Proyecto de Grado

Presentado ante la ilustre Universidad de Los Andes como requisito final para obtener el Título de Ingeniero de Sistemas

Desarrollo de una Plataforma Virtual Deportiva como Sistema de Inscripción y Análisis para la Academia Emeritense F.C. (AEFC)

Por

Br. Romay Casique Juan Jose.

Tutor: Dr. Gerard Páez

Noviembre 2023

Resumen: El presente trabajo tiene como finalidad el desarrollo, implementación y puesta en marcha de una aplicación web alojada en la nube para la inscripción de jugadores y representantes de la Academia Emeritense F.C, así como también el manejo y administración de los mismos por parte de los administradores y personal directivo del club. Dicha plataforma cuenta con un sistema de gestión de contraseñas y autenticación basado en JSON Web Tokens para el flujo de inicio de sesión mediante proveedores OAuth o a través de credenciales usuario/contraseña implementada con la librería NextAuth que se ejecuta sobre el framework JavaScript NextJS basado en React. También se presenta un apartado de interfaz de usuario para el acceso, manejo y administración de los datos ingresados, se diseñó por medio de los frameworks NextJS para el frontend y Express.js para la parte del backend conectando ambas partes por medio de websockets, lo que permite la comunicación bidireccional en tiempo real entre cliente y el servidor en un mismo socket, utilizando una conexión TCP, esto para que la aplicación responda en tiempo real frente a las acciones de usuario, para controlar alertas o notificaciones en el lado del cliente y más importante para proveer un canal para el flujo de datos en el panel del administrador.

Para el almacenamiento de los datos se utiliza el proveedor de servicios MongoDB Atlas, este brinda acceso a una base datos MongoDB alojada en la nube. La aplicación también procesa imágenes cargadas por los usuarios, para esto, se optó por un servicio de almacenamiento especializado en gestión de archivos como lo es Cloudinary. Finalmente se presentan algunas opciones de proveedores PaaS (Plataforma como servicio), los cuales brindan hosting y permiten desplegar y ejecutar el código de la aplicación manejado a través de repositorios git o contenedores docker, tales como Vercel, Railway o Heroku.

Palabras clave: Aplicación Web, Autenticación, Mongo DB, Comunicación bidireccional, Despliegue.

Introducción

La tecnología avanza mucho con el pasar de los días, Las implementaciones de los sistemas actuales distan enormemente de cómo eran hace pocos años, hoy en día el acceso a internet por parte de las aplicaciones es vital para el funcionamiento de las mismas, se vive en una sociedad donde la administración y distribución de información está sujeta cada vez más a la tecnología, a internet, en la actualidad el intercambio de datos entre aplicaciones es un proceso normal, cada vez es más fácil poder transferir información de un lugar a otro con tan solo un clic.

Más aun, la aparición de la computación en la nube que viene a ser el uso de una red de servidores remotos conectados a internet para almacenar, administrar y procesar datos, bases de datos, redes y software. En lugar de depender de un servicio físico instalado, se tiene acceso a una estructura donde el software y el hardware están virtualmente integrados. Esto brinda alance global, además de un óptimo despliegue y servicio de la aplicación sin los pormenores de tratar con dispositivos físicos en algún lugar específico.

Las nuevas tecnologías aportan un amplio abanico de posibles implementaciones al actual sistema de inscripción de la Academia Emeritense F.C. Como primer paso se busca desplegar una aplicación basada en la nube que contenga la base de datos de jugadores pertenecientes a la organización, automatizar el sistema de registro para que cada jugador pueda acceder a la plataforma y suministrar sus datos y a la vez estos datos puedan ser gestionados por administradores y directivos. Las Plataforma como servicio (PaaS) ofrece un entorno de desarrollo completo, lo cual permite a los programadores enfocarse en la aplicación sin que la necesidad de encargarse de los elementos de infraestructura subyacente como el sistema operativo o hardware. Con esto establecido, se priorizan aspectos referentes a la aplicación en sí, como las conexiones de esta con la base de datos, conexiones entre el cliente y el servidor mediante websockets, sistema de autenticación y autorización, estos componentes antes mencionados forman parte del backend implementado en Node.js interactuando mediante llamadas api REST. Por otra parte, para el frontend se aplican librarías de estilo visual y maquetado para lograr un apartado visual agradable he intuitivo.

Índice General

Índice de Figuras

Capítulo 1

Planteamiento del Problema

1.1 El problema

La presente investigación aborda una de las problemáticas que desde hace años afectan a muchas agrupaciones e instituciones formativas, como es el llevar un control eficiente de inscripciones formales de sus participantes, tal es el caso de la Escuela de Formación de Talentos, Academia Emeritense, la cual realiza su proceso de inscripción a través de planillas que son llenadas en forma manual, dando como resultado procedimientos lentos y registros tediosos, aparte de gastos de material de papelería que debían ser archivados, por lo cual dicho proceso se convierte en obsoleto y disfuncional.

Las nuevas tecnologías permiten un manejo más eficiente de este tipo de procesos, el mundo del desarrollo de software avanza velozmente, cada día surgen nuevas alternativas y tendencias que debemos implementar si queremos mayor eficiencia en nuestros procesos, por lo tanto el objetivo principal de este trabajo de investigación es crear un sistema basado en la computación en la nube, como es la implementación de un proceso de inscripción automatizado que permita registros rápidos con información oportuna, precisa y confiable mediante una plataforma virtual, contribuyendo en mejorar las tareas administrativas rutinarias y de gran volumen de información en trabajos más fáciles, eficientes y provechosos.

1.2 Objetivos

1.2.1 Objetivo General

Desarrollar e implementar una plataforma virtual como sistema de registro, inscripción y análisis de datos de jugadores, personal técnico y administradores pertenecientes a la Academia Emeritense F.C.

1.2.2 Objetivos específicos

- Definir el modelo de la base datos para cada tipo de usuario, ya sea usuario jugador o administrador.
- Definir los componentes de seguridad y autenticación del sistema.
- Diseñar la interfaz de usuario de los distintos eventos del apartado de registro e inicio de sesión, así como también el panel de administrador y los formularios de inscripción de los distintos usuarios.
- Desarrollo de las API pertenecientes al backend para el manejo de datos, conexión con los sistemas de almacenamiento remotos y conectividad entre servidores.
- Desplegar la aplicación en las plataformas de computación en la nube previamente seleccionadas y configurar el entorno de alojamiento y despliegue de la aplicación.

1.3 Justificación

La Academia Emeritense no cuenta con un sistema de inscripción y control de jugadores que faciliten un trabajo óptimo por cuanto se basaba en registros de forma manual, repercutiendo en inversiones de tiempo más extensas al momento de realizar las mismas, gastos de inversión en papelería y poco control de datos específicos de los participantes convirtiéndose en un sistema poco eficiente, obsoleto y tardío, por lo cual se hace necesario la implementación de una plataforma virtual como sistema de inscripción y análisis que facilite el proceso de registro de los jugadores y sus representantes, el cual se refleja en un trabajo rápido por gozar de los beneficios de la red al poner la página a disposición de todos los entes participantes de la academia.

La utilización del sistema informático facilitará la gestión de los requerimientos solicitados por los profesores y la academia emeritense en general, permitiendo el fácil acceso y resguardo de los datos del jugador.

1.4 Alcances

El presente proyecto se centra en el diseño, la implementación y puesta en marcha de una plataforma web, con la finalidad de gestionar la data perteneciente a la Academia Emeritense, teniendo en cuenta los siguientes aspectos:

- Diseño de la interfaz de usuario intuitiva y fácil de usar que permita a los distintos tipos de usuarios registrarse en el sistema de manera eficiente.

- El sistema debe permitir a los usuarios ingresar y almacenar información relevante para la academia u otros datos necesarios para el registro.

- Validación de datos en tiempo real para asegurarse de que los datos ingresados sean correctos y cumplan con los criterios establecidos.

- Diseño de la estructura he implementación de un sistema de gestión de bases de datos.

- Generación de identificadores únicos de registro para facilitar la identificación y el acceso posterior a la información almacenada.

- Gestión de permisos de acceso, así como también la implementación de medidas de seguridad adecuadas para prevenir el ingreso no autorizado a los datos.

Capítulo 2
Marco Teórico

2.1 Antecedentes

A lo largo de la presente investigación, se expondrá la importancia e incorporación que ha tenido la tecnología en los diferentes ámbitos de la vida humana y en especial en la planificación y control de los ámbitos administrativos. De esta manera, la tecnología se convierte en un recurso importante y primordial para afrontar las exigencias del mundo actual.

En el área educativa, los investigadores han orientado sus proyectos con la incorporación de la tecnología, para mejorar los procesos de enseñanza-aprendizaje, así como también solventar problemas administrativos como es el registro y control de información, a continuación, se hace referencia a algunos de ellos:

Proyecto de Titulación en la Universidad Estatal Del Sur de Manabí, Ecuador. Nora Tatiana Sornoza Ponce. (Julio 2017), lleva por título **"APLICACIÓN INFORMÁTICA PARA LLEVAR EL CONTROL DEL REGISTRO TÉCNICO Y METODOLÓGICO DE LOS DEPORTISTAS DE LA LIGA DEPORTIVA CANTONAL DEL CANTÓN JIPIJAPA."** El cual tiene como objetivo "Implementar una aplicación informática para llevar el control del registro técnico y metodológico de los deportistas de la LDC Jipijapa." Concluyendo de esta manera que la aplicación informática fue de ayuda para que la información sea bien administrada y el tiempo de respuesta ante procesos sea optimizado.

Rosa Patricia Castillo Chaguay (Junio 2013), presentó a la Universidad De Guayaquil Facultad De Ciencias Matemáticas Y Físicas Carrera De Ingeniería En Sistemas Computacionales, su proyecto de grado relacionado a la **IMPLEMENTACIÓN Y DESARROLLO DE UN PORTAL WEB PARA LA UNIDAD EDUCATIVA** "LUIS FELIPE BORJA DEL ALCÁZAR" con la finalidad de optimizar el proceso comunicacional basado

en herramienta Open Source; el cual consiste en ayudar a la comunidad educativa en el desarrollo de la página web para facilitar el trabajo de los docentes a la hora de entrega de sus notas que se estaban realizando de manera manual, además de ser un mecanismo de comunicación entre toda la comunidad educativa fortaleciendo la información de manera clara y precisa. Esta tesis es relevante ya que da nuevas ideas que en el proceso de creación o con el pasar del tiempo se podría ir creando en la página web de la Institución siendo esta cada vez más completa con mayores servicios para la comunidad educativa.

2.2 Bases Teóricas

En este capítulo estarán los conceptos más importantes de las herramientas y procedimientos que han sido usados para llevar a cabo este trabajo.

2.2.1 SISTEMAS WEB

Se denomina aplicación web al software que reside en un ordenador, denominado servidor web, que los usuarios pueden utilizar a través de Internet o de una intranet, con un navegador web, para obtener los servicios que ofrezca (Zofío Jiménez, 2013).

Las aplicaciones Web se presentan en una amplia variedad de formas y tamaños, están escritas en todo tipo de lenguajes y se ejecutan en cualquier sistema operativo. En el núcleo de todas estas aplicaciones está la base de que todas sus funcionalidades son ejecutadas usando el protocolo HTTP y los resultados son formateados por lo general usando HTML.

El protocolo HTTP se utiliza para el trabajo cliente-servidor. HTTP por sí solo no tiene noción de estado, esto implica que una conexión no tiene relación con otra. Con HTTP la aplicación es la encargada de administrar las sesiones de usuarios y determinar cuáles conexiones están relacionadas con otras. Resulta sencillo capturar una comunicación HTTP y es simple también para los humanos interpretar y entender este protocolo ya que la información es transmitida en texto plano (Hernández Diaz, 2012).

2.2.2 API

Una API o interfaz de programación de aplicaciones es un conjunto de definiciones y protocolos que se usan para diseñar e integrar el software de las aplicaciones.

Las API permiten que sus productos y servicios se comuniquen con otros, sin necesidad de saber cómo están implementados. Esto simplifica el desarrollo de las aplicaciones y permite ahorrar tiempo y dinero. Las API le otorgan flexibilidad; simplifican el diseño, la administración y el uso de las aplicaciones; y ofrecen oportunidades de innovación, lo cual es ideal al momento de diseñar herramientas y productos nuevos (o de gestionar los actuales).

A veces, las API se consideran como contratos, con documentación que representa un acuerdo entre las partes: si una de las partes envía una solicitud remota con cierta estructura en particular, esa misma estructura determinará cómo responderá el software de la otra parte.

En este caso particular de implementación la creación de las API se usará como mecanismo de comunicación para realizar consultas específicas sobre la base de datos (Red Hat, 2007).

2.2.3 MongoDB

MongoDB es una base de datos NoSQL orientada a documentos que apareció a mediados de la década de 2000. Se utiliza para almacenar volúmenes masivos de datos. A diferencia de una base de datos relacional SQL tradicional, MongoDB no se basa en tablas y columnas. Los datos se almacenan como colecciones y documentos (DataScientest, 2022).

2.2.4 Javascript

Es un lenguaje de programación o de secuencias de comandos que te permite implementar funciones complejas en páginas web, cada vez que una página web hace algo más que sentarse allí y mostrar información estática para que la veas, muestra oportunas actualizaciones de contenido, mapas interactivos, animación de Gráficos 2D/3D, desplazamiento de máquinas reproductoras de vídeo, etc., puedes apostar que probablemente JavaScript está involucrado (MDN contributors, 2021).

2.2.5 Node.js

Se trata de un entorno de código abierto (open source) multiplataforma que ejecuta el código JavaScript fuera de un navegador. Este entorno de ejecución de JavaScript se orienta a eventos asíncronos (los eventos no dependen de que otros se hayan ejecutado previamente) y permite construir aplicaciones en red escalables, es decir, tiene la capacidad de realizar muchas conexiones de manera simultánea sin que tenga que leer el código línea a línea, ni abrir múltiples procesos (Henry, 2021).

2.2.6 Base de datos NoSQL

Según afirma Rackspace Technology (s.f): NoSQL se refiere a una base de datos no relacional o no SQL. Una base de datos relacional es un formato de bases de datos muy estructurado basado en una tabla, como MySQL u Oracle. Las bases de datos NoSQL están orientadas a los documentos y le permiten almacenar y recuperar datos en formatos que no sean tablas. Las aplicaciones modernas usan y generan tipos de datos complejos y que cambian constantemente, y las bases de datos relacionales no fueron diseñadas para gestionar este tipo de almacenamiento y recuperación de datos. Las bases de datos NoSQL son más flexibles y escalables.

Al trabajar con una base de datos NoSQL, usted puede agregar datos nuevos, sin tener que definirlos previamente en el esquema de la base de datos, lo que le permite procesar rápidamente grandes volúmenes de datos sin estructura, semiestructurados y estructurados.

El esquema dinámico de bases de datos NoSQL permite realizar desarrollos ágiles, que requieren iteraciones rápidas y significativas y durante los que no puede haber tiempo de inactividad (RACKSPACE TECHNOLOGY).

2.2.7 React

React (también llamada React.js o ReactJS) es una biblioteca Javascript de código abierto diseñada para crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones en una sola página. React intenta ayudar a los desarrolladores a construir aplicaciones que usan datos que cambian todo el tiempo. Su objetivo es ser sencillo, declarativo y fácil de combinar (Wikipedia).

2.2.8 Next.js

Next.js es un framework de React que permite a los usuarios crear aplicaciones Javascript de una sola página. Este framework tiene varias ventajas tanto para las aplicaciones de los clientes como para el equipo de desarrollo. Como usuarios, uno puede sentirse cada vez más irritado al ver que nuestras expectativas no se cumplen con sitios web y aplicaciones que tardan mucho tiempo en cargar. Next.js está muy extendido y es una buena opción por varias razones, la mayoría relacionadas con la velocidad y el rendimiento.

Next.js fue desarrollado por Guillermo Rauch, el CEO de Vercel, en 2016. Actualmente está en la versión 13. Next.js está realmente escrito sobre Node.js, por lo que requiere que tengas Node.js para usarlo con Node Package Manager (npm) (Yulima Hernandez, 2022).

2.2.9 Socket

Los sockets son un método de comunicación entre un programa de cliente y uno de servidor a través de una red. Un socket se define como “el extremo de una conexión”.

Un socket, es un método para la comunicación entre un programa del cliente y un programa del servidor en una red. Se define como el punto final en una conexión. Los sockets se crean y se utilizan con un sistema de peticiones o de llamadas de función a veces llamados interfaz de programación de aplicación de sockets (API, Application Programming Interface).

Un socket es también una dirección de Internet, combinando una dirección IP (la dirección numérica única de cuatro octetos que identifica a un computador particular en Internet) y un número de puerto (el número que identifica una aplicación de Internet particular). Un socket es un punto final de un enlace de comunicación de dos vías entre dos programas que se ejecutan a través de la red (Héctor Julio Fúquene Ardila, 2011).

2.2.10 JSON Web Token

Un JSON Web Token es un token de acceso estandarizado en el RFC 7519 que permite el intercambio seguro de datos entre dos partes. Contiene toda la información importante sobre una entidad, lo que implica que no hace falta consultar una base de datos ni que la sesión tenga que guardarse en el servidor (sesión sin estado).

Por este motivo, los JWT son especialmente populares en los procesos de autentificación. Con este estándar es posible cifrar mensajes cortos, dotarlos de información sobre el remitente y demostrar si este cuenta con los derechos de acceso requeridos. Los propios usuarios solo entran en contacto con el token de manera indirecta: por ejemplo, al introducir el nombre de usuario y la contraseña en una interfaz. La comunicación como tal entre las diferentes aplicaciones se lleva a cabo en el lado del cliente y del servidor (Merge-Gyms, 2022).

2.2.11 Cookies

Una cookie es un fragmento de texto o un pequeño dato que el servidor web proporciona al navegador cuando visitamos una página web; guardando información en el disco duro sobre nuestros hábitos online. Cada vez que visitamos una página que usa cookies, nuestro dispositivo envía los datos para que el navegador pueda recordar quiénes somos y crear, así, una experiencia de usuario más personalizada.

Los sitios de compra utilizan cookies para hacer un seguimiento de los elementos que los usuarios han visto anteriormente, usarlos para sugerir otros que les podrían interesar y guardar los elementos en el carrito de la compra mientras continúan comprando. Las cookies se crean cuando los usuarios visitan un sitio web nuevo y el servidor web envía un pequeño flujo de información a sus navegadores web. Esa cookie se envía solo cuando el servidor quiere que el navegador web guarde la cookie. En ese caso, recordará la cadena nombre=valor y la reenviará al servidor con cada solicitud de seguimiento.

Si un usuario vuelve a ese sitio en el futuro, el navegador web devuelve los datos al servidor web en forma de cookie.El nombre de "cookie" viene de "magic cookies," término acuñado por el programador de navegadores web Lou Montulli. Los términos hacen referencia a paquetes de información que se envían y reciben sin cambios. La analogía con el término utilizado en inglés para galleta (cookie) es pura coincidencia, pero viene muy bien al caso (Cassie Bodnar, 2013).

2.2.12 Computación en la Nube

La computación en la nube, conocida también como servicios en la nube, informática en la nube, nube de cómputo o simplemente «la nube», es un paradigma que permite ofrecer servicios de computación a través de una red, que usualmente es internet.

La computación en la nube ofrece acceso a recursos informáticos, de almacenamiento y de redes a pedido. Estos recursos pueden venir ya sea de su propio centro de datos o de un proveedor de servicios en la nube. Según los tipos de servicio y los modelos de implementación que seleccione, la computación en la nube puede ayudarlo a administrar los costos al tiempo que permite lanzar con rapidez nuevos productos o servicios, expandirse a nuevas ubicaciones y maximizar el desempeño y la productividad (Tatiana Grapsas, 2018).

2.2.13 PaaS

PaaS, o plataforma como servicio, es un modelo de cloud computing que proporciona a los clientes una plataforma de cloud completa —hardware, software e infraestructura— para desarrollar, ejecutar y gestionar aplicaciones sin el coste, la complejidad y la inflexibilidad que suelen ir asociadas a la creación y el mantenimiento de dicha plataforma en local.

El proveedor PaaS lo aloja todo en su centro de datos: servidores, redes, almacenamiento, software de sistema operativo, bases de datos, herramientas de desarrollo. Normalmente, los clientes pueden pagar una tarifa fija para proporcionar

una cantidad específica de recursos para un número específico de usuarios, o pueden elegir el precio según uso para pagar solo los recursos que utilicen. Cualquiera de las opciones permite a los clientes de PaaS crear, probar, desplegar, ejecutar, actualizar y escalar aplicaciones de forma más rápida y económica que si tuvieran que crear y gestionar su propia plataforma local (Tatiana Grapsas, 2018).

2.2.14 SaaS

El software como servicio (SaaS) es un modelo de distribución y de licencias usado para entregar aplicaciones de software a través de Internet, es decir, como un servicio.

Los usuarios suelen tener acceso a aplicaciones basándose en un modelo de suscripción, lo que hace que SaaS sea la plataforma ideal para aplicaciones de software empresarial tales como el correo electrónico, la mensajería instantánea y la gestión de relaciones con los clientes (CRM).

El SaaS es una evolución del modelo de ASP: en esta solución, los proveedores y distribuidores gestionan su propio software y no se requiere instalación, ya que el software se distribuye de manera instantánea a través de Internet, es decir, a través de la nube. La informática en la nube permite a las empresas consumir recursos informáticos a través de Internet como un servicio (de la misma forma que la electricidad o el agua). El modelo de informática en la nube basada en SaaS ofrece a las empresas un nivel de eficiencia y ahorro de costes significativos (Tatiana Grapsas, 2018).

2.2.15 IaaS

La infraestructura como servicio (IaaS) es un método para ofrecer funcionalidades de computación, almacenamiento, redes y de otros tipos a través de Internet. La IaaS permite a las empresas utilizar sistemas de funcionamiento, aplicaciones y almacenamiento basados en la web sin tener que comprar, administrar y brindar soporte a la infraestructura de nube subyacente. Los ejemplos más utilizados de plataformas IaaS son Amazon Web Services (AWS) y Microsoft Azure.

La infraestructura como servicio (IaaS) es una de las tres categorías principales del cómputo en la nube, junto con el software como servicio (SaaS) y la plataforma como servicio (PaaS). IaaS es el segmento de nube de crecimiento más rápido (Tatiana Grapsas, 2018).

2.2.16 Sistemas de Autenticación

La autenticación es el proceso que debe seguir un usuario para tener acceso a los recursos de un sistema o de una red de computadores. Este proceso implica identificación (decirle al sistema quién es) y autenticación (demostrar que el usuario es quien dice ser). Esto evita que cualquiera pueda entrar en un determinado sistema o iniciar sesión en alguna plataforma de forma indebida, sin que realmente sea el usuario legítimo que tiene el poder para hacerlo (Merge-Gyms, 2022).

Capítulo 3
Marco Metodológico

Se describe la metodología que será utilizada para planificar y llevar un registro de las tareas a completar, lo cual permitirá tener una mejor organización y manejo a la hora de desarrollar el producto.

Esta Investigación estará basada en la Metodología Kanban, la cual permitirá un mejor control y desarrollo de las tareas.

3.1 Metodología Kanban

Kanban es una forma de ayudar a los equipos a encontrar un equilibrio entre el trabajo que necesitan hacer y la disponibilidad de cada miembro del equipo. La metodología Kanban se basa en una filosofía centrada en la mejora continua, donde las tareas se "extraen" de una lista de acciones pendientes en un flujo de trabajo constante.

La metodología Kanban se implementa por medio de tableros Kanban. Se trata de un método visual de gestión de proyectos que permite a los equipos visualizar sus flujos de trabajo y la carga de trabajo. En un tablero Kanban, el trabajo se muestra en un proyecto en forma de tablero organizado por columnas. Tradicionalmente, cada columna representa una etapa del trabajo. El tablero Kanban más básico puede presentar columnas como Trabajo pendiente, En progreso y Terminado. Las tareas individuales —representadas por tarjetas visuales en el tablero— avanzan a través de las diferentes columnas hasta que estén finalizadas. Kanban para el desarrollo de software es una subcategoría de gestión ágil de proyectos.

La filosofía ágil fomenta la planificación adaptativa, el desarrollo evolutivo, la entrega temprana y la mejora continua, para ayudar a los equipos a responder de manera flexible al cambio.

Hoy en día, los tableros Kanban son en su mayoría tableros virtuales con columnas que representan las etapas del trabajo, aunque aún se pueden dibujar tableros Kanban en pizarras blancas y dar seguimiento al trabajo con post-it!). En un tablero, una "tarjeta Kanban" representa una tarea, y esta tarjeta de tarea avanza a través de las etapas del trabajo a medida que se finaliza. Los equipos que usan un sistema Kanban tienden a colaborar en un único tablero Kanban, aunque las tareas generalmente se asignan a miembros individuales del equipo.

3.2 Principios Kanban

Existen cuatro principios básicos que te ayudarán a guiar a tu equipo al momento de implementar la metodología Kanban:

Empieza con lo que haces ahora: Puedes implementar el marco Kanban a cualquier proceso o flujo de trabajo actual. A diferencia de algunos procesos de gestión ágil más definidos como Scrum, el marco Kanban es lo suficientemente flexible como para adaptarse a las prácticas centrales de tu equipo de trabajo.

Comprométete a buscar e implementar cambios progresivos y evolutivos: Los grandes cambios pueden ser perjudiciales para tu equipo y, si intentas cambiar todo a la vez, es posible que el nuevo sistema no funcione como esperabas. Ese es el motivo por el cual el marco Kanban está diseñado para fomentar la mejora continua y el cambio progresivo. En lugar de cambiar todo de una vez, empieza por buscar cambios progresivos para lograr que los procesos de tu equipo realmente evolucionen con el tiempo.

Respeta los procesos, los roles y las responsabilidades actuales: A diferencia de otras metodologías Lean, Kanban no tiene roles integrados y puede funcionar con la estructura y los procesos actuales de tu equipo. Además, tu proceso actual podría tener elementos excelentes, que se perderían si intentaras modificar completamente tu sistema de trabajo de un momento a otro.

Impulsa el liderazgo en todos los niveles: Con el espíritu de mejora continua, el método Kanban reconoce que el cambio puede provenir de cualquier dirección y no solo “de arriba abajo”. Con Kanban, se alienta a los miembros del equipo a participar, proponer nuevas formas para lograr que los procesos evolucionen y emprender nuevas iniciativas de trabajo.

3.3 Metodología Kanban

Los principios básicos de Kanban te ayudan a guiar la mentalidad de tu equipo al momento de abordar un flujo de trabajo Kanban. Para implementar un proceso Kanban, sigue estas seis prácticas que usan las grandes empresas y que te permitirán ayudar a tu equipo a implementar una mentalidad de mejora continua y a lograr un crecimiento progresivo: los principios esenciales del marco Kanban.

Visualizar el trabajo: Una de las principales ventajas de Kanban es que puedes visualizar cómo el trabajo “avanza” a través de las etapas. Una tarjeta Kanban de tarea comenzará su viaje en el lado izquierdo de tu tablero y, a medida que tu equipo trabaja en ella, recorrerá lentamente las siguientes etapas hasta que aterrice en la columna Finalizadas. Esta práctica no solo te brinda una idea general de cómo el trabajo avanza a través de las etapas, sino que también te permite obtener información en tiempo real y apreciar de un vistazo el estado de los proyectos.

Limitar el trabajo en curso: Como metodología ágil, Kanban se centra en un principio de entrega temprana, lo que implica que las tareas deben moverse rápidamente de una columna a otra en lugar de estancarse en un estado ambiguo de “trabajo en progreso” (wip). Aunque no existe un requisito establecido sobre cuántas tareas deben estar “en progreso” en un momento dado, es importante establecer los límites del trabajo; por lo que anima a tu equipo a centrarse en finalizar tareas individuales y a evitar realizar varias tareas a la vez.

Gestionar el flujo de trabajo: La práctica n° 2 recomienda limitar la cantidad de trabajo en curso, y la mejor manera de hacerlo es con la optimización del flujo de tareas dentro del tablero Kanban. Gestionar y mejorar el flujo de trabajo te permitirá controlar el tiempo predestinado para el trabajo y así poder reducir el tiempo de entrega (el tiempo que pasa entre el inicio de una tarea hasta que llega a la columna Finalizadas de tu tablero Kanban) y garantizar que estás entregando tareas o enviando nuevos productos mientras siguen siendo relevantes.

Implementar políticas de procesos explícitas: Debido a la rapidez con la que se mueven las tareas en Kanban, asegúrate de que tu equipo haya establecido y comunicado claramente las convenciones. Las políticas de tu proceso deben guiar a tu equipo en la implementación de la metodología Kanban. Además, se debe alentar a todos en el equipo a participar e innovar las políticas Kanban, tal como se establece en el cuarto principio básico de Kanban: Impulsar el liderazgo en todos los niveles.

Implementar ciclos de comentarios: En Kanban, necesitas recopilar comentarios de dos grupos distintos: tus clientes y tu equipo.

Recopila comentarios de tus clientes sobre la calidad y eficacia de la solución que produjo el equipo. ¿Fue el producto adecuado? ¿Hubo algún problema? En el caso de que haya surgido algún problema, como errores en un código o cualquier otro defecto del producto, revisa tu flujo Kanban y agrega más tiempo para la revisión, los ajustes y la evaluación.

Realiza consultas frecuentes con el equipo sobre el proceso de ejecución de un marco Kanban. ¿Cómo se sienten con los resultados? Aquí tienes otra oportunidad para fomentar el liderazgo en todos los niveles y mejorar las políticas de procesos del equipo.

Mejorar colaborando y evolucionar experimentando: En esencia, Kanban se trata de una mejora continua. Sin embargo, también significa que otros sistemas podrían funcionar bien junto con Kanban. Ya sea Scrum o alguna otra metodología, debes estar siempre dispuesto a colaborar, experimentar y desarrollar tus procesos si es necesario.

3.4 Implementación de la metodología Kanban

De acuerdo con el primer principio básico de Kanban (Empieza con lo que haces ahora), puedes aplicar Kanban a cualquier flujo de trabajo. Puedes crear un tablero Kanban con una pizarra y algunos post-it o con una hoja de cálculo y un formato dado, pero la mejor manera de visualizar un tablero Kanban es con una herramienta de gestión del trabajo.

3.5 Los beneficios de Kanban

Kanban es una herramienta excelente y flexible que puede ayudar a los equipos a encontrar un equilibrio entre la demanda de trabajo y la disponibilidad del equipo. Si se usa correctamente, Kanban puede:

• Ofrecer un panorama que permite ver de un vistazo el trabajo de tu equipo. Como un método visual de gestión de proyectos, Kanban puede ayudarte a poner en marcha el trabajo y obtener una visión clara de los flujos de trabajo de tu equipo.

• Aumentar la claridad, especialmente en los equipos remotos. Si tu equipo se encuentra trabajando de forma remota, puede resultar difícil obtener visibilidad del trabajo que realizan tus compañeros. Al centralizar las tareas y reducir la cantidad de trabajo en curso en un momento dado, los tableros Kanban pueden ayudarte a ti y a tu equipo a obtener información de un vistazo sobre quién está haciendo qué.

• Fomentar la flexibilidad. Debido a que el marco Kanban se basa en un proceso de mejora continua, los equipos que implementan Kanban pueden volverse más flexibles y dinámicos con el tiempo. Si sigues los cuatro principios básicos y las seis prácticas esenciales, tu equipo podrá volverse más ágil y más predispuesto al cambio. Cuando hay algo que cambiar, el equipo se ajusta para conseguir el objetivo del sprint.

Capítulo 4
Desarrollo e Implementación del Proyecto

4.1 Introducción

Durante este capítulo se presenta el desarrollo del software, análisis e implementación de la capa administrativa y registros de la plataforma virtual deportiva de la institución Academia Emeritense F.C.

4.2 Planificación

A lo largo de esta primera fase se realizaron diversas reuniones y entrevistas con la dirección administrativa de la Academia Emeritense F.C. en donde se pudo especificar y consolidar las necesidades y requerimientos, tanto a lo referente de su administración como para el momento de inscripción de sus jugadores como padres y representantes.

4.3 Descripción de los Componentes e Implementación

La aplicación fue desarrollada en Next.js en conjunto con Node. Next.j.js es un framework de React con el cual se puede desarrollar tanto la parte del Frontend como la del Backend de una aplicación, mientras que Node.js es un entorno de ejecución de Javascript de lado del servidor que en conjunto con el framework Express.js conforma una parte del Backend de la aplicación. Dicha aplicación está desplegada en diferentes proveedores de servicios de computación en la nube, cada uno cumpliendo una función específica de manera independiente.

Para la base de datos, se utilizó Mongo Atlas. Asimismo, se utilizó un servidor de backend escrito en Express que se aloja en Railway. Otro proveedor también apto para este tipo de entorno es Render, en ambos se desplegó y ejecutó correctamente. Como servidor de alojamiento web se utilizó Vercel que brindaba una implementación fácil y práctica para desplegar rápidamente el código desarrollado por medio de Next.js.

Debido a que Vercel implementa una arquitectura sin servidor (Serverless), es necesario también el uso de un servidor tradicional desplegado en la nube, como se especificó anteriormente con Railway o Render para ejecutar APIs que no son soportadas por funciones sin servidor (Serverless Functions) y que son necesarias para el correcto funcionamiento del sistema.

Se utilizó tambien Cloudinary para almacenar y gestionar imágenes que los usuarios suministran a la plataforma, este proveedor es ampliamente reconocido y brinda una buena facilidad de uso.

4.4 Interfaces GUI

La aplicación cuenta con dos interfaces, la primera es de autenticación y la otra permite al usuario o administrador acceder e ingresar o gestionar sus datos. La librería utilizada en la parte de autenticación es NetxAuth. Es importante mencionar que esta librería no brinda soporte para la gestión completa de usuarios a través de credenciales, como se utilizó en la aplicación en cuestión. La documentación de NextAuth recomienda utilizar autenticación OAuth u otros tipos de autenticación más seguros, o también reforzar el proceso con una autenticación de dos factores. A pesar de esto, se decidió permitir la autenticación con credenciales de usuario por ser una alternativa muy acostumbrada por las personas y para mejorar la experiencia de usuario. Como resultado, se tuvo que gestionar la autenticación de usuarios mediante credenciales. Esto quiere decir que tanto el manejo de los datos de la cuenta como la encriptación y desencriptación de estos viene dado a criterio del desarrollador.

4.5 Base de Datos y Almacenamiento

Dentro de la base de datos se crean distintas tablas para almacenar los datos referentes a las cuentas de usuarios registrados, una de estas tablas contiene el correo electrónico, contraseña y tipo de cuenta entre otros datos, en otra tabla se almacenan los códigos de verificación que se envían al correo al momento de crear una cuenta o cambiar la contraseña y en las tablas restantes se guardan datos de los jugadores o administradores según sea el caso, en una de estas se guarda la información personal, en otra datos de los padres del jugador y en otra información médica.

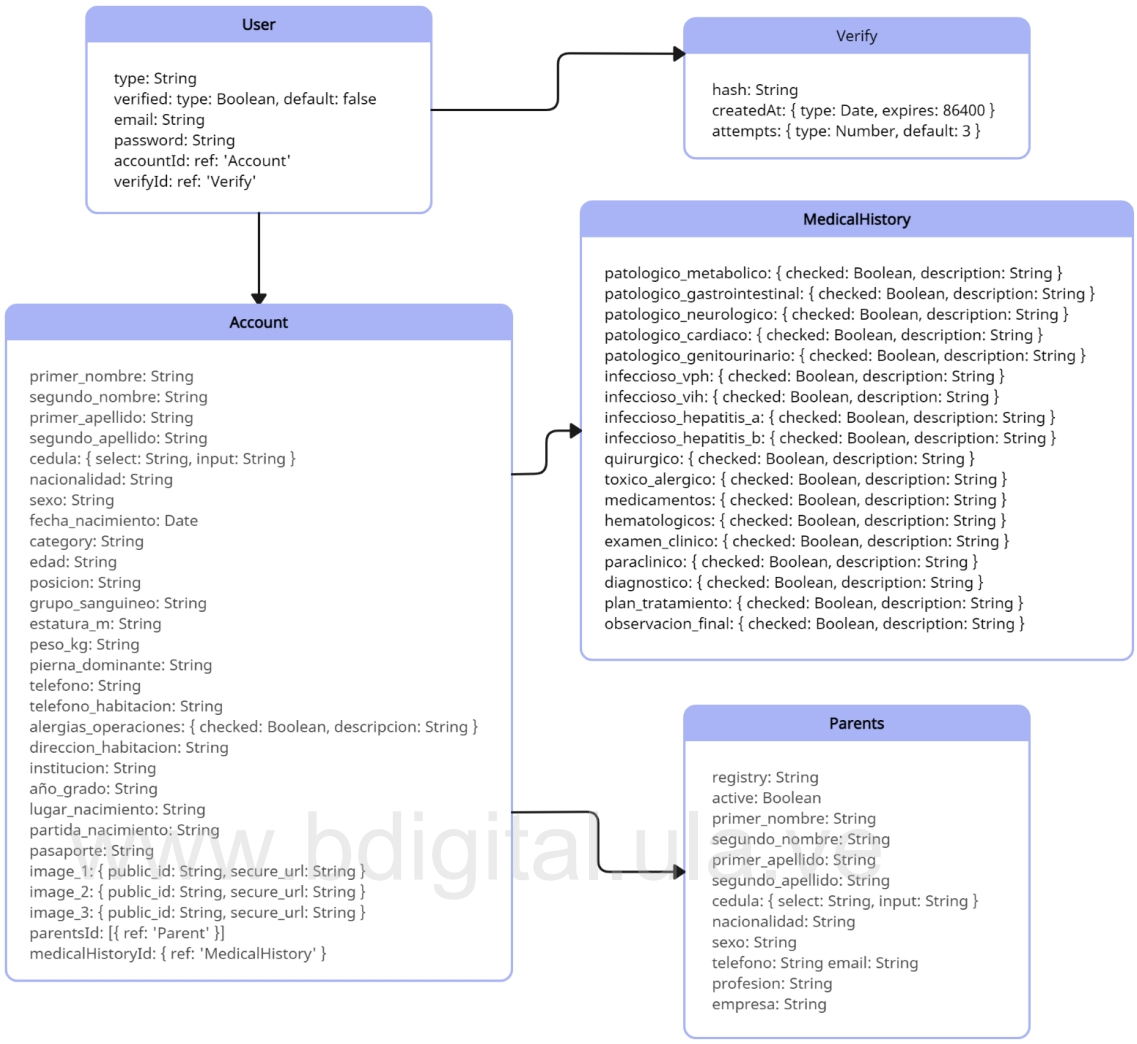

Figura 1. Modelo de datos para usuarios.

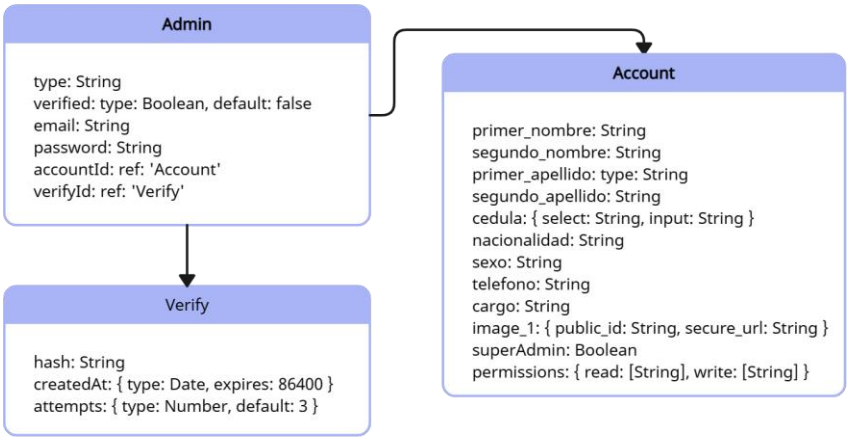

Figura 2. Modelo de datos para administradores.

4.6 Autenticación y Registro

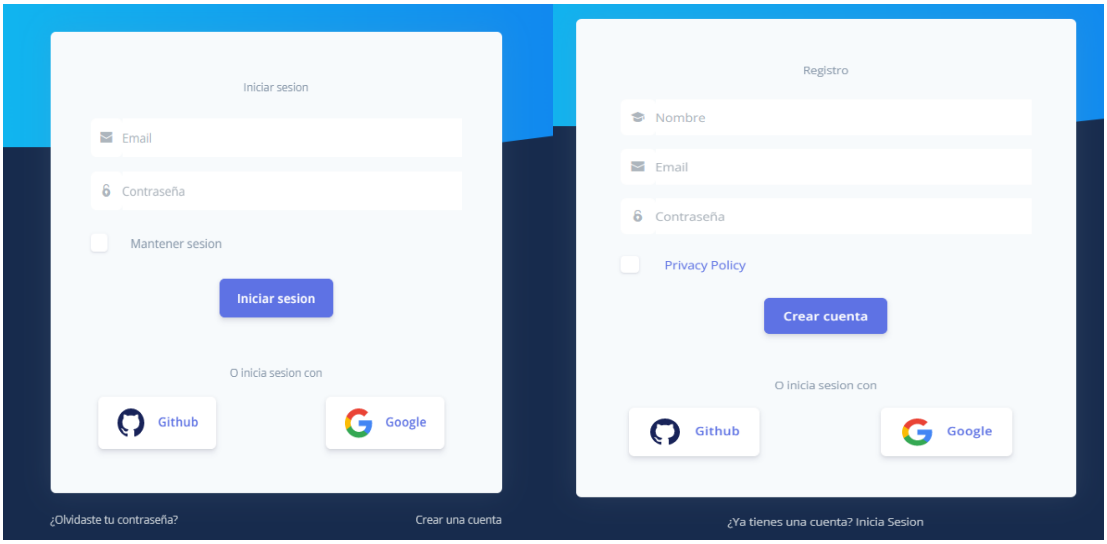

Figura 3. Interfaz de autenticación y registro.

4.6.1 Registro por Credenciales

Para explicar el manejo de los recursos vamos a detallar el proceso de creación de una cuenta por medio de credencias (correo electrónico y contraseña). Este algoritmo se ejecuta como una función sin servidor (Serverless Function) en servidores provistos por Vercel, el cual es el mismo que se encarga de suministrar la página al cliente, por lo que cuando un cliente solicita la creación de una nueva cuenta y envía los datos (por medio de una solicitud POST), estos son recibidos por medio de una API, que al ejecutarse primero envía una solicitud de búsqueda a la base de datos con el correo suministrado.

En caso de que la búsqueda genere un resultado, tal acción significa que ya el correo esta registrado, por ende, la aplicación envía una respuesta 401 al cliente para indicar que la operación no se completó por el intento de duplicación.

Si dicho correo no exista se continua con el proceso de registro creando una serie de documentos en la base de datos que se detallaran a continuación. Primero se genera un documento que contiene una cadena de caracteres pseudoaleatorios como código de verificación que posteriormente se envía al correo. Luego se crea un documento que va a contener los datos personales de la cuenta, de momento este documento estará vacío y finalmente estos documentos recién creados se enlazan a otro que contiene el correo electrónico, el hash de la contraseña y un dato que indica si la cuenta ha sido verificada. Después que todos los registros son creados satisfactoriamente dentro de la base de datos se manda un correo con el código de

verificación con la ayuda de la librería NodeMailer y se envía una respuesta 201 a la solicitud del usuario. Con esta respuesta la interfaz del usuario cambia y solicita el código enviado al correo electrónico.


```
_id: ObjectId('64360b25489ad81c66b6973c')
verified: true
type: "credential"
email: "j2romay@gmail.com"
password: "$2b$10$U0By1UEoFY2MZmTbXzd45eg4yX9xlU9teQQKmzkFCrhHPFZhEJaiS"
accountId: ObjectId('64360b25489ad81c66b69739')
verifyId: null
__v: 0
```

```
_id: ObjectId('64360b25489ad81c66b69739')
primer_nombre: "Juan"
▸ parentsId: Array
__v: 0
```


Figura 4. Colección de datos creada bajo credenciales de usuario después de la verificación.

4.6.2 Autenticación OAuth

Un procedimiento muy similar ocurre cuando el cliente se registra por medio del protocolo OAuth, se crea en la base de datos un documento vacío que contendrá los datos personales, también otro documento que almacena el correo electrónico y los tokens de respuesta de autorización. El documento de verificación se omite en este caso. Luego que el usuario haya sido verificado, la librería NextAuth se encarga de culminar el flujo de autenticación, automatizando la gestión por medio de cookies seguras para manejar las sesiones de usuario. Esta librería gestiona la creación, actualización y eliminación de cookies de sesión, lo que le simplifica al programador, el proceso de gestión de sesiones. Entonces, una vez el proceso se complete exitosamente la cookie de sesión es enviada al cliente y pasa a la vista de usuario autenticado.

```
_id: ObjectId('6425eedc489ad81c66b69732')
iss: "https://accounts.google.com"
azp: "400902463661-cn2pjsgv4p0utvlg1fssm4fk0ntgrbs1.apps.googleusercontent.c…"
aud: "400902463661-cn2pjsgv4p0utvlg1fssm4fk0ntgrbs1.apps.googleusercontent.c…"
sub: "110061819245121679261"
email: "j2romay@gmail.com"
email_verified: true
at_hash: "6pqsFpeogB8s3NYstX-FoA"
name: "Juan Romay"
picture: "https://lh3.googleusercontent.com/a/AGNmyxZLJ9-gVluadye_j8Qv7lQ_dSZ7ns…"
given_name: "Juan"
family_name: "Romay"
locale: "es-419"
iat: 1680207578
exp: 1680211178
type: "oauth"
verified: true
accountId: ObjectId('6425eedc489ad81c66b69731')
```


Figura 5. Colección de datos creada con autenticación OAuth.

4.6.3 Cambio de Contraseña

También ocurre algo parecido en cuanto a la gestión cuando el usuario inicia sesión o solicita cambiar su contraseña en el apartado de autenticación. Al momento de iniciar sesión se envía una solicitud con el correo y la contraseña suministrada, el servidor recibe la información y busca un documento en la base de datos cuyo correo electrónico coincida con el suministrado, si lo consigue genera un hash de la contraseña para compararlos entre sí, y si todo resulta correcto la función culmina con una respuesta 201 y envía una cookie de autorización al cliente. De igual modo, si el usuario solicita el cambio de contraseña, se vuelve a generar un documento con un código de verificación que se envía al correo. El usuario luego ingresa el código y este se valida en la base de datos de la misma forma como se explicó anteriormente, si los códigos coinciden se sustituye la contraseña antigua con la suministrada.

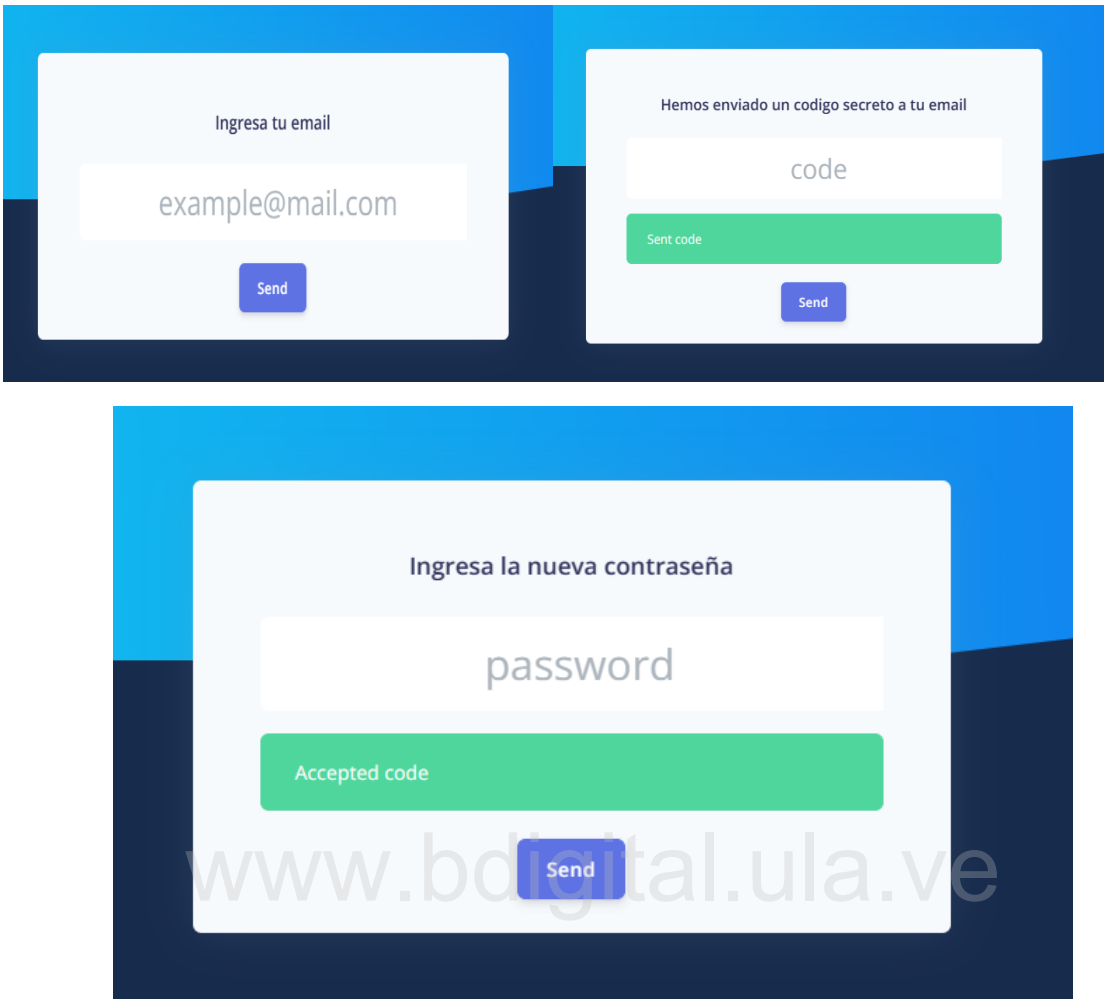

Figura 6. Interfaz para el cambio de contraseña.

4.6.4 Verificaciones de Usuario

En la base de datos los documentos que contienen códigos de verificación poseen un tiempo de vida (índice TTL) de 86400 segundos, es decir, la base de datos los elimina automáticamente después de 24 horas, y también son eliminados por el servidor al concluir una verificación exitosa.

Todo el módulo de autenticación se lleva a cabo en servidores de Vercel como funciones sin servidor en conjunto con la nube de Mongo Atlas para manejar la base de datos por medio de llamadas API.

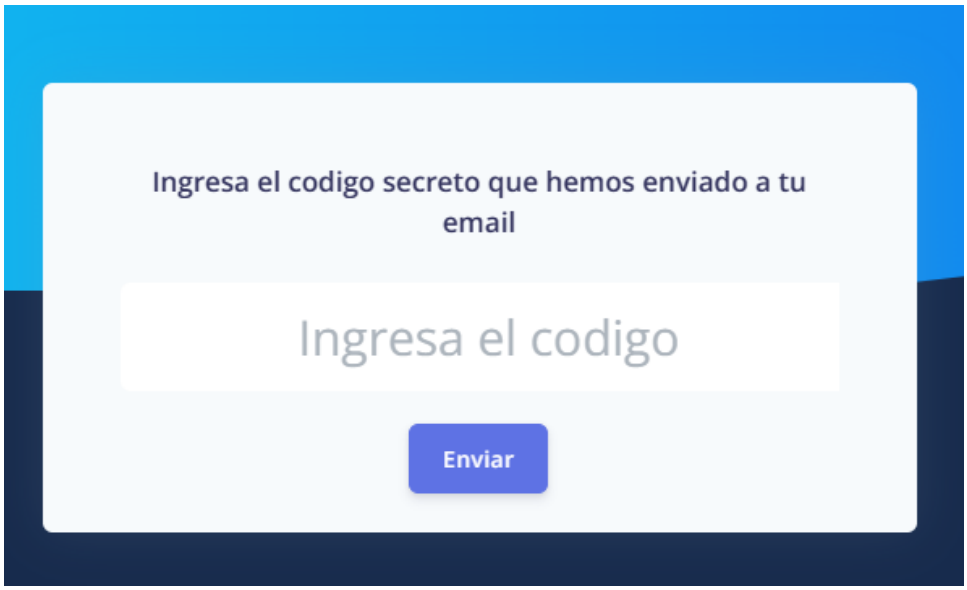

Figura 7. Vista de verificación de usuario


```
_id: ObjectId('64360b25489ad81c66b69734')
attempts: 3
createdAt: 2023-04-12T01:36:37.206+00:00
hash: "LG6vkh"
__v: 0
```


Figura 8. Datos de verificación en la base de datos.

4.6.5 Gestión de Cookies de Sesión

Las cookies son parte fundamental del funcionamiento de la plataforma, después de que el usuario se autentica debe enviar la cookie en cada solicitud que haga de ahora en adelante para ser atendido. La cookie una vez llega al servidor es desencriptada y contiene información referente al usuario que realizó la petición. De esta manera se gestionan adecuadamente las peticiones según quien y que tipo de usuario realiza la solicitud ya que también se hace un manejo distinto entre administradores y usuarios convencionales. Vamos a detallar en profundidad este funcionamiento.

4.6.6 Manejo de Sesiones de Usuario

NextAuth permite personalizar la información que se almacena en las cookies de sesión, lo que permite adaptar la gestión de sesiones y comportamiento de la plataforma según esto, en este caso la información que se agrega es el id del

documento almacenado en la base de datos que contiene el correo electrónico, también el id del documento que contiene la información personal de la cuenta y por último el dato que valida si la cuenta ha sido verificada o no (en caso de que el usuario intente iniciar sesión sin antes verificar la cuenta, al revisar este dato se redirecciona al usuario a la pantalla de verificación). Esta información adicional permite al servidor en un primer momento después de que el usuario ya se encuentra autenticado y solicita los recursos de la página, verificar la cookie y desencriptarla para luego extraer de la base de datos la información concerniente al usuario y de esta manera recuperar la información que anteriormente el usuario registro en la plataforma.

4.7 Interfaces de Usuario

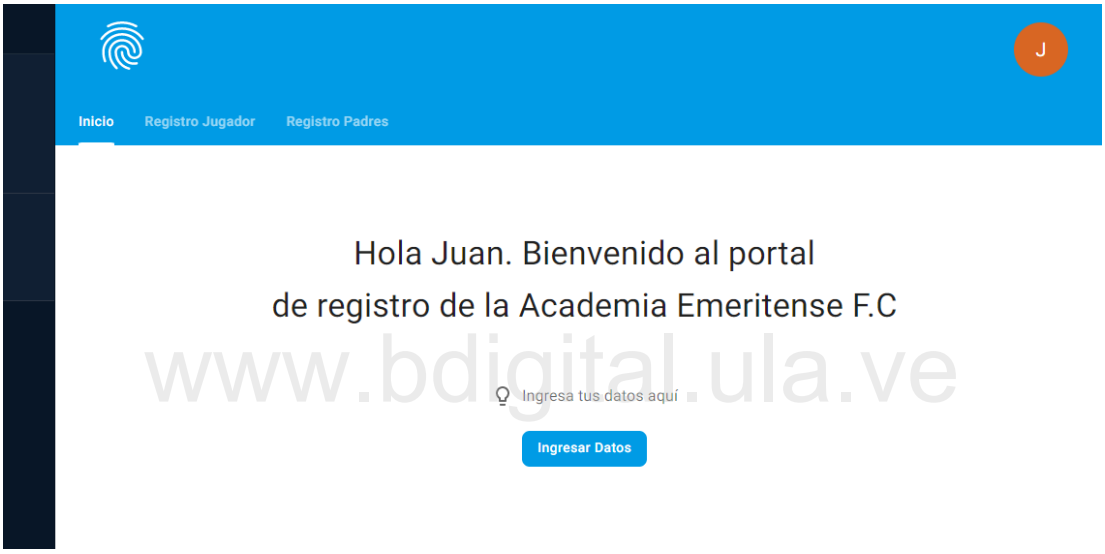

Figura 9. Pantalla de inicio.

El mecanismo de autenticación antes mencionado es idéntico tanto para usuarios convencionales como para administradores en cuanto al manejo de la cookie, la consulta y el envió de datos personales, ahora bien, el sistema envía recursos distintos según el tipo de usuario, para el usuario convencional que viene a ser el jugador en este caso, el recurso web de respuesta conforma una interfaz con un menú de inicio, un apartado para ingresar datos personales y subir una foto, otro para ingresar la información de padres o representantes y otro para registrar datos médicos. En cambio, para un administrador la interfaz recibida también la compone un apartado donde puede ingresar algunos datos personas y subir una foto, pero se omiten los apartados para anexar información del representante o datos médicos y en su lugar se agrega un apartado donde puede visualizar, editar o eliminar distintas cuentas registradas en la plataforma a las cuales el administrador tenga permiso ya sea de lectura o escritura.

4.8 Tipos de Cuentas

Entonces la plataforma está construida para registrar usuarios, que son jugadores de la academia emeritense en este caso, dichos usuarios pueden almacenar su información personal que luego es gestionada por el club mediante los administradores, los cuales también se registran, crean una cuenta y se les asigna ciertos permisos de acceso (ya sea de lectura o escritura) con los que van a tener a su disposición una lista de cuentas autorizadas para su manejo según el permiso. Por eso un tipo de administrador se podría tomar como un entrenador de la academia, al cual se le asignan permisos según la categoría que tenga a su cargo, de esta forma tiene acceso solo a los jugadores pertenecientes a cierta categoría. Hay que hacer una acotación muy importante en cuanto a este dato, la "categoría" es el valor por el cual se rigen los permisos de acceso de administradores que no poseen acceso total a los datos, en este caso pueden ser los entrenadores o profesores del club, en cambio, un administrador con acceso total a los datos, que viene siendo alguien encargado de la parte administrativa de club por ejemplo, se conoce como un administrador principal y tiene acceso a los jugadores y administradores (entrenadores) registrados en la plataforma, teniendo permisos de visualización, escritura y eliminación de los mismos.

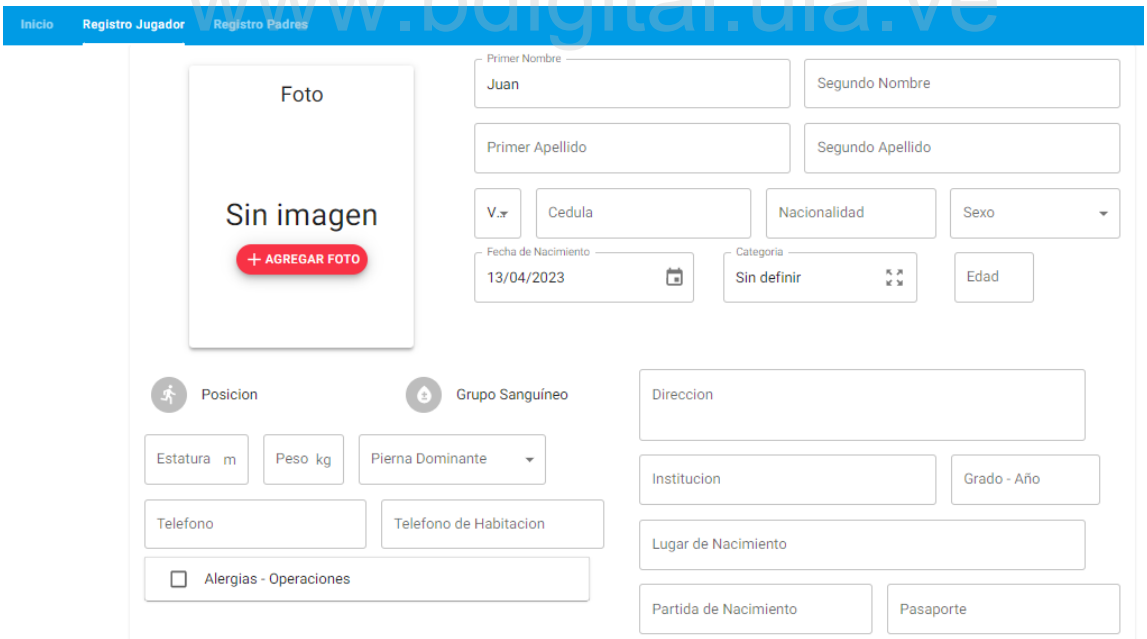

Figura 10. Formulario de jugador.

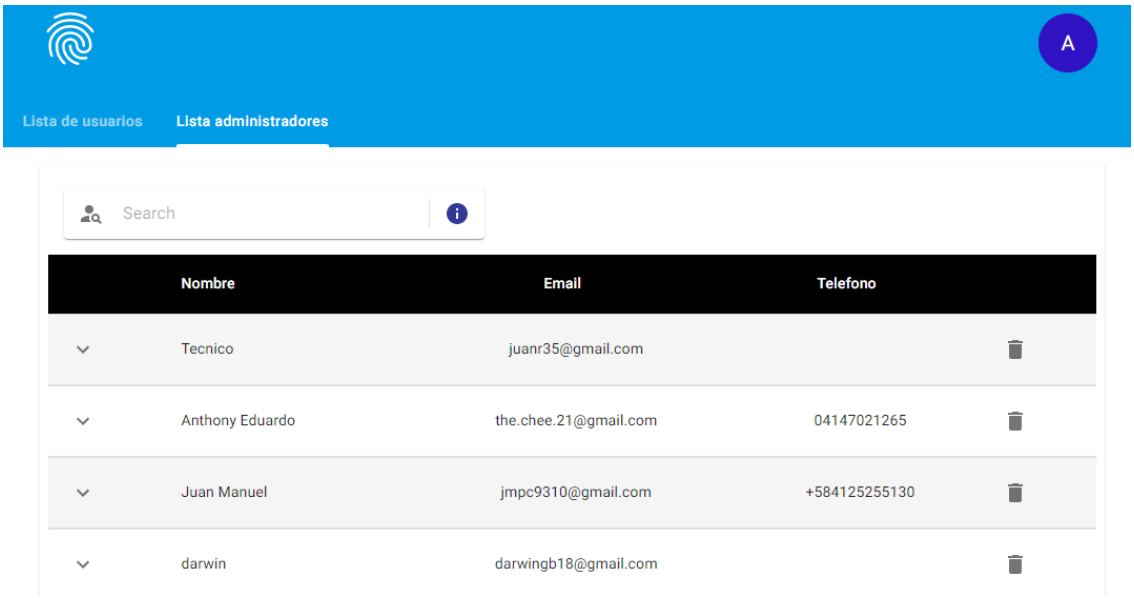

Figura 11. Interfaz de administrador de usuarios registrados.

4.9 Permisos de Acceso

Un administrador principal posee un apartado grafico donde puede consultar las cuentas de los demás administradores y modificar los permisos que estos van a tener de acuerdo a la categoría, ya sea de escritura o de lectura. Para ilustrar mejor esto podemos plantear la situación en donde el administrador principal asigna permisos solo de lectura al entrenador de la categoría sub-13, este únicamente solo puede visualizar los datos y conocer las características de cada jugador, por otro lado, está el médico del club el cual necesita modificar o escribir observaciones medicas de los jugadores, a este el administrador principal debe asignar permisos tanto de lectura como de escritura para poder desempeñarse correctamente. Cabe destacar que no tiene sentido asignar permisos de escritura y no de lectura a un administrador para una misma categoría, porque el administrador no lograra visualizar las cuentas incluso si posee permisos para editarlas.

Entonces podemos tener en cuenta que existen cuentas de usuarios que vienen a ser las cuentas de los jugadores del club, luego vienen las cuentas con permisos de acceso, la de mayor privilegio es la del administrador principal que posee acceso total a los datos de cuentas registradas y puede asignar y modificar los permisos de administradores secundarios, los cuales tienen ciertos permisos de acceso y no pueden modificar sus permisos ni el de otros administradores, un administrador principal no puede tampoco visualizar ni editar la cuenta de otro administrador principal, ya para esto el desarrollador debe modificarlo directamente en la base de datos.

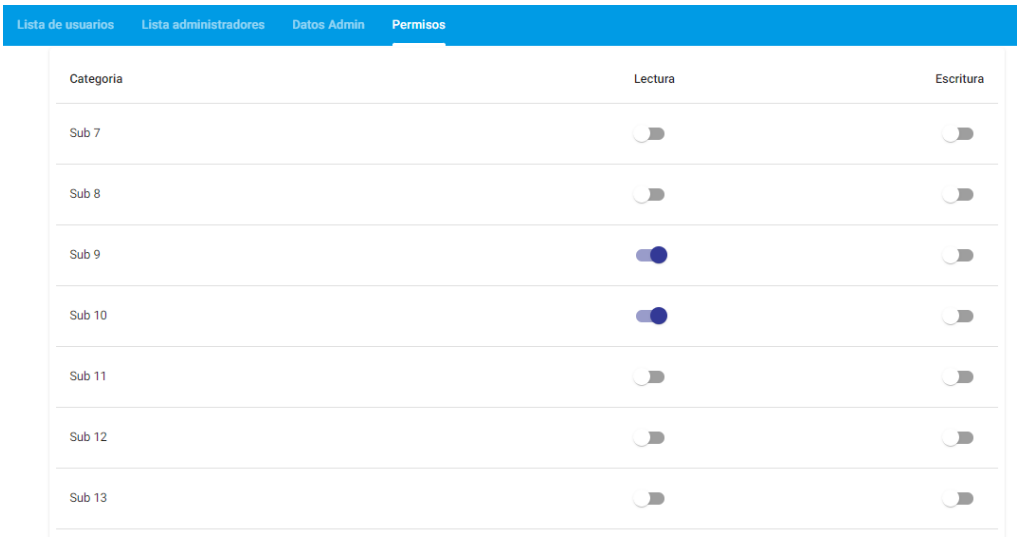

Figura 12. Menú de asignación de permisos del administrador.

4.10 Comunicación mediante Sockets

Hasta ahora no se ha hablado de un componente muy importante para el funcionamiento de la plataforma como lo son los sockets. La librería usada en cuestión es Socket.IO la cual permite la comunicación en tiempo real y la transmisión bidireccional de datos entre el servidor y el cliente. Todo el flujo de datos y la ejecución de acciones que el cliente realiza después de recibir los componentes de la página son llevados a cabo a través de sockets.

4.11 Servidor Backend

4.11.1 Punto de Acceso API

Aquí es donde entra en juego el servidor desplegado en el proveedor de Railway o Render que crea una conexión en tiempo real basada en WebSockets entre el servidor y el cliente. Este tipo de conexión antes mencionada no se puede llevar a cabo en servidores de Vercel, ya que estos no soportan la implementación de sockets debido a que implementan funciones sin servidor las cuales deben ejecutarse en un periodo de tiempo corto y luego destinar los recursos a otras tareas, esto entra en conflicto con la implementación de los sockets, ya que estos ameritan tiempos de ejecución prolongados en servidores que deben mantener una conexión constante con el cliente. Vercel recomienda ciertas alternativas en cuanto al uso de sockets en arquitecturas serverless pero para este caso, se tomó esta decisión por ser un despliegue más tradicional y con mayor soporte.

4.11.2 Comunicación entre Servidores

Las funciones que cumplen los sockets también varían según el tipo de usuario, pero cabe acotar, que sin importar el tipo de cuenta los mecanismos de validación vienen siendo los mismos, el servidor desplegado en Railway maneja las conexiones de sockets y también atiende solicitudes API para la gestión de los datos, en ambos casos se debe comprobar la legitimidad del usuario. Como se mencionó antes, la pieza fundamental para la autenticación de usuarios en esta plataforma son las cookies, pero inicialmente esto hace que surja un error en cuanto a la validación de usuarios entre los servidores, mejor dicho, que no se puede realizar, ya que las configuraciones que tienen ambos servidores no permiten el envió de las cookies entre ellos ya que estos poseen distintos dominios.

Detallando mejor esto, tenemos un servidor desplegado en Vercel el cual provee un dominio al cual los usuarios acceden y obtienen los recursos de la página, luego tenemos un servidor desplegado en otro proveedor (ya sea Railway o Render) que automáticamente se conecta con los usuarios autenticados que acceden a la plataforma mediante una conexión websocket y aparte de eso, también está a la escucha de llamadas api que los usuarios le hagan, entonces este servidor cuando recibe la solicitud de algún usuario para comprobar la legitimidad de mismo, envía la cookie que este posea al servidor alojado en Vercel mediante una llamada api entre servidores, el servidor de Vercel le dice al de Railway si la cookie por la que se le está haciendo la consulta es legítima, esto se debe a que el servidor de Railway no tiene manera de comprobar dicho resultado por sí solo ya que es el servidor de Vercel el que asigna las cookies a los usuarios y posee los hashes para desencriptar el contenido de las mimas, aunado a esto como se describió antes la librería de NextAuth se encarga de este proceso y dicha librería no es compatible con Node el cual es ejecutado en el servidor de Railway.

4.11.3 Solicitudes CORS

Ahora bien, para lograr el proceso de autenticación entre el cliente y los dos servidores debemos tener en cuenta ciertas configuraciones, ya que se efectúa una solicitud de origen cruzado (Cross-Origin Resource Sharing) o CORS entre otras cosas. Entonces primero debemos permitir que el usuario pueda acceder a los recursos del servidor de Railway desde un dominio diferente (en este caso el dominio provisto por Vercel), para esto habilitamos las solicitudes CORS en la configuración del servidor al momento de instanciarlo, para que acepte llamadas de origen cruzado únicamente del dominio en el que se aloja la plataforma. Esto produce que el servidor modifique las

cabeceras http para que el navegador tenga permitido enviar solicitudes al servidor de Railway.

4.12 Mapeo de Dominios

Después de lograr una comunicación exitosa entre el servidor de Railway y el cliente, este debe pasarle la cookie que posee, naturalmente el navegador del usuario no enviara una solicitud con la cookie al dicho servidor porque este se encuentra en un dominio diferente, una solución fue utilizar la función `rewrite` de Next.js que sirve para reescribir la URL de una solicitud entrante antes de que llegue al servidor. Esto permite que una URL sea mapeada a otra URL o recurso sin que el cliente se dé cuenta de la reescritura. De este modo podemos hacer que el cliente en vez de realizar una solicitud directamente al servidor de Railway la haga a una ruta que especifiquemos en el servidor de Vercel por ejemplo “example.vercel/express” y esa solicitud posteriormente se redirigirá al servidor de Railway, de esta manera se logra que el navegador envié la cookie de sesión ya que la URL pertenece al mismo dominio. El uso de la función rewrite permite la unificación de las distintas partes de la plataforma bajo una misma URL ya que la sección de creación de cuentas y autentificación está aislada del apartado donde el usuario registra su información o la gestiona, ambas implementaciones están desplegadas en Vercel pero cada una es independiente y posee su propio dominio o URL, pero con ayuda de la función rewrites el acceso a todos los recursos se mapea a una única dirección a través de subdominios.

4.13 Envió de Cookies

El mapeo de direcciones que establece la función rewrite brinda una solución al problema planteado anteriormente en cuanto a solicitudes api, pero surge un problema para las conexiones de sockets, en estos no podemos utilizar la reescritura de direcciones y por ende se debe definir directamente el envió de la cookie a través del canal del WebSocket, aquí la configuración de la propiedad httpOnly por defecto que posee la cookie impide que sean accedidas o manipuladas por JavaScript en el lado del cliente. Para mitigar esto se establece la propiedad httpOnly en la implementación del servidor a “false”, este es el único cambio sensible que se efectúa en la configuración de las cookies y es necesario para que desde en cliente se pueda enviar la cookie al servidor de Railway y ser procesada. Esta acción se ejecuta cada vez que el cliente abre una nueva conexión WebSocket con el servidor.

Nombre	Valor	Domain	Path	Expires / ...	Tamaño	HttpOnly	Secure	SameSite
__Secure-next-auth.session-token	eyJhbGciOiJkaXIiLCJlbmMiOiJBMjU2R0NNIn0..sUoHaQh7...	auth-use...	/	2023-05-...	375		✓	Lax
__Secure-next-auth.callback-url	https%3A%2F%2Fauth-user-production.up.railway.app	auth-use...	/	Sesión	80	✓	✓	Lax
__Host-next-auth.csrf-token	2f91730ee18cb80d0bb4b66af895062fa0d186ca2a5ef3f1c...	auth-use...	/	Sesión	158	✓	✓	Lax

Figura 13. Cookies almacenadas en el navegador.

4.14 Uso de Sockets

4.14.1 Transporte de Información

Solucionados los problemas de autorización y conexión del cliente con los servidores, podemos explicar las funciones que cumplen los sockets en el sistema, estos forman una red entre los clientes que se encuentren conectados y el servidor, una de las funciones de los sockets es informar el éxito o fallo de una operación, por ejemplo, cuando un usuario introduce algún dato personal y presiona el botón de guardar, los datos se envían por medio de una llamada api al servidor de Railway, este efectúa la verificación del usuario y después se conecta a la base de datos para guardar o modificar la información, si todo este proceso acaba exitosamente se envía una respuesta a través de la conexión WebSocket al cliente que por medio de la interfaz gráfica es alertado que la operación fue completada con éxito, o en caso contrario que algo salió mal. Normalmente este tipo de procesos se realizan simplemente con la llamada POST a la api, en la cual según la respuesta sabemos el resultado de la operación, pero esto resulta exitoso cuando se efectúan llamadas breves a la api, que pasa si en lugar de enviar algunas cadenas cortas de texto se envía un imagen la cual es mucho más pesada, en ese caso las pruebas que se hicieron en varias ocasiones excedían en tiempo de espera de una llamada POST a la api, resultando en una respuesta al cliente fallida a pesar de que la operación se completara exitosamente. Para este tipo de eventos en donde la acción de subir una imagen se prolongue por más de 30 segundos no hay que preocuparse de que alguna llamada caduque por exceso de tiempo ya que la conexión del socket puede prolongarse indefinidamente.

4.14.2 Comunicación en Tiempo Real

Los sockets también mantienen la información actualizada en tiempo real con el ultimo estado de la base de datos, a manera de ejemplo si se diera el caso en el que un jugador accede a la plataforma y al mismo tiempo un administrador edita algún dato perteneciente a la cuenta del jugador antes mencionado, ese dato se actualiza tanto en la vista del jugador como en la del administrador ya que el servidor después de realizar cierto cambio en la base de datos envía una señal por medio de sockets a la cuenta modificada en cuestión si se encuentra conectada en ese momento y también a todas las cuentas administrativas que estén conectadas y que tengan permisos de lectura de la cuenta que se modificó, los señales varían, pueden ser alertas que informan el estado de culminación de una operación o también pueden ser datos provenientes de la base de datos que serán actualizados en todas las interfaces conectadas. Las alertas incluso pueden referirse a la eliminación de alguna cuenta, en dicho caso los datos de la base de datos son eliminados, también la cuenta deja de visualizarse en las listas de los administradores y en caso de estar conectada su sesión es cerrada automáticamente.

Hasta ahora se han descrito las funciones que cumplen los sockets al momento de actualizar algún dato, pero los sockets también cumplen una tarea de suma importancia en el apartado de los administradores la cual es suministrar el listado de cuentas a la que tiene acceso un administrador. Ilustremos mejor esto, cuando un administrador accede a la plataforma tiene un apartado en el que visualiza todas las cuentas a la que tiene acceso, esta información podría cargarse al principio mediante una llamada api como si se solicitaran los datos del propio administrador, pero que pasaría si existiesen unos 5000 registros o más a los cuales el administrador tiene acceso, evidentemente no se puede cargar toda esa data en una única llamada, la mejor manera de obtener los datos es por medio de un cursor ya que cuando se recupera un documento el cursor avanza al siguiente documento en la colección. Esto permite acceder y manipular los documentos de manera eficiente y escalable. Ahora bien, en este caso el cursor solo puede ser manejado por el servidor y no por el cliente, por lo que se emplean los sockets para enviar cada documento recuperado por el cursor al cliente, pudiendo de esa manera obtener colecciones de datos sin sobrecargar los recursos del sistema.

4.15 Frameworks de Diseño

Finalmente pasemos a nombrar las herramientas utilizadas en el apartado gráfico, para la sección de autenticación se utilizó el framework de Bootstrap el cual tiene una librería llamada reactstrap, que incluye componentes de Bootstrap preestilizados para React y también proporciona integraciones para manejar eventos y actualizar el estado. De igual modo para el resto de la aplicación se utiliza Material UI, la cual es una biblioteca de componentes de interfaz de usuario (UI) para también de React. Esta librería facilita el diseño de una estética moderna y minimalista, con colores brillantes y planos, y sombras para crear una sensación de profundidad, de igual modo proporciona una amplia variedad de componentes de UI predefinidos, como botones, barras de navegación, tarjetas, formularios, diálogos y muchos otros. Estos componentes están diseñados para ser altamente personalizables y adaptativos a diferentes dispositivos y tamaños de pantalla.

Capítulo 5
Conclusiones y Recomendaciones

5.1 Conclusiones

En este trabajo se describe el desarrollo del software de la plataforma virtual deportiva de la Academia Emeritense F.C. enfocándose en la implementación de las herramientas de programación utilizadas. Se detalla el despliegue en diferentes proveedores de servicios de computación en la nube y bases de datos. La aplicación cuenta con dos interfaces de autenticación y una base de datos con diversas tablas

para almacenar datos relacionados con las cuentas de usuarios registrados, información personal, datos de los padres del jugador y datos médicos, entre otros. Se explica el manejo de la autenticación y el registro mediante protocolo OAuth y credenciales (correo y contraseña), el cambio de contraseña, las verificaciones de usuario y la gestión de cookies de sesión en servidores de Vercel como funciones sin servidor en conjunto con la nube de Mongo Atlas. También se detalla cómo se manejan las sesiones de usuario y cómo se personaliza la información que se almacena en las cookies de sesión para adaptar la gestión de sesiones y el comportamiento de la plataforma.

Además, se menciona que la aplicación utiliza la librería NextAuth para la autenticación y se explica que la aplicación cuenta con una gestión diferenciada entre administradores y usuarios convencionales, lo que permite el manejo adecuado de las solicitudes según el tipo de usuario que las realiza. En general, se detallan los procesos y algoritmos utilizados en la aplicación para garantizar su correcto funcionamiento y la gestión adecuada de la información de los usuarios y de la plataforma en sí.

Los usuarios pueden almacenar su información personal, la cual es gestionada por el club mediante los administradores. Los administradores tienen diferentes niveles de permisos de acceso y un administrador principal tiene acceso total a los datos de las cuentas registradas, pudiendo asignar o modificar permisos de administradores secundarios. La comunicación en tiempo real y la transmisión bidireccional de datos entre el servidor y el cliente se lleva a cabo a través de sockets usando la librería Socket.IO.

En cuanto a la gestión de permisos de acceso, se menciona que un administrador principal posee un área gráfica donde puede consultar las cuentas de los demás administradores y modificar los permisos que estos van a tener de acuerdo a la categoría, ya sea de escritura o de lectura.

El servidor backend desplegado en Railway o Render crea una conexión en tiempo real basada en WebSockets entre el servidor y el cliente, lo que no es posible en servidores de Vercel debido a que implementan funciones sin servidor que no pueden mantener una conexión constante con el cliente. Se habilitan las solicitudes CORS en la configuración del servidor de Railway además del uso de otras herramientas como rewrite que permiten que el navegador envíe solicitudes al servidor de Railway. Los sockets forman una red entre los clientes que se encuentran conectados y el servidor, y una de sus funciones es informar el éxito o fallo de una operación en tiempo real, lo que es fundamental para la gestión adecuada de la información de los usuarios y de la plataforma en sí. También se encargan del transporte de la información

En cuanto a los frameworks de diseño, se utilizaron las librerías de reactstrap y Material UI para el diseño de la interfaz visual de la plataforma, los cuales proporcionan una amplia variedad de componentes de interfaz de usuario predefinidos y personalizables.

5.2 Recomendaciones

Esta plataforma, al igual que muchos proyectos de desarrollo de programación, ofrece una amplia variedad de opciones y herramientas para que los desarrolladores puedan construir cada componente del mismo. Como resultado, los mecanismos que implemente el sistema pueden variar enormemente según las herramientas utilizadas, cada una con sus pros y contras.

Teniendo en cuenta lo anterior, se plantean dos alternativas que, a simple vista para los usuarios, no suponen ningún cambio, pero que a nivel interno buscan implementar un sistema más compacto y, según el caso, con un menor gasto de recursos. Como se describió en capítulos anteriores, el sistema hace uso de servidores de backend de distintos proveedores, lo que conlleva el manejo y mantenimiento de diferentes servidores que son fundamentales para el funcionamiento de la plataforma. Una recomendación es tener un sistema más centralizado y volcar todos los procesos de backend en un único servidor. Aquí es donde se presentan las dos alternativas mencionadas:

La primera opción es trasladar todos los procesos de backend al servidor de Vercel. En este caso, se debe descartar el uso de sockets, ya que Vercel implementa una arquitectura sin servidor. Para suplir esto, la misma página de Vercel recomienda algunos servicios de terceros (Ably, Convex, Pusher, PubNub, Firebase, TalkJS, Supabase) que permiten establecer una comunicación en tiempo real de la misma manera que funciona la plataforma. Sin embargo, el uso de estos servicios implica algunas consideraciones: algunos de ellos son de pago, otros requieren el uso de una base de datos diferente, como Firebase, y finalmente, estos servicios de terceros establecen un servidor privado que siempre está a la escucha para manejar la comunicación en tiempo real entre el cliente y el servidor. Es decir, algo parecido al servidor propio implementado en Railway. Cabe destacar que estas herramientas son más especializadas y ahorran al desarrollador el mantenimiento del servidor.

Otra opción es descartar el uso de las funciones sin servidor y volcar todos los procesos de backend al servidor de Railway. El principal inconveniente de esto es que ya no se podrá hacer uso de la librería NextAuth para la autenticación de los usuarios. Debido a que el servidor ejecuta una instancia de Node, se pueden implementar otras alternativas como Passport. De esta manera, se logra que la plataforma funcione del

mismo modo haciendo uso de un servidor convencional sin utilizar las funciones sin servidor. Aunque igual se utiliza y despliega el servidor de Vercel para servir la página al usuario, esto también podría ser cambiado por el desarrollador sin mayor problema y optar por simplemente desarrollar el apartado gráfico en React únicamente (en vez de Next).

Estas alternativas permiten simplificar los componentes internos de la plataforma, hay otras cosas que se dejan de lado, pero se tratan estas en especial por ser las que influyen directamente en el despliegue de la aplicación.

Referencias

Zofío Jiménez, J. (2013). Aplicaciones web. Madrid. Madrid: ES: Macmillan Iberia, S.A.

Hernández Diaz, L. R. (2012). Un modelo para la implementación de la seguridad de una aplicación Web con el uso de la programación orientada a aspectos.

Red Hat .(31 de Octubre de 2017). ¿Qué es una API?https://www.redhat.com/es/topics/api/what- are-application-programming-interfaces

DataScientest (7 de abril 2022). MongoDB : todo sobre la base de datos NoSQL orientada a documentos. https://datascientest.com/es/mongodb-todo-sobre-la-base-de-datos-nosql-orientada-a- documentos

MDN contributors. (20 de abril de 2021). JavaScript : ¿Quieres convertirte en un desarrollador web front-end? https://developer.mozilla.org/es/docs/Web/HTML

Henry (1 de nov. de 2021). ¿Qué es Node.js y para qué se utiliza?. https://blog.soyhenry.com/que-es-node-js-y-para-que-se-utiliza

RACKSPACE TECHNOLOGY (s.f). ¿Qué son las bases de datos NoSQL?.https://www.rackspace.com/es/library/what-is-a-nosql-database

Wikipedia (s.f). React. https://es.wikipedia.org/wiki/React

Yulima Hernandez. (28 de nov. de 2022). ¿Qué es Next js y para qué sirve? https://www.dongee.com/tutoriales/next-js-que-es-y-para-que-sirve/

Héctor Julio Fúquene Ardila. (4 de nov. de 2011). Implementación cliente servidor mediante sockets https://core.ac.uk/download/pdf/229160848.pdf

Merge-Gyms. (20 de mayo de 2022). AUTENTICACIÓN VS AUTORIZACIÓN https://github.com/fernandosegoviadev/Merge-Gyms#tokens-de-autenticaci%C3%B3n

e/

Cassie Bodnar. (29 de mayo. de 2013). Todo lo que necesitas saber sobre las cookies https://www.kaspersky.es/blog/todo-lo-que-necesitas-saber-sobre-las-cookies/892/

Tatiana Grapsas. (16 de septiembre de 2018). ¿Qué es cloud computing o computación en la nube? https://rockcontent.com/es/blog/computacion-en-la-nube/