# INFORME DEL PROYECTO: "APP WEB DE NOTICIAS"

**Introducción**

Este informe describe el desarrollo de una aplicación web diseñada para proporcionar noticias organizadas en categorías y accesibles a los usuarios de forma dinámica e interactiva. La App busca convertirse en una fuente confiable y accesible de información, destacándose por su diseño intuitivo, estructura eficiente y capacidad para adaptarse a las necesidades de los usuarios.

# Objetivos del Proyecto Objetivo General:

* Desarrollar una aplicación web intuitiva y atractiva que permita a los usuarios acceder a noticias categorizadas de manera eficiente y personalizada.

# Objetivos Específicos:

* Crear una interfaz intuitiva que facilite la navegación entre categorías de noticias.
* Implementar funcionalidades avanzadas como búsquedas por categoría, agregar noticias a favoritos y refrescar contenido en tiempo real.
* Diseñar una estructura de datos clara y flexible, optimizada para futuras expansiones.
* Integrar un sistema de navegación eficiente, adaptado a dispositivos móviles y escritorio.
* Incorporar funcionalidades que promuevan la interacción del usuario, como botones de redes sociales y marcadores personalizados.

# Descripción del Proyecto

La aplicación web utiliza una estructura de datos JSON organizada para manejar las noticias. Cada noticia incluye propiedades como título, descripción, imágenes, cuerpo, categorías y créditos del autor. Además, cuenta con diversas funcionalidades que hacen de esta app una herramienta interactiva y moderna:

# Componentes Principales:

* **Card:** Muestra detalles del clima y las fechas asociadas con las noticias.
* **Buttons:** Facilitan la navegación entre categorías y páginas específicas.
* **List:** Proporciona un menú desplegable con opciones organizadas por categorías. App Bar: Incluye una barra de herramientas con buscador integrado y opciones de navegación.
* **Bottom Navigation:** Permite agregar noticias a favoritos y refrescar contenido.
* **Footer:** Contiene información de contacto, redes sociales y derechos de autor.
* **Speed Dials:** Incluye botones adicionales para funcionalidades especiales.

# Funcionalidades adicionales a Futuro

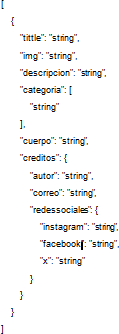
Se espera el futuro integrar, en nuestra página web de noticias características avanzadas para mejorar la experiencia del usuario, y ofrecer un contenido más personalizado y accesible.

* **Modo oscuro:** donde los usuarios puedan alternar entre temas claro y oscuro.
* **Favoritos:** Opciones para guardar noticias preferidas.
* **Filtros:** Búsqueda avanzada por categorías y palabras clave.
* **Interacción social:** Enlaces directos a redes sociales.

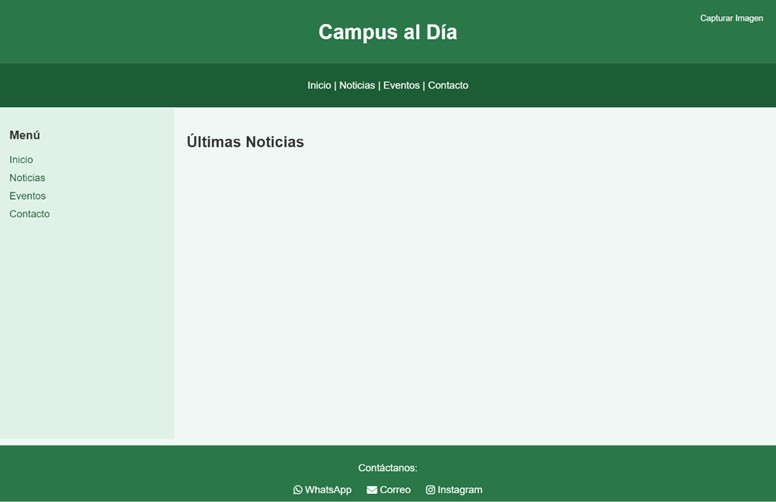
# Página Principal:

* **Carrusel de Noticias Destacadas:** para presentar noticias relevantes en un formato visual atractivo.
* **Sistema de Publicación:** Las noticias se almacenaran en un formato JSON, lo que facilitara la manipulación y la presentación de datos.
* **Actualización Dinámica:** Los datos se obtendrán de APIs externas y se presentaran en tiempo real

# Estructuración.

Su estructura json que se empleara para la aplicación web comienza con un arreglo que contiene varios objetos que contienen las siguientes propiedades o se estructura de esta manera:

# Mockup:



**Componentes que se planean usar:**

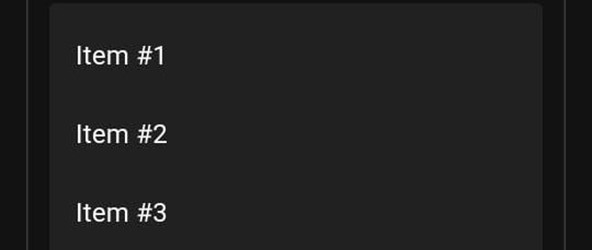
* **Card:** Este componente se utilizará para mostrar detalles sobre el clima y las fechas, aunque estos datos pueden estar sujetos a cambios durante el proceso de desarrollo de la App.



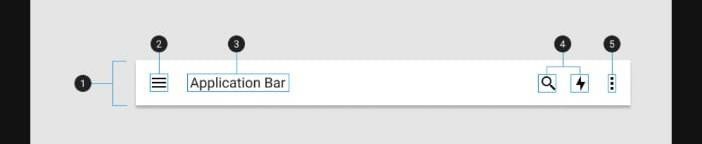
* **Buttons:** Se utilizarán para la navegación entre páginas, como las diferentes categorías, y podrían incluirse en menús desplegables u otros ajustes de navegación.



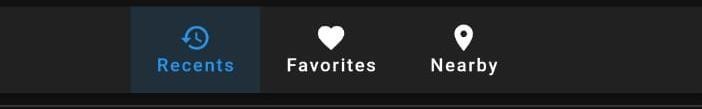
* **List:** Este componente se empleará para una lista de menús desplegables que estará ubicada en el lado izquierdo de la App.



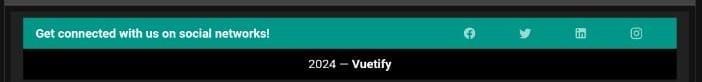
* **App Bar:** En conjunto con el componente "List" y una parte del "App Bar", se creará una lista de categorías. Otras partes del App Bar, como la lupa, se utilizarán exclusivamente para la búsqueda de noticias.



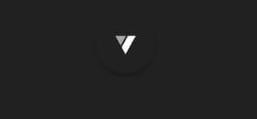
* **Bottom Navigation:** Este componente permitirá al usuario agregar elementos a favoritos y refrescar la página.



* **Footer:** Incluirá los contactos de las redes sociales (Facebook, Instagram y número de WhatsApp), junto con el copyright, logo y nombre de la app. El orden de estos elementos está aún en desarrollo.



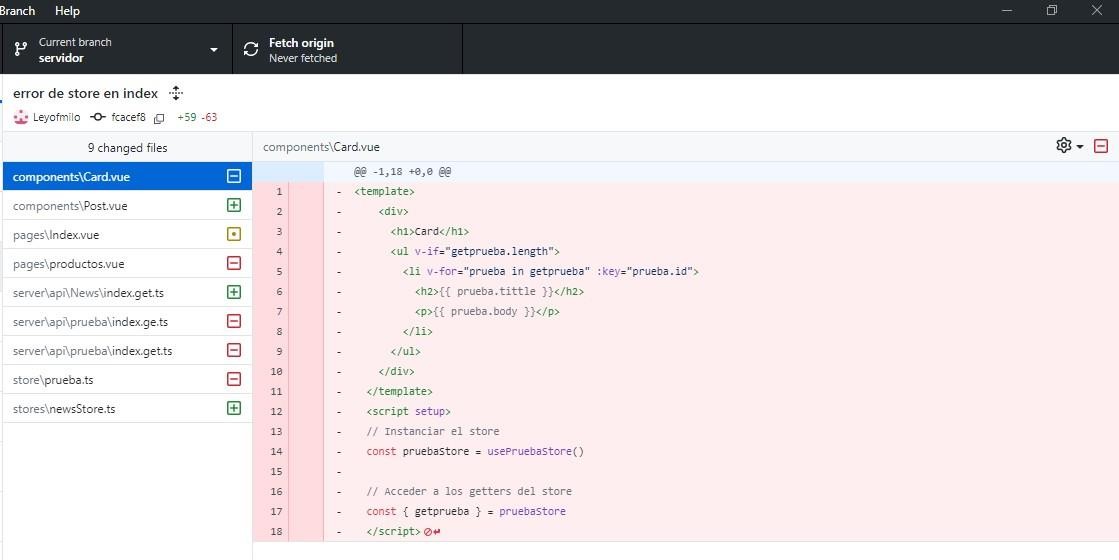
* **Speed Dials:** Se usará para añadir botones adicionales, aunque aún no se ha definido exactamente qué botones se incluirán; esto se decidirá conforme avance el proyecto**.**



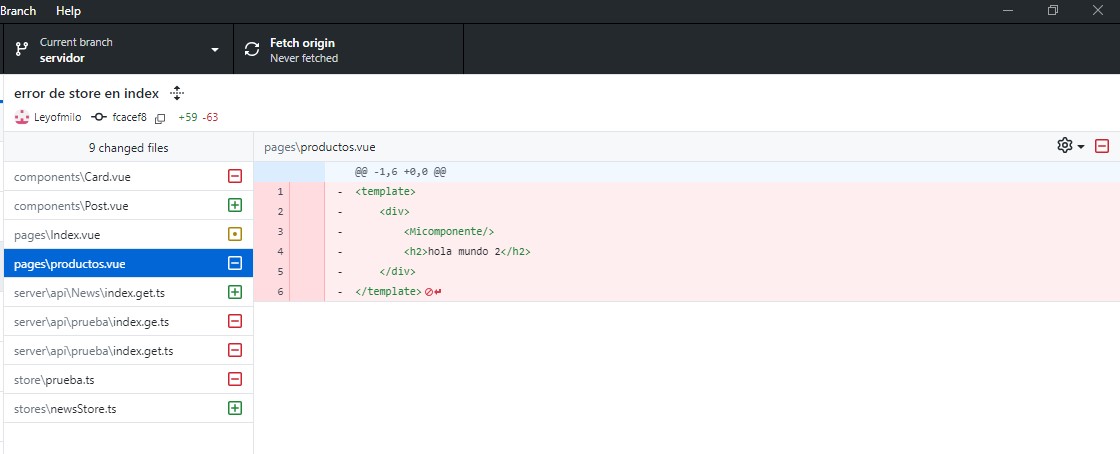
# NOTA: Todos los componentes mencionados pueden estar sujetos a cambios durante el desarrollo de la app.

**ALGUNAS MODIFICACIONES.**

* Este componente obtiene una lista de objetos prueba del store pruebaStore. Si la lista contiene elementos, el componente muestra cada uno en una tarjeta con un título y un cuerpo. La lógica de mostrar u ocultar depende de la cantidad de elementos en getprueba.



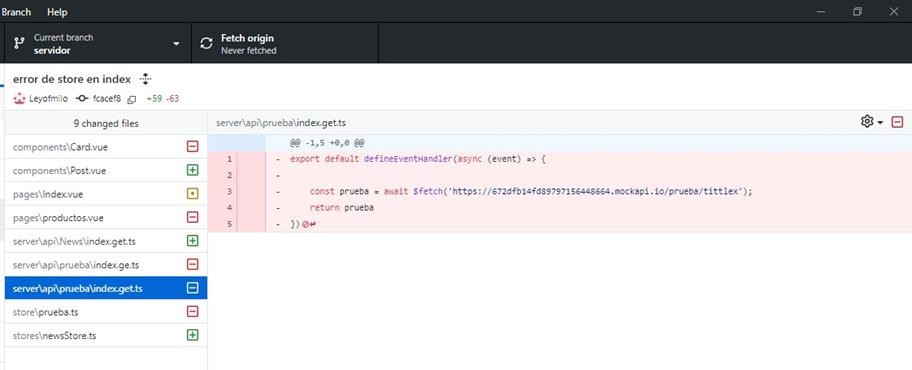
* Este código tiene una estructura básica y solo renderizará el contenido de MiComponente junto con el texto. No tiene ninguna lógica adicional ni comportamiento interactivo, solo sirve para estructurar y mostrar estos elementos.



Este código define un endpoint de API en Nuxt que:

* Realiza una solicitud a una API externa para obtener datos.
* Espera la respuesta de esa API.
* Devuelve los datos obtenidos al cliente que hizo la solicitud.

Cuando alguien accede al endpoint /api/prueba/index, esta función se ejecuta, obtiene los datos de la API externa y devuelve esos datos al cliente. Esto es útil cuando se necesitan datos de otra fuente y se quiere encapsular esa lógica en un endpoint del servidor.

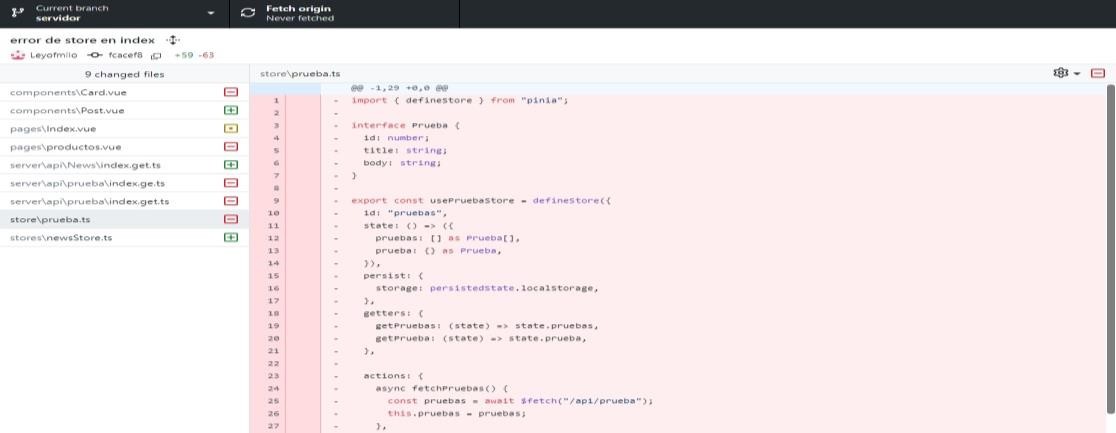


# EXPLICACION DEL CÓDIGO

Este store proporciona una forma centralizada de manejar y almacenar datos de tipo Prueba en la aplicación. Las características clave son:

* + Estado reactivo para almacenar la lista pruebas y el objeto prueba.
  + Persistencia en localStorage para conservar los datos.
  + Getters para acceder al estado.
  + Acción fetchPruebas para obtener datos de una API y actualizar el estado.

Este store facilita la gestión de datos prueba en los componentes Vue, ya que permite acceder y modificar el estado desde cualquier componente que lo use.



# EXPLICACION DEL CÓDIGO.

Utiliza el componente v-list con el modelo de datos opened="open", lo cual indica un control reactivo de apertura/cierre.

Primer elemento: v-list-item con el icono mdi-home y título Home. Grupos de listas (v-list- group):

Grupo "Users":

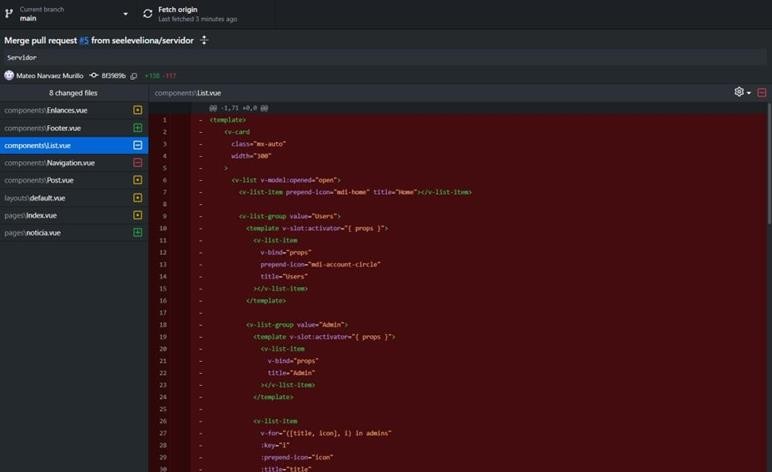
Usa v-slot:activator="{ props }" para manejar la activación del grupo.

Incluye un elemento v-list-item con ícono mdi-account-circle y título Users. Grupo "Admin":

Mismo enfoque que "Users", pero con el título Admin.

Usa una iteración v-for para recorrer un arreglo admins que renderiza los títulos e íconos dinámicamente.

* **Conclusión:** Este componente List.vue está diseñado para mostrar listas organizadas en grupos desplegables, con iconos e información dinámica, probablemente usando Vuetify (por el uso de prefijos v-).



# EXPLICACION DEL CÓDIGO

* Este componente proporciona una barra de navegación personalizada con:
* Un título estático.
* Un icono de navegación.
* Un espacio para personalizar contenido mediante slots.
* La integración del componente Enlaces para manejar o mostrar enlaces de navegación adicionales.

# ESTRUCTURA

Template

<div>: Contenedor principal del componente.

<v-app-bar :elevation="2">: Barra de aplicación con elevación visual de nivel 2.

<template v-slot:prepend>: Slot para insertar contenido antes del contenido principal.

<v-app-bar-nav-icon>: Icono de navegación (generalmente usado para menús laterales).

<v-app-bar-title>: Título de la barra, en este caso "Unicordoba al día".

<Enlaces />: Componente personalizado "Enlaces", que se importa y renderiza dentro de la barra. Script

