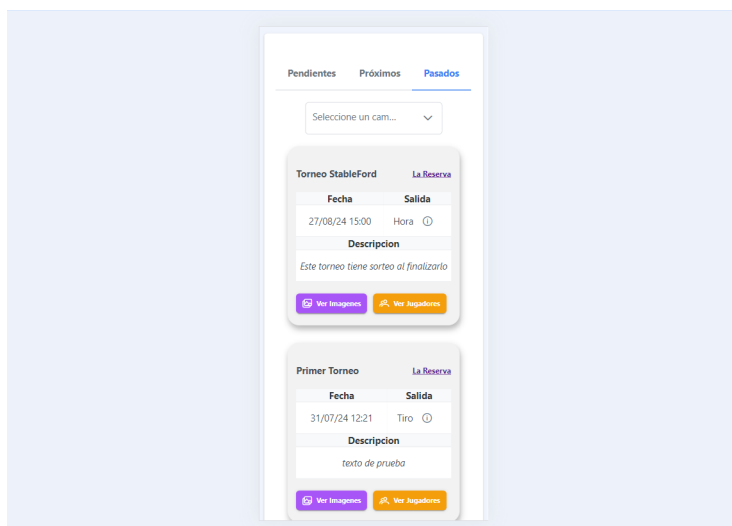


Figura 38: Torneos Pasados

14. Partidas

Este último apartado es muy similar al anterior tanto visual como funcionalmente. Se podrán ver las partidas tanto pendientes como pasadas, al igual que aquellas que se van a disputar y al menos hay un amigo apuntado. Los jugadores podrán filtrar las partidas a partir de un campo de golf o de un amigo en concreto. (Figuras 39, 40 y 41)

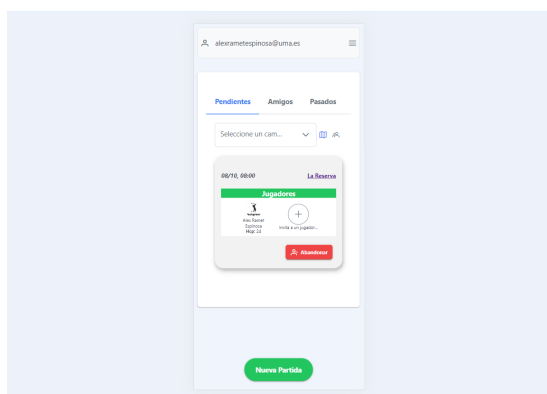


Figura 39: Partidas Pendientes

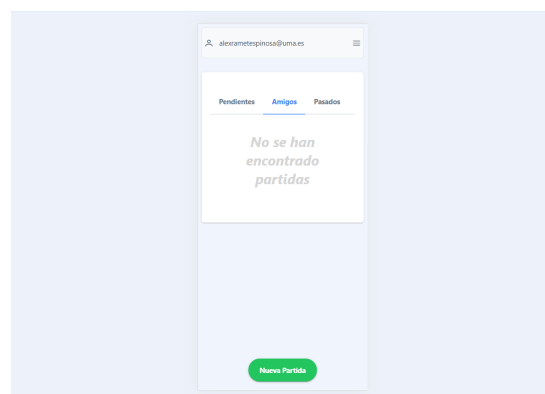


Figura 40: Partidas de Amigos

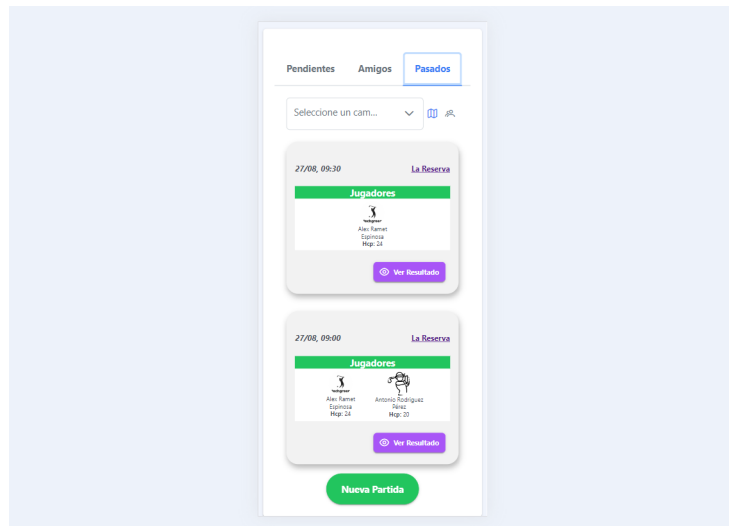


Figura 41: Partidas Pasadas

Adicionalmente, desde la vista de partidas pendientes se pueden encontrar tres funcionalidades extras.

- Invitar jugadores: Si se pulsa el botón de invitar se podrá invitar a un amigo siempre que no haya más de cuatro jugadores.
- Eliminar jugadores: Si el creador de la partida ve a un jugador no deseado en ella, tendrá la opción de eliminarlo pulsando sobre la imagen de este.
- Modificar ajustes: Si el usuario pulsa sobre sí mismo, podrá cambiar su handicap y su color de salida para esa partida exclusivamente.

Y por último, los usuarios podrán crear una partida o unirse a una ya creada desde el calendario de un campo en una de las franjas horarias que tengan habilitadas. (Figura 42)

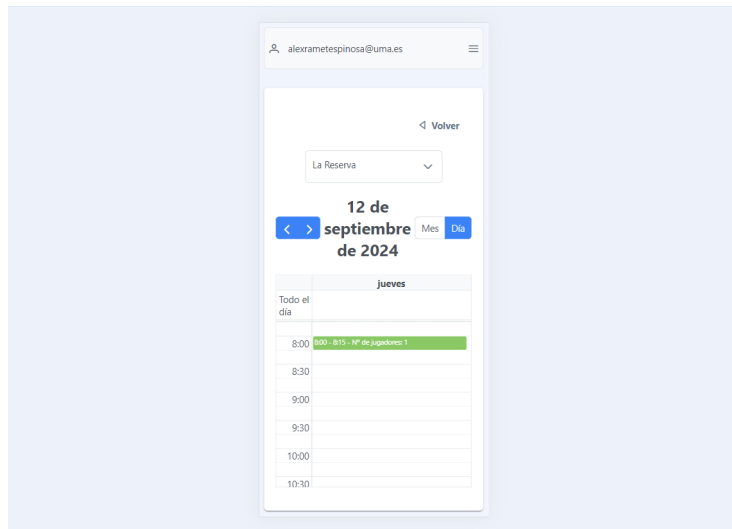


Figura 42: Calendario Partidas



UNIVERSIDAD
DE MÁLAGA

| uma.es

E.T.S de Ingeniería Informática
Bulevar Louis Pasteur, 35
Campus de Teatinos
29071 Málaga

E.T.S. DE INGENIERÍA INFORMÁTICA