

Polideportivo martínez

Javier Martínez Manjavacas

Miguel Martínez Manjavacas



14 de junio de 2022

2º Desarrollo de Aplicaciones Web

****

****

**Índice**

1. [Contenido del documento 1](#_bookmark0)
   1. [Introducción 1](#_bookmark1)
   2. [Justificación del proyecto y objetivos 1](#_bookmark2)
   3. [Planificación 1](#_bookmark3)
   4. [Parte experimental 1](#_bookmark4)
      1. [Análisis. ¿Qué hará la aplicación? 1](#_bookmark5)
      2. [Diseño. ¿Cómo se hará la aplicación? 2](#_bookmark6)
      3. [Implementación y pruebas 2](#_bookmark7)
      4. [Implantación y documentación 2](#_bookmark8)
      5. [Resultados y discusión 2](#_bookmark9)
   5. [Conclusiones 2](#_bookmark10)
   6. [Bibliografía y referencias 2](#_bookmark11)
   7. [Anexos 2](#_bookmark12)
2. [Formato de entrega del proyecto 3](#_bookmark13)

# Contenido del documento:

## Introducción

Nuestro proyecto consiste en desarrollar una aplicación web en lenguaje Java, concretamente de tipo Spring para un polideportivo, donde los usuarios podrán darse de alta mediante una matriculación y apuntarse a los diversos deportes ofrecidos por el establecimiento. Dicho portal web tendrá dos vistas: una para los clientes y otra para la administración de la página.

La arquitectura de dicho proyecto se logrará mediante el uso de la conocida tecnología Docker. En este caso, implementándolo como gestor de base de datos MySQL del que se proveerá la aplicación.

El proyecto reunirá la mayoría de módulos/asignaturas cursadas a lo largo de los dos años del ciclo formativo. Concretamente son los siguientes:

1er curso

* **Bases de Datos**. Motor MySQL.
* **Entornos de desarrollo**. Uso de sistema de control de versiones (GitHub).
* **Lenguaje de marcas y sistemas de gestión de información**. Uso de la tecnología HTML y XML principalmente, pues la primera compondrá parte del Front-end y la segunda establecerá la configuración de la aplicación (Librerías a utilizar de Maven).

2º curso

* **Desarrollo web entorno servidor**. Gestión de la capa de datos y negocio de la aplicación mediante la técnica MVC (Modelo-Vista-Controlador). Parte Back-end de la aplicación.
* **Despliegue de aplicaciones web**. Implementación de la arquitectura Spring y alojo de motor de base de datos MySQL en Docker.
* **Diseño de aplicaciones web**. Está relacionada con la asignatura de lenguaje de marcas mencionada previamente. Se encargará de la parte Front-end de la aplicación. Hará uso de algunas tecnologías como HTML, CSS, Bootstrap y algún que otro framework, como NiceAdmin.
* **Desarrollo web entorno cliente.** Implementación de tecnología JavaScript para añadir ciertas funcionalidades a la aplicación web.

## Justificación del proyecto y objetivos.

La decisión de realizar de tal forma el proyecto es porque reúne contenidos de la mayoría de asignaturas cursadas a lo largo del ciclo formativo. Además, pensamos que así construimos una arquitectura de aplicación bastante completa ya que intervienen diversos servicios y tecnologías. Por otra parte, ya tenemos algo de experiencia trabajando con Spring debido a la realización de proyectos similares (pero no tan complejos) con algunas de estas tecnologías.

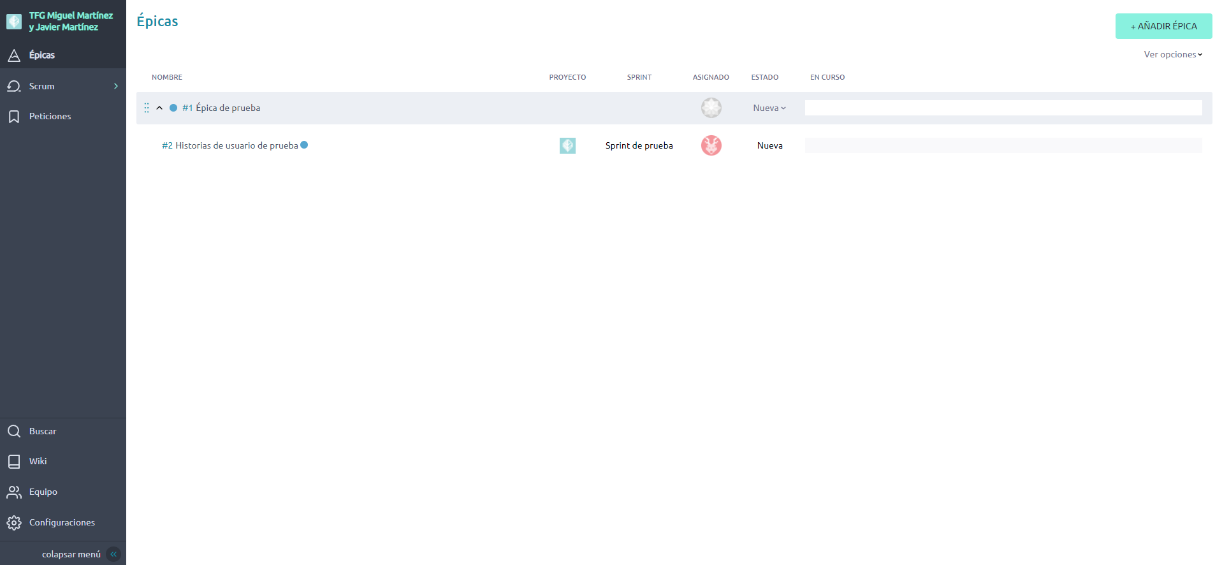
En cuanto a las opciones de cara al mercado, desde los últimos años, está aumentando la realización de ejercicio, por ende, la apertura de negocios que ofrecen diversos deportes, siendo algunos de ellos cada vez más populares independientemente de la edad de cada persona. Sin ir más lejos, aquí en nuestro municipio (Rivas Vaciamadrid, Madrid) hay más de 40 establecimientos o gimnasios de los cuales casi la mitad no tienen ni 4 años de antigüedad. Si miramos la estadística se trata de un porcentaje que ha ido aumentando de forma muy notable, y como desarrolladores de aplicaciones, creemos que podría ser una buena oportunidad de negocio.

El objetivo primordial que se pretende alcanzar con la realización de este proyecto se basa en la consolidación de los conocimientos, prácticas y habilidades adquiridas a lo largo del ciclo, sobre todo, en la parte de programación de código. No obstante, también existe la finalidad de adquirir nuevos saberes relacionados con la arquitectura de una aplicación web, así como los distintos servicios que se requieren para elaborar de forma correcta esta última.

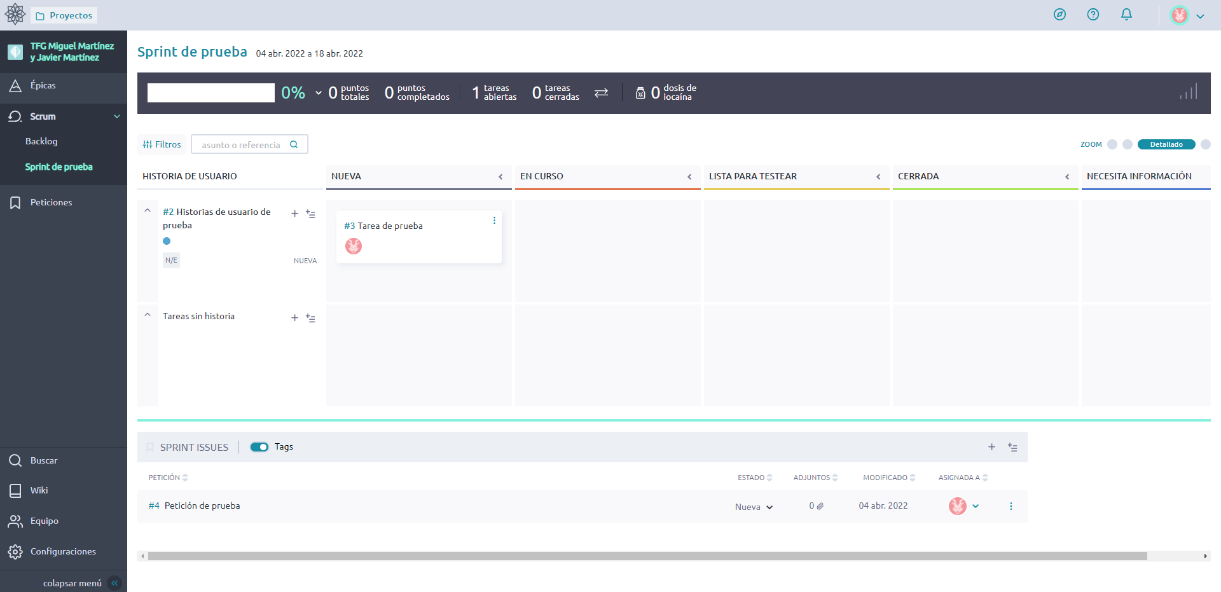
Si tuviésemos que definir el tipo o perfil de público al que va dirigido nuestro software, sería para uno profesional. Es decir, sólo podría ser de utilidad para aquellos negocios que vayan destinados a ofertar la práctica de uno o más deportes a sus clientes.

Este proyecto ha sido elegido por varias razones: en primer lugar, consideramos que dispone de las características necesarias para poder consolidar lo aprendido en el ciclo. Por otro lado, creemos necesario trabajar sobre una idea que nos motive y que sea llamativa para nosotros, por eso, la idea de un polideportivo es idónea ya que es un ámbito de nuestro agrado y con el que estamos familiarizados.

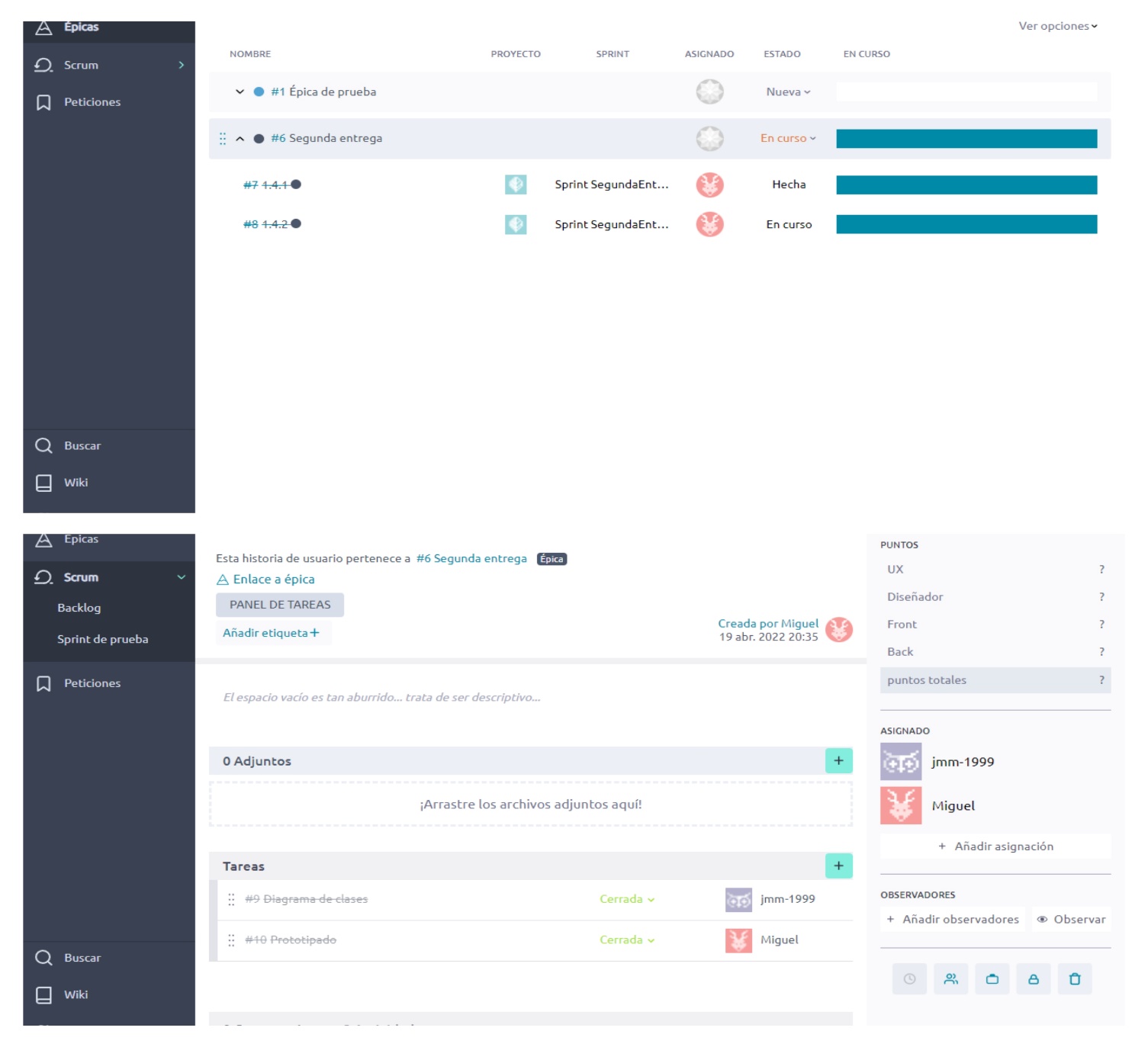
## Planificación

Primeras pruebas en taiga (creación de épicas, historias de usuario, sprints, tareas y peticiones). Agregación de miembros:

(Imagen 1: Primera Épica de prueba)

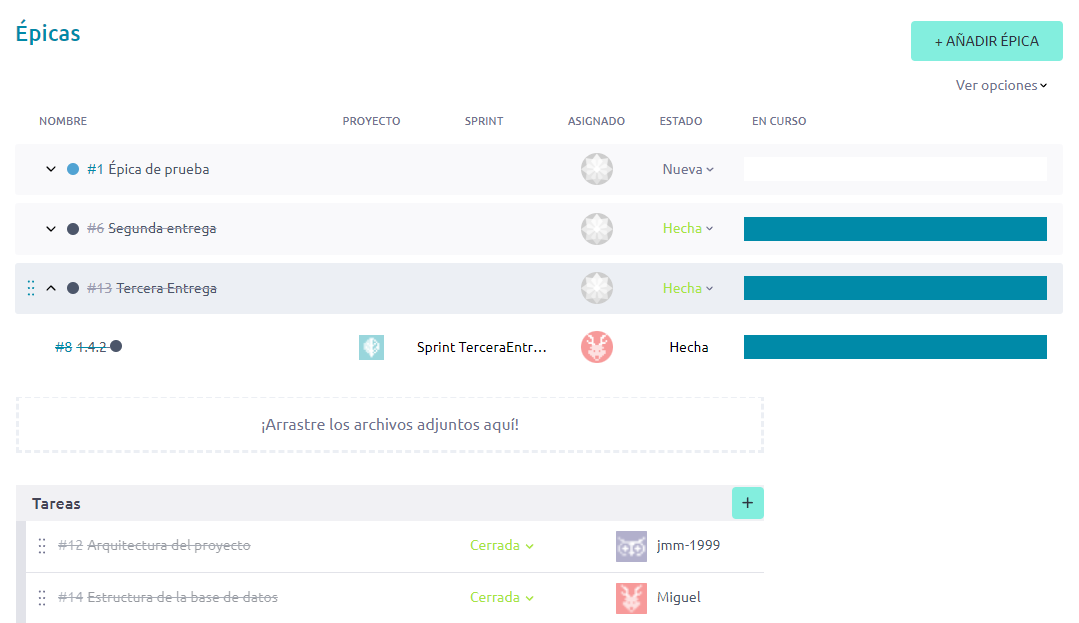


(Imagen 2: Primer Sprint de prueba)

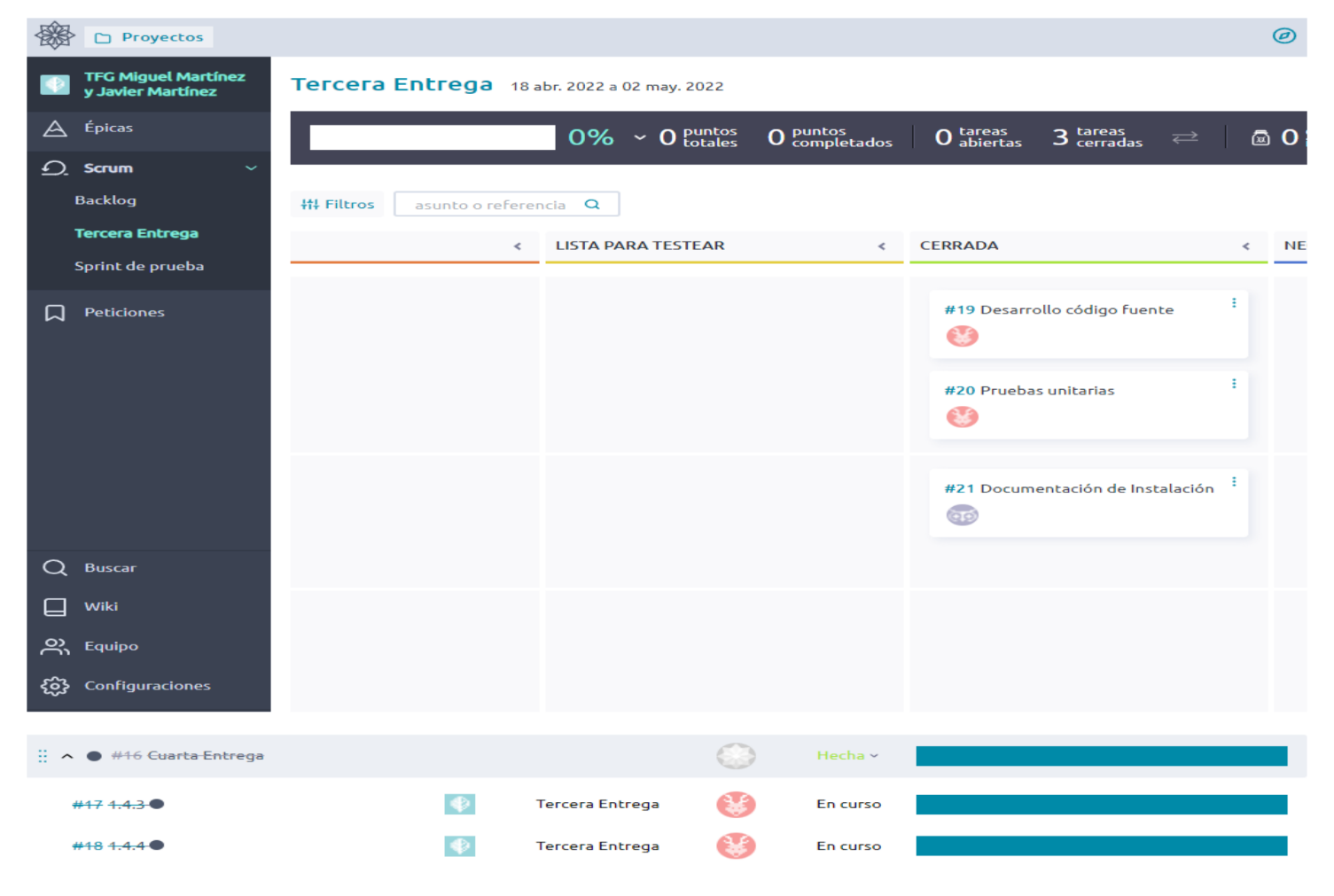
Segunda entrega:

(Imagen 3: Épica equivalente a la segunda entrega)

Tercera entrega:

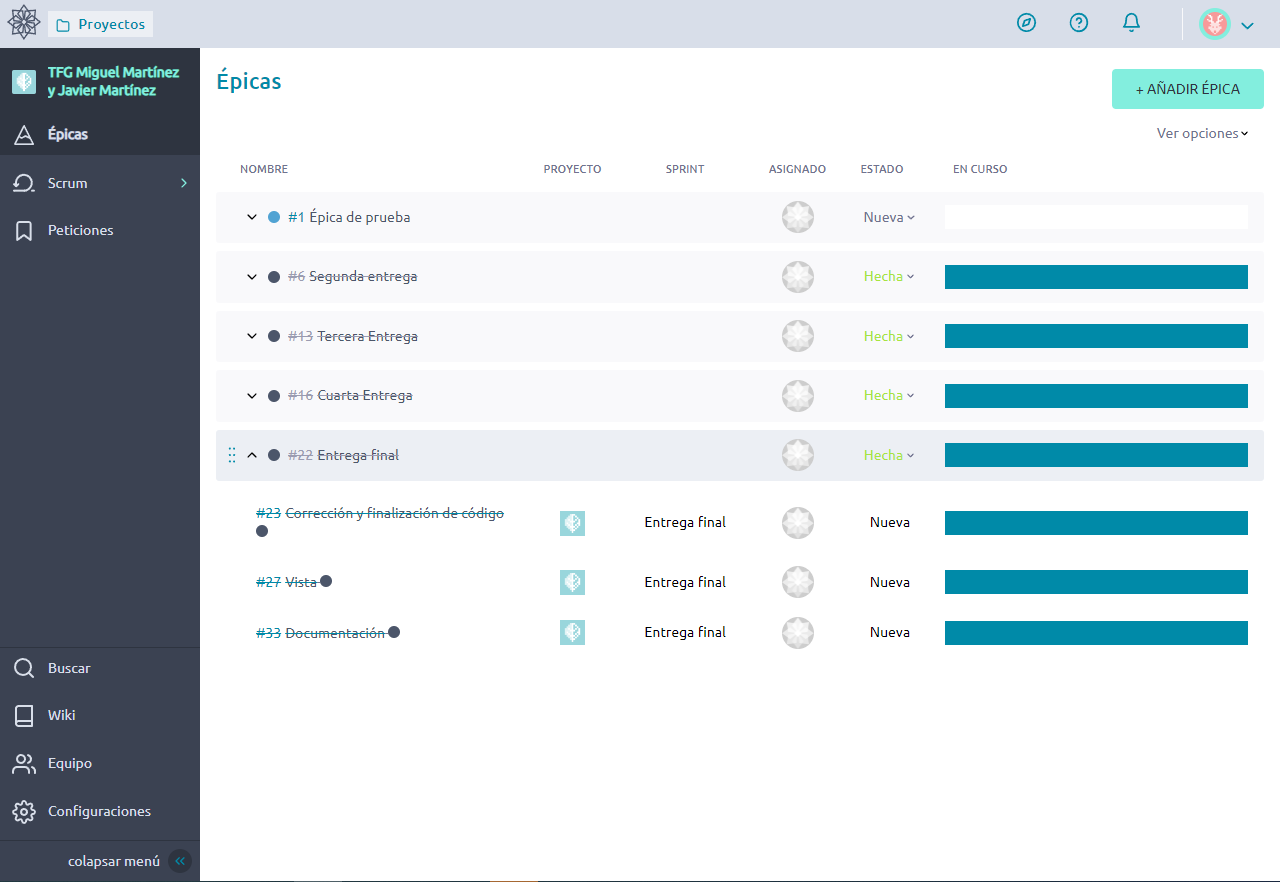


(Imagen 4: Épica equivalente a la tercera entrega)

Cuarta entrega:

(Imagen 5: Épica equivalente a la cuarta entrega)

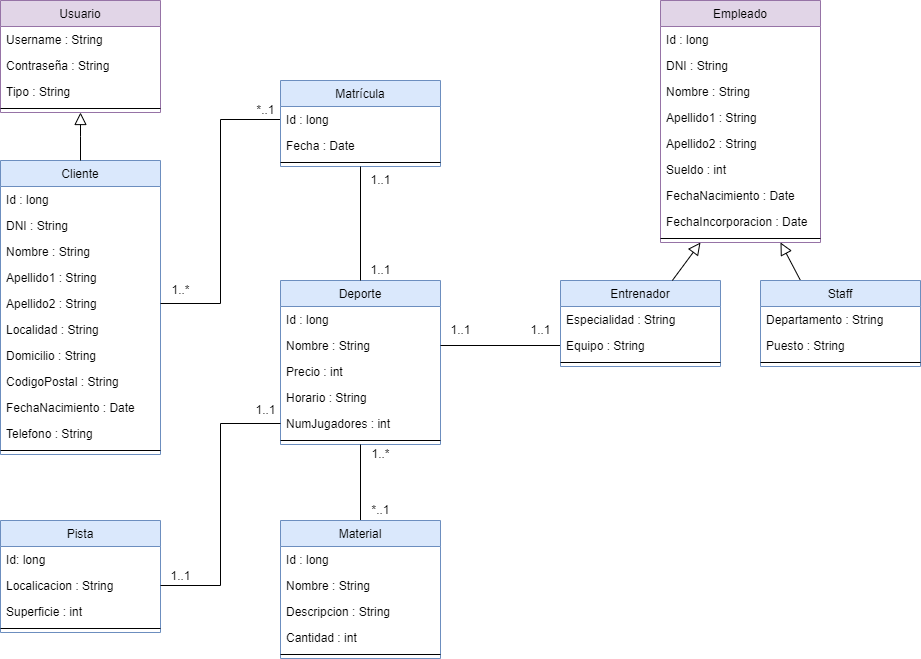
Última entrega:



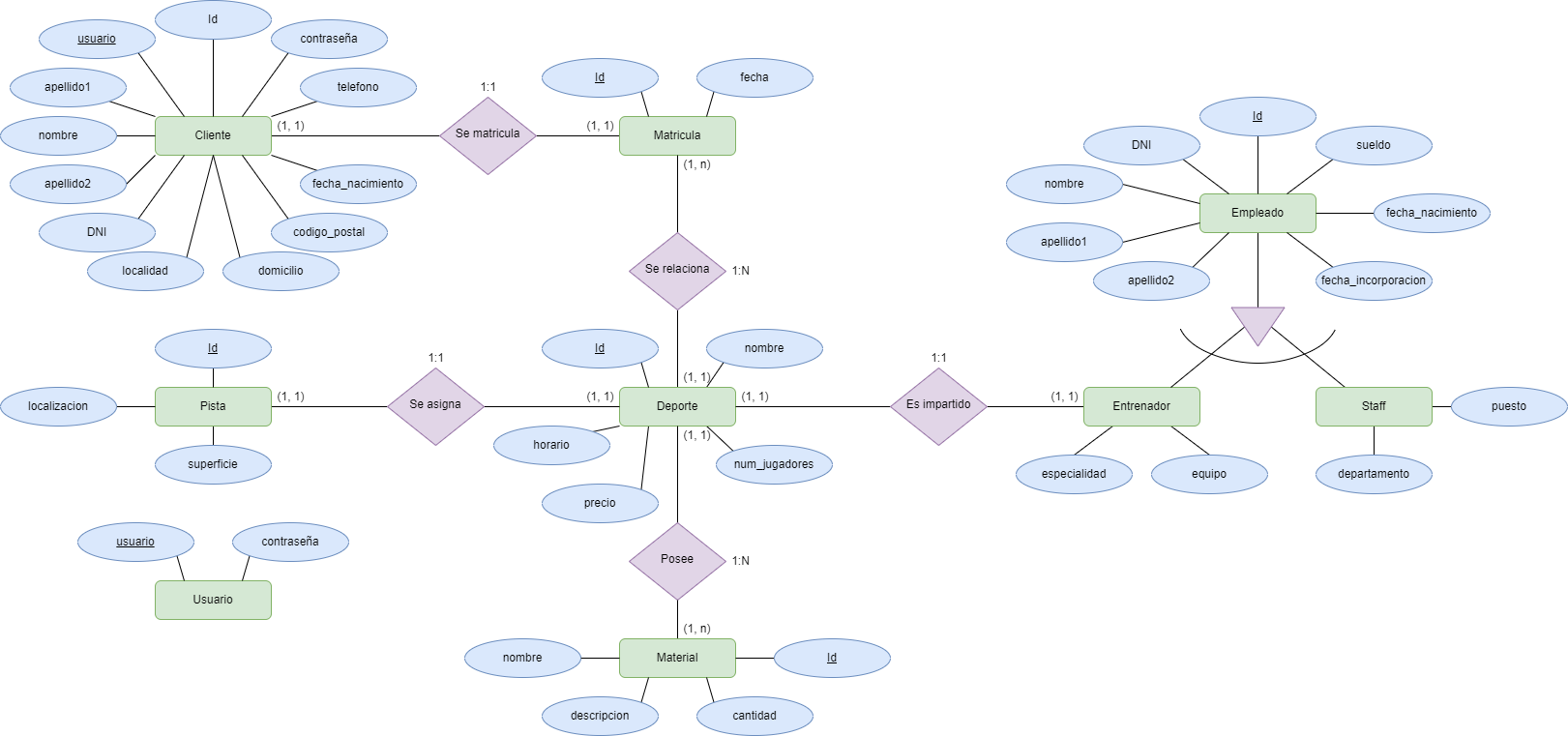
(Imagen 6: Épica equivalente a la entrega final)

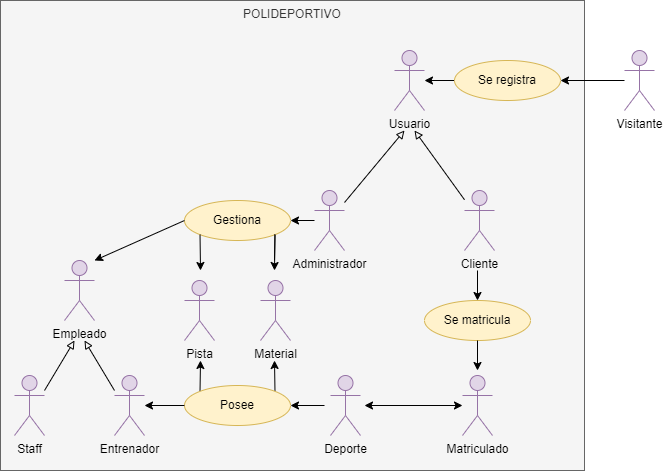
## Parte experimental

### Análisis. ¿Qué hará la aplicación?



(Imagen 7: Diagrama de Clases)

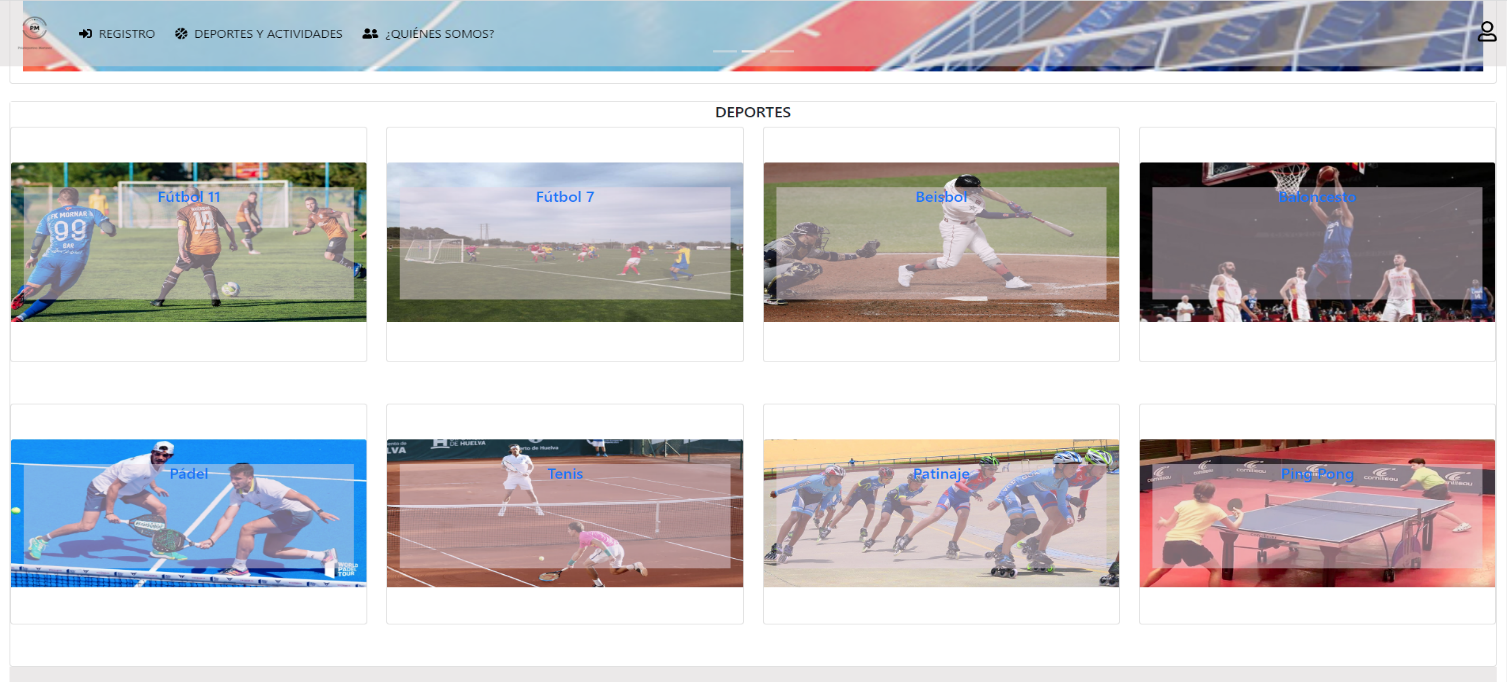
(Imagen 8: Diagrama Entidad Relación)

(Imagen 9: Diagrama de Casos de Uso)

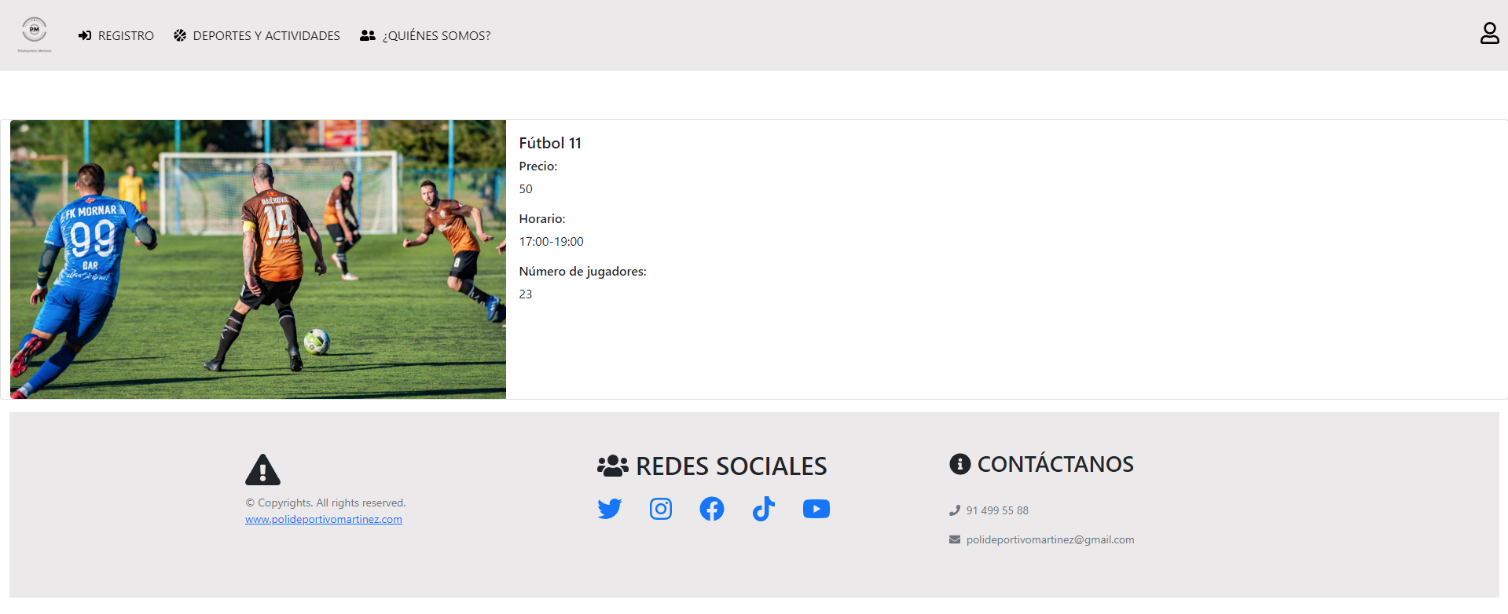
En cuanto a las pantallas, la web cuenta con un total de nueve vistas, pudiéndolas clasificar en vistas del usuario, vistas del cliente y vistas del administrador.

Vistas usuario:

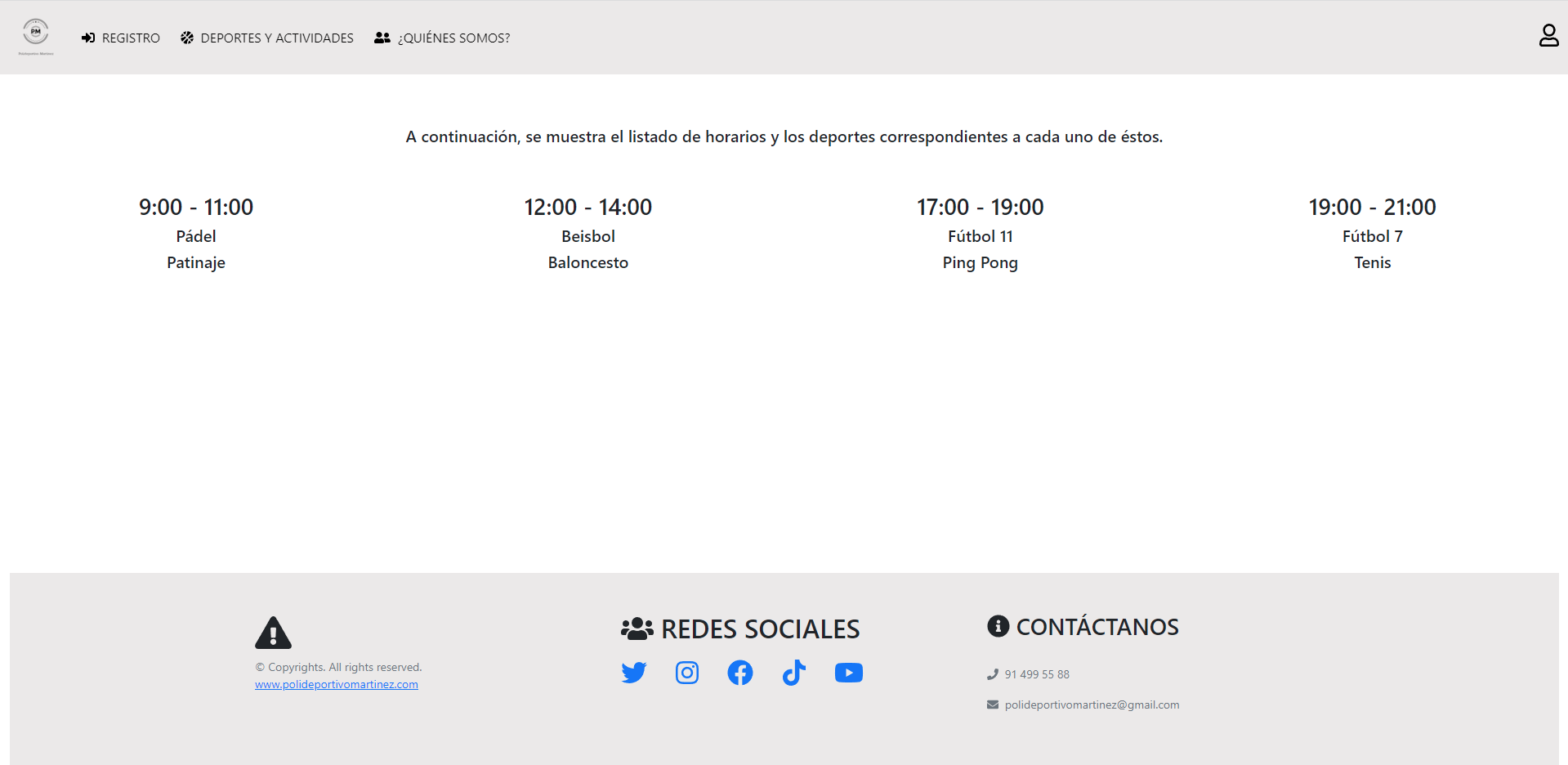
(Imagen 10: Carroussel página principal)



(Imagen 11: Cards página principal)



(Imagen 12: Detalles de los deportes)

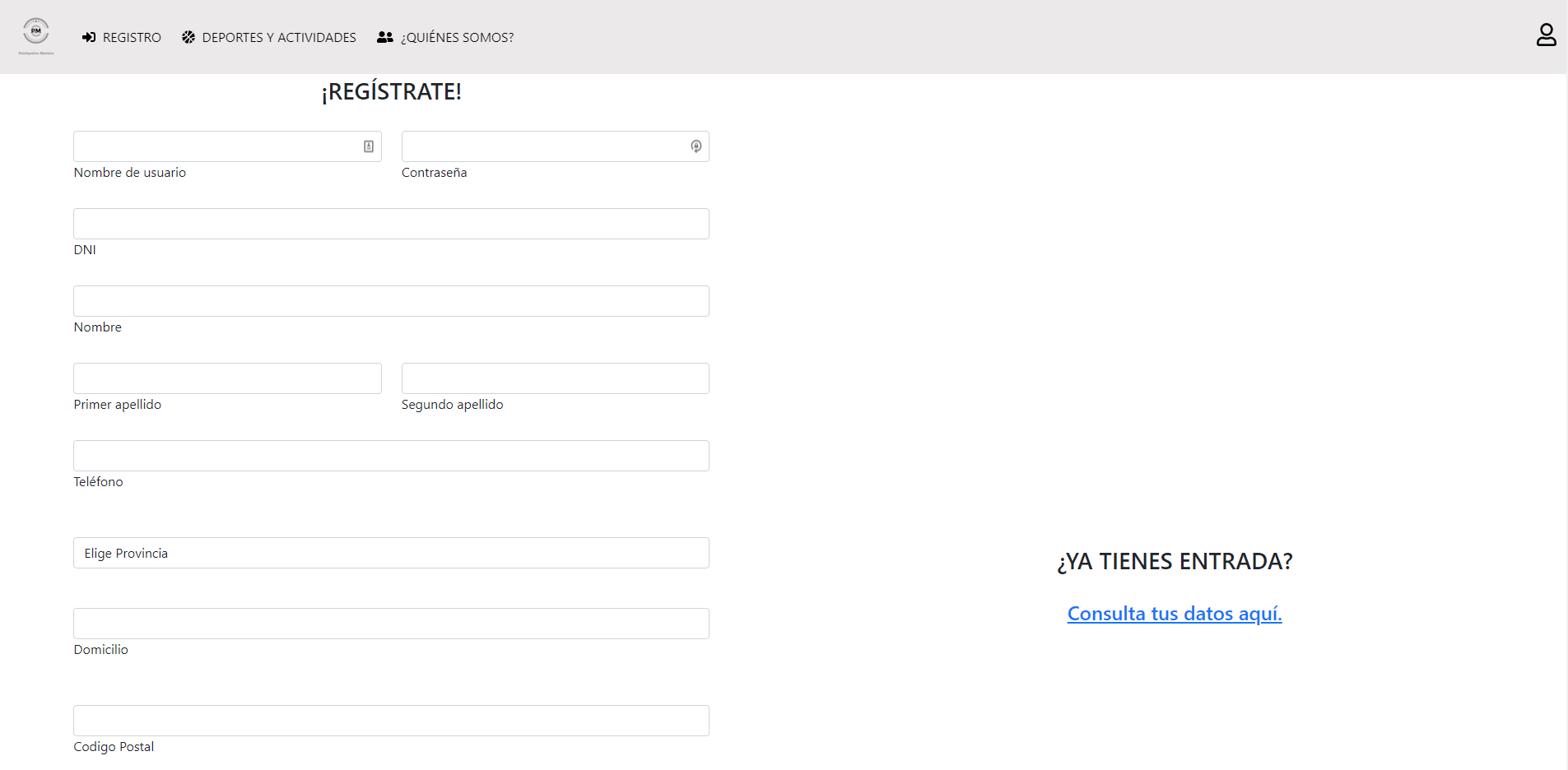


(Imagen 13: Lista de deportes y horarios)



(Imagen 14: Quienes somos)

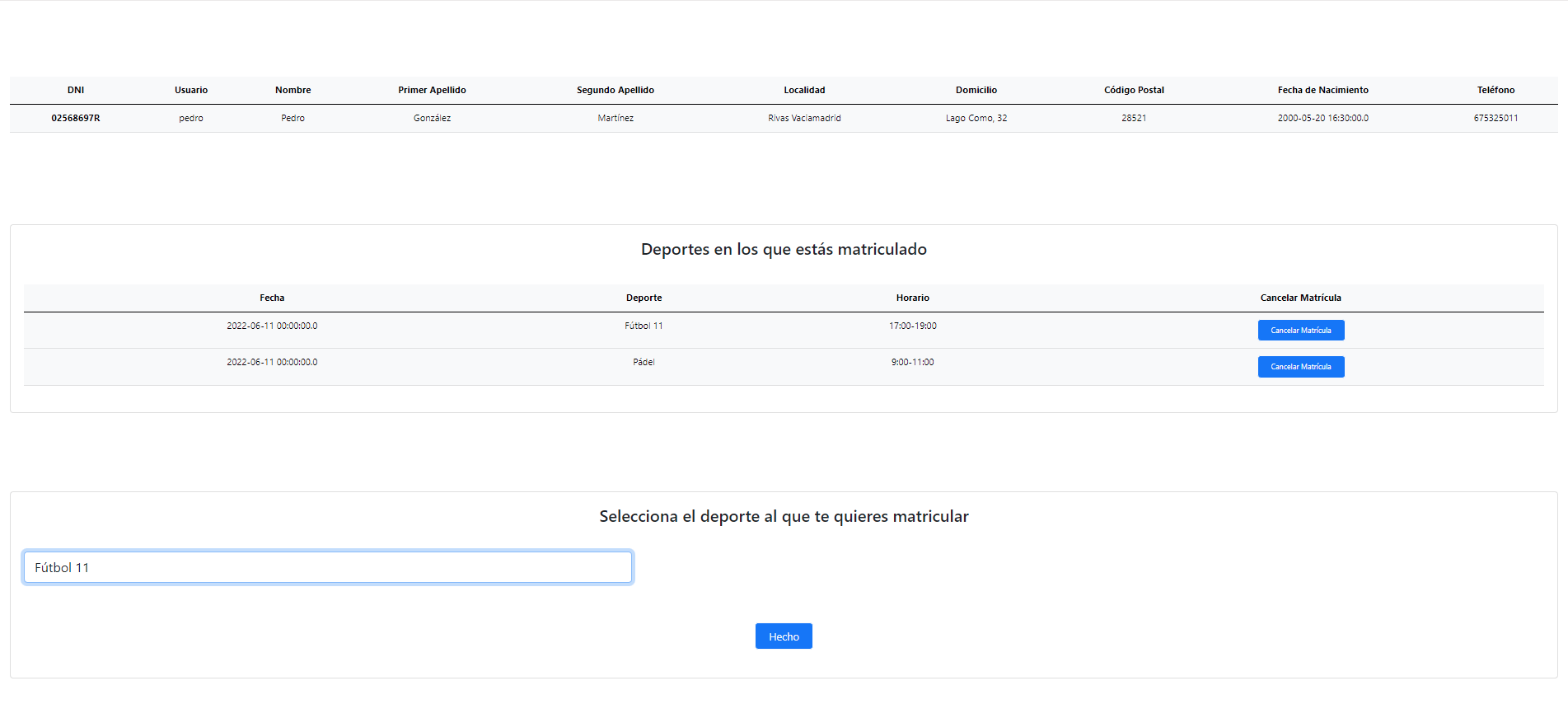
Vistas cliente:



(Imagen 15: Registro)

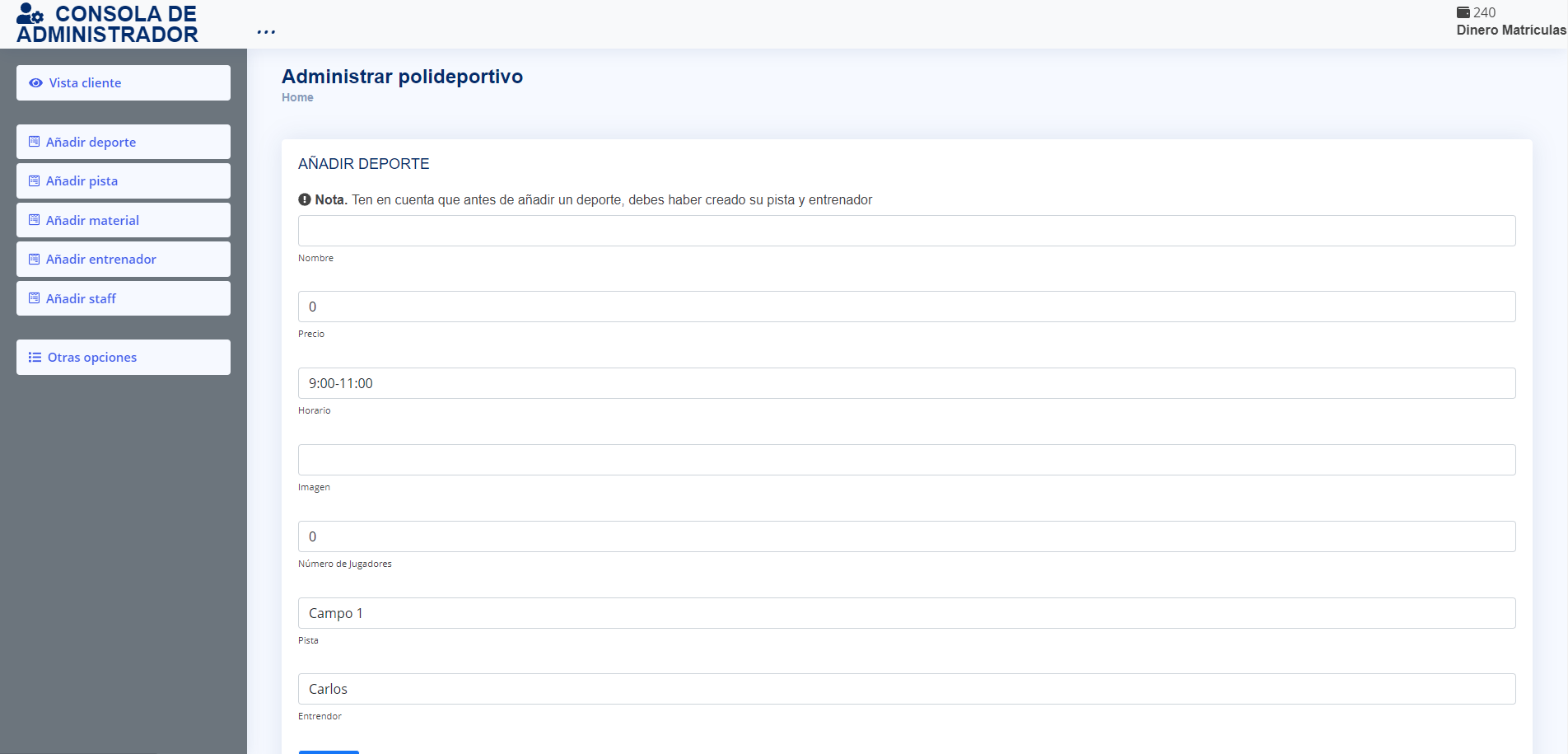


(Imagen 16: Iniciar sesión)

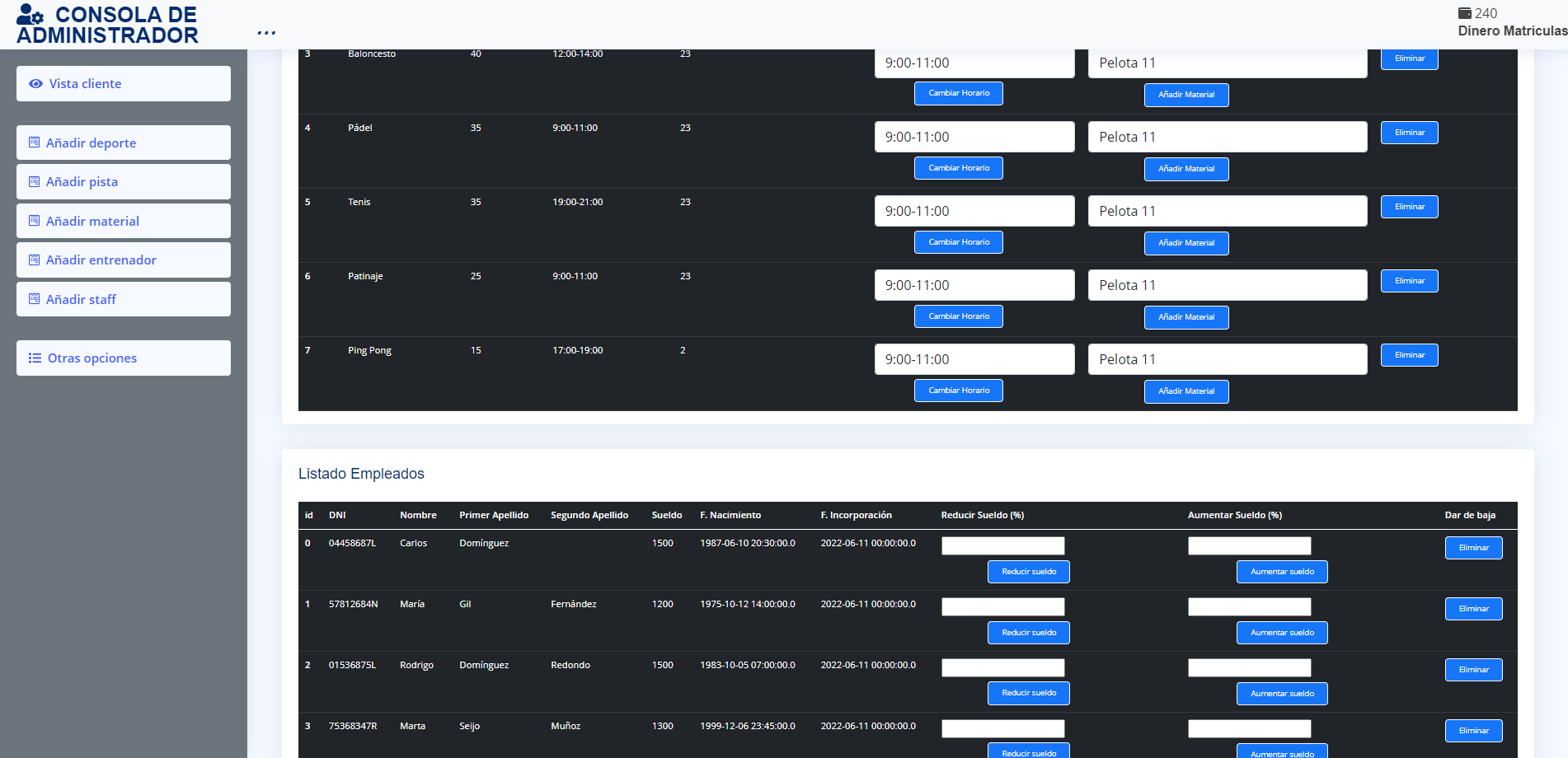


(Imagen 17: Datos del cliente)

Vista administrador:



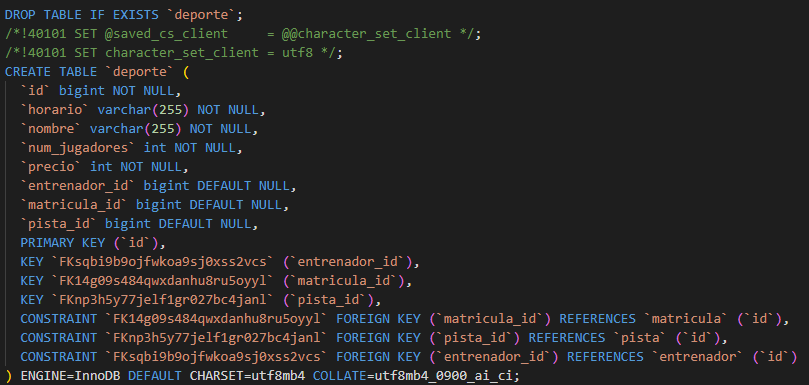
(Imagen 18: Formularios administrador)

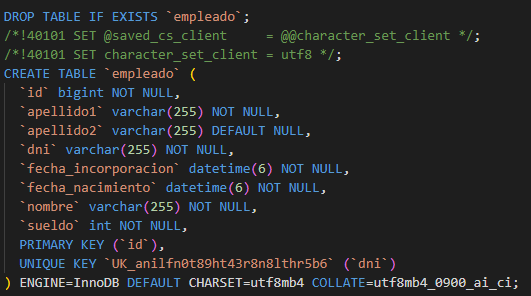


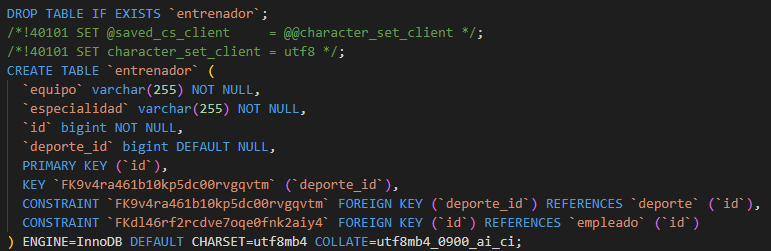
(Imagen 19: Listados administrador)

### Diseño. ¿Cómo se hará la aplicación?

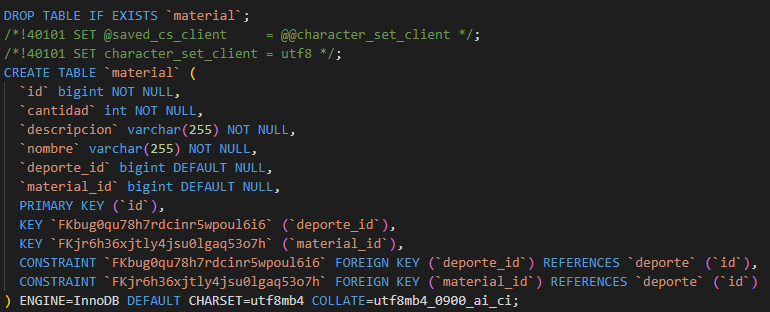
En lo referido a la estructura, la organización de las tablas será la siguiente:

(Imagen 20: Tabla “deporte” en SQL)

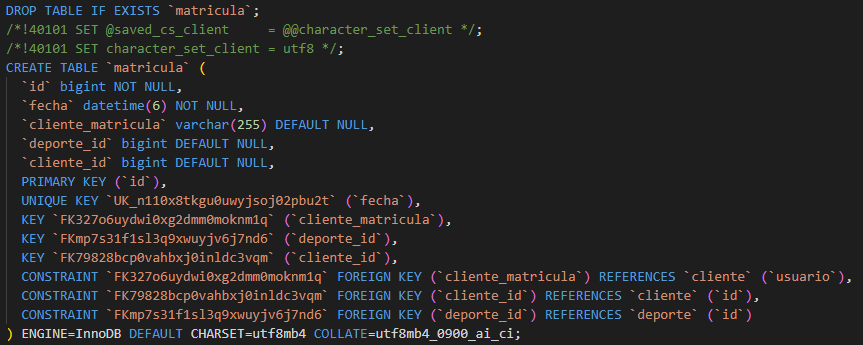
(Imagen 21: Tabla “empleado” en SQL)



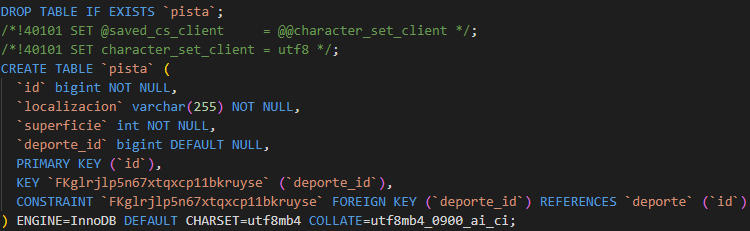
(Imagen 22: Tabla “entrenador” en SQL)



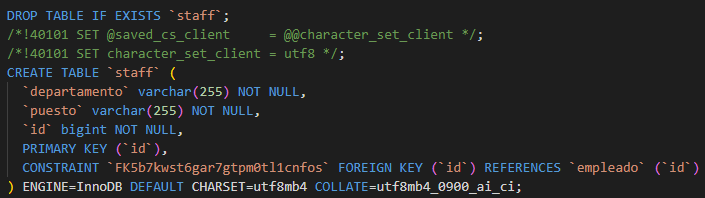
(Imagen 23: Tabla “material” en SQL)



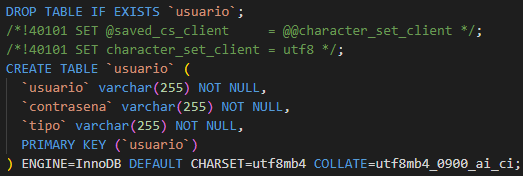
(Imagen 24: tabla “matricula” en SQL)



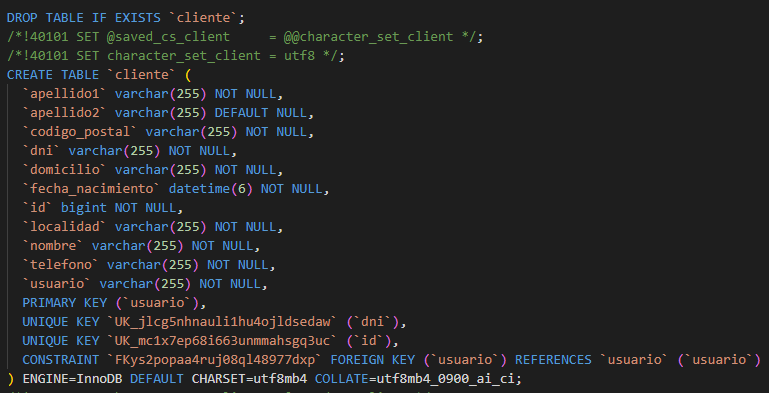
(Imagen 25: Tabla “pista” en SQL)



(Imagen 26: Tabla “staff” en SQL)



(Imagen 27: Tabla “usuario” en SQL)



(Imagen 28: Tabla “cliente” en SQL)

Para el desarrollo de la aplicación, haremos uso de diversas tecnologías:

* **Parte web.** La página o servicio web será desarrollada con Java, concretamente con Spring Boot, recurriendo al patrón MVC. Para la parte de la vista haremos uso del conocido motor de plantillas Thymeleaf. Además, también influirán otras técnicas como HTML5, CSS3, Bootstrap5 y JavaScript.
* **Gestión de BD.** Mediante un fichero de configuración (Docker-compose) incluiremos un gestor de bases de datos (MySQL).

### Implementación y pruebas

En lo referido al desarrollo de la aplicación, hemos optado por un lenguaje java apoyado en Spring. La arquitectura del proyecto cuenta con una serie de capas que podemos clasificar en:

* **Entidades**: cada una con sus distintas propiedades. Entre todas formarán la estructura de la base de datos.
* **Controladores**:permiten la relación entre los servicios y la vista, definiendo la URL pertinente y la vista asociada
* **Servicios:** son llamados desde los controladores, contienen los diferentes métodos funcionales y las llamadas a los repositorios.
* **Repositorios:** son llamados desde los servicios. Apoyados en el framework de Spring Data JPA, permiten las llamadas a la base de datos, realizando en ésta diversas consultas (algunas ya definidas y otras creadas por nosotros), inserciones, borrados y actualizaciones.

Para la elaboración de pruebas hemos optado por realizar un controlador REST que realice llamadas a los distintos servicios creados. Así, ayudándonos de una tecnología aprendida durante las FCT (Postman) realizamos las peticiones deseadas y observamos las distintas respuestas obtenidas (éxito, errores…).

### Implantación y documentación

En primer lugar, hay que ejecutar el fichero **docker-compose.yml**. Para ello, hay que abrir un terminal cmd o PowerShell en la carpeta del proyecto (a la altura del propio fichero) y ejecutar “docker-compose up -d”.

Tras esto, se habrá ejecutado el servicio de base de datos y podremos ejecutar la aplicación, que, tras configurar la conexión a la base de datos, realizará la creación de las distintas tablas en esta.

Abrir un navegador web y poner el puerto y la dirección IP o dominio donde se aloja dicho programa.

### Resultados y discusión

En cuanto al desarrollo, ambos miembros estamos de acuerdo en que la sensación del proyecto final es buena, pero con un cierto grado de decepción. Esta decepción es debida a que no se han logrado alcanzar las expectativas propuestas inicialmente, ya que teníamos intención de hacer más uso de la herramienta Docker, instalando un DNS y un proceso de despliegue de la aplicación con tomcat. En conclusión, ya sea porque nuestras expectativas eran demasiado ambiciosas o porque no hemos estado a la altura, obtenemos un proyecto que en lo referido a la aplicación web estamos satisfechos, pero no en lo referido al despliegue de la misma.

En lo referido a las dificultades encontradas, más allá de ciertas excepciones que se han ido generando en el código o ciertos errores generales, la principal dificultad ha sido, como se ha mencionado en el punto anterior, los distintos problemas generados en el despliegue de la aplicación. Aun habiéndonos puesto en contacto con los profesores para pedir consejo, no hemos sido capaces de desarrollar un correcto despliegue.

En un futuro, cuando contemos con más experiencia, se podría realizar un despliegue de la aplicación óptimo. Por otro lado, en cuanto al tema de usuarios de registro y logeo, se podría trabajar con sesiones, en vez de con entidades de la base de datos, permitiendo así mantener el perfil abierto durante toda la ejecución de la aplicación.

## Conclusiones

Este proyecto nos ha permitido consolidar todos los conocimientos obtenidos durante el ciclo, y a su vez, poder obtener junto a las FCT una experiencia de trabajo en equipo más orientada al mundo laboral. El haber sido los únicos responsables de la arquitectura y desarrollo de nuestro proyecto hemos obtenido una mayor capacidad de actuar frente a ciertas situaciones, rediseñando la arquitectura o buscando soluciones a errores.

En cuanto a conocimientos ampliados, es cierto que hemos obtenido nuevos conocimientos, pero consideramos que más allá de los conocimientos, lo que más hemos obtenido es experiencia para abordar problemas y solucionarlos.

# 

## Bibliografía y referencias

Listado de páginas web y documentos a partir de los cuales hemos resuelto dudas sobre el proyecto:

* Documentos en el aula virtual del centro de los diferentes módulos.
* El popular foro de resolución de conflictos: <https://stackoverflow.com/>
* La famosa página web de aprendizaje y documentación: <https://www.w3schools.com/>
* La página web oficial de Spring: <https://spring.io/projects/spring-data-jpa>
* El portal de Thymeleaf: <https://www.thymeleaf.org/>
* La página de información de Docker: <https://docs.docker.com/compose/>
* El sitio web de Bootstrap: <https://getbootstrap.com/>

## Anexos

# Formato de entrega del proyecto

* Extensión: 25-30 páginas, en folios tamaño DIN A4, incluyendo las figuras y tablas, que se integrarán en el texto con subtítulos y orden de numeración. Anexos excluidos.
* Tipo de letra: Arial o Times New Roman, tamaño 12, interlineado 1.5
* Portada: Contendrá el título del proyecto, el nombre del autor/es, ciclo formativo, curso académico, fecha de entrega y el logotipo del IES.
* Encabezado y pie de página: Entre los dos contendrán la siguiente información: Logo del proyecto, logo del IES, nombre del alumno, curso y ciclo, año escolar, número de página.
* Soporte: Se entregará una copia en papel del documento, encuadernada en espiral.

Además, se subirá a la plataforma virtual del curso el documento en pdf, el archivo de la presentación que se vaya a utilizar en la exposición, y el código fuente de la aplicación

* Licencia (Opcional): Al proyecto se le otorgará la licencia Creative Commons más restrictiva.
* Índice y paginación: Serán obligatorios e incluirán al menos los contenidos especificados en el punto anterior, entre otros que se estimen convenientes por el alumno.