



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

– **TELECOM** ESCUELA
TÉCNICA **VLC** SUPERIOR
DE INGENIERÍA DE
TELECOMUNICACIÓN

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería de
Telecomunicación

Aplicación web para la gestión, coordinación y creación de
eventos deportivos para la UPV

Trabajo Fin de Grado

Grado en Ingeniería de Tecnologías y Servicios de
Telecomunicación

AUTOR/A: Puchades Jiménez, Sergio

Tutor/a: López Patiño, José Enrique

CURSO ACADÉMICO: 2022/2023



Resumen

Aplicación web que facilitará participar en eventos deportivos sin necesidad de tener una lista previa de personas para poder disfrutar de cualquier evento deportivo en la UPV. En esta, el alumno se inscribirá en un evento de grupo de manera individual, seleccionando su posición/posiciones y, en algunos eventos su nivel, así con el resto de clientes de tal manera que una vez que el evento se complete, avise a los participantes, confirmándoles que el evento finalmente se hará.

En caso de que no se llegue a completar antes de una fecha límite, la aplicación web también avisará y cancelará el evento. Gracias a esto, por ejemplo, no será preciso el tener 14 personas de antemano para poder jugar, por ejemplo un partido de fútbol.

Además, fomentará conocer al resto de estudiantes del campus a través del deporte. Esta aplicación web estará disponible para un gran número de deportes que la UPV ofrece. Para inscribirse en un evento, únicamente se registrarán en la plataforma o bien entrarán con sus credenciales en el caso de ya tener una cuenta, elegirán el deporte, el día y la hora y, finalmente algunos datos relacionados con la naturaleza del deporte

Resum

Aplicació web que facilitarà la participació en esdeveniments esportius sense necessitat d'una llista prèvia de persones per a poder gaudir de qualsevol esdeveniment esportiu a la UPV. En aquesta aplicació, l'alumne s'inscriurà en un esdeveniment de grup de manera individual, seleccionant la seua posició/posicions i, en alguns esdeveniments el seu nivell, juntament amb la resta de clients per tal que una vegada l'esdeveniment es complete, els notifique als participants, confirmant que finalment es durà a terme.

En cas que l'esdeveniment no s'haja completat abans d'una data límit, l'aplicació web també avisarà i cancel·larà l'esdeveniment. Gràcies a açò, per exemple, no serà necessari tindre 14 persones prèviament per a poder jugar, per exemple, un partit de futbol.

A més, fomentarà conèixer la resta d'estudiants del campus a través de l'esport. Aquesta aplicació web estarà disponible per a un gran nombre d'esports que la UPV ofereix. Per a inscriure's en un esdeveniment, només s'hauran de registrar en la plataforma o bé accedir amb les seues credencials en cas de ja tindre un compte, triar l'esport, el dia i l'hora i, finalment, proporcionar alguns dades relacionats amb la naturalesa de l'esport.



Abstract

A web application that will make it easier to participate in sports events without the need for a previous list of people to be able to enjoy any sports event at UPV. In this application, the student will register for a group event individually, selecting their position/positions and, in some events, their level, along with the other clients so that once the event is completed, it will notify the participants, confirming that the event will finally take place.

In case the event is not completed before a deadline, the web application will also notify and cancel the event. Thanks to this, for example, it will not be necessary to have 14 people in advance to play, for example, a football match.

Additionally, it will encourage students to get to know each other on campus through sports. This web application will be available for a large number of sports offered by UPV. To register for an event, they will simply register on the platform or log in with their credentials if they already have an account, select the sport, day, and time, and finally provide some data related to the nature of the sport.



Índice de contenidos

1. Introducción.....	6
1.1. Motivación.....	6
1.2. Objetivos.....	6
1.3. Metodología empleada.....	7
1.4. Desarrollo de la memoria.....	10
2. Contexto Actual.....	11
2.1. Desarrollo web.....	11
2.2. Eventos deportivos.....	12
3. Tecnologías Utilizadas.....	14
3.1. Introducción.....	14
3.2. Linux.....	14
3.2.1. Introducción.....	14
3.2.2. Historia.....	14
3.3. Servidor Apache.....	15
3.3.1. Introducción.....	15
3.3.2. Historia.....	15
3.4. MYSQL.....	15
3.4.1. Introducción.....	15
3.4.2. Historia.....	15
3.5. PHP.....	16
3.5.1. Introducción.....	16
3.5.2. Historia.....	16
3.6. Java Script.....	17
3.6.1. Introducción.....	17
3.6.2. Historia.....	17
3.7. HTML.....	18
3.7.1. Introducción.....	18
3.7.2. Historia.....	18
3.8. CSS.....	19
3.8.1. Introducción.....	19
3.8.2. Historia.....	19
3.9. Herramientas complementarias.....	20
3.9.1. WinSCP.....	20
3.9.2. HeidiSQL.....	22
3.9.3. PuTTY.....	23
4. Descripción Del Proyecto Web.....	24
4.1. Introducción.....	24
4.2. El Servidor.....	24
4.2.1. Apache.....	25



4.2.2. Base De Datos.....	27
4.3. Estructura de directorios.....	31
4.4. Desarrollo de la aplicación.....	33
4.4.1. Front-end.....	34
4.4.2. Back-end.....	36
5. Funcionalidad de la aplicación.....	39
5.1. Introducción.....	39
5.2. El cliente.....	39
5.3. El administrador.....	54
6. Conclusiones.....	56
7. Bibliografía.....	57

Índice de figuras

Figura 1.....	7
Figura 2.....	8
Figura 3.....	8
Figura 4.....	9
Figura 5.....	21
Figura 6.....	21
Figura 7.....	22
Figura 8.....	22
Figura 9.....	23
Figura 10.....	23
Figura 11.....	26
Figura 12.....	26
Figura 13.....	27
Figura 14.....	30
Figura 15.....	30
Figura 16.....	31
Figura 17.....	33
Figura 18.....	39
Figura 19.....	40
Figura 20.....	41
Figura 21.....	42
Figura 22.....	43
Figura 23.....	43
Figura 24.....	43
Figura 25.....	44
Figura 26.....	44
Figura 27.....	44
Figura 28.....	44



Figura 29.....	45
Figura 30.....	45
Figura 31.....	45
Figura 32.....	45
Figura 33.....	46
Figura 34.....	46
Figura 35.....	46
Figura 36.....	46
Figura 37.....	47
Figura 38.....	47
Figura 39.....	47
Figura 40.....	48
Figura 41.....	48
Figura 42.....	49
Figura 43.....	49
Figura 44.....	50
Figura 45.....	50
Figura 46.....	51
Figura 47.....	51
Figura 48.....	52
Figura 49.....	52
Figura 50.....	53
Figura 51.....	54
Figura 52.....	55
Figura 53.....	34
Figura 54.....	36
Figura 55.....	37

1. Introducción

1.1. Motivación

Existen dos motivos principales que hicieron arrancar este proyecto. Primeramente mi afición al deporte, concretamente a los deportes de equipo. He sido deportista durante toda mi vida y lo sigo siendo, estoy enormemente agradecido el que en su día alguien me inculcara la disciplina deportiva e hiciera que el deporte fuese una parte fundamental para mí. Dicho esto, los deportes de equipo son un entorno ideal para conocer gente y aprender a trabajar en equipo por un mismo fin.

El segundo motivo es mi pasión por las tecnologías, concretamente las basadas en electrónica como la computación. Me parece fascinante como una cantidad abrumadora es capaz de crear cosas como las de hoy en día basando toda su complejidad al uso de un sistema binario.

Llegado el día, me dí cuenta de lo difícil que era concretar cualquier evento deportivo. Primeramente, entras a un entorno nuevo y no conoces prácticamente a nadie. Segundo, una vez creas un grupo de personas que son de tu agrado, te das cuenta de la complicación que supone acordar un día, una hora y completar el cupo de personas mínimo que necesitaríais para poder disfrutar del deporte en equipo.

Por ello, gracias al momento contextual tecnológico en el que nos encontramos, concretamente dentro del desarrollo web, pensé en lo que supondría el poder inscribirse cuando quisieras en un evento deportivo, sin tener que realizar ningún tipo de captación previa. Además, siendo conocedor de las grandes amistades que me ha dado el deporte en equipo, no veía límites en la escalabilidad social que podía tener dicha aplicación web dentro de la UPV.

1.2. Objetivos

Este proyecto tiene como objetivo promover el deporte así como facilitar a las personas el poder disfrutar de, por ejemplo, un partido de balonmano sin la necesidad de conocer y coordinar a entre diez y doce personas para poder jugar el partido. Partiendo del ejemplo, se escala a una gran cantidad de deportes que ofrece la UPV.

Existe un objetivo que es transversal. El proyecto pretende, de manera indirecta aumentar la cantidad de personas que puedes llegar a conocer durante tus estudios dentro de la UPV así como ampliar tus conocimientos a cerca de lo que ofrece, es seguro que, a más gente conozcas más cosas aprenderás sobre lo que te ofrece tu etapa universitaria, serás conocedor del contexto en otras carreras, los servicios que tu desconoces, posibles futuros compañeros para afrontar exámenes y motivarte a seguir con tus estudios y, quien sabe, quizás futuros compañeros en el mundo laboral. Todo ello gracias al uso de las nuevas tecnologías y al deporte.

1.3. Metodología empleada

Durante la elaboración del proyecto en su totalidad, incluida la memoria, me ha sido de gran ayuda emplear metodologías ágiles como herramientas de etiquetado. Concretamente, el seguimiento de las tareas que tenía que realizar, plazos de entrega autoimpuestos con el objetivo de ser riguroso, posibles mejoras, etc. Para todas estas cosas, se ha utilizado la herramienta de etiquetado llamada Trello. Esta herramienta consiste en la creación de listas, a las cuales les vas añadiendo tarjetas en función de tus necesidades. Además, estas tarjetas disponen de una gran cantidad de opciones para ayudarte con tu proyecto en este caso.

En este apartado se hablará sobre dicha herramienta y sobre como se ha explotado para hacer un seguimiento riguroso del proyecto. Dentro del seguimiento, se optó por crear las siguientes listas: “Lista de tareas”, “En local”, “En el entorno de prueba”, “En producción”, “DONE”. Además habían 5 tipos de etiquetas para marcar los diferentes tickets: “PRIORIDAD ALTA”, “PRIORIDAD MEDIA”, “SQL”, HTML y CSS” y por último “PHP”. En función del ticket que se creaba se le asignaban una serie de etiquetas, en las figuras 1, 2, 3 y 4 se aprecia una pequeña evolución del proyecto gracias a la herramienta, así como el contenido de algunos tickets.

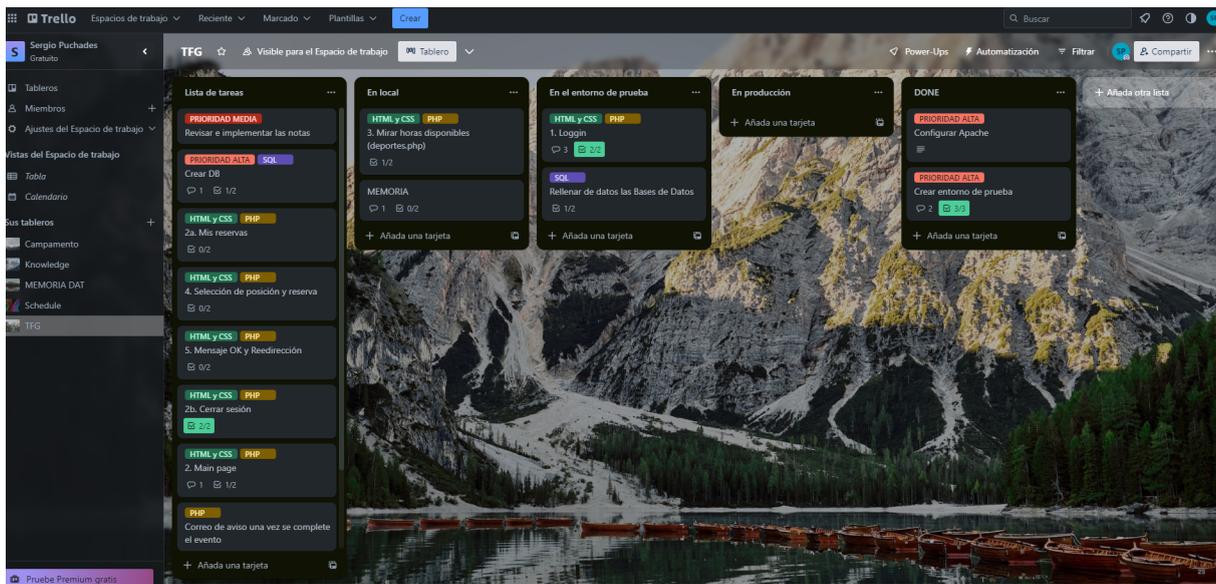


Figura 1. Herramienta de etiquetado Trello. El estado de las tareas en la fase inicial del proyecto.

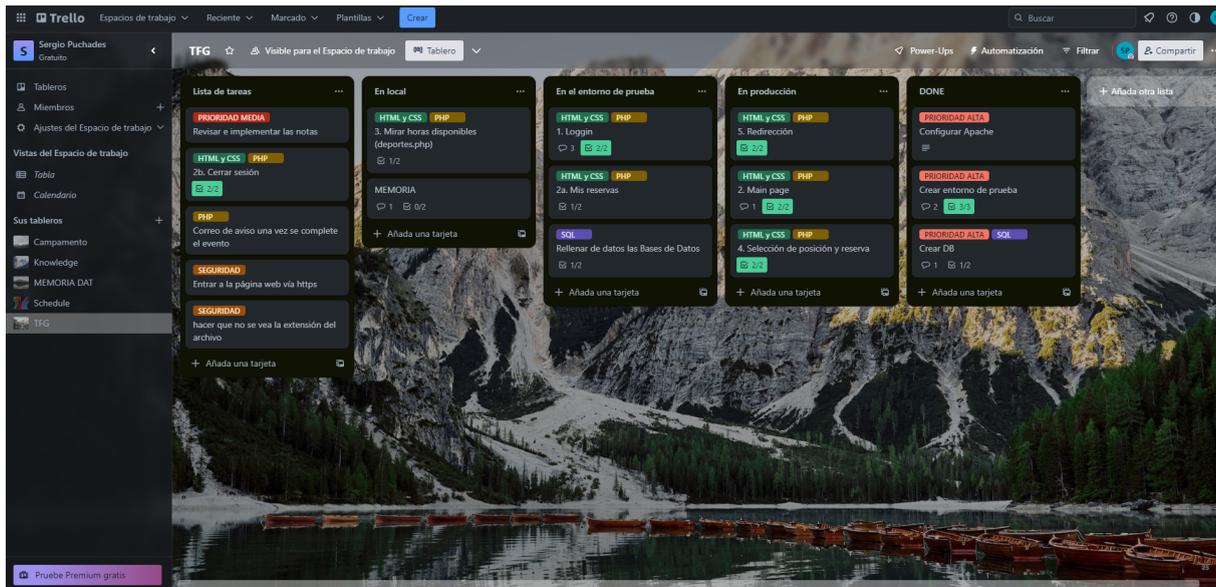


Figura 2. Herramienta de etiquetado Trello. El estado de las tareas en una fase intermedia del proyecto.

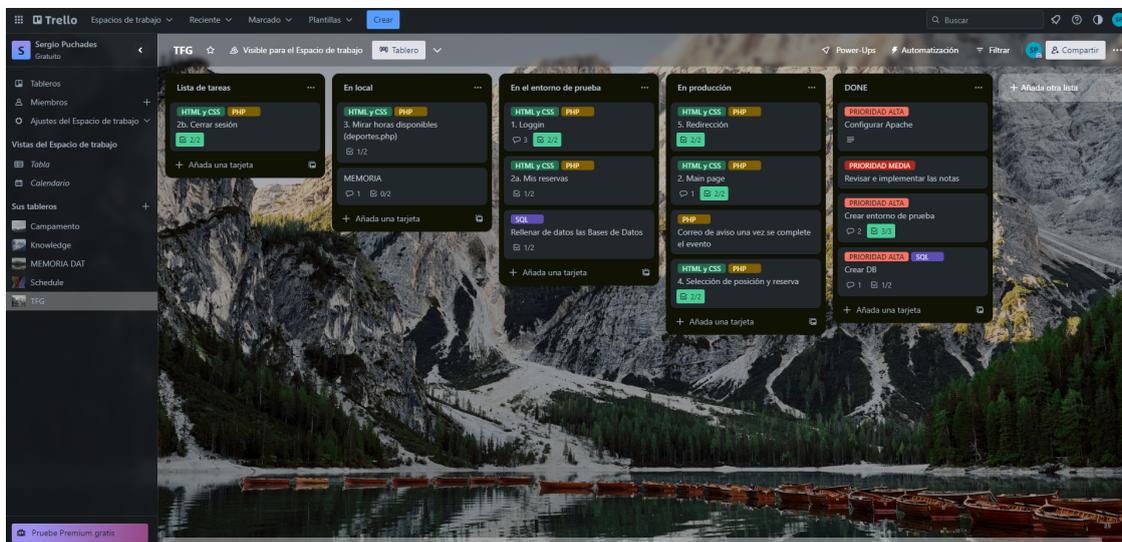


Figura 3. Herramienta de etiquetado Trello. El estado de las tareas en la fase final del proyecto.

Tal y como se aprecia en las figuras anteriores, las tarjetas van oscilando entre unas listas y otras en función de su estado. Dentro de “Lista de tareas” se encontraba al inicio todas las tarjetas. Una vez puesto en marcha el desarrollo del proyecto, el siguiente nivel por el que pasaban era por la lista “En local”, esta lista estaba para cuando tenía archivos de código en el ordenador personal y no subidos al servidor. A continuación aparecen dos listas más, “en entorno de pruebas” y la de “entorno de producción”. Ambas listas sirven para saber donde en

que entorno está el proyecto y por tanto ver lo que aún no se ha conseguido llegar a su versión final. A medida que pasaba el tiempo las tarjetas iban oscilando de una lista a otra ya que se realizaban pequeños parches desde el entorno de pruebas con código que ya estaba en el entorno de producción. Por último, tenemos la lista “DONE” esta lista es la encargada de recoger todas aquellas tareas que no eran directamente la programación de la página web y, por lo tanto no les correspondía estar en ningún entorno. Tareas como por ejemplo la creación de las bases de datos, instalación y funcionamiento del servidor apache, etc.

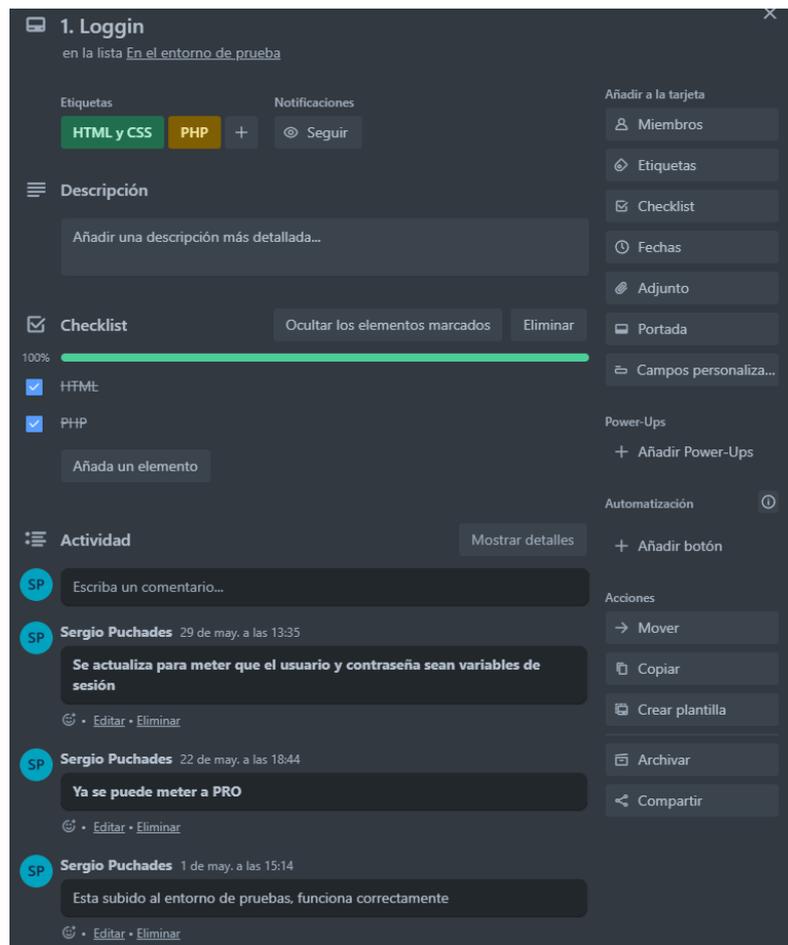


Figura 4. Herramienta de etiquetado Trello. Tiquet expandido.

La figura 4 nos muestra una de las tarjetas expandida. Se aprecia la gran capacidad de descripción que tiene para poder seguir rigurosamente el estado de la tarjeta. Primeramente, tenemos el título, con un vistazo puedes saber de que se trata. A continuación aparecen las etiquetas de la tarjeta, otro golpe visual para rápidamente saber a que se refiere. Debajo de las etiquetas, la herramienta concede un espacio para hacer una descripción de la tarjeta si es necesario.

A la derecha, aparece una columna con todas las opciones descriptivas y decorativas que puedes realizar. Las que han sido utilizadas es la implementación de una “Checklist”, adjuntar archivos e insertar fechas.

La “checklist” aparece justo debajo de la descripción, como se aprecia, en este caso es una comprobación de como avanza internamente la página de “Loggin”, es decir, inicio de sesión, a nivel de código. Por último, hay un registro de actividad, donde se pueden poner comentarios paulatinamente en función del progreso de la tarea. esto nos permite llevar un control bastante detallado del estado donde se quedó la última vez que se trabajó sobre ella.

1.4. Desarrollo de la memoria

La memoria de este proyecto se podría dividir en dos grandes bloques, un bloque contextual, donde atacamos al análisis del contexto actual dentro del tema del proyecto. En este mismo bloque se atañe también a las tecnologías que se han empleado, un breve texto con la historia de cada tecnología nos pondrá en situación sobre la misma.

El segundo bloque atañe al proyecto en cuestión. Se detallará el mismo, sus partes y funcionalidades, más concretamente. Estará dividido en dos. Primeramente, la descripción del entorno y del desarrollo de la aplicación, apartado 4. Este apartado sobre el desarrollo atacará al servidor utilizado, base de datos y a los ficheros que toman parte en el proyecto web y sus desempeños separando estos ficheros en *front-end* y *back-end*. A continuación, se hablará sobre la funcionalidad de la aplicación, apartado 5, explicándose aquí la experiencia por la que pasa un cliente desde que accede hasta que cierra la sesión, y las funciones administrativas que deberá desempeñar el administrador de la página.

Finalmente nos encontraremos dos apartados más. Uno de ellos estará dedicado a las conclusiones obtenidas una vez realizado el proyecto. El segundo y último apartado será la bibliografía.

2. Contexto Actual

2.1. Desarrollo web

Hoy en día el desarrollo de software está a la orden del día. Dentro de este mundo y, por como está evolucionando la sociedad, uno de las ramas más demandadas dentro este es el desarrollo web.

El desarrollo web se refiere al proceso de crear y mantener sitios web. Implica la planificación, diseño, desarrollo y puesta en marcha de una aplicación web. El desarrollo web abarca una amplia gama de actividades, desde la creación de una página web simple con contenido estático hasta el desarrollo de aplicaciones web complejas y dinámicas.

El desarrollo web ha tenido una gran influencia en nuestras vidas, he aquí un par de ejemplos sobre dicha influencia:

Comunicación y accesibilidad: Las páginas web han transformado la forma en que nos comunicamos y accedemos a la información. Proporcionan una plataforma global para compartir y acceder a contenido.

Presencia en línea: Las páginas web son una herramienta crucial para establecer una presencia en línea. Ya sea para empresas, organizaciones o individuos, tener un sitio web permite mostrar información relevante, productos, servicios y lograr una mayor visibilidad en el mundo digital.

Comercio electrónico: El desarrollo web ha dado lugar al auge del comercio electrónico. Las tiendas en línea permiten a las empresas vender productos y servicios las 24 horas del día, los 7 días de la semana, a una audiencia global. Esto ha transformado la forma en que las personas compran y venden, abriendo nuevas oportunidades de negocio.

Interactividad y experiencia del usuario: Las páginas web interactivas y atractivas mejoran la experiencia del usuario. El desarrollo web ha evolucionado para ofrecer diseños intuitivos, navegación fluida y contenido interactivo, lo que brinda a los visitantes una experiencia enriquecedora y agradable.

Compartir conocimientos y colaboración: Las páginas web permiten compartir conocimientos, recursos educativos, investigaciones y colaborar en proyectos en línea. Plataformas como blogs, foros y sistemas de gestión de contenidos facilitan la creación y el intercambio de información, fomentando la colaboración y el aprendizaje.

Algo que destaca en el mejor de los sentidos respecto al desarrollo web es la gran variedad de tecnologías que tenemos para la creación y mantenimiento. Por resumir un poco estos serían los principales componentes dentro de un proyecto de desarrollo web. Más adelante se hablará de algunos de estos componentes.

Lenguajes de marcado: como HTML (HyperText Markup Language) para estructurar el contenido de la página y CSS (Cascading Style Sheets) para definir la apariencia y el diseño.

Lenguajes de programación: como JavaScript, PHP, Python, Ruby, entre otros, que permiten agregar funcionalidad y realizar operaciones dinámicas en el lado del servidor o del cliente.

Bases de datos: para almacenar y recuperar información de manera eficiente.

Servidores web: como Apache o Nginx, que son responsables de entregar las páginas web al navegador del usuario.

Frameworks y bibliotecas: como React, Angular, Django o Ruby on Rails, que proporcionan herramientas y estructuras para facilitar el desarrollo web y acelerar el proceso de construcción de aplicaciones.

En resumen, las páginas web desempeñan un papel fundamental en la forma en que nos comunicamos, accedemos a la información y realizamos transacciones en la era digital. Su influencia se extiende a diversos campos y sectores, y el desarrollo web es un proceso multidisciplinario que combina diseño programación y gestión de bases de datos para crear sitios web funcionales y atractivos.

2.2. Eventos deportivos

Los eventos deportivos son una parte integral de la cultura global y desempeñan un papel significativo en el ámbito del entretenimiento y el deporte.

La popularidad y audiencia masiva en los eventos deportivos por ejemplo. Atraen a una audiencia masiva en todo el mundo. Competiciones como los Juegos Olímpicos, la Copa Mundial de la FIFA y los torneos de tenis de Grand Slam generan un gran interés y atraen a millones de espectadores tanto en los estadios como a través de las transmisiones televisivas.

Por otro lado, el impacto económico, los eventos deportivos importantes tienen un impacto económico significativo en las ciudades y países anfitriones. Generan ingresos a través del turismo, la inversión en infraestructuras y el gasto de los espectadores en alojamiento, transporte, alimentación y compras.

Como se ha mencionado anteriormente, la promoción del deporte. Los eventos deportivos internacionales promueven la práctica deportiva y la actividad física. Inspiran a las personas a participar en diferentes disciplinas deportivas y pueden tener un impacto positivo en la salud y el bienestar de la población.

Cada evento tiene sus características y elementos distintivos, pero en general, estos eventos desempeñan un papel importante en la cultura global, generando emoción, promoviendo la unidad y dejando un impacto significativo en los ámbitos económico, social y deportivo.



En resumen, los eventos deportivos son un gran motor mundial, por ello, aunque en menor medida que grandes organizaciones y competiciones, tener la oportunidad de promocionarlos aportando mi grano de arena a la montaña me satisface enormemente.

3. Tecnologías Utilizadas

3.1. Introducción

En este apartado se detallarán las tecnologías que se han empleado durante el trabajo. Junto a cada tecnología se incluye una breve introducción a cerca de la misma y un contexto histórico para conocer de donde vienen. Por último, se han recogido en una sección las herramientas más relevantes que han sido usadas en el trabajo.

3.2. Linux

3.2.1. Introducción

Linux es un sistema operativo de código abierto y libremente distribuible que ha revolucionado el panorama de la tecnología. Desde su creación, ha ganado popularidad en diversos campos, incluyendo servidores web. En este apartado, exploraremos en qué consiste Linux y cómo ha influido en el desarrollo de la estructura LAMP.

3.2.2. Historia

La historia de Linux se remonta a 1991, cuando un estudiante finlandés llamado Linus Torvalds inició su desarrollo como un proyecto personal. Inspirado en el sistema operativo UNIX, Torvalds comenzó a trabajar en un núcleo de sistema operativo que pronto recibiría contribuciones de programadores de todo el mundo. La colaboración abierta y la licencia de código abierto permitieron que Linux creciera rápidamente y se adaptara a diversas necesidades.

En la década de 1990, Linux se consolidó como una opción viable para servidores web debido a su estabilidad, seguridad y flexibilidad. Empresas y organizaciones comenzaron a adoptar Linux para sus infraestructuras de servidor, aprovechando sus capacidades de personalización y su amplia compatibilidad con una variedad de aplicaciones y herramientas. Hoy en día, Linux es el sistema operativo dominante en servidores web, potenciando una amplia gama de aplicaciones y servicios en línea.

3.3. Servidor Apache

3.3.1 Introducción

Apache es un servidor web de código abierto reconocido mundialmente por su fiabilidad y rendimiento. En este apartado, exploraremos cómo Apache ha contribuido a la estructura LAMP y su importancia en el alojamiento de sitios web y aplicaciones.

3.3.2 Historia

Apache nace en 1995 como un proyecto impulsado por la comunidad. Su nombre proviene de la tribu nativa americana Apache, conocida por su habilidad en la guerra y su resistencia. Al igual que su nombre, el servidor web Apache demostró ser valiente y resistente en un mercado emergente y altamente competitivo.

Inicialmente, Apache fue creado como una alternativa al servidor web NCSA HTTPd. Su popularidad creció rápidamente debido a su licencia de código abierto, lo que permitía a los desarrolladores adaptarlo y mejorarlo libremente. La arquitectura modular de Apache permitió a los administradores de servidores personalizarlo según sus necesidades específicas, brindando flexibilidad y escalabilidad.

A lo largo de los años, Apache ha evolucionado para enfrentar los desafíos cambiantes de la web, agregando características de seguridad, rendimiento y compatibilidad con estándares web. Su estabilidad y capacidad para manejar altas cargas de tráfico han hecho de Apache una opción predilecta en la industria, respaldando millones de sitios web y aplicaciones en todo el mundo.

3.4. MYSQL

3.4.1. Introducción

MySQL es un sistema de gestión de bases de datos relacional ampliamente utilizado en la estructura LAMP. En esta sección, exploraremos cómo MySQL ha influido en el desarrollo de aplicaciones web y cómo se ha convertido en una piedra angular en la gestión de datos en línea.

3.4.2. Historia

MySQL, una de las tecnologías más populares para la gestión de bases de datos relacionales, fue creada en 1995 por Michael "Monty" Widenius, Allan Larsson y David Axmark. Su objetivo principal era desarrollar una solución de base de datos rápida, confiable y de código abierto que pudiera adaptarse a diversas aplicaciones y entornos.

En sus inicios, MySQL fue diseñado para satisfacer las necesidades de los sitios web en crecimiento, proporcionando una opción accesible y escalable para almacenar y recuperar datos. La primera versión de MySQL, conocida como "Version 3.21", fue lanzada en mayo de 1996.

En 2000, la compañía MySQL AB fue fundada para proporcionar soporte comercial y servicios relacionados con MySQL. Esto contribuyó a una mayor adopción de MySQL en empresas y organizaciones de todo el mundo. MySQL se convirtió en una opción confiable para aplicaciones de misión crítica, ya que ofrecía un rendimiento sólido, escalabilidad y una amplia compatibilidad con diferentes sistemas operativos y lenguajes de programación.

En 2008, Sun Microsystems adquirió MySQL AB y, posteriormente, en 2010, Oracle Corporation adquirió Sun Microsystems. Esta adquisición generó cierta incertidumbre en la comunidad de usuarios de MySQL debido a las preocupaciones sobre el futuro de la tecnología de código abierto. Sin embargo, Oracle ha continuado desarrollando y manteniendo MySQL, y se ha comprometido a seguir proporcionando una versión de MySQL de código abierto.

A lo largo de los años, MySQL ha evolucionado y ha introducido numerosas características y mejoras. Hoy en día sigue siendo una de las bases de datos relacionales más utilizadas en el mundo. Su popularidad se debe a su estabilidad, velocidad y la gran comunidad de usuarios y desarrolladores que respaldan la tecnología. MySQL se utiliza en una amplia gama de aplicaciones y casos de uso, desde pequeños sitios web y aplicaciones empresariales hasta sistemas de alta carga y escalabilidad.

3.5. PHP

3.5.1. Introducción

PHP es un lenguaje de programación de código abierto ampliamente utilizado en el desarrollo web. En este apartado, exploraremos la importancia de PHP en la estructura LAMP y cómo ha contribuido a la creación de sitios web dinámicos y aplicaciones interactivas.

3.5.2. Historia

El origen de PHP se remonta a 1994, cuando Rasmus Lerdorf creó un conjunto de scripts para rastrear las visitas a su currículum vitae en línea. Estos scripts evolucionaron y se convirtieron en Personal Home Page Tools, más tarde renombrados como PHP Hypertext Preprocessor.

Inicialmente, PHP fue diseñado como un lenguaje de scripting simple para la creación de páginas web dinámicas. A medida que ganaba popularidad, Lerdorf lanzó la primera versión pública en 1995. El enfoque de PHP en la facilidad de uso y la integración con HTML lo convirtió rápidamente en una opción atractiva para los desarrolladores web.

Con el tiempo, PHP se expandió y maduró, agregando características como el soporte para bases de datos y la capacidad de interactuar con servidores web. Su compatibilidad con una amplia gama de sistemas operativos y servidores web contribuyó a su adopción generalizada.

La comunidad de desarrolladores de PHP desempeñó un papel fundamental en su crecimiento. La creación de una amplia variedad de bibliotecas, frameworks y herramientas adicionales permitió a los desarrolladores aprovechar al máximo PHP, acelerando el desarrollo de aplicaciones web y simplificando tareas comunes.

Hoy en día, PHP es uno de los lenguajes de programación más utilizados en el ámbito web. Ha demostrado ser un componente esencial en la estructura LAMP, brindando a los desarrolladores la capacidad de crear sitios web dinámicos, aplicaciones interactivas y sistemas de gestión de contenidos, todo ello integrado con Linux, Apache y MySQL.

3.6. Java Script

3.6.1. Introducción

JavaScript es un lenguaje de programación interpretado que se utiliza principalmente en el desarrollo web. Con su capacidad para agregar interactividad y funcionalidad dinámica a las páginas web, JavaScript se ha convertido en un componente esencial para crear experiencias web modernas y ricas en contenido.

3.6.2. Historia

JavaScript fue creado originalmente por Brendan Eich en 1995 mientras trabajaba en Netscape Communications Corporation. Inicialmente, el lenguaje se llamaba LiveScript, pero poco después se renombró a JavaScript para aprovechar el crecimiento de popularidad del lenguaje de programación Java en ese momento.

La primera versión de JavaScript se introdujo en Netscape Navigator 2.0 en septiembre de 1995. Su objetivo principal era proporcionar capacidades de scripting en el lado del cliente para interactuar con el HTML y modificar dinámicamente los elementos de una página web.

La adopción de JavaScript se aceleró con la aparición de nuevas versiones de los navegadores web, como Internet Explorer de Microsoft. En 1997, Netscape decidió estandarizar el lenguaje y se asoció con Ecma International, una organización de estándares, para crear una especificación oficial. Así nació ECMAScript, el estándar en el que se basa JavaScript.

Hoy en día, JavaScript es ampliamente utilizado en el desarrollo web y ha expandido su alcance más allá del navegador. Ahora se puede encontrar en servidores web (Node.js), aplicaciones móviles híbridas (PhoneGap, React Native), aplicaciones de escritorio (Electron) y en muchos otros contextos. La comunidad de desarrolladores ha creado una amplia variedad de bibliotecas y frameworks, como React, Angular y Vue.js, que han impulsado aún más la popularidad y la versatilidad de JavaScript.

JavaScript se ha convertido en un lenguaje esencial para el desarrollo web moderno, permitiendo la creación de interacciones dinámicas, validaciones de formularios, manipulación del DOM, solicitudes asíncronas (AJAX) y mucho más. Su evolución continua y su amplio soporte en los navegadores modernos lo hacen imprescindible para los desarrolladores que buscan crear experiencias web interactivas y dinámicas.

3.7. HTML

3.7.1. Introducción

HTML (Lenguaje de Marcado de Hipertexto) es un lenguaje fundamental en el desarrollo de páginas web. Proporciona la estructura y el formato básico para el contenido en línea, permitiendo a los desarrolladores crear documentos accesibles y enlazados. En esta sección, exploraremos la importancia de HTML y su papel en la creación de la World Wide Web.

3.7.2. Historia

El origen de HTML se remonta a la década de 1980, cuando Tim Berners-Lee propuso la idea de un sistema de hipertexto para compartir información en línea. En 1990, Berners-Lee desarrolló el primer lenguaje de marcado llamado HTML para describir la estructura y el formato de los documentos web. La especificación inicial de HTML permitía el enlace de documentos mediante hipervínculos y el uso de etiquetas para definir la estructura básica del contenido.

A medida que la web ganaba popularidad, HTML evolucionó para incluir nuevas características y mejoras. La especificación HTML 2.0 se publicó en 1995, seguida por HTML 3.2 en 1997 y HTML 4.01 en 1999. Cada versión introdujo elementos y atributos adicionales, mejorando la semántica y la presentación de los documentos web.

En 2000, la World Wide Web Consortium (W3C) asumió la responsabilidad de estandarizar HTML. Trabajando en colaboración con la comunidad web, el W3C desarrolló XHTML, una versión más estricta y basada en XML de HTML. Sin embargo, debido a su complejidad, XHTML no reemplazó por completo a HTML.

En 2014, el W3C lanzó HTML5, una importante revisión de HTML que introdujo nuevas características y capacidades. HTML5 proporcionó un conjunto de elementos semánticos, soporte para contenido multimedia, capacidades de almacenamiento local y más, lo que permitió a los desarrolladores crear experiencias web más ricas y dinámicas.

Hoy en día, HTML sigue siendo el lenguaje fundamental en el desarrollo web. Su versatilidad y compatibilidad con una amplia gama de plataformas y dispositivos han contribuido a su uso generalizado en la creación de páginas web modernas y accesibles.

3.8. CSS

3.8.1. Introducción

CSS (Hojas de Estilo en Cascada) es un lenguaje utilizado para controlar la apariencia y el diseño de las páginas web. Al separar el contenido de la presentación, CSS permite a los desarrolladores definir estilos consistentes y aplicar cambios de diseño de manera eficiente. En esta sección, exploraremos la importancia de CSS y cómo ha mejorado la estética y la experiencia del usuario en la web.

3.8.2. Historia

CSS surgió como una solución a la necesidad de separar la estructura y el diseño de los documentos web. Antes de la introducción de CSS, los estilos se aplicaban directamente en el código HTML, lo que dificultaba la gestión y la coherencia del diseño en diferentes páginas.

En 1996, el World Wide Web Consortium (W3C) lanzó la primera versión de CSS. Esta especificación permitía a los desarrolladores separar la información de estilo del contenido HTML, lo que facilitaba la actualización y la reutilización de los estilos en todo el sitio web.

A lo largo de los años, CSS ha experimentado varias revisiones y mejoras. CSS2 se introdujo en 1998, y CSS3, una versión importante y más completa, se publicó en módulos a partir de 2007. CSS3 introdujo una amplia gama de nuevas características, como la capacidad de aplicar animaciones, efectos visuales, sombras y diseños flexibles.

La adopción de CSS ha sido fundamental para el desarrollo web moderno. Al permitir un mayor control sobre la presentación y el diseño, CSS ha mejorado significativamente la apariencia y la usabilidad de los sitios web. Además, la capacidad de utilizar CSS en combinación con HTML y JavaScript ha permitido la creación de experiencias web interactivas y altamente personalizadas.

Hoy en día, CSS es una parte integral del desarrollo web y se ha convertido en un estándar ampliamente aceptado. Los navegadores modernos brindan un amplio soporte para las características de CSS, lo que permite a los desarrolladores crear diseños atractivos y responsivos en una variedad de dispositivos y tamaños de pantalla.

3.9. Herramientas complementarias

3.9.1. WinSCP

WinSCP es una aplicación de código abierto y libre. Consiste en un cliente SFTP o FTP gráfico para Windows. Su función principal es facilitar una transferencia segura de archivos entre dos máquinas.

Más concretamente, el cliente FTP es una herramienta utilizada para transferir archivos entre un cliente y un servidor a través del protocolo FTP. Este protocolo, ampliamente utilizado en la transferencia de archivos en redes informáticas, permite a los usuarios cargar, descargar, administrar y organizar archivos en servidores remotos. En esta sección, exploraremos la importancia y funcionalidad del cliente FTP en el intercambio eficiente de archivos en el entorno digital.

El protocolo FTP ha sido ampliamente utilizado en el desarrollo web y en la administración de sitios web. Los desarrolladores pueden utilizar clientes FTP para cargar archivos de un sitio web local al servidor web remoto, actualizar y modificar archivos existentes, y administrar la estructura de carpetas y archivos en el servidor. Esto es especialmente útil en entornos donde se necesita una transferencia eficiente y confiable de archivos, como en el caso de sitios web con mucho contenido multimedia o en el despliegue de aplicaciones web.

En resumen, el cliente FTP desempeña un papel crucial en la transferencia de archivos eficiente y confiable en redes informáticas. Con su funcionalidad para cargar, descargar y administrar archivos en servidores remotos, el cliente FTP se ha convertido en una herramienta fundamental para los desarrolladores web, administradores de sistemas y usuarios en general que necesitan intercambiar archivos de manera segura y eficiente en el entorno digital.

En las figuras 5 y 6 se pueden observar ambos menús, tanto la interfaz una vez conectada con el servidor como el inicio de sesión.

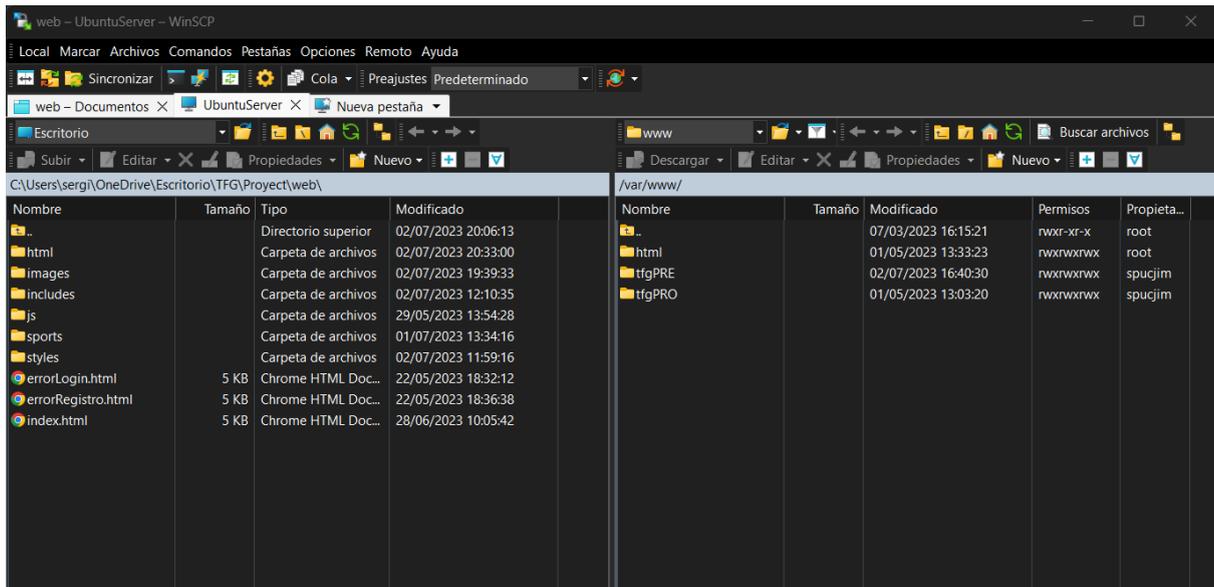


Figura 5. Herramienta WinSCP donde se muestra el escritorio local a la izquierda y el remoto a la derecha.

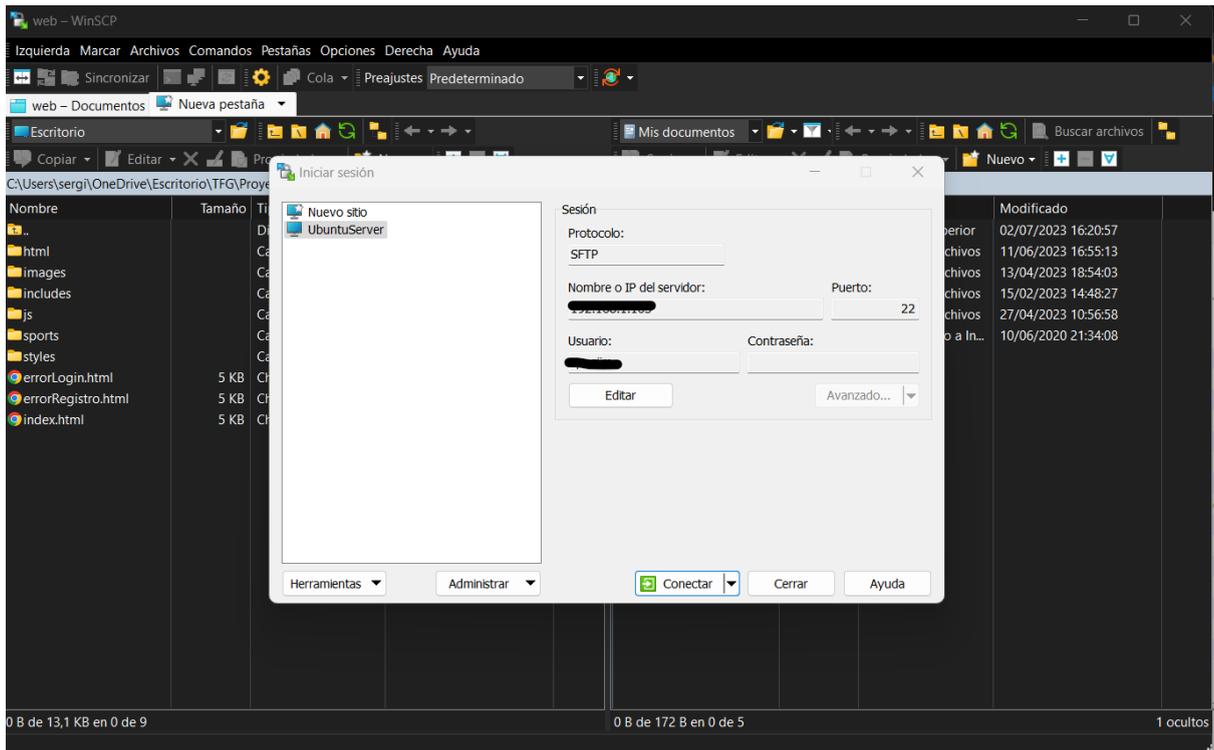


Figura 6. Herramienta WinSCP. Se muestra la ventana para poder acceder al servidor que hospeda la página.

3.9.2. HeidiSQL

HeidiSQL es una herramienta de administración de bases de datos de código abierto y gratuita que permite a los desarrolladores y administradores de bases de datos interactuar y administrar bases de datos MySQL, Microsoft SQL Server, PostgreSQL y SQLite de manera intuitiva y eficiente. Con su interfaz gráfica fácil de usar y sus numerosas características. En las figuras 7 y 8 se aprecian ambas interfaces, la de conexión previa y una vez dentro. Más adelante se hablará sobre las bases de datos.

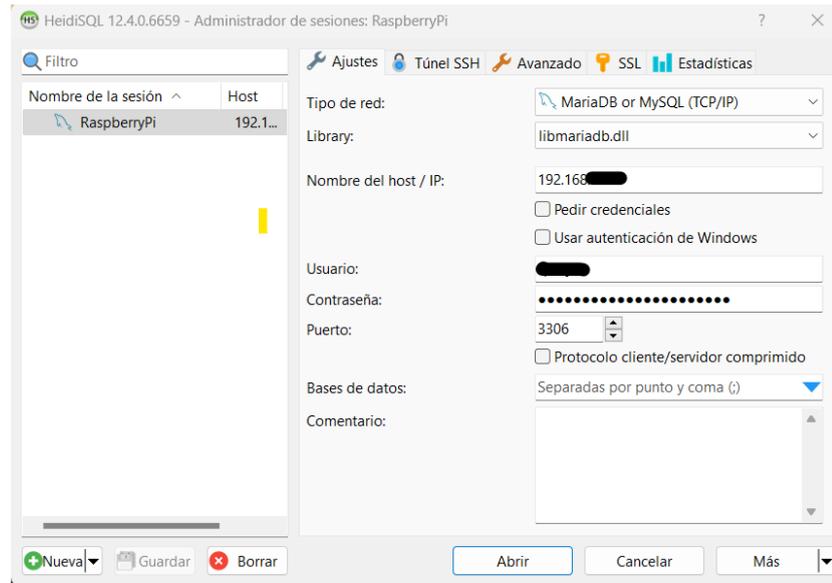


Figura 7. Herramienta HeidiSQL. Ventana con los ajustes de conexión de la base de datos

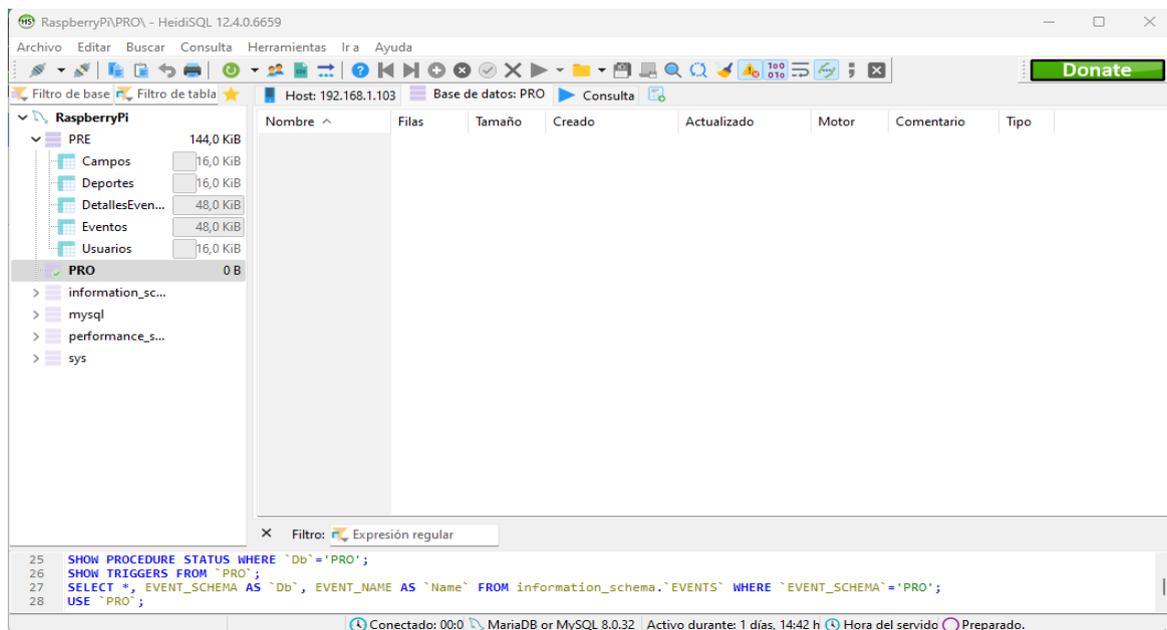


Figura 8. Herramienta HeidiSQL. Ventana donde aparecen las bases de datos, una de ellas expandida.

3.9.3. PuTTY

PuTTY es una aplicación de código abierto y gratuita que ofrece una solución de cliente de terminal y emulador de SSH (Secure Shell) para sistemas operativos Windows y Unix. Diseñado originalmente para proporcionar conectividad segura a través de SSH, PuTTY ha evolucionado para admitir una variedad de protocolos de red, como Telnet, SCP y rlogin. Es una herramienta utilizada por administradores de sistemas, desarrolladores y entusiastas de la tecnología para establecer conexiones seguras y administrar sistemas remotos. En las figuras 9 y 10 aparecen imágenes sobre la interfaz para conectarse y la terminal una vez conectado.

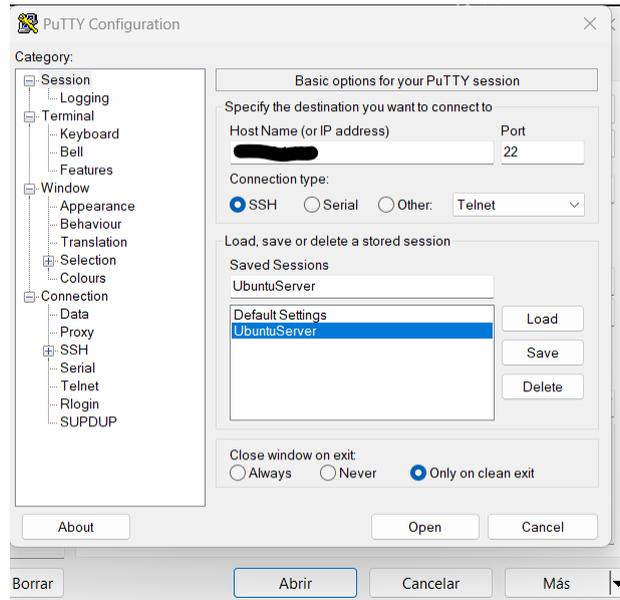


Figura 9. Herramienta PuTTY. Ventana con los ajustes de conexión al servidor.

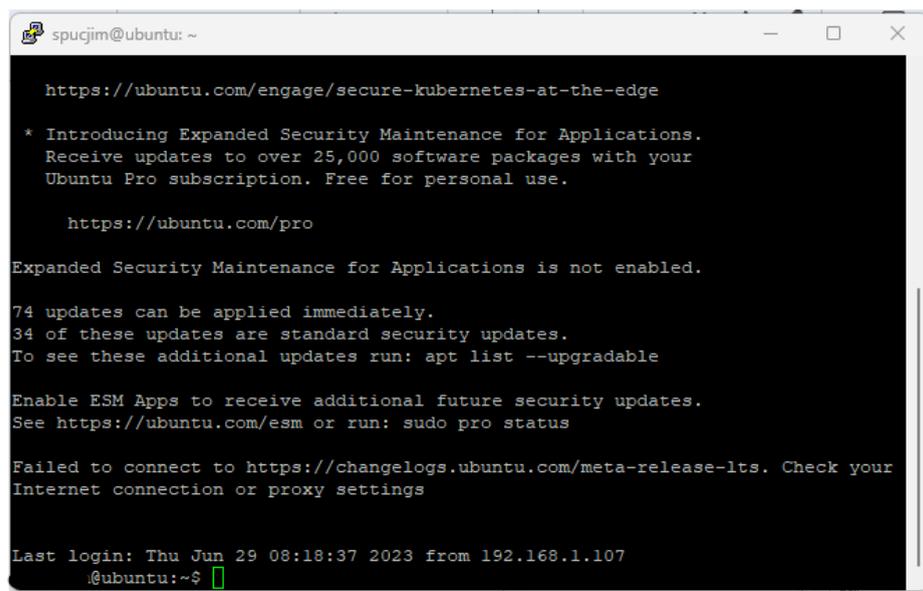


Figura 10. Herramienta PuTTY. Terminal que imprime la herramienta una vez conectado al servidor.

4. Descripción Del Proyecto Web

4.1. Introducción

Ya conocido el tema del proyecto, este apartado tratará sobre la parte técnica, tanto en el sitio del cliente como en el sitio del servidor. Por ello, es pertinente distinguir ambos espacios.

El frontend, sitio del cliente, se refiere a la parte de un sitio web o una aplicación que los usuarios interactúan directamente. Incluye todo lo que se ve y con lo que se interactúa en el navegador o la interfaz de usuario. Esto implica el diseño, la disposición de los elementos, la tipografía, los colores, las imágenes, los botones y cualquier otra cosa que veas en la pantalla. Parte que ha sido desarrollada con HTML, CSS y JavaScript.

Por otro lado, el backend, la parte del servidor, se refiere a la parte del desarrollo web que se ocupa de la lógica y la funcionalidad detrás de escena. El backend se encarga de procesar las solicitudes del usuario, realizar operaciones en la base de datos, gestionar la autenticación y la seguridad, y proporcionar los datos necesarios al frontend. Esta parte se ha desarrollado con el lenguaje de programación PHP.

Para la gestión general del proyecto, se optó por la creación de dos entornos. Un entorno llamado PRE y el otro PRO. El primero hace referencia a la preproducción, en este entorno se ha realizado la gran parte del proyecto. El segundo, a la producción, este entorno es el que estaría abierto al cliente, por lo tanto, debe salir sin ningún fallo.

La creación de ambos entornos es una práctica obligatoria cuando de desarrollo se habla. Por último, mencionar que cada entorno tiene su propia base de datos, así como su archivo de configuración dentro del servidor web haciéndolos entornos totalmente aislados el uno del otro y sin interconexión entre ambos.

Así pues, en este apartado se hablará sobre el desarrollo de la aplicación. Para ello tomará protagonismo el servidor donde se hospeda la página web. Seguidamente, la base de datos y la estructura de los directorios dentro del servidor. Por último, los ficheros de los que está compuesta, ficheros que han sido separados en *front-end* (4.4.1) y *back-end* (4.4.2).

4.2. El Servidor

El servidor elegido es concretamente una Raspberry Pi 4 Model B.

La Raspberry Pi 4 B es una placa de desarrollo de tamaño reducido. Esta placa cuenta con un procesador de cuatro núcleos a 1.5 GHz, opciones de memoria RAM que van desde 2 GB hasta 8 GB, puertos USB, conexión Ethernet, Wi-Fi integrado y una ranura para tarjetas microSD para almacenamiento. Esta se ha combinado con el sistema operativo Ubuntu.

Cuando se combina con el sistema operativo Ubuntu, la Raspberry Pi 4 B se convierte en una plataforma potente para el desarrollo web. Ubuntu es una distribución de Linux ampliamente utilizada y reconocida por su estabilidad, seguridad y facilidad de uso.

Con la Raspberry Pi 4 B y Ubuntu, puedes instalar software como Apache y PHP para crear un entorno de servidor web completo. También puedes utilizar bases de datos como MySQL para almacenar y administrar los datos de tus aplicaciones web.

Lo más relevante de dicha máquina es que ofrece un bajo consumo de energía y un tamaño compacto, lo que la hace ideal para implementaciones de desarrollo web de baja escala o proyectos personales.

Sin embargo, es importante tener en cuenta que, si bien la Raspberry Pi 4 B es lo suficientemente capaz para tareas de desarrollo web ligeras, puede tener limitaciones en cuanto a rendimiento y capacidad de procesamiento en comparación con servidores web de alta gama.

En resumen, la combinación de la Raspberry Pi 4 B con el sistema operativo Ubuntu me ha proporcionado una plataforma accesible y económica para el desarrollo web. Con ella, he podido crear y alojar mi aplicación web sin mucha complicación.

4.2.1. Apache

Para que el proyecto saliera adelante, otra de las partes indispensables era el servidor web. En mi caso, el servidor web Apache.

Apache se basa en un modelo cliente-servidor, donde el servidor escucha las solicitudes de los clientes web y les devuelve las respuestas correspondientes. Es compatible con una amplia gama de sistemas operativos dentro de estos se encuentra Linux.

La instalación del mismo se ha realizado vía terminal, usando la herramienta de PuTTY ya mencionada anteriormente.

La instalación era relativamente sencilla con los comandos que se describen a continuación.

```
sudo apt update
```

```
sudo apt install apache2
```

El primer comando sirve para poner a punto la última versión de nuestro sistema operativo, el segundo comando sirve para la instalación del apache.

```
sudo systemctl status apache2
```

Por último comprobamos el estatus de nuestro servidor web, si está todo en orden podremos conectarnos a través de nuestro navegador con la url “http://ip_del_servidor”.

El siguiente paso sería ajustar la ruta de donde coge los archivos y se los envía al cliente, para ello se modificó un archivo de configuración dentro de:

```
servidor@ubuntu:/etc/apache2/sites-available
```

Concretamente se crearon dos archivos, *pre.conf* y *pro.conf*. Archivos que hacen referencia a cada uno de los entornos que se describen anteriormente. En la figura 11 se muestra el contenido del archivo *pre.conf*. En la figura 12 se muestra el contenido del archivo *pro.conf*.

A la hora de realizar la configuración se añadió una ruta de acceso “www.pre.es” o “www.pro.es” para hacer más rápido el mismo y evitar tener que poner la dirección ip cada vez que se quisiera acceder. El único cambio adicional para su correcto funcionamiento fué la modificación de un par de líneas dentro del archivo “host” local de nuestro pc. Modificación que aparece en la figura 13.

```
spucjim@ubuntu: /etc/apache2/sites-available
GNU nano 6.2 pre.conf
VirtualHost *:80>
    ServerName pre.es
    ServerAdmin root@pre.es
    ServerAlias www.pre.es
    DocumentRoot /var/www/tfgPRE
    DirectoryIndex index.html
</VirtualHost>
```

Figura 11. Contenido del archivo “pre.conf”.

```
spucjim@ubuntu: /etc/apache2/sites-available
GNU nano 6.2 pro.conf
VirtualHost *:80>
    ServerName pro.es
    ServerAdmin root@pro.es
    ServerAlias www.pro.es
    DocumentRoot /var/www/tfgPRO
    DirectoryIndex indexPRO.html
</VirtualHost>
```

Figura 12. Contenido del archivo “pro.conf”.

```
1 # Copyright (c) 1993-2009 Microsoft Corp.
2 #
3 # This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
4 #
5 # This file contains the mappings of IP addresses to host names. Each
6 # entry should be kept on an individual line. The IP address should
7 # be placed in the first column followed by the corresponding host name.
8 # The IP address and the host name should be separated by at least one
9 # space.
10 #
11 # Additionally, comments (such as these) may be inserted on individual
12 # lines or following the machine name denoted by a '#' symbol.
13 #
14 # For example:
15 #
16 #       102.54.94.97       rhino.acme.com       # source server
17 #       38.25.63.10      x.acme.com         # x client host
18 #
19 # localhost name resolution is handled within DNS itself.
20 #   127.0.0.1       localhost
21 #   ::1             localhost
22 #
23 192.168.1.103      www.pre.es
24 192.168.1.103      www.pro.es
```

Figura 13. Contenido del fichero “host” del PC local. Son de interés las líneas 23 y 24 de este fichero

4.2.2. Base De Datos

Para la base de datos se ha utilizado MySQL, ya introducida anteriormente. Para la creación y administración de las dos bases de datos ha sido de gran ayuda la herramienta de HeidiSQL. En este apartado hablaremos sobre las bases de datos concretamente, que tablas contienen, columnas que cada una, funcionalidad, etc.

En la figura 16 aparece un esquemático de la base de datos, con flechas indicando que campos están guardan relación entre sí. En los siguientes párrafos se procede a hablar sobre el motivo de cada tabla así como su contenido. En las explicaciones de las tablas se hablará sobre los campos que las componen además de detallar el nombre del campo, formato de datos y una breve descripción.

La tabla “Usuarios” está compuesta por 4 columnas. Su función es guardar los datos de los usuarios que se registran:

- IdUser, autonumérico, el identificativo el cual se usará para hacer las operaciones pertinentes en la aplicación. Esto es así porque se reduce considerablemente la carga algorítmica al buscar un número en vez de un campo de texto o cualquier otra cosa.
- Nick, texto, guarda el nombre con el que quiere ser identificado el cliente dentro de la aplicación
- Correo, texto, correo electrónico del cliente.
- Pwd, texto, contraseña del cliente.

La tabla “Campos”, compuesta por 2 columnas. Es una tabla sencilla, cuyo propósito es almacenar el reflejo de las instalaciones deportivas que existen. Destacar que esta tabla no varía en base a la aplicación.

- IdCampo, autonumérico, el identificativo que se usará para hacer las operaciones pertinentes en la base de datos. El motivo por el cual se usa este es el mismo que en la tabla “Usuarios”.
- Nombre, texto, nombre que se le dan a los diferentes campos de los que se disponen, en la figura 14 aparece los datos que hay en esta tabla.

La tabla “Deportes”, formada por 4 columnas. Al igual que en la tabla de “Campos”, esta tabla plasma el abanico de deportes del que se dispone. En la figura 15 se ve dicha tabla con sus datos, al igual que la tabla anterior, esta no varía con la aplicación.

- IdDeporte, autonumérico, identificativo usado para, como anteriormente se ha hecho, facilitar los procesos de la base de datos y así optimizar su rendimiento.
- Nombre, texto, nombre del deporte para que la administración de dicha tabla sea más sencilla.
- PlazasMin, numérico (int), es el número de plazas mínimo que se requiere para poder realizar dicho deporte, este valor existe con la misión de que la aplicación reconozca cuando debe avisar a los participantes de que el evento se va a realizar.
- PlazasMax, numérico (int), número máximo de plazas por deporte. Se necesita para evitar una sobrepoblación en un evento concreto y que sea inviable jugarlo, salvo excepciones, se ha estipulado una media de 5 cambios por equipo. Las excepciones se recogen en deportes como por ejemplo tenis.

La tabla “Eventos”, tabla con mayor densidad, recogiendo en ella 9 columnas de datos y siendo la tabla que más entradas tenga. Esta tabla es gestionada por los administradores de la aplicación, pues estos son los responsables de crear eventos.

- IdEvento, autonumérico, identificativo del evento en cuestión.
- Lleno, bit, su valor oscila en función de si se ha llegado a las plazas máximas del deporte el cual está relacionado con ese evento en concreto. Esto sirve para avisar a la aplicación y hacer los cambios pertinentes.
- Finalizado, bit, su valor oscila en función del campo “Fecha” y “HoraFin” dentro de esta misma tabla. Gracias a esto el programa puede fácilmente liberar datos en caso de que se requiera con una simple consulta.
- Fecha, date, fecha del evento.
- HoraInicio, time, hora de inicio del evento.
- HoraFin, time, hora a la que concluye el evento.
- IdCampo, numérico (int), relación con la tabla “Campos”. El objetivo es que en un evento quede reflejado el lugar donde se efectuará el mismo, así se evitan solapamientos entre deportes que compartan campo.
- IdDeporte, numérico (int), relación con la tabla “Deportes”. En función del deporte que sea, se precisa una disposición u otra de jugadores, código en *front-end* y número de plazas. Por ello, en cada evento debe haber constancia del deporte.
- Plazas, numérico (int), número inicialmente a cero, a cada registro de un cliente, aumentará en una plazas. Este campo se va comparando con el número de plazas mínimo y máximo de cada deporte para realizar las operaciones pertinentes.

La tabla “DetallesEventos”, tabla complementaria para relacionar un evento con un jugador, de este modo, cada vez que un jugador se inscriba a un evento se le dará uso a esta tabla. Así se reduce notablemente el tamaño de la tabla anterior. Esta tabla está compuesta por 4 campos los cuales son modificados enteramente por el cliente a través del *front-end* de la aplicación.

- IdEvento, numérico (int), relación con la tabla “Eventos”.
- IdUser, numérico (int), relación con la tabla “Usuarios”.
- Posición, numérico (int), posición elegida por el usuario en cada deporte. Dependiendo del deporte estarán disponibles unas posiciones u otras, incluyendo posiciones de reserva o no. Más adelante se detallarán estos puntos.

- Nivel, numérico (int), nivel elegido por el usuario en cada evento. Este nivel va del uno al cinco, siendo el uno el número más bajo. Más adelante se detallará la importancia de este valor.

RaspberryPi(PRE\Campos) - HeidiSQL 12.4.0.6659

Host: 192.168.1.103 Base de datos: PRE Tabla: Campos

PRE.Campos: 15 filas en total (aproximadamente)

IdCampo	Nombre
1	futbolCesped1
2	futbolCesped2
3	pabellon1
4	pabellon2
5	tenis1
6	tenis2
7	tenis3
8	padel1
9	padel2
10	padel3
11	padel4
12	squash1
13	squash2
14	voley1
15	voleyPlaya1

```

37 SELECT * FROM information_schema.KEY_COLUMN_USAGE WHERE TABLE_SCHEMA='PRE' AND TABLE_NAME='Campos' AND REFERENCED_TABLE_NAME IS NOT NULL;
38 SHOW CREATE TABLE `PRE`.`Campos`;
39 SELECT tc.CONSTRAINT_NAME, cc.CHECK_CLAUSE FROM `information_schema`.`CHECK_CONSTRAINTS` AS cc, `information_schema`.`TABLE_CONSTRAINTS` AS t;
40 SELECT `IdCampo`, LEFT(`Nombre`, 256) FROM `PRE`.`Campos` LIMIT 1000;
  
```

Figura 14. Contenido de la tabla “Campos” dentro de las bases de datos “PRE” y “PRO”.

RaspberryPi(PRE\Deportes) - HeidiSQL 12.4.0.6659

Host: 192.168.1.103 Base de datos: PRE Tabla: Deportes

PRE.Deportes: 9 filas en total (aproximadamente)

IdDeporte	Nombre	PlazasMin	PlazasMax
1	baloncesto	10	20
2	balonmano	14	24
3	futbolSala	10	20
4	futbol7	14	24
5	padel	4	4
7	squash	2	2
8	tenis	4	4
9	voleibol	12	20
10	voleyPlaya	4	8

```

48 SHOW CREATE TABLE `PRE`.`Campos`;
49 SELECT `IdCampo`, LEFT(`Nombre`, 256) FROM `PRE`.`Campos` LIMIT 1000;
50 SHOW CREATE TABLE `PRE`.`Deportes`;
51 SELECT `IdDeporte`, LEFT(`Nombre`, 256), `PlazasMin`, `PlazasMax` FROM `PRE`.`Deportes` LIMIT 1000;
  
```

Figura 15 . Contenido de la tabla “Deportes” dentro de las bases de datos “PRE” y “PRO”.

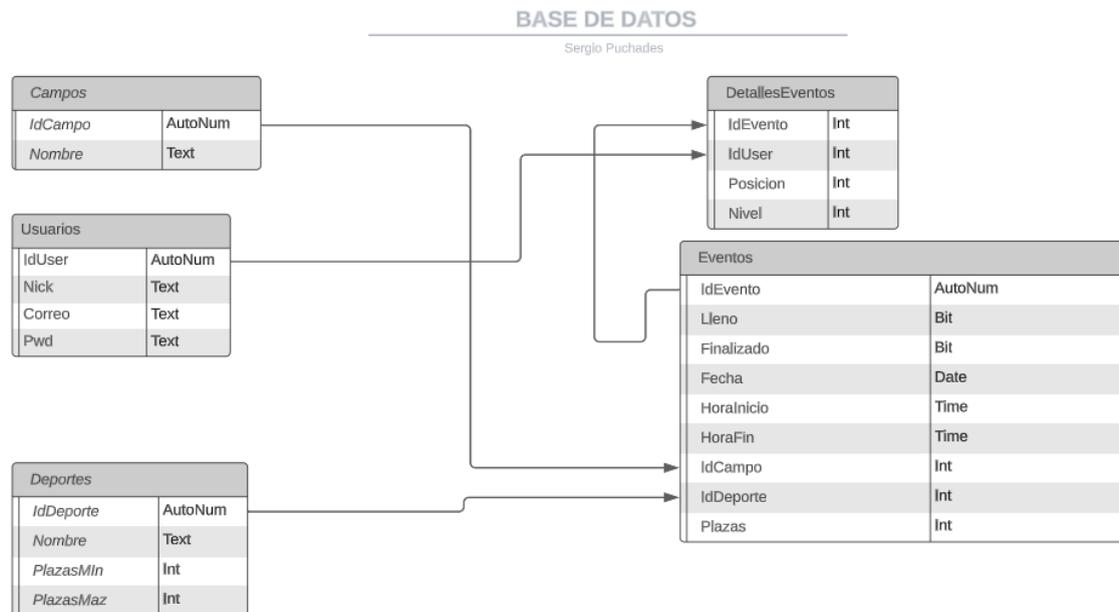


Figura 16. Esquemático de la base de datos.

4.3. Estructura de directorios

Un punto fundamental a la hora de desarrollar correctamente algo es tener una estructura de directorios sólida y flexible, de tal forma que nos permita ir agregando los archivos necesarios sin necesidad de hacer grandes cambios en un futuro. En ese apartado se detallará dicha estructura explicando en que consiste cada uno de ellos.

Dentro de nuestros entornos de prueba, la estructura de carpetas es exactamente la misma, el único cambio es el nombre de la carpeta donde el servidor apache apunta para coger los archivos.

La estructura de carpetas está expuesta en la figura 17. Como hemos mencionado antes, la diferencia entre los entornos estaría en cambiar la ruta donde apunta. La ruta del entorno de pruebas sería “/var/www/tfgPRE”. Mientras que en el entorno de producción apuntaría a “/var/www/tfgPRO”.

Nada más entrar en dichas carpetas, nos encontramos tres archivos fuera, estos son:

- index.html. Es la página de registro y de inicio de sesión, esta es el archivo por defecto que coge nuestro servidor web para mostrar al navegador.
- errorLogin.html. Página que se muestra en caso de no haber realizado el inicio de sesión correctamente por culpa del cliente.
- errorRegistro.html. Página que se muestra después de no haber tenido un registro de nuevo usuario exitoso por culpa del cliente.

A continuación, aparecen las siguientes carpetas:

- html. Carpeta donde se almacenan los archivos donde existe contenido html que se le envía a la parte de cliente.
- images. Carpeta encargada de almacenar las todas las imágenes que vamos a utilizar.
- includes. Carpeta donde únicamente aparecen archivos php. Esta carpeta es el centro logístico de toda la aplicación. Dentro de esta hay archivos encargados de recibir y enviar datos, sin texto html que mostrar por pantalla.
- js. Carpeta donde se almacenan los archivos de JavaScript para la parte estética de la página web.
- sports. Carpeta donde se almacenan archivos con extensión php, en esta encontramos las diferentes páginas que nos muestran el formulario para reservar tu posición y nivel dentro de un evento. Existe un archivo diferente para cada deporte, adecuándose a las necesidades del mismo.
- styles. Carpeta donde se almacenan los archivos con extensión CSS. Archivos responsables del estilo y pulcritud de la página.

Nombre	Tamaño	Modificado	Permisos	Propiet...
..		01/05/2023 13:33:23		
html		02/07/2023 16:40:31		
images		04/07/2023 19:37:16		
includes		02/07/2023 16:40:31		
js		01/07/2023 13:35:11		
sports		01/07/2023 13:35:11		
styles		02/07/2023 16:40:31		
errorLogin.html	5 KB	22/05/2023 18:32:12		
errorRegistro.html	5 KB	22/05/2023 18:36:38		
index.html	5 KB	28/06/2023 10:05:42		

Figura 17. Estructura de los directorios dentro del servidor.

4.4. Desarrollo de la aplicación

En los siguientes apartados se atacará al desarrollo del proyecto. Aquí se recogerán todos los ficheros de código que forman parte del proyecto junto a una breve explicación sobre el desempeño que tienen en la aplicación.

Se vuelve a remarcar la importancia de separar estos ficheros en el *front-end* y *back-end* puesto que dependiendo de donde actúen, tendrán unas pautas de actuación u otras, esta separación quedará reflejada en los puntos 4.5.1 y 4.5.2.

4.4.1. Front-end

En este apartado abordaremos todos aquellos archivos enfocados a la parte del cliente dentro del proyecto. Estos archivos se identifican a nivel de extensión, o bien su extensión es HTML o, por otro lado es un archivo PHP el cual imprime código HTML en función de las demandas del cliente. La figura 53 es un flujograma que representa el salto entre archivos.

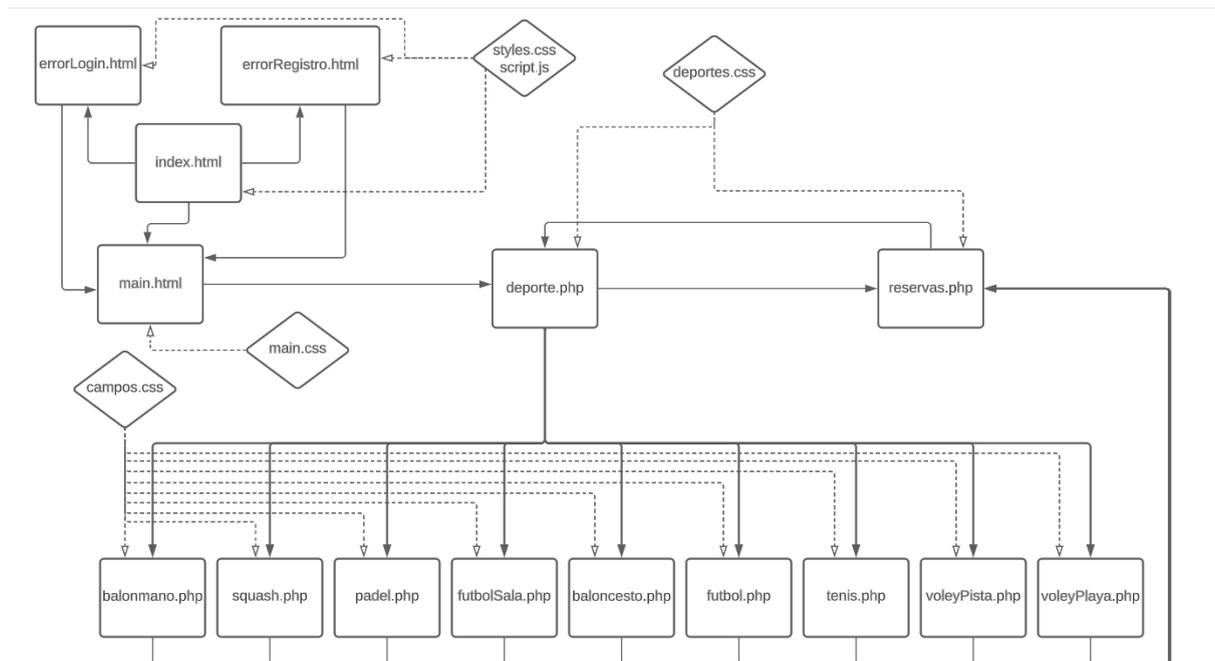


Figura 53. Flujograma que muestra las acciones entre archivos y de qué archivos cogen los estilos.

Hablaremos primeramente de los archivos que se encuentran fuera de cualquier carpeta, estos archivos son “index.html”, “errorLogin.html”, “errorRegistro.html”:

- index.html. Primer archivo de código HTML que se le envía al cliente, en este está contenido la página de inicio de sesión y de registro. Dentro del mismo se llaman a dos archivos más dedicados a la estética. Los archivos son “style.css” y “script.js”.
- errorLogin.html. Archivo muy similar al de “index.html” se le envía al cliente si el servidor identifica una incidencia por culpa del cliente en el momento de iniciar sesión. Llama también a los archivos “style.css” y “script.js”.
- errorRegistro.html. Archivo muy similar al de “index.html” se le envía al cliente si el servidor identifica una incidencia por culpa del cliente en el momento de registrarse. Llama también a los archivos “style.css” y “script.js”.

La primera carpeta que se describe anteriormente en esta memoria se nombra como “html”, aquí encontraremos archivos o bien con extensión HTML o archivos PHP que imprimen código HTML. Ambos por y para el cliente. Dentro de esta carpeta aparecen los siguientes ficheros:

- main.html. Página que muestra todos los deportes disponibles tal y como aparecen en la figura 26. Esta página toma toda su estética llamando a “main.css” archivo el cual se detallará más adelante. Hablando más en profundidad de la misma, cada deporte representa un formulario que envía a “deporte.php” un valor diferente en función del deporte. Gracias a este valor el archivo “deporte.php” sabe que peticiones debe realizar a la base de datos y que elementos cargar por pantalla al cliente.
- deporte.php. La misión de esta página es presentarle al cliente todos los eventos disponibles que tiene el deporte que ha seleccionado previamente. Esto se logra gracias a una petición “SELECT” a la base de datos, una vez nos devuelve la información deseada, se pasa a variables de PHP para que puedan ser tratadas. Un punto importante de esto es que, una vez tenemos los datos en variables, se libera la consulta para evitar consumir memoria de manera innecesaria. Dentro del mismo se llama a “deportes.css” para la parte estética. Por último, en el mismo archivo encontramos incrustado un script de JavaScript trabajando junto al archivo CSS.
- reservas.php. Como su propio nombre indica este es el archivo responsable de imprimir en el navegador las reservas que tiene el cliente que ha iniciado sesión. Esto lo hace gracias a unas variables de sesión que se instancian cuando entras a la aplicación. Variables las cuales se hablará más adelante, en el apartado dedicado al lado del servidor. Dentro del mismo se llama a “deportes.css” para la parte estética. Por último, como en el archivo anterior, también encontramos incrustado un script de JavaScript trabajando junto al archivo CSS.

Toma protagonismo la carpeta donde se almacenan todos los archivos que cargan los diferentes campos y posiciones en función de la elección del cliente. Esta es la carpeta de “sports”. Encontraremos pues un archivo por cada deporte que esté disponible, un total de nueve archivos, “baloncesto.php”, “balonmano.php”, “futbol.php”, “futbolSala.php”, “padel.php”, “squash.php”, “tenis.php”, “voleyPista.php” y por último “voleyPlaya.php”.

Los archivos comparten la misma lógica y contenido, lo único que los diferencia es la imagen del campo que cargan, la colocación en el navegador de las posiciones y el número de las mismas. Para poder hacer estos archivos más estéticos se llama a un CSS, “deportes.css”. Dentro de todos y cada uno de los archivos hay dos funciones de JavaScript incrustadas con la misión de guardar el valor que el cliente ha seleccionado, tanto de la posición en el campo como del nivel. Así pues, introduce estos valores en un formulario y lo envía al PHP de *back-end* llamado “procesarReserva.php”.

Llegando ya al final del apartado, queda mencionar a todos estos archivos que hacen posible una experiencia agradable a la vista. Primeramente, la carpeta “js” que contiene un archivo

JavaScript llamado “script.js” el cual ya se ha mencionado previamente, su misión en proporcionar animaciones a dichas páginas.

La segunda carpeta es llamada “styles”, en ella se recogen todos los archivos CSS que serán llamados como anteriormente se ha descrito. Estos archivos se componen de clases que luego en el HTML son reconocidas con la etiqueta “class=’Nombre_de_la_clase’ ”. Los archivos son; “campos.css”, “deportes.css”, “main.css”, “style.css”.

Por último la carpeta de “images” en ella están todas y cada una de las imágenes que la página web puede cargar desde el fondo y el logo hasta los campos que de cada uno de los deportes.

4.4.2. Back-end

Los archivos dedicados a la parte de servidor son aquellos que, primeramente no imprimen ningún código HTML y, por tanto son archivos que el cliente no va a tener acceso ni saber de su existencia.

Debido a la naturaleza del proyecto, hay dos bloques de procesos en la parte de servidor. El primer bloque se ocupa de la entrada y salida de un cliente a la página. Esto se consigue gracias a cuatro archivos PHP cuyo flujo de vida se aprecia en la figura 54.

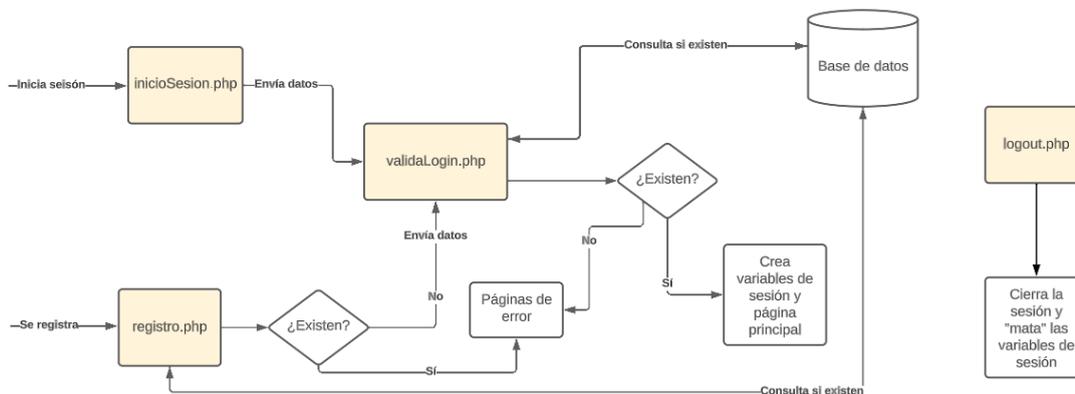


Figura 54. Flujograma que muestra la logística del proyecto en la parte del servidor en los inicios o registros de sesión.

Los archivos implicados están marcados en color dentro de la figura 54. Como se aprecia estos son “inicioSesion.php”, “logout.php”, “registro.php”, “validaLogin.php” los cuales se encuentran dentro de la carpeta llamada “includes”.

- inicioSesion.php. Archivo que recoge los datos del inicio de sesión, los transforma en variables y las escapa, al escapar las variables evitamos posibles ataques de “SQL injection” sobre información tan sensible como son las credenciales de un usuario y los envía a “validaLogin.php”.

- registro.php. Al igual que el archivo anterior, recoge datos, esta vez de un registro nuevo, transforma los datos a variables y los escapa. Sin embargo antes de enviarle la información a “validaLogin.php” hace una llamada a la base de datos para comprobar que las credenciales de correo electrónico y nombre de usuario no coinciden con ninguna si esto es así y, por lo tanto, la consulta vuelve vacía, envía los datos a “validaLogin.html”, en caso de que si existan, redirige al cliente a “errorRegistro.html”.
- validaLogin.php. El artífice de la entrada a la web, este archivo PHP recibe la información correspondiente de los ficheros anteriores y, con una consulta a la base de datos comprueba si estas credenciales son válidas. En caso de ser así, transforma las variables que ha recibido en variables de sesión junto a el “IdUsuario”, también transformado en una variable de sesión y envía al cliente a la página principal. En caso de no ser así, redirige al cliente a la página de “errorLogin.html”.
- logout.php. Archivo mediante el cual se “mata” una sesión iniciada por el cliente, cerrando las variables de sesión que previamente se habían abierto. Se pone en funcionamiento cuando el cliente clicca sobre el botón de “Cerrar Sesión” que muestra la figura 24. Este archivo redirige al cliente automáticamente a la página de inicio de sesión.

El segundo y último bloque corresponde a la gestión de las reservas. Esta gestión la lleva un solo archivo PHP llamado “procesarReserva.php”, a rasgos generales, la figura 55 representa mediante un flujograma el trabajo que realiza el mismo.

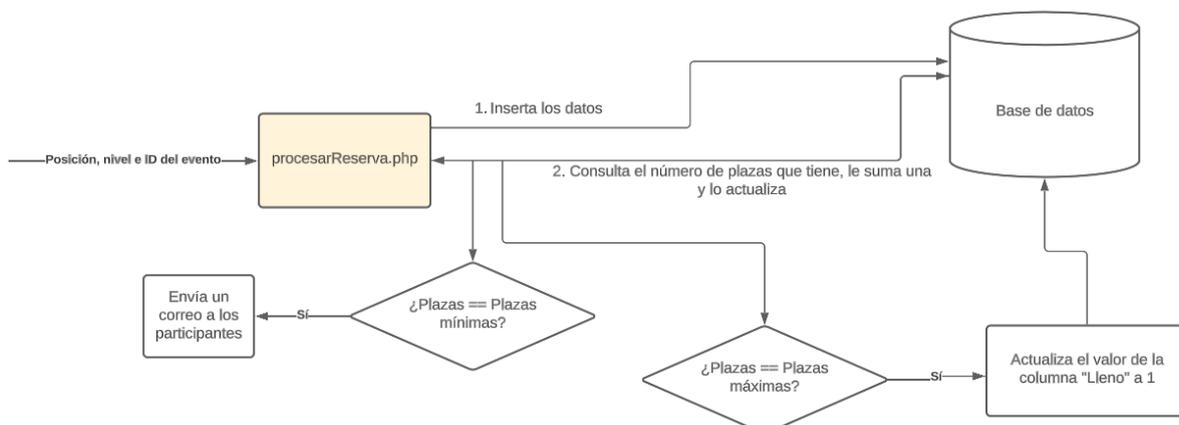


Figura 55. Flujograma que muestra la logística del proyecto en la parte del servidor las reservas de plazas.

El archivo “procesarReserva.php” primeramente recibe los datos de la posición, el nivel y el identificador del evento, este último se le pasa mediante la url. Seguidamente los transforma en variables y junto a la variable de sesión del “IdUsuario” que se instanció previamente. Escapa las variables y los añade a las tablas correspondientes en la base de datos.



Después de esto, realiza una consulta para conocer el número de personas inscritas al evento, y las plazas máximas y mínimas del deporte al cual está vinculado el evento. Suma uno al número de plazas que tiene el evento y lo actualiza en la base de datos. Una vez hecho esto realiza dos comparaciones. La primera es si las plazas actuales son iguales a las mínimas, si esta sentencia se cumple realiza otra consulta pidiendo los correos de todos los usuarios afectados por el evento y les envía un correo informativo diciéndoles que el evento se va a realizar, en el correo añade la fecha, hora y campo dónde se realizará dicho evento. Si la sentencia no se cumple, no realiza nada.

La segunda compara el número de plazas actuales con las máximas del deporte, si estos valores coinciden actualizará el campo de “Lleno” dentro de la tabla “Eventos” y lo pondrá a uno. El administrador podrá recoger esta información para saber los eventos que se van a realizar y los que no.

5. Funcionalidad de la aplicación

Como se ha descrito anteriormente, esta es la parte responsable de la explicación funcional del proyecto. Aquí se recogerán todas las opciones de las que dispone tanto los clientes como el administrador de la página.

Es importante enfatizar la división entre la parte de cliente y la de administrador a la que los puntos 5.1 y 5.2 corresponden respectivamente.

5.1. Introducción

Como se ha descrito anteriormente, esta es la parte responsable de la explicación funcional del proyecto. Aquí se recogerán todas las opciones de las que dispone tanto los clientes como el administrador de la página.

Es importante enfatizar la división entre la parte de cliente y la de administrador a la que los puntos 5.1 y 5.2 corresponden respectivamente.

5.2. El cliente

La experiencia de un cliente debe ser interactiva a la par que intuitiva. De un vistazo se debería entender cada una de las funciones en las diferentes etapas por las que pasa el mismo dentro de la página web. Por ello, se ha optado por la transmisión de dicha información a través de imágenes descriptivas en lugar de textos. Para empezar a entrar en detalle, la figura 18 representa en un flujograma el ciclo de vida por el que pasa un cliente en la aplicación.

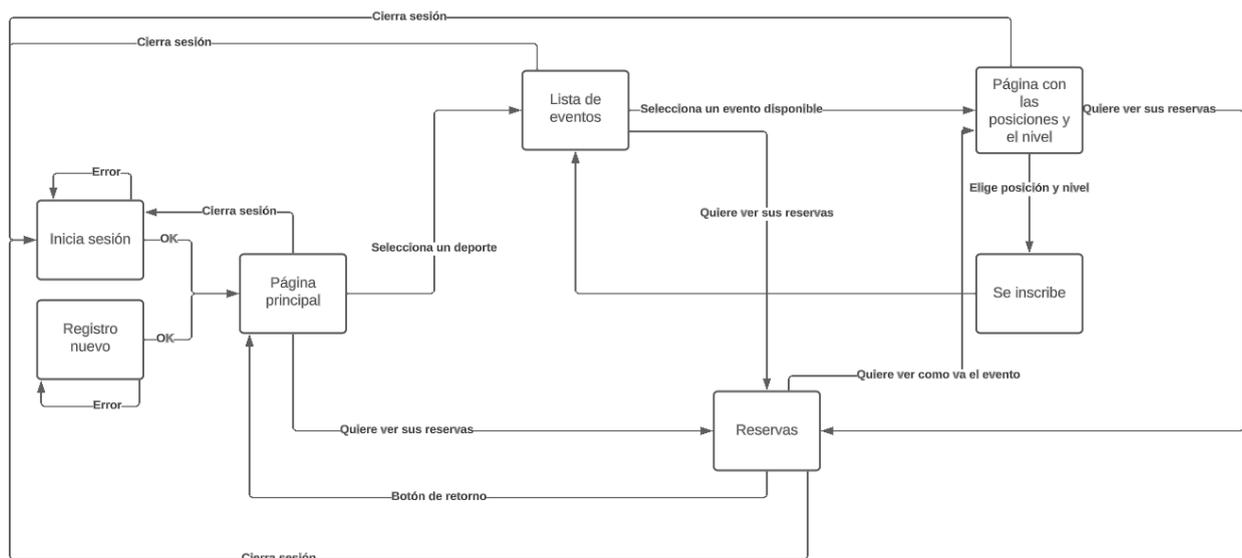
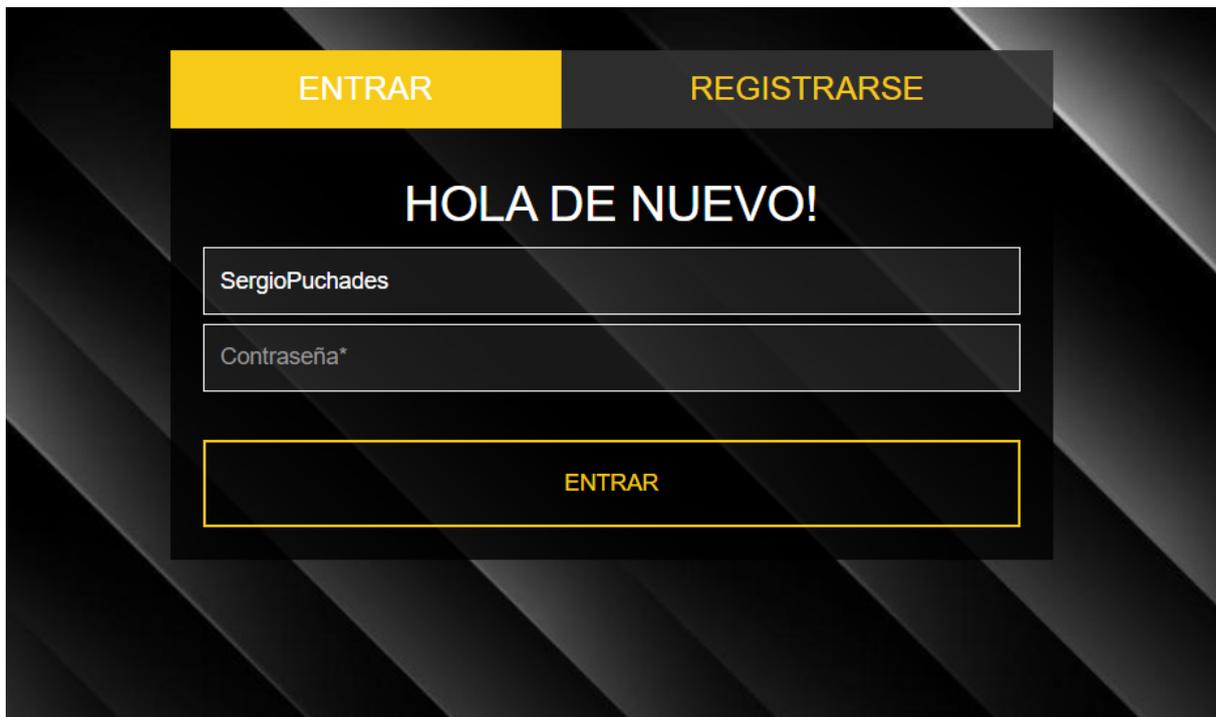


Figura 18. Flujograma sobre el ciclo que recorre un cliente en la aplicación.

Lo primero que debe hacer un cliente al ingresar a la página será iniciar sesión si ya posee una cuenta o, registrarse en caso de que no sea así. En caso de ya tener un usuario, solo deberá ingresar el nombre con el que se registró y su contraseña, tal y como se muestra en la figura 19. Si requiere de un usuario nuevo, deberá introducir el nombre con el que quiere registrarse, un correo electrónico y una contraseña tal y como se muestra en la figura 20.



The image shows a login form on a dark background with diagonal stripes. At the top, there are two buttons: 'ENTRAR' (highlighted in yellow) and 'REGISTRARSE'. Below these is the text 'HOLA DE NUEVO!'. There are two input fields: the first contains the text 'SergioPuchades' and the second is labeled 'Contraseña*'. At the bottom, there is a large yellow-outlined button labeled 'ENTRAR'.

Figura 19. Página de inicio de sesión.



The image shows a registration form on a dark background. At the top, there are two buttons: 'ENTRAR' (grey) and 'REGISTRARSE' (yellow). Below them is the heading 'REGISTRATE, ES GRATIS!'. The form consists of three input fields: 'Usuario*' (short), 'Correo electrónico' (long), and 'Contraseña*' (long). At the bottom is a yellow button labeled 'CREAR USUARIO'.

Figura 20. Página para registrarse como nuevo usuario.

También existe la posibilidad de que el cliente cometa algún error, en el inicio de sesión. Esto se le mostrará por pantalla y no podrá acceder a la página hasta que ponga sus credenciales correctamente. Figura 21.



The image shows a login interface with a dark background and yellow accents. At the top, there are two buttons: 'ENTRAR' (highlighted in yellow) and 'REGISTRARSE'. Below these, the text 'HOLA DE NUEVO!' is displayed in large white letters, followed by the error message 'Usuario o correo electrónico incorrectos' in red. There are two input fields: 'Nombre de Usuario*' and 'Contraseña*'. At the bottom, there is a large yellow-outlined button labeled 'ENTRAR'.

Figura 21. Página que muestra un error ocurrido por parte del en el inicio de sesión.

Si a la hora de registrarse el cliente desea hacerlo con un correo o nombre de usuario que ya existen, se le notificará también para que proceda a cambiarlos. Hasta que no se registre con un nombre y correo electrónico nuevos no podrá acceder. Figura 22.



Figura 22. Página que muestra un error ocurrido por parte del en el inicio de sesión.

Una vez el cliente se ha iniciado sesión, aparece en la página principal de la aplicación, sin embargo, antes de pasar a las siguientes páginas, tomaremos un tiempo para hablar de la barra superior que tiene la página. En esta barra superior aparece el logo de la página, presente en la figura 23, y tres textos clicables autodescriptivos, presentes en la figura 24. El primero, “Deportes”, en caso de no estar en la página de los deportes, aparecerá este botón para poder ir a ella. Después, “Reservas” como su nombre indica, enviará al cliente a una página donde se muestren los eventos a los que está inscrito, página de la que se hablará más adelante. El segundo enlace, “Cerrar Sesión” llevará al cliente a la página de inicio de sesión después de cerrar correctamente su sesión. Esta barra superior, mostrada en la figura 25, aparecerá a lo largo de las siguientes páginas para que la experiencia de navegación del cliente sea lo más fluida y flexible posible.

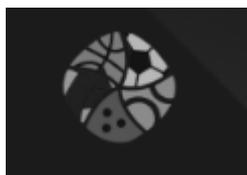


Figura 23. Logo

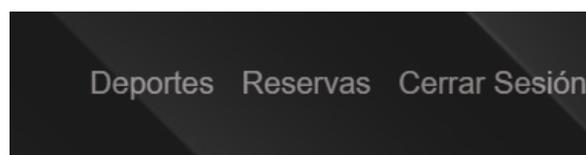


Figura 24. Botones de navegación

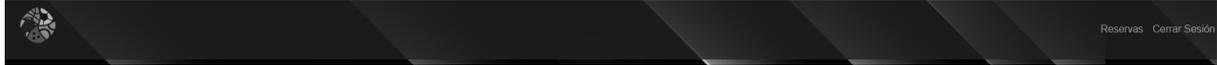


Figura 25. Barra superior con el logo y los botones de navegación.

La página principal está compuesta por una disposición de bloques equidistantes entre si, como se aprecia en la figura 26. Cada uno de estos bloques tiene en su interior una imagen descriptiva con el tipo de deporte al que hace referencia, además, gracias tecnologías como JavaScript, al deslizar el ratón por encima de los mismos sufren una animación de “hover” y se iluminan para hacer la experiencia de navegación más interactiva. Dicha animación se aprecia en las figuras 27 y 28. Por último, aparecen las figuras de la 29 a la 36 mostrando los iconos restantes en detalle que están a disposición del cliente.



Figura 26. Página donde se muestran todos los deportes disponibles.



Figura 27. Ratón fuera del icono. Fútbol 7.

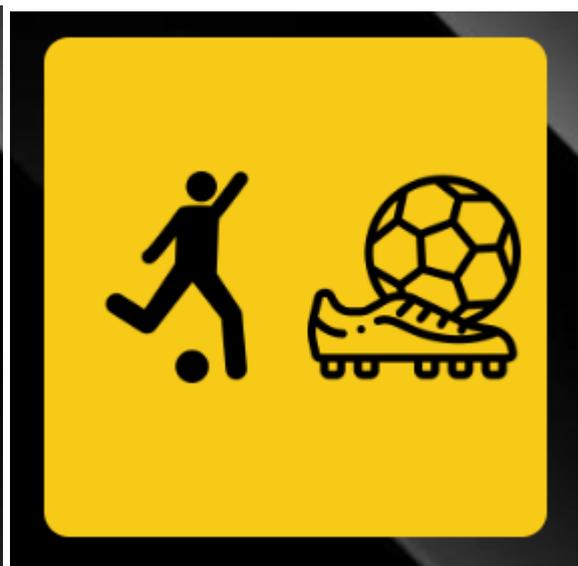


Figura 28. Ratón encima del icono.



Figura 29. Baloncesto.



Figura 30. Fútbol Sala.



Figura 31. Pádel.

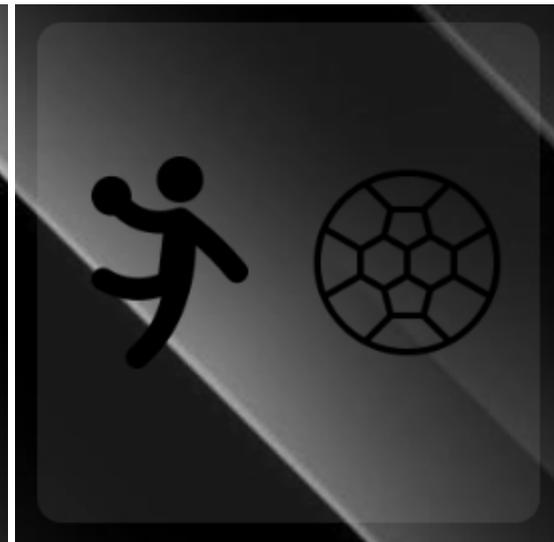


Figura 32. Balonmano.



Figura 33. Squash.



Figura 34. Tennis.



Figura 35. Voleibol playa.



Figura 36. Voleibol Pista.

Una vez el cliente un deporte pinchando sobre el icono correspondiente la aplicación le enviará una pagina con 5 columnas, correspondientes a los días de la semana, excluyendo fines de semana, vea la figura 37. Dentro de cada una de las columnas aparecerán tantos bloques como eventos hayan ese día. Los bloques contendrán información sobre la fecha del evento, la hora y el número de plazas que quedan disponibles tal y como aparece en la figura 38.

Es importante aclarar que en cada día de la semana aparecen eventos tanto de la misma semana como de semanas futuras. Esta decisión en el diseño se tomó para evitar que el cliente tuviera que ir navegando de una semana a a otra. De esta forma, con un simple vistazo el cliente puede ver todos los eventos que, por ejemplo están disponibles los lunes.

Por último, así como los botones de selección de deporte, estos botones de selección de evento comparten la misma animación, apreciada en la figura 39.

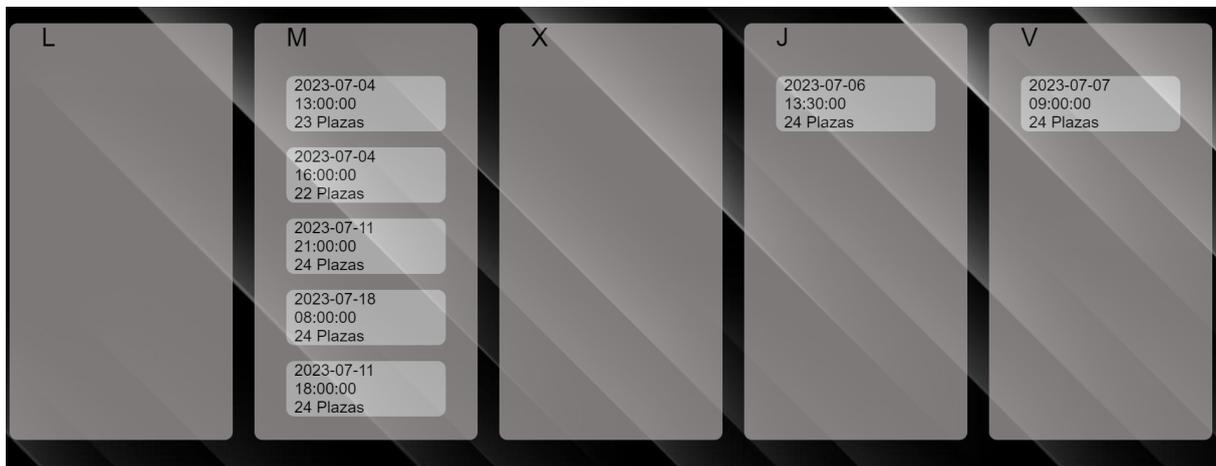


Figura 37. Página que muestra los eventos disponibles en función del deporte seleccionado.

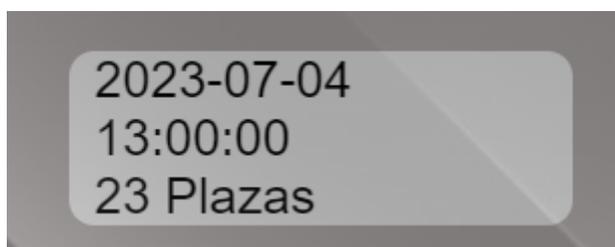


Figura 38. Evento sin el ratón por encima

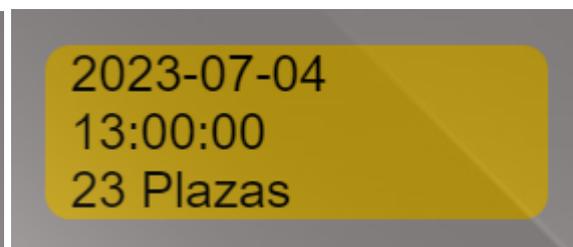


Figura 39. Evento con el ratón por encima

Una vez el cliente haya seleccionado un evento en concreto, la página web le mostrará el campo del deporte correspondiente, unos círculos representarán las posiciones, estos, en función del deporte, adoptarán un lugar u otro en el espacio, estos campos con los círculos mencionados se pueden apreciar viendo desde la figura 40 hasta la 48. Además de esto, el cliente deberá seleccionar su nivel de destreza según su propia percepción visible en la figura 49.

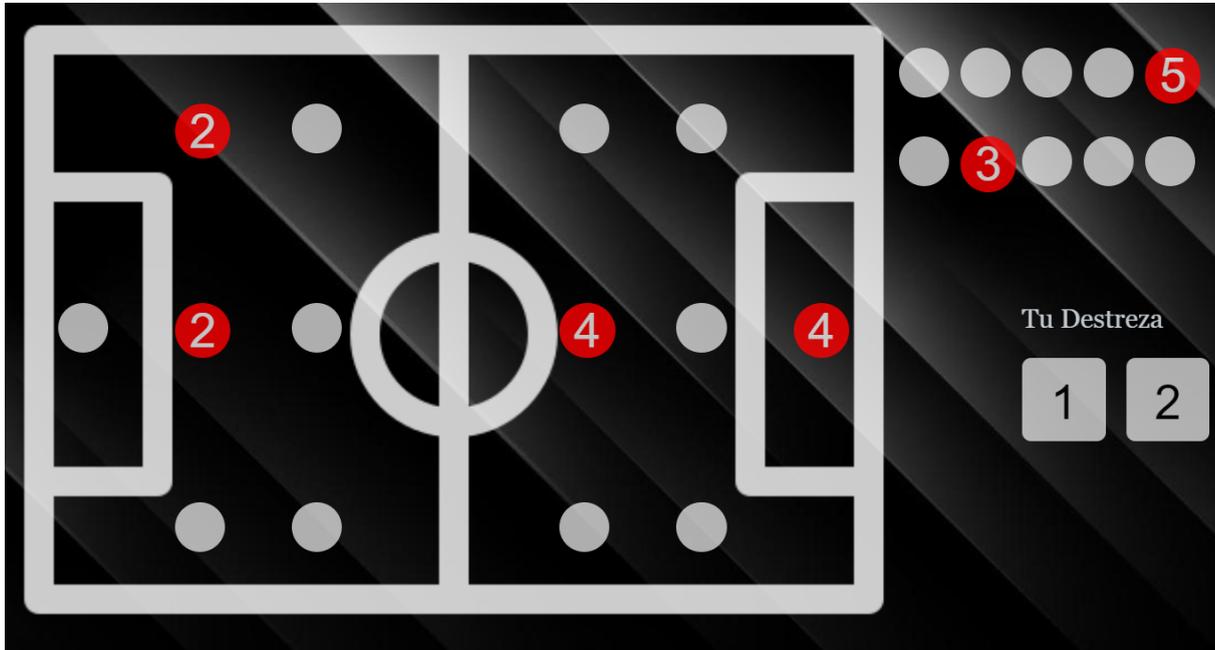


Figura 40. Evento de fútbol 7.

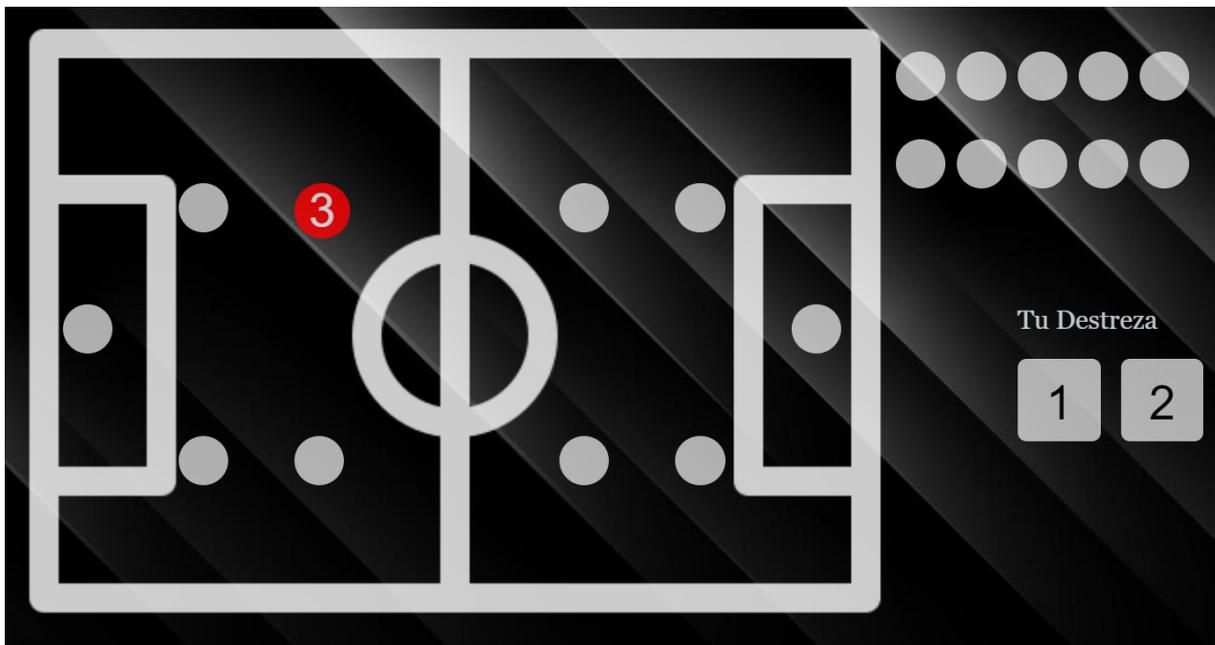


Figura 41. Evento de fútbol sala.

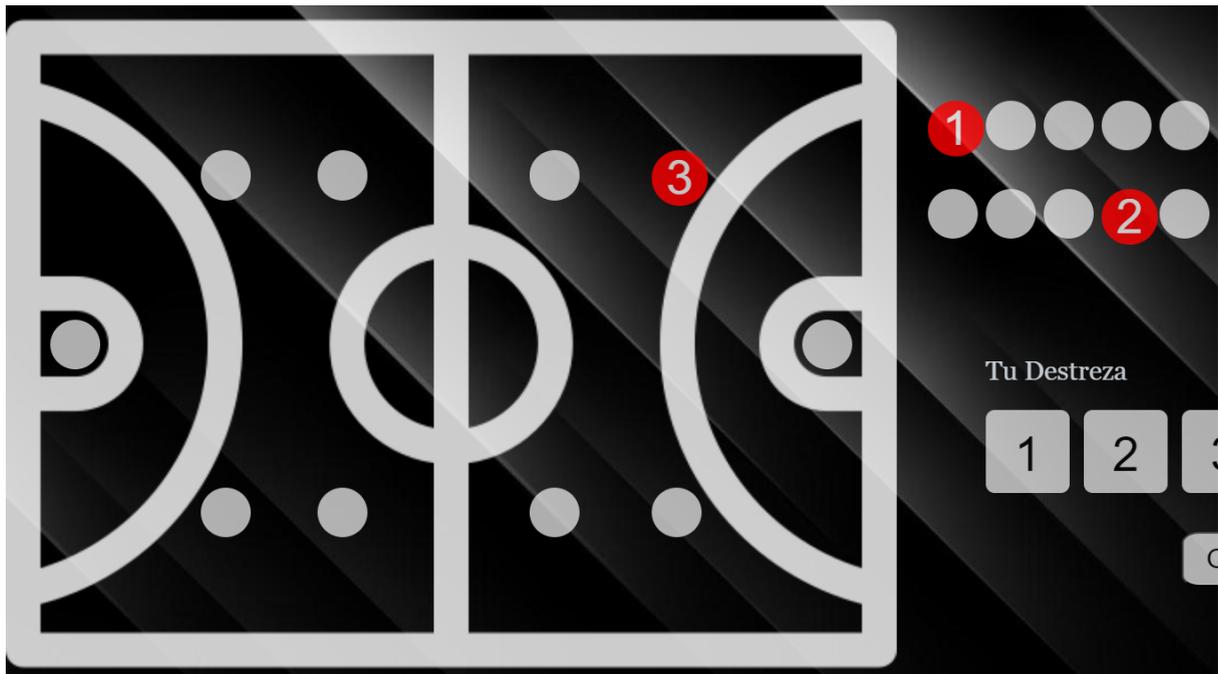


Figura 42. Evento de baloncesto.

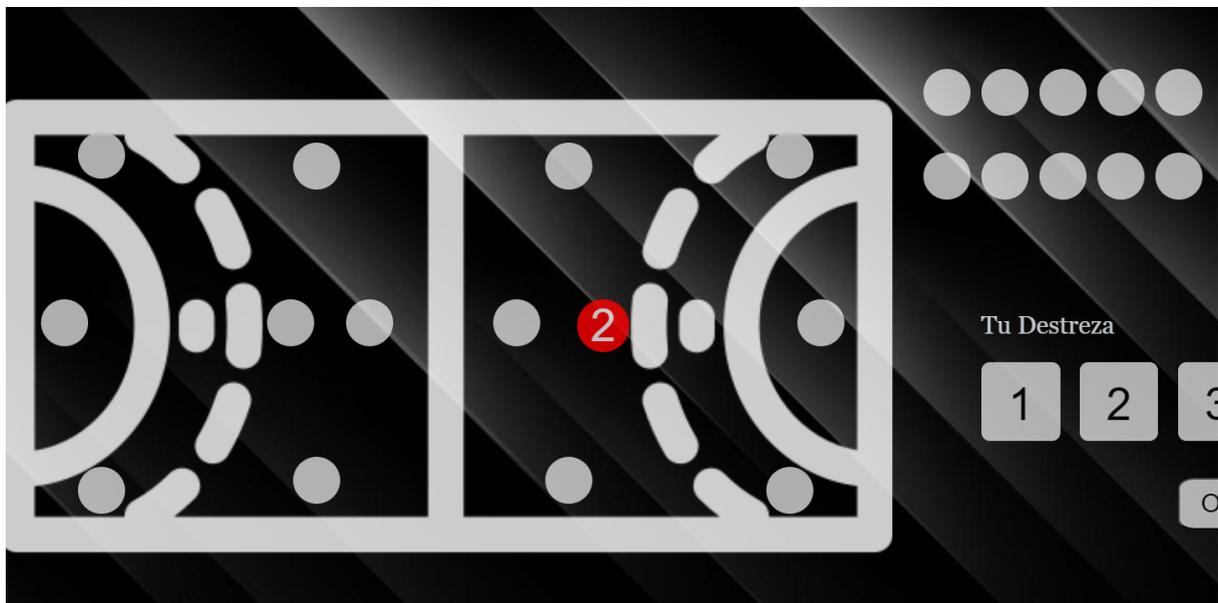


Figura 43. Evento de balonmano.

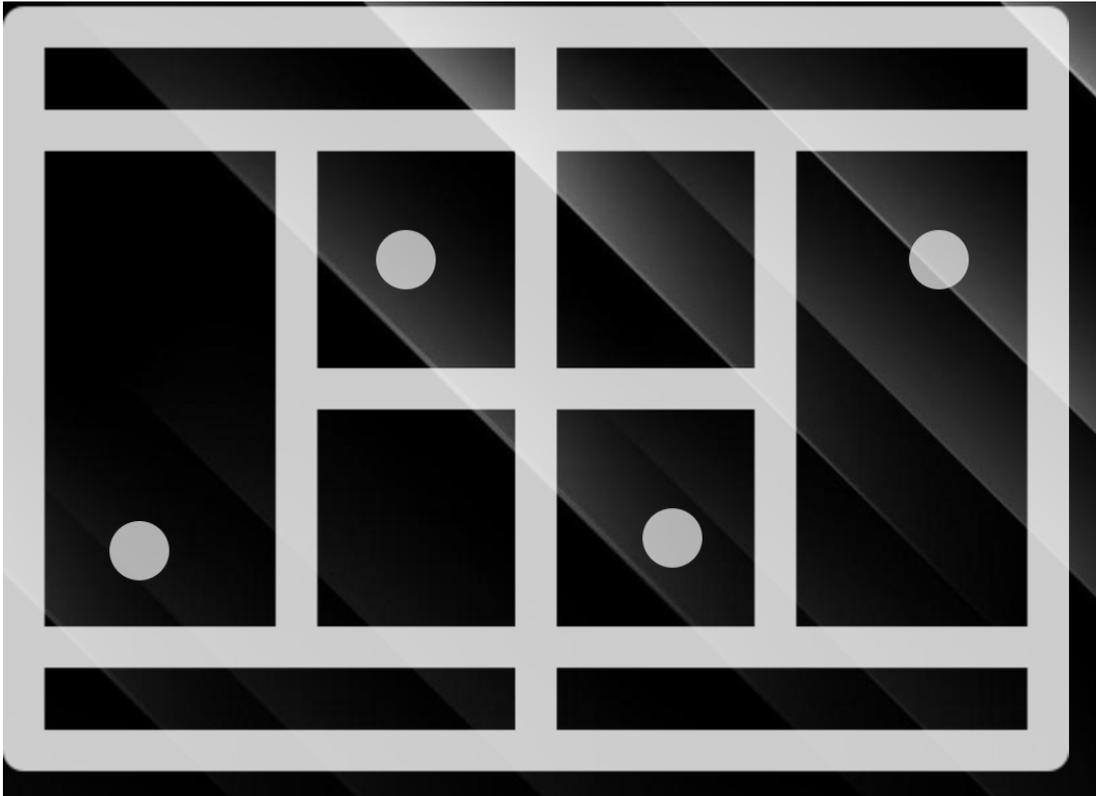


Figura 44. Evento de padel.

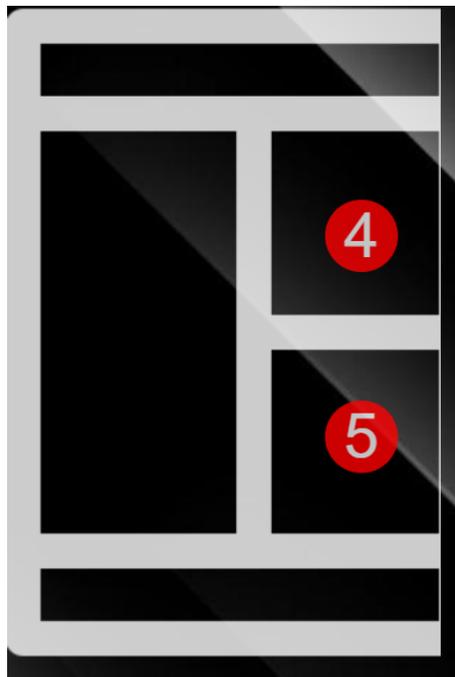


Figura 45. Evento de squash.

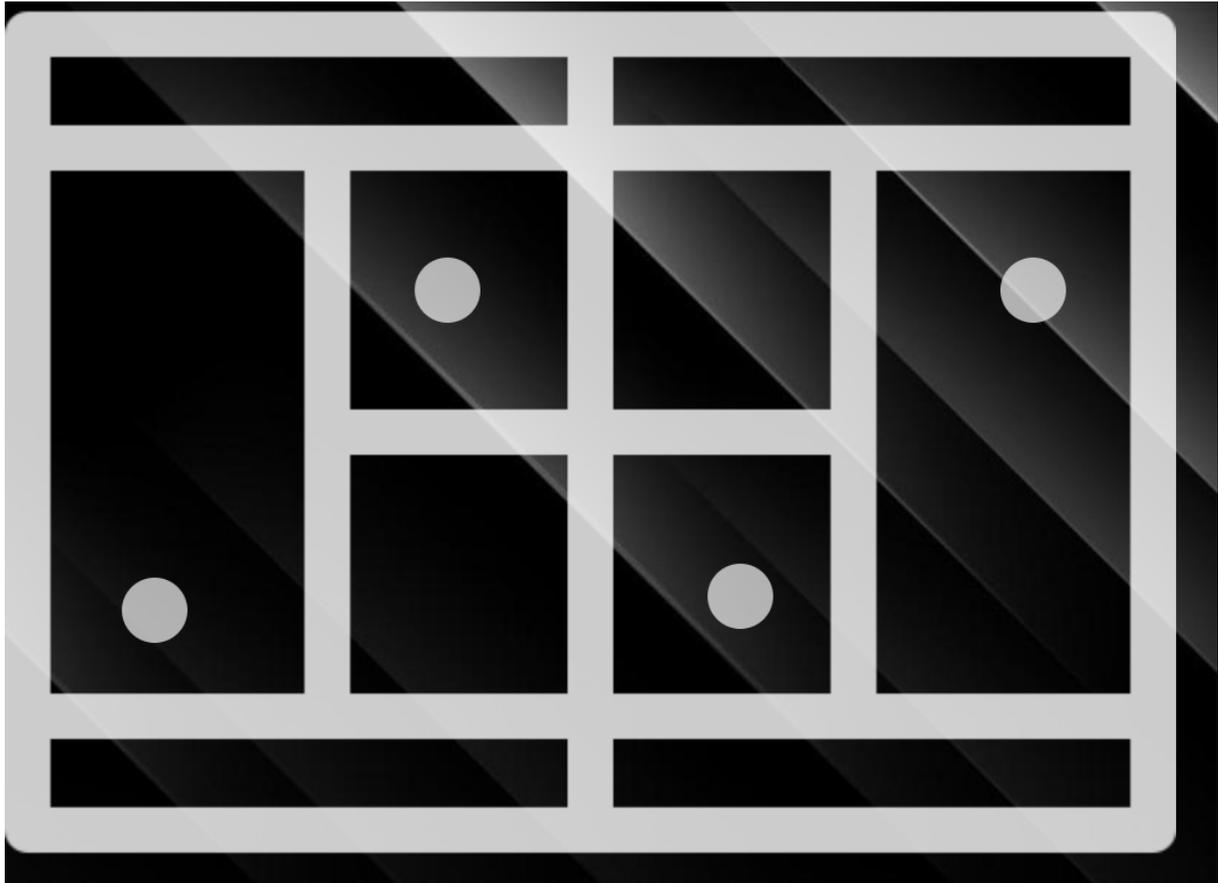


Figura 46. Evento de tenis.

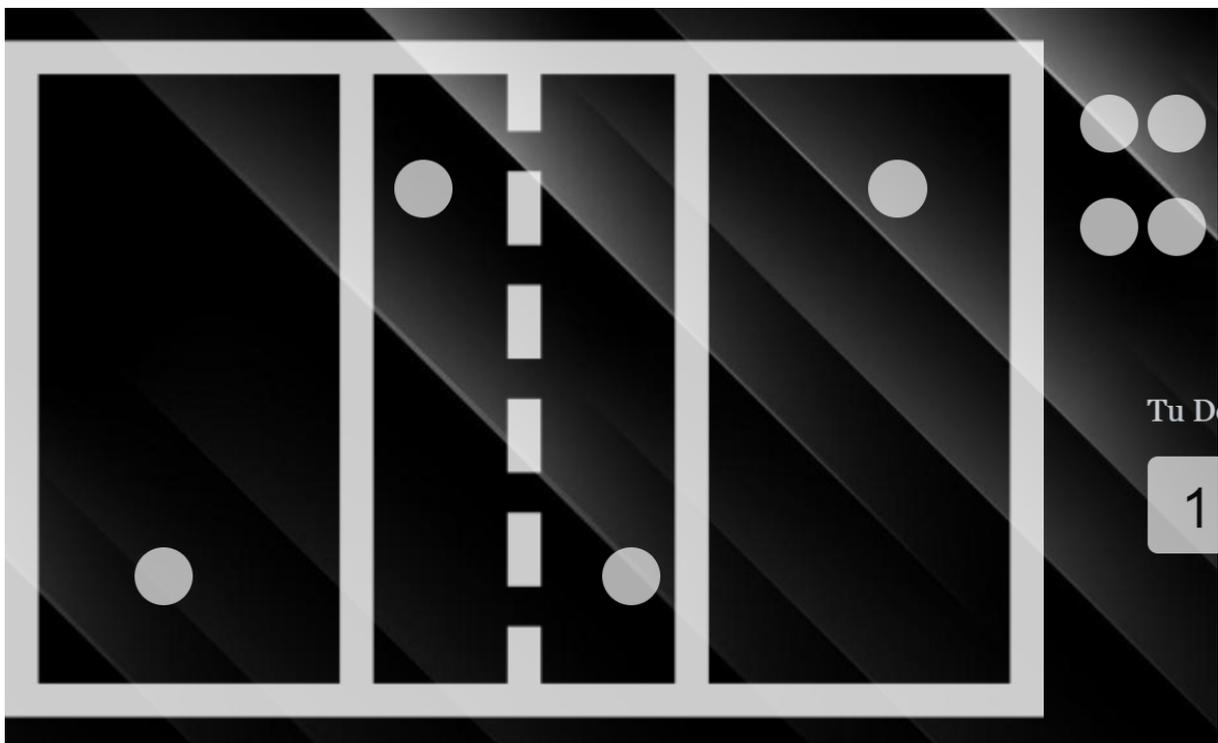


Figura 47. Evento de voleibol playa.

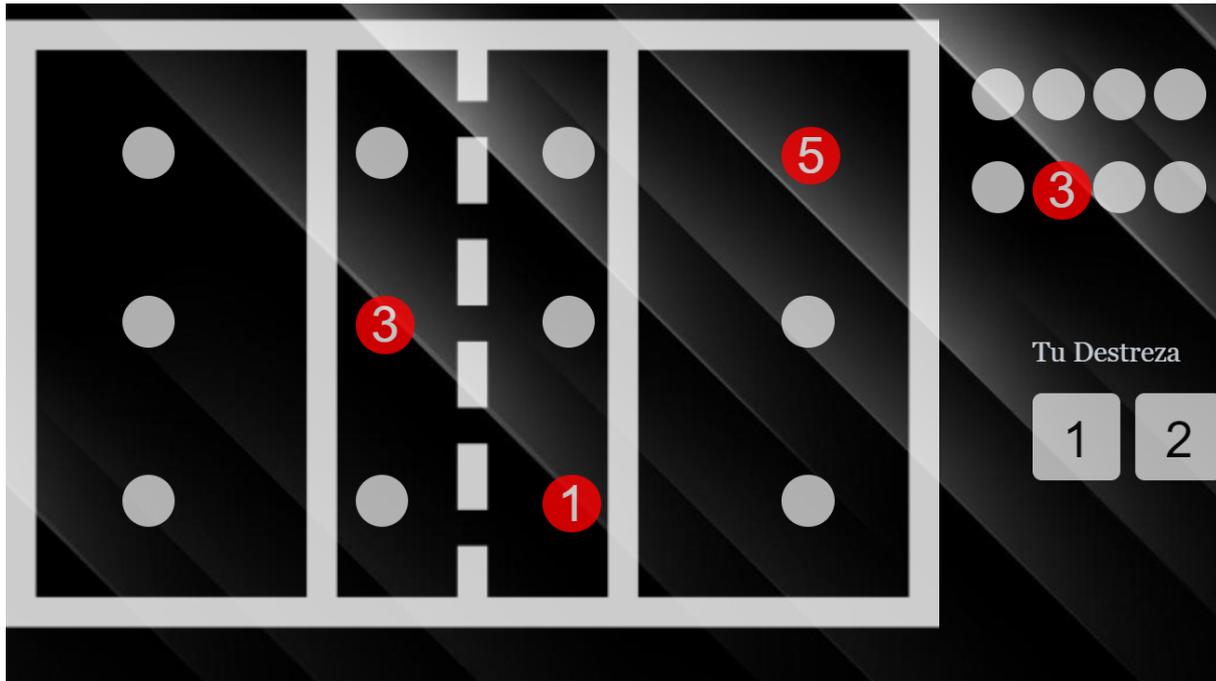


Figura 48. Evento de voleibol pista.

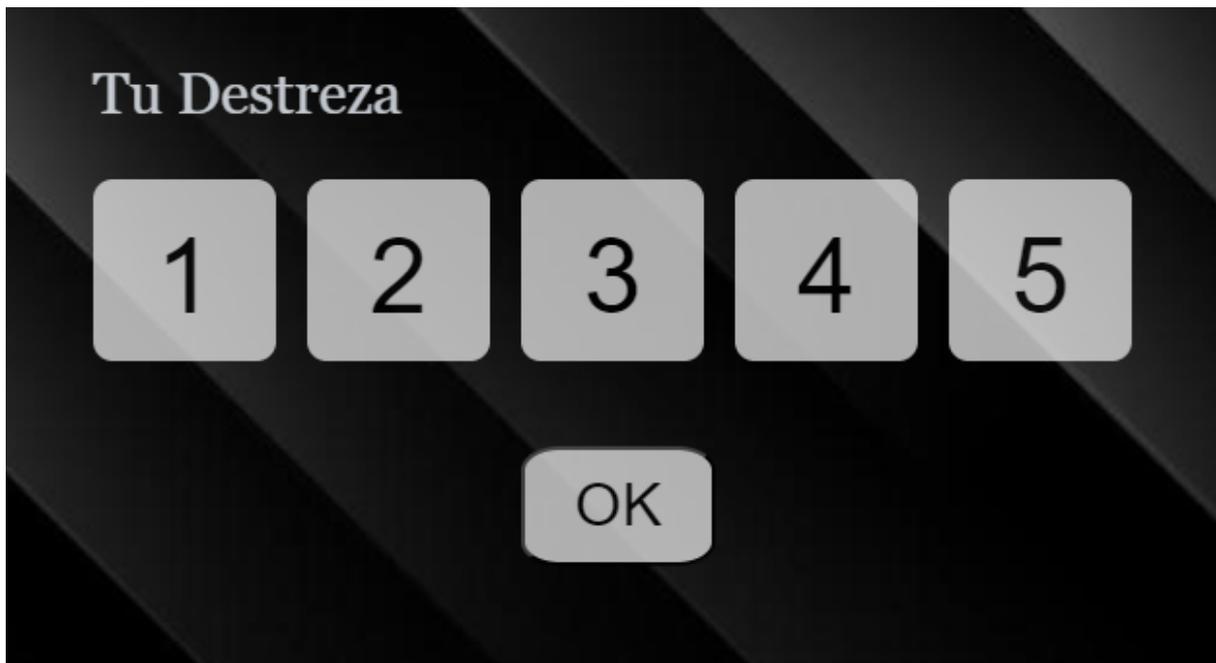


Figura 49. Selección de nivel y botón de “OK”.

En el formulario para reservar tu plaza solo se podrá elegir una posición y un nivel. De esta manera, cuando el cliente pulse el botón de “OK”, figura 49, se guardará su plaza automáticamente y será redirigido a la página donde se procede a la selección del deporte.

Lo más importante de este sistema es que el cliente puede ver a tiempo real que posiciones están seleccionadas y cual es el nivel de la persona que figura en cada posición. En las figuras mencionadas se pueden apreciar dichas plazas ya cubiertas pues están le aparecerán al cliente en rojo con un número en su interior, que indicará el nivel, además estas posiciones no serán seleccionables por el cliente.

En la figura 50 podemos ver un ejemplo desde un cliente que quiere inscribirse a un evento, seleccionando su posición y nivel, con el ratón encima del botón de “OK” para enviar la solicitud.

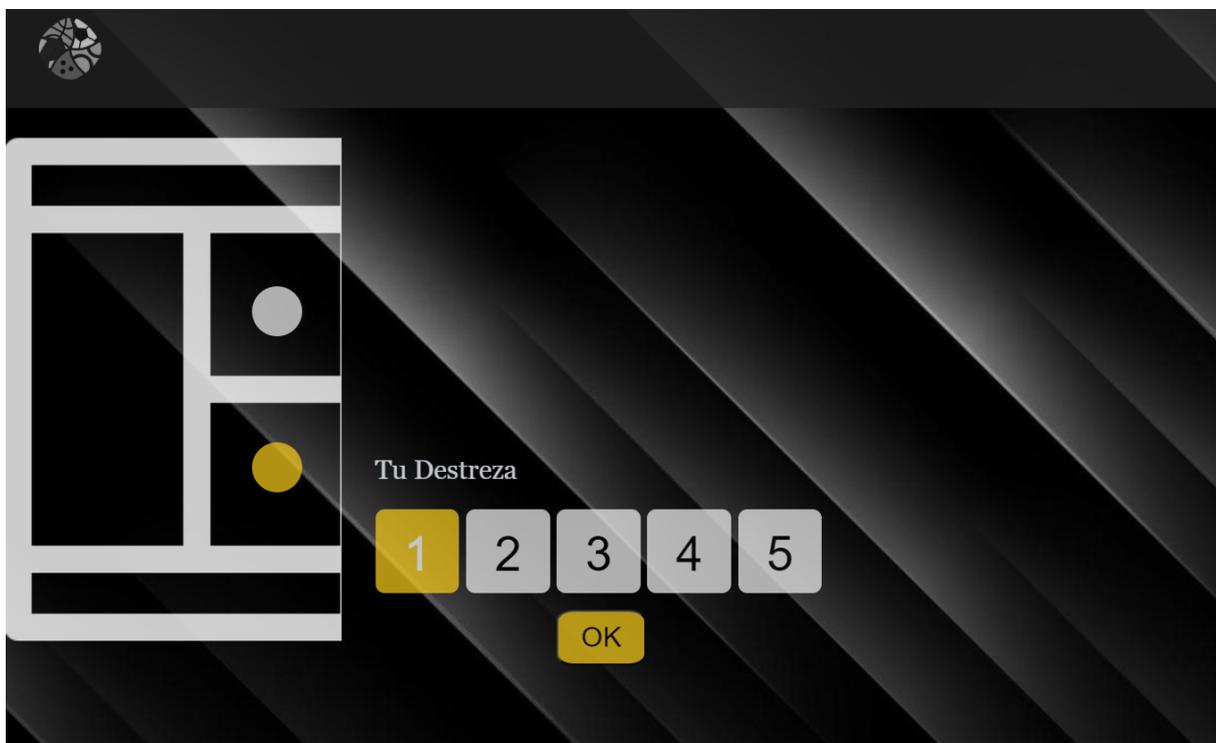


Figura 50. Vista del cliente a la hora de querer reservar una posición y poner su destreza.

Una vez el cliente ha enviado el formulario para reservar su plaza, se le guardará en su apartado de reservas. Único para cada cliente, es el apartado donde aparecen todos y cada uno de los eventos a los que el cliente se ha inscrito. Estéticamente es similar al apartado de eventos, mostrado en la figura 37, sin embargo ahora en cada bloque de evento se mostrará adicionalmente el tipo de deporte. Con esto conseguimos primeramente que el cliente por un lado, al reconocer la estructura le sea más sencillo encontrar un evento en concreto y, por otro, conocer el deporte al cual pertenece el evento en vez de tener que clicar sobre el mismo para saberlo.

Como muestra la figura 51, aquí aparecería una página de reservas de un cliente. Caben mencionar varias cosas, primeramente, al seguir la misma estética que en la figura 37, las animaciones y colores se mantienen y, segundo, con el objetivo de mejorar la experiencia en la aplicación, el cliente puede clicar sobre cualquier evento de sus reservas, provocando que la página lo lleve al evento en cuestión. De esta forma puede ver como se va completando el evento.

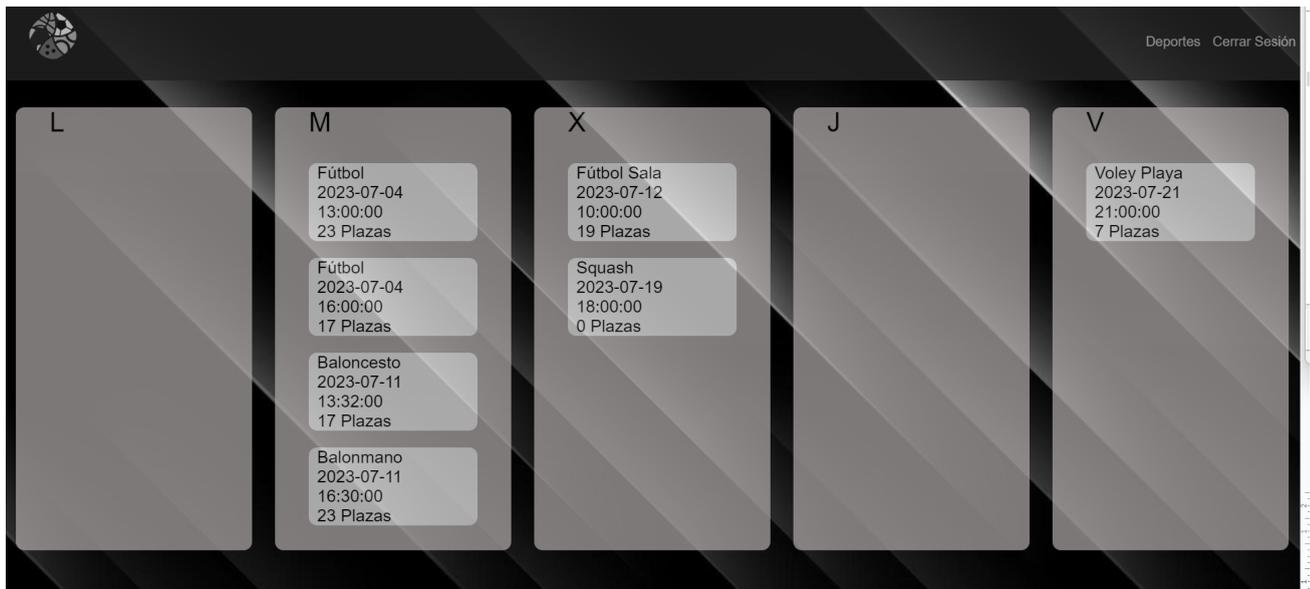


Figura 51. Página que muestra las reservas del cliente que haya iniciado sesión.

Con esto concluye la experiencia de un cliente a través de la aplicación web. A modo de alerta, la propia aplicación se encarga de enviar un correo a los inscritos en los eventos una vez se completen las plazas mínimas para que dicho evento pueda llevarse a cabo. En el correo figura el día, la hora y el tipo de deporte al que pertenece el evento.

5.3. El administrador

Este apartado está dedicado a la misión que tiene el administrador de la página. Debido a la naturaleza de la aplicación, son pocas y breves las tareas que debe realizar.

El administrador de la página tiene acceso al código fuente, permisos de administrador en directorios que afecten a la aplicación y, por supuesto a la base de datos.

A nivel de código, es responsable del mantenimiento de la aplicación, como la solución de problemas o la creación de actualizaciones.

Enlazando con lo anterior, debe tener acceso a los directorios para pasar los archivos pertinentes de cualquier cambio ya sea de adición o eliminación.

Por último, lo que será su función principal y le ocupará la mayor parte del tiempo, la gestión de la base de datos. Dentro de sus funciones, desempeñará tareas administrativas con las credenciales de los usuarios, pudiendo eliminar las mismas, crear o modificar las mismas. Sin embargo, su misión fundamental será la creación de eventos deportivos. Será el artífice que dicte cuando y dónde se harán los eventos realizando un seguimiento de los mismos y asegurándose de su correcta evolución. Como se ha mencionado anteriormente, dentro de la base de datos existe una tabla llamada “Eventos” a la que le corresponde una columna de tipo bit llamada “Finalizado”, será tarea del administrador, con el fin de liberar espacio en la base de datos, de eliminar dichos eventos ya finalizados. Esto se realiza de manera manual por si surge cualquier incidente posterior al evento y se ha de tomar mano de la información del mismo, si se eliminara mediante un automatismo podría ocurrir un incidente y no ser posible identificar a las personas inscritas al mismo por ejemplo. Tal y como se muestra en la figura 52, siendo esta una imagen de la tabla “Eventos” y exponiéndose en la misma los campos que debe rellenar dicho administrador.

IdEvento	Lleno	Finalizado	Fecha	HoraInicio	HoraFin	IdCampo	IdDeporte	Plazas
8	0	1	2023-07-04	13:00:00	14:00:00	1	4	1
9	1	1	2023-07-04	16:00:00	17:00:00	1	4	7
10	1	1	2023-07-06	13:30:00	14:30:00	2	4	0
11	0	0	2023-07-07	09:00:00	10:00:00	1	4	0
12	0	0	2023-07-11	13:32:00	14:32:00	3	1	3
13	0	0	2023-07-11	16:30:00	17:30:00	4	2	1
14	0	0	2023-07-12	10:00:00	11:00:00	3	3	1
15	0	0	2023-07-18	12:00:00	13:00:00	8	5	0
16	0	0	2023-07-19	08:00:00	09:00:00	6	8	0
17	1	0	2023-07-19	18:00:00	19:00:00	13	7	2
18	0	0	2023-07-21	21:00:00	20:00:00	15	10	1
19	0	0	2023-07-11	21:00:00	22:00:00	2	4	0
20	0	0	2023-07-18	08:00:00	09:00:00	2	4	0
21	0	0	2023-07-11	18:00:00	19:00:00	1	4	0
22	0	0	2023-07-11	09:40:45	09:40:46	14	9	4
23	0	0	2023-07-11	09:52:33	09:52:33	13	7	1

Figura 52. Tabla “Eventos”.

6. Conclusiones

Este proyecto se ha realizado con la intención de facilitar el acceso para realizar deporte a todo estudiante que lo desee sin necesidad de tener la carga de gestionar dicho evento, otro motivo ha sido la intención de fomentar la socialización de un campus tan abrumador como el de la Universidad Politécnica de Valencia.

Además, gracias al proyecto se pretende aumentar el uso de las instalaciones deportivas y dar a conocer el mundo del deporte en equipo. La facilidad que otorga participar en cualquier evento únicamente seleccionando tu posición y tu nivel dentro del deporte consigue que sea aún más llamativo para los estudiantes.

Para que la aplicación no se quede obsoleta, de cara al futuro se plantea seguir actualizando la misma. Entre las nuevas mejoras de la aplicación se plantea una interfaz para los administradores, de tal forma que no tengan que trabajar sobre la misma base de datos y lo puedan hacer directamente en la página web utilizando sus credenciales. Mirando por el cliente, otra actualización planteada es introducir un chat interno de cada evento, tanto de equipo como global, así los clientes pueden comunicarse entre sí.

Durante el desarrollo del proyecto ha supuesto un reto contemplar todas las opciones que debería otorgar el mismo, así como sumergirse en tecnologías desconocidas y aprender a usarlas en base a prueba y error. Problemas con instalaciones de servidores o bases de datos, fallos en la arquitectura del proyecto que suponían reestructurarlo de nuevo, fallos en los propios archivos de código cuya solución se encontraba gracias a documentación en internet y la capacidad de abstracción de uno mismo. Estas, entre otras muchas han sido las enseñanzas que se han aparecido de manera indirecta gracias a la puesta en marcha y finalización del proyecto lo cual es de agradecer y valorar.



7. Bibliografía

Tokio. <https://www.tokioschool.com/noticias/que-es-mysql/>

Tokio. <https://www.tokioschool.com/noticias/que-es-mysql>

W3schools. <https://www.w3schools.com/>.

Wikipedia. <https://www.wikipedia.org>

Apache. <https://httpd.apache.org/>

PHP. <https://www.php.net/manual/es/history.php>

KeepCoding. <https://keepcoding.io/blog/la-historia-de-javascript/>

Uniwebsidad. <https://uniwebsidad.com/libros/css/capitulo-1/breve-historia-de-css>

Raspberry Pi. <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>

DigitalOcean

<https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-20-04-es>