

Título: Club Atletismo Maracena

Autor: Daniel Escobar Molina

El proyecto consiste en el desarrollo de una página web informativa para el Club de Atletismo Maracena, utilizando React y Firebase. El objetivo principal es proporcionar a los miembros y al público en general información actualizada sobre eventos, entrenamientos, resultados y noticias del club, aparte de poder contactar e inscribirse de forma online. La plataforma está diseñada para ser intuitiva y accesible, ofreciendo una experiencia de usuario óptima en diferentes dispositivos. Con la implementación de Firebase, se asegura una gestión eficiente de los datos y una integración fluida con las aplicaciones móviles. Este proyecto mejorará significativamente la comunicación y la organización dentro del club, facilitando el acceso a la información relevante para todos los interesados.

The project involves the development of an informative website for the Maracena Athletics Club, using React and Firebase. The main objective is to provide members and the general public with updated information about events, training sessions, results, and club news. The platform is designed to be intuitive and accessible, offering an optimal user experience across different devices. With the implementation of Firebase, efficient data management and seamless integration with mobile applications are ensured. This project will significantly enhance communication and organization within the club, making relevant information easily accessible to all interested parties.

Índice

4. Justificación:	2
a. Características generales	2
b. Restricciones generales, si las hay	3
c. Aspectos a cubrir, junto con los que no se van a tratar	3
d. Estudio de las prestaciones de la herramienta que se propone frente a otras existentes en la misma categoría	3
5. Justificación de la tecnología empleada	3
6. Requerimientos hardware y software	5
7. Análisis y diseño	6
Diagrama de casos de uso	7
Diagrama de clases:	8
b. Base de datos no relacional:	9
8. Implementación	11
9. Evaluación y prueba	12
10. Manual de estilos en el que se recojan	13
a. Sketches páginas principales	13
b. Criterios de Accesibilidad	17
c. Criterios de Usabilidad Diseño intuitivo y navegación clara. Consistencia en el diseño y la interacción. Feedback claro para acciones del usuario (por ejemplo, al enviar un formulario). Minimización de la carga cognitiva para los usuarios.	17
d. Tipografía Fuentes principales: Bebas Neue en los títulos de las secciones, Montserrat, en la foto inicial de los componentes y textos de todo el proyecto. Tamaños de fuente: el tamaño de las funetes de los títulos suele ser responsivo para que se ajuste al ancho de la pantalla, y el de los textos es de 16px.	18

e. Mapa de colores	18
f. Dispositivos/Vistas para las que se ha Diseñado el Proyecto	18
11. Software utilizado	18
12. Mejoras posibles y aportaciones	19

4. Justificación:

a. Características generales

Interfaz intuitiva y amigable: Diseñada con React para asegurar una experiencia de usuario fluida y responsiva.

Accesibilidad: Compatible con dispositivos móviles y de escritorio, asegurando que los usuarios puedan acceder a la información desde cualquier lugar.

Funcionalidades específicas: Secciones dedicadas a eventos, entrenamientos, resultados, y noticias del club.

Seguridad: Implementación de medidas de seguridad para proteger los datos de los usuarios y la integridad de la información.

b. Restricciones generales, si las hay

Dependencia tecnológica: La plataforma depende de los servicios proporcionados por Firebase, lo que puede implicar limitaciones en función de los costos y la disponibilidad del servicio.

c. Aspectos a cubrir, junto con los que no se van a tratar

Aspectos a cubrir:

Información actualizada del club: Eventos, entrenamientos, resultados y noticias.

Interacción con miembros: Sección de contacto y formularios para feedback.

Registro y gestión de miembros: Sistema básico de registro de usuarios para recibir actualizaciones y boletines.

Aspectos que no se van a tratar:

Comercio electrónico: No se incluirá una tienda en línea para la venta de productos del club.

Aplicaciones móviles nativas: No se crearán aplicaciones móviles independientes; el enfoque será en una web responsiva.

d. Estudio de las prestaciones de la herramienta que se propone frente a otras existentes en la misma categoría.

React vs. Angular/Vue:

Ventajas: React ofrece una curva de aprendizaje más sencilla y una comunidad de desarrollo amplia, lo que facilita la resolución de problemas y la implementación de nuevas funcionalidades.

Desventajas: Puede requerir más configuraciones iniciales comparado con frameworks como Angular que son más completos.

Firestore vs. Backend tradicional :

Ventajas: Firestore proporciona una solución integral con autenticación, base de datos en tiempo real, y alojamiento, lo que acelera el desarrollo y simplifica la arquitectura del sistema.

Desventajas: Dependencia en un proveedor de servicios externos y posibles costos elevados en función del uso y la escalabilidad.

5. Justificación de la tecnología empleada

Tecnologías Utilizadas

React

Firestore

HTML5 y CSS3

TypeScript

Git y GitHub

React

Descripción: React es una biblioteca de JavaScript para construir interfaces de usuario, desarrollada por Facebook. Es conocida por su enfoque en componentes reutilizables y su capacidad para manejar la vista de manera eficiente con un virtual DOM.

Justificación: React fue elegido por su flexibilidad, rendimiento y gran comunidad de apoyo. Su arquitectura basada en componentes facilita la creación de una interfaz de usuario modular y mantenible. Además, la popularidad de React asegura una gran cantidad de recursos y bibliotecas de terceros que pueden acelerar el desarrollo.

Firestore

Descripción: Firestore es una plataforma de desarrollo de aplicaciones móviles y web desarrollada por Google. Proporciona una variedad de servicios como base de datos en tiempo real, autenticación, almacenamiento y hosting.

Justificación: Firestore se seleccionó por su capacidad para proporcionar una solución integral que simplifica la gestión del backend. La base de datos, el almacenamiento y la autenticación son particularmente útiles para aplicaciones que requieren actualizaciones en tiempo real, almacenar archivos y gestión de usuarios. La integración fluida con React y su facilidad de uso permiten un desarrollo más rápido y eficiente.

HTML5 y CSS3

Descripción: HTML5 es el lenguaje de marcado estándar para crear páginas web, mientras que CSS3 es el lenguaje de estilos utilizado para diseñarlas y darles formato.

Justificación: HTML5 y CSS3 son tecnologías fundamentales para cualquier desarrollo web. Se utilizaron por su capacidad para crear páginas web estructuradas y estilizadas, proporcionando una base sólida para la aplicación React. La compatibilidad con navegadores modernos y la capacidad de crear diseños responsivos son cruciales para asegurar una buena experiencia de usuario.

TypeScript

Descripción: TypeScript es un lenguaje de programación desarrollado y mantenido por Microsoft. Es un superconjunto de JavaScript que añade tipos estáticos y otras características avanzadas.

Justificación: TypeScript se eligió por su capacidad para mejorar la calidad y mantenibilidad del código. La tipificación estática ayuda a detectar errores en tiempo de desarrollo, lo que reduce los bugs y facilita el refactoring. Además, su integración con React es excelente, permitiendo un desarrollo más robusto y predecible.

Git y GitHub

Descripción: Git es un sistema de control de versiones distribuido, y GitHub es una plataforma de alojamiento de código que facilita la colaboración y el control de versiones.

Justificación: Git y GitHub se utilizaron para gestionar el código fuente del proyecto, permitiendo un control de versiones eficiente y colaboración entre los desarrolladores. Estas herramientas son esenciales para cualquier proyecto de desarrollo moderno, asegurando la integridad del código y facilitando el trabajo en equipo.

6. Requerimientos hardware y software

Requerimientos del Cliente (Usuario Final)

Hardware:

Dispositivo: Ordenador de escritorio, portátil, tablet o smartphone.

Procesador: Cualquier procesador moderno compatible con la navegación web.

Memoria RAM: Mínimo 2 GB.

Almacenamiento: Suficiente espacio para el sistema operativo y navegador web.

Conectividad: Conexión a Internet estable (mínimo 1 Mbps).

Software:

Sistema Operativo: Windows, macOS, Linux, iOS o Android.

Navegador Web:

Google Chrome .

Mozilla Firefox.

Microsoft Edge.

Safari .

Otros navegadores compatibles con HTML5, CSS3.

Requisitos adicionales: Compatibilidad con JavaScript habilitada, soporte para cookies y almacenamiento local (localStorage).

Requerimientos del Servidor

Hardware:

CPU: Procesador de múltiples núcleos.

Memoria RAM: Mínimo 4 GB.

Almacenamiento: Espacio de almacenamiento suficiente para el sistema operativo, la base de datos, archivos de aplicación y registros .

Conectividad: Conexión a Internet de alta velocidad con suficiente ancho de banda para manejar múltiples solicitudes concurrentes.

Software:

Sistema Operativo: Linux, o Windows.

Servidor Web:

Firebase Hosting: Utilizado para desplegar la aplicación web.

Base de Datos:

Firebase Firestore: Base de datos NoSQL en tiempo real utilizada para almacenar y sincronizar datos.

Entorno de Desarrollo y Lenguajes:

TypeScript: Lenguaje utilizado para el desarrollo de la aplicación.

Control de Versiones y Gestión de Código:

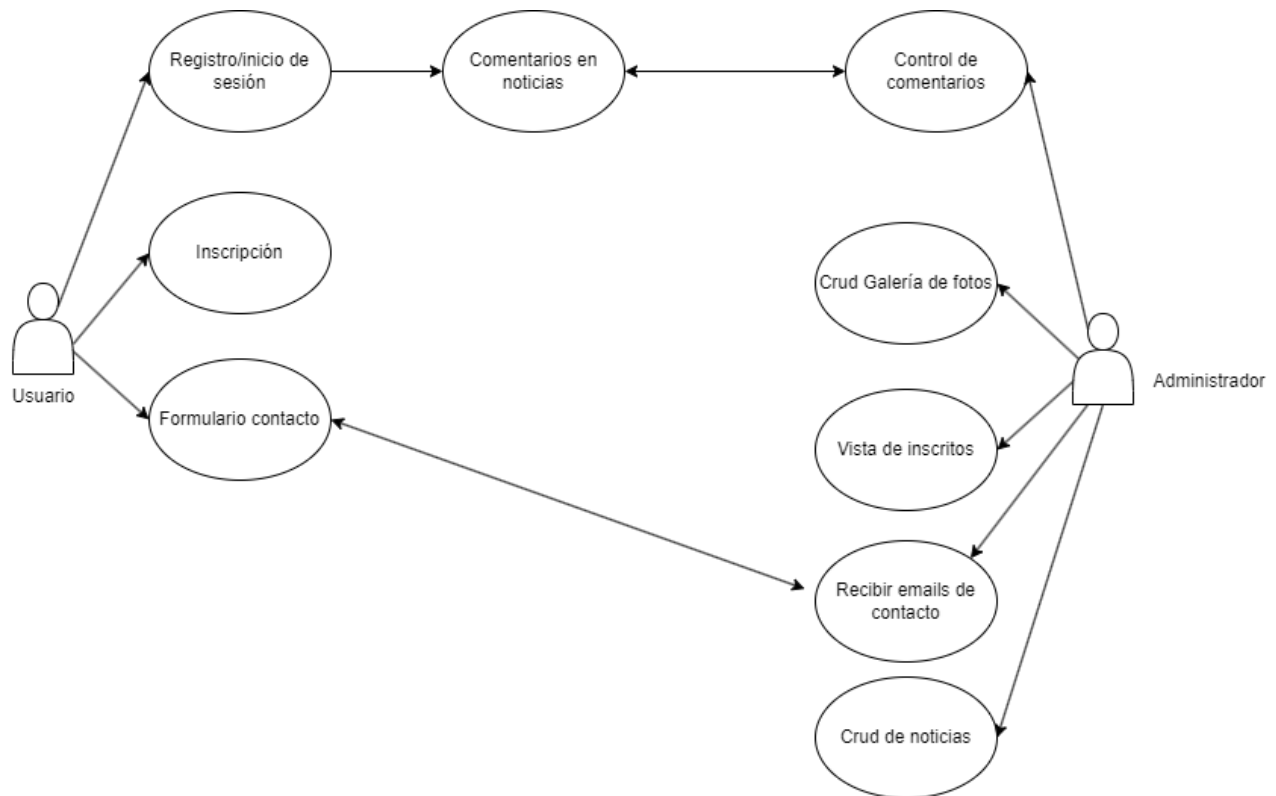
GitHub: Plataforma de alojamiento de repositorios.

Requisitos adicionales:

Firebase CLI: Herramienta de línea de comandos de Firebase para desplegar y gestionar el proyecto.

Certificados SSL: Para asegurar la comunicación entre el cliente y el servidor.

7. Análisis y diseño

Diagrama de casos de uso**Actores:****Usuario**

Representa a los usuarios finales del sitio web, que pueden ser miembros del club, atletas, o visitantes interesados en la información proporcionada por el club.

Administrador

Representa a los administradores del sitio web, responsables de gestionar el contenido y las funcionalidades del sitio web.

Casos de Uso para el Usuario**Registro/Iniciar sesión**

Permite a los usuarios registrarse en el sitio web o iniciar sesión en sus cuentas existentes.

Inscripción

Los usuarios pueden inscribirse el club mensualmente.

Formulario de contacto

Los usuarios pueden enviar consultas o mensajes al club a través de un formulario de contacto.

Comentarios en noticias

Los usuarios pueden dejar comentarios en las noticias publicadas en el sitio web.

Casos de Uso para el Administrador

CRUD de noticias

El administrador puede crear, leer, actualizar y eliminar noticias en el sitio web.

CRUD de galería de fotos

El administrador puede gestionar la galería de fotos del sitio web, incluyendo la creación, lectura, actualización y eliminación de imágenes.

Vista de inscritos

Permite al administrador ver la lista de usuarios inscritos en los eventos o actividades del club.

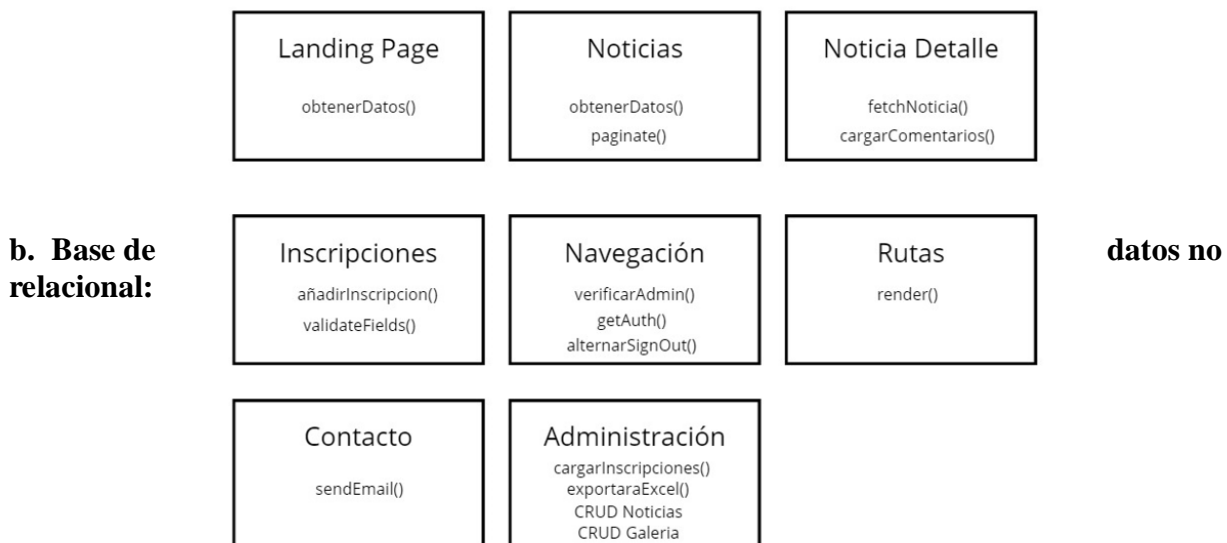
Recibir emails de contacto

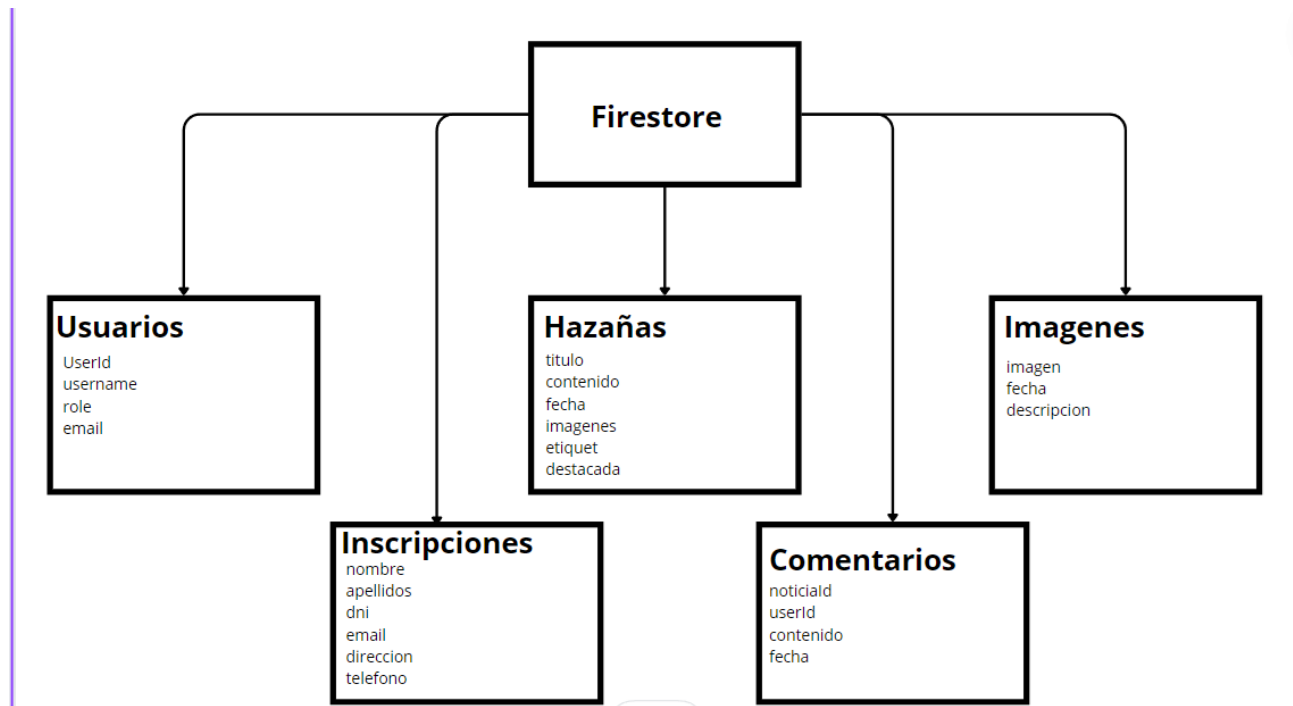
El administrador puede recibir y gestionar los correos electrónicos enviados a través del formulario de contacto.

Control de comentarios

El administrador puede moderar los comentarios dejados en las noticias, incluyendo la aprobación, eliminación o respuesta a los comentarios.

Diagrama de clases:





Usuarios

userId: INTEGER (Identificador único del usuario, clave primaria).

username: VARCHAR(50) (Nombre de usuario).

role: VARCHAR(20) (Rol del usuario, puede ser 'usuario' o 'administrador').

email: VARCHAR(100) (Dirección de correo electrónico del usuario).

Descripción: Esta tabla se ha hecho para asignar roles a los usuarios. De este modo, se diferencia de un rol básico del administrador.

Hazañas

titulo: VARCHAR(100) (Título de la noticia).

contenido: TEXT (Contenido detallado de la noticia).

fecha: DATE (Fecha de publicación de la noticia).

imagenes: TEXT (URLs de las imágenes asociadas a la noticia, almacenadas como una lista separada por comas).

etiqueta: VARCHAR(50) (Etiquetas o categorías asignadas a la noticia).

destacada: BOOLEAN (Indica si la noticia está destacada, true/false).

Descripción: Esta tabla se ha hecho para guardar todas las noticias del club a lo largo de su recorrido. Algo resaltable es el campo destacada que es booleano, y sirve para poner destacada una noticia más relevante. De esta forma tiene otro formato en la página.

Imágenes

imagen: VARCHAR(255) (URL de la imagen).

fecha: DATE (Fecha en que la imagen fue subida).

descripcion: TEXT (Descripción de la imagen).

Descripción: Esta tabla se ha hecho para guardar las fotos del club que se van a exponer en la galería.

Inscripciones

nombre: VARCHAR(50) (Nombre del inscrito).
apellidos: VARCHAR(100) (Apellidos del inscrito).
dni: VARCHAR(20) (Documento Nacional de Identidad del inscrito).
email: VARCHAR(100) (Correo electrónico del inscrito).
direccion: VARCHAR(255) (Dirección del inscrito).
telefono: VARCHAR(20) (Número de teléfono del inscrito).

Descripción: Esta tabla se ha hecho para guardar los datos de las personas que se quieran inscribir en el club.

Comentarios

noticiaId: INTEGER (Identificador de la noticia a la que pertenece el comentario, clave foránea).
userId: INTEGER (Identificador del usuario que hizo el comentario, clave foránea).
contenido: TEXT (Contenido del comentario).
fecha: TIMESTAMP (Fecha y hora en que se hizo el comentario).

Descripción: Esta tabla se ha hecho para guardar los comentarios de las noticias. Sus campos permiten saber qué usuario lo ha puesto y cuándo.

8. Implementación

Hojas de Estilo:

CSS Modules: Utilizados para estilizar los componentes de React de manera modular y evitar conflictos de nombres de clases. React permite importar estilos CSS directamente en componentes individuales.

Plantillas:

Componentes de React: Utilizados para estructurar la interfaz de usuario de manera reutilizable y modular. Por ejemplo, componentes como Header, Footer, Navigation, etc., para mostrar diferentes partes de la información del club.

Formularios:

Uso de formularios HTML dentro de componentes de React para capturar información de los usuarios o administradores del club.
Uso de **EmailJS** para captar los emails.

Funciones de Envío de Datos:

Firebase SDK: Uso de las funciones proporcionadas por Firebase SDK para enviar datos a la base de datos en tiempo real Firestore o en la base de datos en tiempo real de Firebase.

Conexión y Consultas a la Base de Datos:

Firestore: Uso de Firestore para almacenar y consultar datos estructurados como noticias, imágenes, miembros del club, etc.

Selección de la Base de Datos:

Firestore: Ideal para datos estructurados y consultas más complejas.

Uso de Fichero de Configuración:

Archivo de Configuración: Existe un archivo `firebase.ts` para configurar la aplicación de React para que se conecte a Firebase.

9. Evaluación y prueba

Pruebas Unitarias y de Integración:

Prueba de que los componentes rendericen correctamente, manejen el estado y las props adecuadamente, y que respondan correctamente a eventos y cambios.

Pruebas de Funcionalidad:

Pruebas para verificar el funcionamiento de los formularios.

Simular entradas de usuario y verifica que la validación de campos (si la hay) funcione correctamente.

Ejemplo: En el formulario de añadir noticias, verificar que no se permitan campos vacíos o que la longitud de los campos sea la adecuada.

Pruebas de Rendimiento:

Evaluación el rendimiento de la aplicación utilizando Lighthouse para verificar la velocidad de carga, la accesibilidad y las mejores prácticas.

Pruebas de Interfaz de Usuario (UI):

Pruebas manuales para verificar que la interfaz de usuario sea intuitiva y fácil de usar para los usuarios finales.

Asegurarse de que los estilos se apliquen correctamente y que la navegación entre páginas o secciones funcione sin problemas.

Pruebas de Navegación:

Verificar que la navegación entre diferentes secciones o páginas de la aplicación sea coherente y que los enlaces y botones funcionen como se espera.

Testeo de la Base de Datos (Firebase)

Integridad de Datos:

Verificar que los datos almacenados en Firebase Firestore sean correctos y consistentes.

Pruebas para asegurar que los datos se almacenen correctamente al agregar, modificar o eliminar registros desde la aplicación web.

Consultas y Operaciones CRUD:

Ejecución de consultas para obtener datos y verifica que devuelvan los resultados esperados.

Prueba de operaciones de creación, lectura, actualización y eliminación (CRUD) en la base de datos para asegurar que funcionen correctamente.

Seguridad y Autenticación:

Prueba de que los roles y permisos de acceso funcionen según lo previsto.

Verificación de que solo los usuarios autorizados puedan realizar ciertas operaciones CRUD o acceder a ciertos datos.

Funciones de Verificación de Campos y Realimentación al Usuario

Validación de Campos:

Implementación de funciones de validación en los formularios para verificar la entrada de datos del usuario.

Manejo adecuado de casos como campos requeridos, longitud de datos, formato de datos (por ejemplo, fechas), etc.

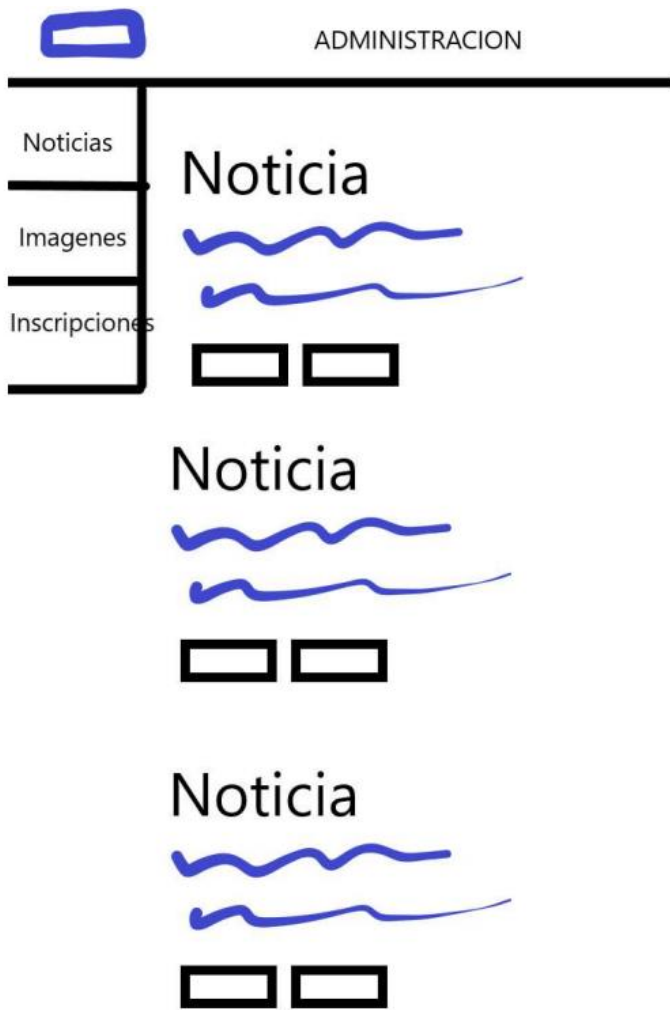
Realimentación al Usuario:

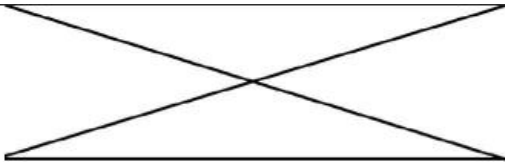
Proporción de mensajes claros y útiles para informar al usuario sobre el estado de sus acciones.

Por ejemplo, muestra mensajes de éxito al agregar una nueva noticia o errores específicos si hay problemas con los datos ingresados.

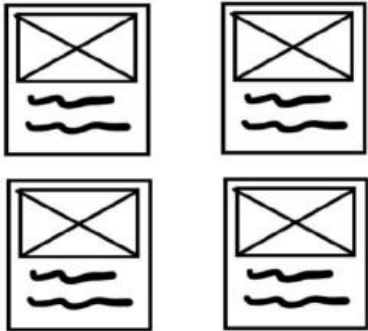
10. Manual de estilos en el que se recojan

a. Sketches páginas principales

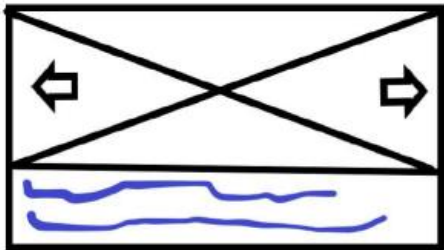




Sección

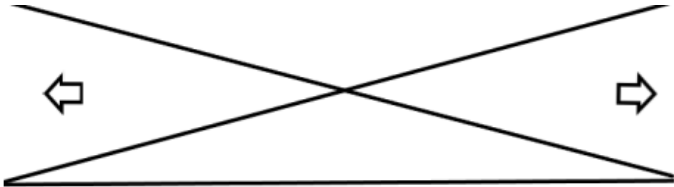


Sección



Sección

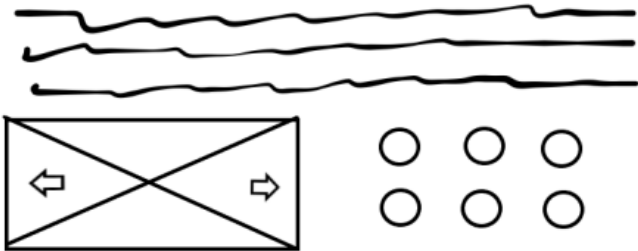




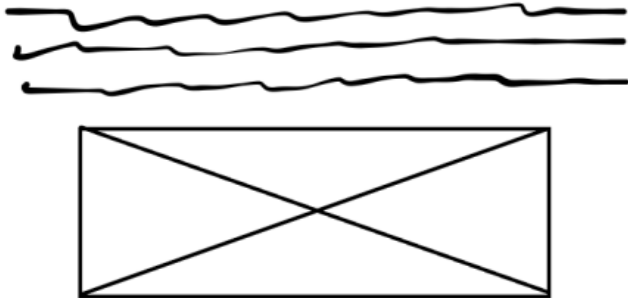
Seccion

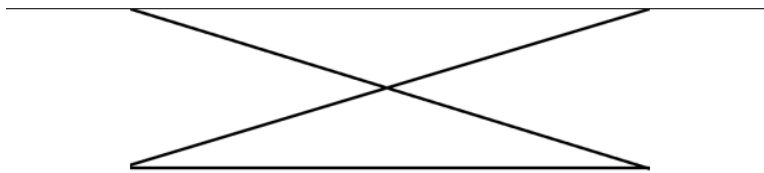


Seccion

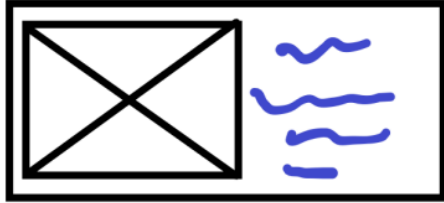


Seccion

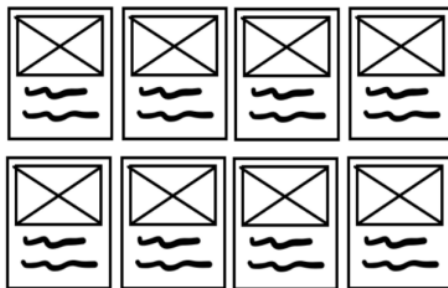




Sección



Sección



1 2 .. 10

b. Criterios de Accesibilidad

Uso de colores con suficiente contraste.
 Texto alternativo para imágenes.
 Navegación fácilmente accesible mediante teclado.

c. Criterios de Usabilidad

Diseño intuitivo y navegación clara.
 Consistencia en el diseño y la interacción.
 Feedback claro para acciones del usuario (por ejemplo, al enviar un formulario).
 Minimización de la carga cognitiva para los usuarios.

d. Tipografía

Fuentes principales: Bebas Neue en los títulos de las secciones, Montserrat, en la foto inicial de los componentes y textos de todo el proyecto.

Tamaños de fuente: el tamaño de las fuentes de los títulos suele ser responsivo para que se ajuste al ancho de la pantalla, y el de los textos es de 16px.

e. Mapa de colores

#1F1F1F 31, 31, 31 negro grisáceo. Se aplica al menú, las letras, alguna sección, y el footer.

#FFFFFF 255, 255, 255 blanco. Se aplica a todo el background del menu y a las letras con background en negro.

#4062f7 64, 98, 247 azul. Se aplica a los rectángulos de los títulos, enlaces, y diversos detalles.

f. Dispositivos/Vistas para las que se ha Diseñado el Proyecto

Diseño Responsivo: Para los tres tipos de vistas, móvil, tablet y ordenador.

Tecnología Usada

Frontend: React para la creación de componentes dinámicos y reactivos.

Backend: Firebase para el almacenamiento de datos y la gestión de usuarios.

Estilos: CSS .

Herramientas de Diseño: Sketch.

11. Software utilizado

Editor de Código:

Visual Studio Code: Utilizado como entorno de desarrollo principal para escribir y editar código JavaScript, JSX, HTML, CSS, etc.

Gestión de Versiones:

Git: Utilizado para el control de versiones del código fuente de la aplicación, permitiendo el seguimiento de cambios y colaboración en equipo.

Librerías y Frameworks:

React: Biblioteca de JavaScript utilizada para construir la interfaz de usuario de manera dinámica y reactiva.

React Router: Utilizado para la gestión de las rutas y la navegación dentro de la aplicación React.

Firebase SDK: SDK proporcionado por Firebase para integrar Firebase Authentication, Firestore, y otras funcionalidades de Firebase en la aplicación web.

Gestión de Dependencias y Paquetes:

npm: Utilizado para gestionar las dependencias de la aplicación, instalar paquetes de npm, y

ejecutar scripts de construcción y desarrollo.

Herramientas de Estilo y Diseño:

CSS Modules: Utilizados para el manejo de estilos CSS a nivel de componente en React, evitando conflictos de nombres y facilitando el mantenimiento del código.

Herramientas de Pruebas:

Lighthouse: Utilizado para evaluar el rendimiento, la accesibilidad, las mejores prácticas y la SEO de la aplicación web.

Herramientas de Despliegue y Hosting:

Firebase Hosting: Utilizado para desplegar la aplicación web y alojarla en los servidores de Firebase, proporcionando una URL pública accesible.

Herramientas de Comunicación y Colaboración:

GitHub: Utilizado para almacenar el repositorio de código, gestionar problemas, etc.

12. Mejoras posibles y aportaciones

Implementación de Funcionalidades Adicionales:

Calendario de Eventos: Integrar un calendario interactivo donde los usuarios puedan ver y registrar eventos deportivos próximos.

Notificaciones: Implementar notificaciones para informar a los usuarios sobre nuevas noticias, eventos o actualizaciones importantes.

Documentación y Comentarios:

Documentar el código fuente de manera clara y completa, incluyendo comentarios donde sea necesario para facilitar la comprensión y la colaboración futura.

Optimización del Rendimiento:

Cache: Implementa estrategias de caching para contenido estático y dinámico utilizando Firebase Hosting y servicios de CDN.