

INFORME PROYECTO

Grupo 1

INTRODUCCIÓN

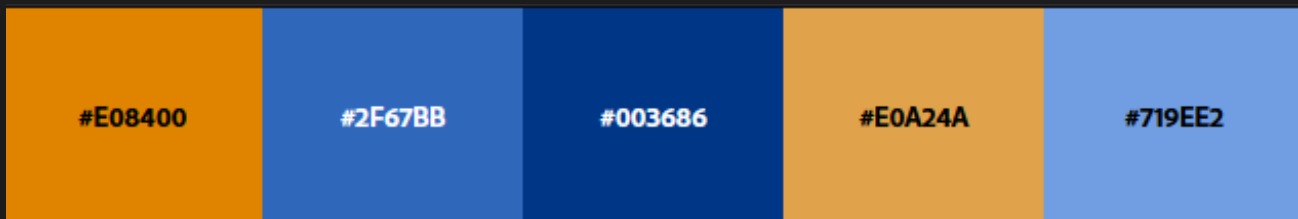
Este documento tiene como objetivo ofrecer una visión detallada y compacta del desarrollo de una aplicación web implementada en REACT.JS, inspirada en el conocido servicio "Netflix". A través de este análisis técnico, pretendemos presentar tanto el diseño como la funcionalidad integral de la aplicación.

OBJETIVO

El propósito fundamental de este documento es proporcionar una guía detallada sobre los pasos esenciales para el desarrollo de una aplicación web basada en las tecnologías fundamentales de React y JavaScript. La idea principal es explicar paso a paso cómo construir y desarrollar este proyecto innovador.

DISEÑO

Consideramos usar como esquema de colores el azul y el naranja; el primero, porque representa la tecnología, confianza, seguridad, calma. Y el segundo porque simboliza diversión, felicidad, modernidad, ocio, energía, entre otras cosas.



Tipografía seleccionada: Roboto, ya que es la más clara, moderna y legible para dispositivos tecnológicos y descansa la vista. Por lo tanto, se emplea Bold para los títulos y subtítulos, semibold para botones y regular para texto, utilizando distintos tamaños acorde al diseño responsivo.

Título

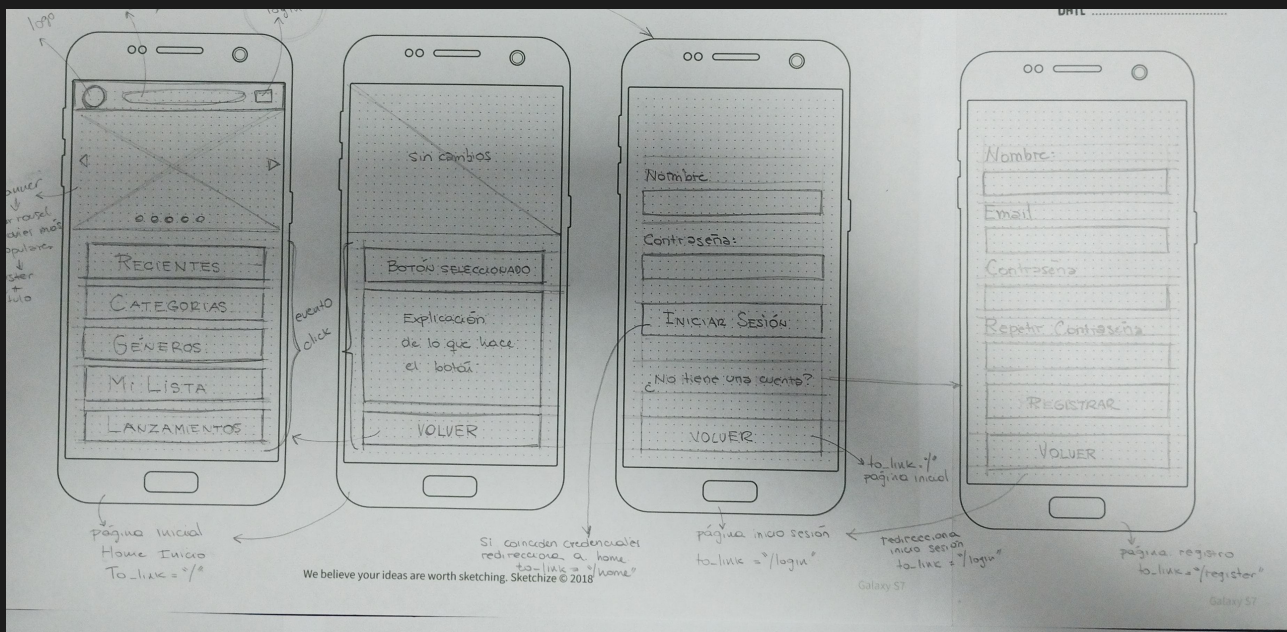
Subtítulo

Botón

Texto

DISEÑO DE INTERFACES

Estos bocetos son la primera visión de nuestra aplicación. Son esbozos iniciales que definen el diseño y estilo de las interfaces. A través de ellos, empezamos a visualizar cómo se verán las páginas y cómo interactuarán los elementos.



VER INTERFACES



VER DISEÑO DE INTERFAZ EN FIGMA

PROTOTIPO

Se desarrolló un prototipo utilizando la herramienta "Figma" para definir el diseño y establecer las bases del proyecto. El enfoque se centra en diversas páginas clave, abarcando tanto la experiencia del usuario invitado como la del usuario logueado.

MOBILE



[VER FIGMA](#)

PROTOTIPO

PÁGINAS PRINCIPALES:

1. LOGIN:

- Interfaz de acceso para los usuarios registrados.

2. REGISTRO:

- Página dedicada a nuevos usuarios para crear cuentas.

3. HOME - INVITADO:

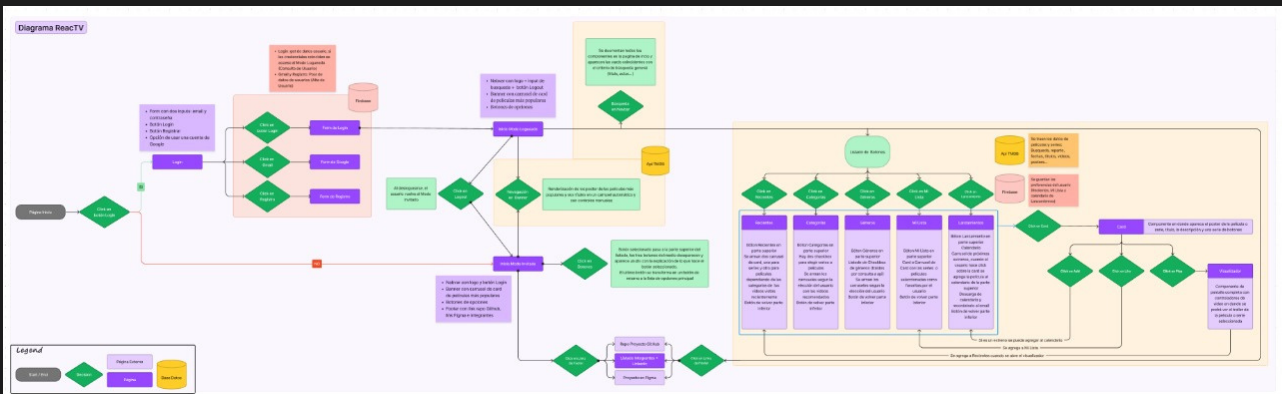
- Recientes: Destaca las últimas incorporaciones.
- Categorías: Organiza el contenido por categorías.
- Géneros: Explora películas por género.
- Mi Lista: Espacio para que los usuarios guarden sus selecciones.
- Lanzamientos: Muestra las últimas novedades.

4. MAIN - LOGUEADO:

- Recientes: Destacando novedades.
- Categorías: Organización por categorías.
- Géneros Lista: Lista detallada para explorar géneros.
- Géneros Carrusel: Presentación visual de géneros.
- Mi Lista: Espacio personalizado para las selecciones del usuario.
- Lanzamientos: Destacando los últimos estrenos.
- Búsqueda: Facilita la búsqueda de contenido.
- Card - Ver Sinopsis: Vista previa de una película con opción para ver la sinopsis completa.
- Card - Con Sinopsis: Vista completa de una película, incluyendo la sinopsis.
- Acerca de: Información sobre la aplicación.
- Contacto: Formulario para ponerse en contacto.

DIAGRAMA DE FLUJO

Este esquema visual actúa como guía, revelando la secuencia lógica de interacciones y direcciones dentro de la aplicación, al mismo tiempo que proporciona una explicación detallada sobre el uso de Firebase y la API. Ofrece una visión rápida y clara de cómo los usuarios navegarán entre las diferentes secciones, simplificando la comprensión de la estructura esencial.



VER DIAGRAMA DE FLUJO

API TMDB

La plataforma de streaming se basa en la API de The Movie Database (TMDB) para obtener información esencial sobre películas y series. Esta API se utiliza para realizar búsquedas de contenidos, proporcionando detalles como títulos, reparto, fechas de lanzamiento, videos y pósters. Este enfoque garantiza que los usuarios accedan a información completa y actualizada sobre los contenidos disponibles en la plataforma.

En la página de inicio, se implementa un banner interactivo que muestra pósters de las películas más populares. La información obtenida de TMDB se utiliza para renderizar los pósters y títulos en un carrusel.

La función de búsqueda en la barra de navegación (Navbar) utiliza la API de TMDB para desmontar todos los componentes de la página de inicio y mostrar tarjetas coincidentes con el criterio de búsqueda general, como títulos o actores.

API TMDB

La sección "Recientes" muestra dos carruseles de tarjetas, uno para series y otro para películas. Estos carruseles se generan basándose en las categorías de videos vistos recientemente por el usuario, utilizando la información de TMDB.

En la sección "Categorías" se ofrecen opciones a los usuarios para elegir entre series o películas mediante checkboxes. También el usuario puede elegir los generos de su interes. La plataforma utiliza la API de TMDB para construir carruseles de tarjetas con contenido recomendado según las preferencias del usuario.

En la sección "Lanzamientos", la plataforma utiliza la API de TMDB para obtener información sobre las fechas de estreno de las próximas películas. Esta información se utiliza para construir un calendario interactivo que muestra los próximos estrenos.

DDBB

En nuestra plataforma de streaming, utilizamos **Firebase** para gestionar la autenticación y personalizar la experiencia del usuario.

Al hacer clic en "Registro", los usuarios completan un formulario cuyos datos se envían a Firebase mediante una solicitud POST. También ofrecemos la opción de registro con Google.

El proceso de inicio de sesión se ejecuta a través de una solicitud GET a Firebase, donde se verifica la autenticidad de las credenciales proporcionadas por el usuario.

En caso de éxito en la autenticación, se accede al "Modo Logueado". En esta fase, se efectúa una segunda solicitud GET a Firebase para recuperar los datos específicos del usuario.

Si la autenticación es exitosa, el "Modo Logueado" brinda acceso a información personalizada del usuario, con sus preferencias almacenadas. Estas preferencias pueden incluir detalles sobre contenido reciente y una lista personalizada de favoritos.

HERRAMIENTAS UTILIZADAS

@emotion/react:	Bibliotecas y paquetes relacionados con Material-UI. Emotion es una biblioteca de estilo para React que se utiliza para escribir estilos CSS en tu aplicación.
@emotion/styled:	Bibliotecas y paquetes relacionados con Material-UI. Esta es otra parte de la biblioteca Emotion, específicamente para la creación de componentes estilizados
@fontsource/roboto:	Bibliotecas y paquetes relacionados con Material-UI. Este paquete proporciona la fuente Roboto
@mui/icons-material:	Bibliotecas y paquetes relacionados con Material-UI. Este paquete contiene iconos predefinidos
@mui/lab:	Bibliotecas y paquetes relacionados con Material-UI.
@mui/material:	Bibliotecas y paquetes relacionados con Material-UI. Esta es la biblioteca principal de Material-UI que contiene todos los componentes de interfaz de usuario y herramientas necesarias
@testing-library/jest-dom:	Bibliotecas y paquetes relacionados con Material-UI. Biblioteca de utilidades para realizar pruebas en entornos Jest.
@testing-library/react:	Bibliotecas y paquetes relacionados con Material-UI. Esta biblioteca se utiliza para realizar pruebas en componentes React
@testing-library/user-event:	Bibliotecas y paquetes relacionados con Material-UI. Esta biblioteca complementa @testing-library/react y proporciona funciones para simular eventos de usuario de manera más realista en las pruebas.

HERRAMIENTAS UTILIZADAS

axios:	Librería de JavaScript para consumir APIs. Es un Cliente HTTP basado en promesas para node.js y el navegador
bootstrap:	Proporciona componentes predefinidos.
emailjs-com:	Biblioteca que facilita el envío de correos electrónicos directamente desde el lado del cliente en una aplicación web, simplificando la integración de la funcionalidad de correo electrónico, como formularios de contacto.
firebase:	Se utiliza para establecer una conexión con Firebase. Esto permita la autenticación de usuarios (loguearse) y la recuperación de datos desde la base de datos de Firebase.
html-to-image:	Convierte elementos HTML a imágenes. Se utiliza para descargar imagenes, como el calendario o los posters.
html2canvas:	Biblioteca para convertir elementos HTML en imágenes, similar a html-to-image. Se utiliza para descargar imagenes, como el calendario o los posters.
react-bootstrap:	Implementación de Bootstrap para React, que proporciona componentes React predefinidos basados en el diseño de Bootstrap.
react-calendar:	Componente de calendario para React.
react-dom:	Biblioteca para manejar el renderizado de componentes React en el DOM.

HERRAMIENTAS UTILIZADAS

TMDB	API llamada al proyecto para la bases de datos de cine y televisión
react-icons:	Proporciona un conjunto de iconos de diversas bibliotecas para utilizar en los componentes.
react-loader-spinner:	Componente React que muestra un indicador de carga o spinner durante la carga de datos o procesos asíncronos.
react-player:	Reproductor multimedia para React que admite varios formatos de archivo y servicios de transmisión. Se utiliza para los trailers.
react-router-dom:	Biblioteca de enrutamiento para React que facilita la navegación y la gestión de la URL
react-youtube:	Componente React para incorporar videos de YouTube. Se utiliza para los trailer
sweetalert2:	Biblioteca que se utiliza para crear ventanas modales (pop-ups) personalizadas. Se utiliza cuando se elimina una card en la sección "Mi Lista" y "Recientes"
video.js:	Un reproductor de video HTML5 personalizable. Se utiliza para los trailers.
web-vitals:	Biblioteca para medir y reportar métricas web vitales, como el tiempo de carga y la interactividad del usuario

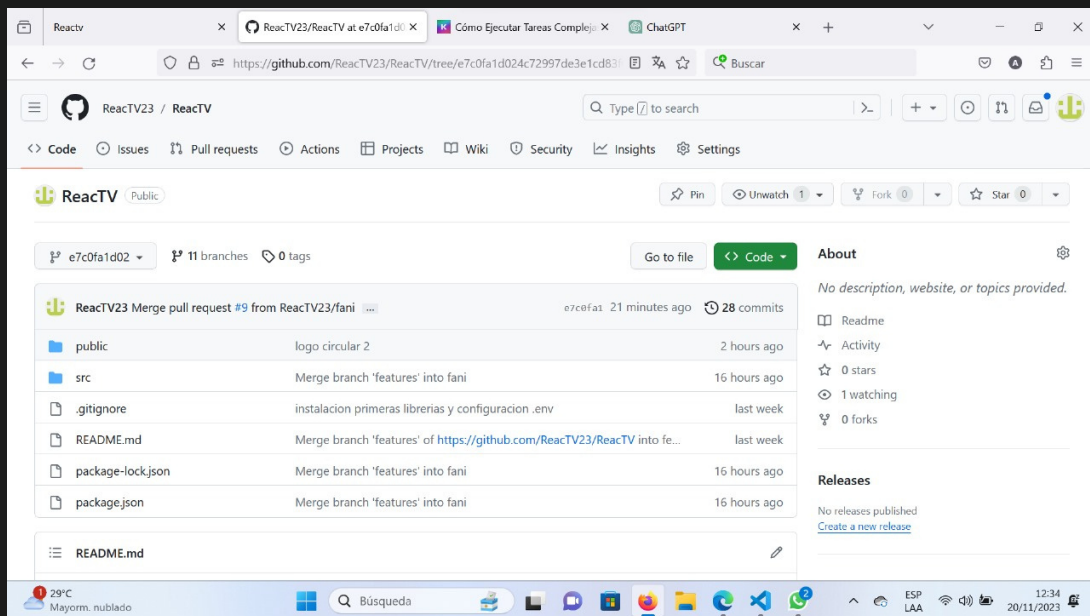
PROBLEMAS

CONFLICTO:

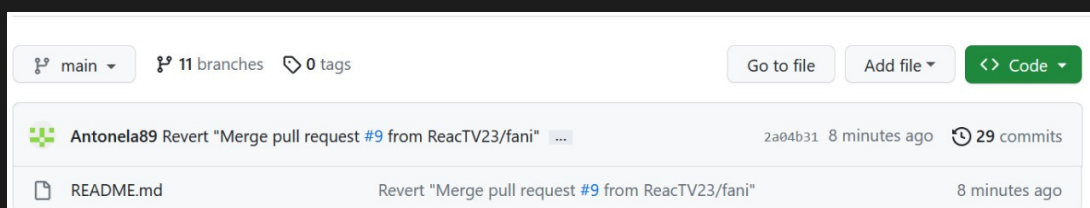
Pull request a main en vez de rama features.

SOLUCIÓN:

Git revert. Revertir el merge: e7c0fa1



```
Revert "Merge pull request #9 from ReactTV23/fani"
PS C:\Users\atena\OneDrive\Escritorio\ReactV-1> git revert e7c0fa1
error: your local changes would be overwritten by revert.
hint: commit your changes or stash them to proceed.
fatal: revert failed
PS C:\Users\atena\OneDrive\Escritorio\ReactV-1> git revert e7c0fa1
error: commit e7c0fa1d024c72997de3e1cd83f0c1af54bdac7b is a merge but no -m option was given.
fatal: revert failed
PS C:\Users\atena\OneDrive\Escritorio\ReactV-1> git revert -m 1 e7c0fa1
>>
[main 2a04b31] Revert "Merge pull request #9 from ReactTV23/fani" This reverts commit e7c0fa1d024c72997de3e1cd83f0c1af
54bdac7b, reversing changes made to 310acea2ebb2a6b88f697000414bcd1e30ae5c74.
41 files changed, 20603 deletions(-)
delete mode 100644 .gitignore
delete mode 100644 package-lock.json
delete mode 100644 package.json
delete mode 100644 public/assets/img/imagen/logo-circular-2.png
delete mode 100644 public/assets/img/imagen/logo-circular.png
delete mode 100644 public/assets/img/imagen/logo-reactv.png
delete mode 100644 public/assets/img/videos/intro-1-usuario.mp4
```



PROBLEMAS

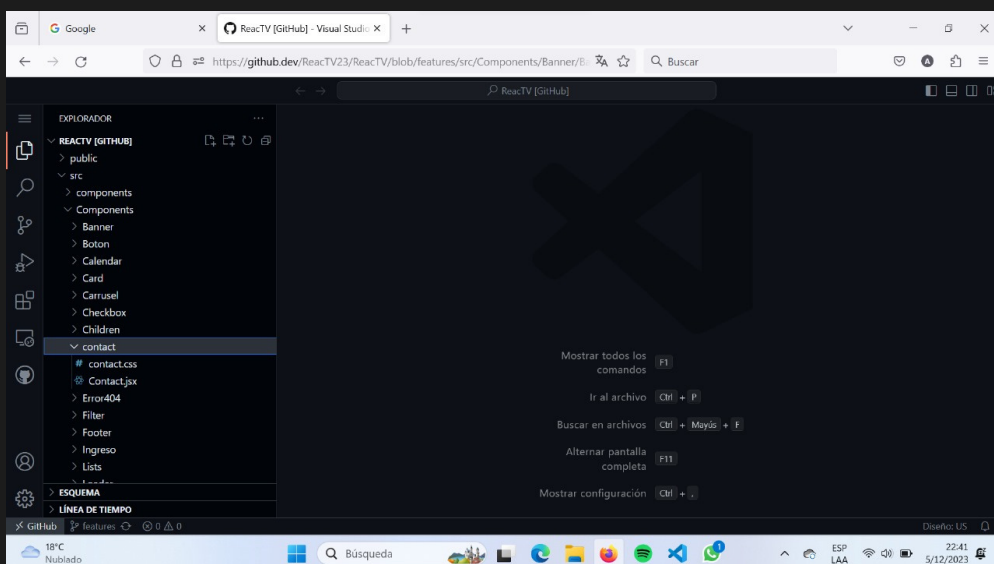
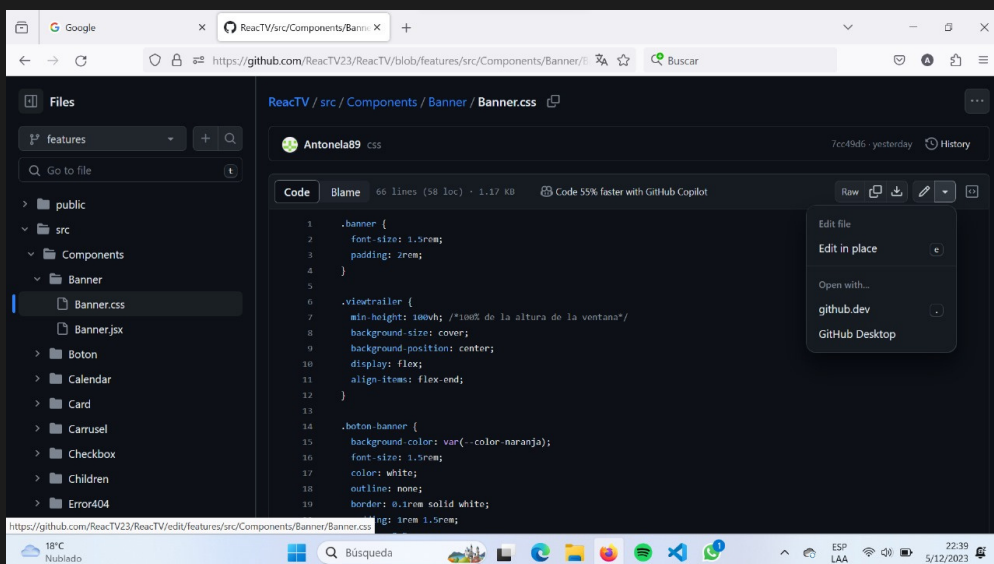
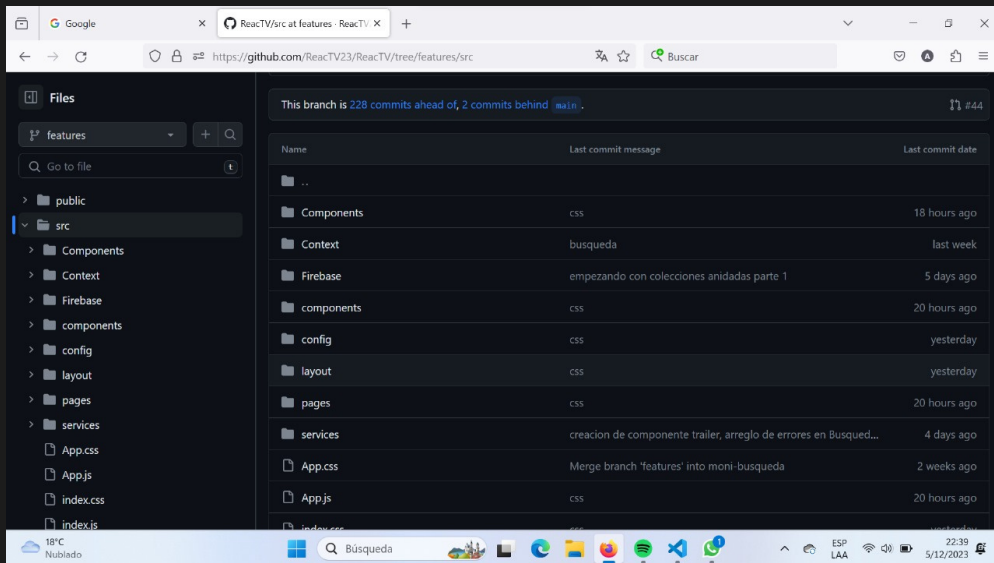
CONFLICTO:

El renombrado de carpetas implica cambios en las rutas a nivel local y remoto, pero la carpeta en sí no se renombra automáticamente localmente. Por ejemplo, al renombrar la carpeta "components" a "Components" en una máquina, se ajusta la ruta correspondiente, se realiza el push, se solicita el pull request y se hace el merge en la rama "features" para compartir el código. Sin embargo, al recibir el código, otros pueden enfrentar errores según cómo tengan la carpeta en su máquina, ya que Git toma la nueva ruta, pero no renombra la carpeta localmente.

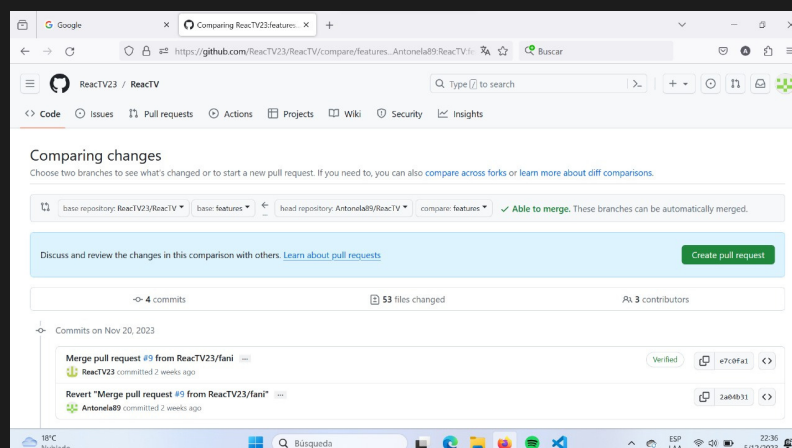
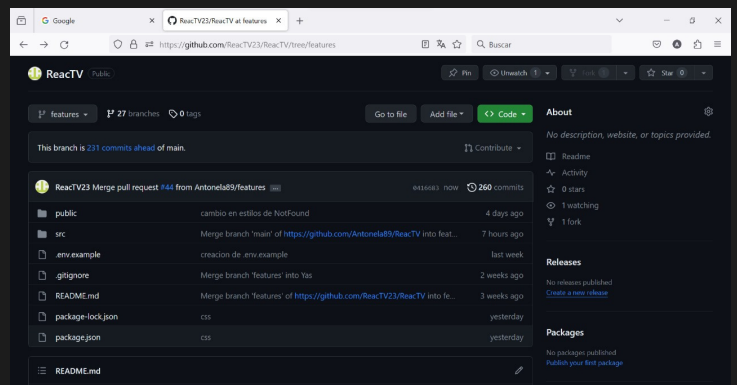
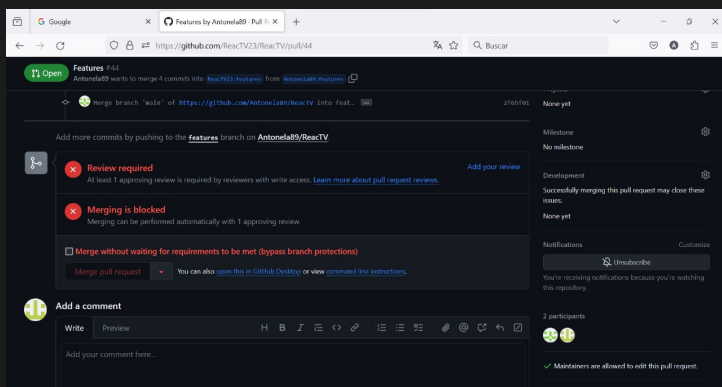
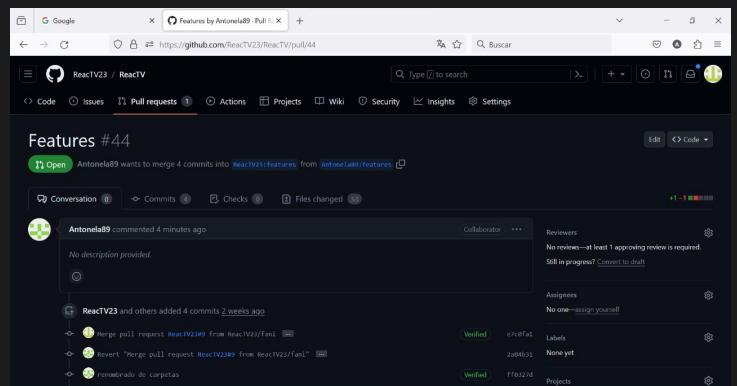
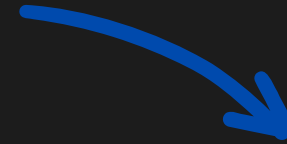
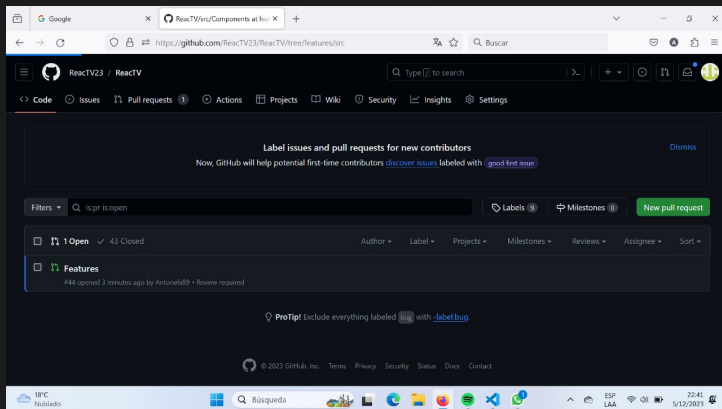
SOLUCIÓN:

Accedimos a la interfaz visual de GitHub online para abordar las discrepancias en las carpetas. Un miembro identificó inconsistencias en las carpetas de la rama "features". Se realizó un fork del proyecto en el repositorio personal de un colaborador en GitHub. Se efectuaron las correcciones necesarias en las carpetas dentro del fork del colaborador. Para verificar la solución, se implementó un cambio en la rama "features" y se transitó a la rama "main" para identificar posibles errores. Se identificó un problema vinculado con un merge anterior en la rama "main" (merge número 9). Se resolvieron cuidadosamente los conflictos. Se ajustaron las carpetas correctamente y se solicitó un pull request desde el repositorio del colaborador hacia el repositorio principal (ReacTV). Desde ReacTV se revisó el pull request para asegurar que este correcta la nueva estructura de carpetas.

carpetas en GitHub



vistas desde el GitHub de ReactV



pedido que se hace a ReactV para un pull request desde Antonela89, donde se hicieron las correcciones.

LINKS DEL PROYECTO

Repositorio: <https://github.com/ReactV23/ReactV>

Figma-Proyecto:

<https://www.figma.com/file/rq4z9JNH8yNPbiDvtk3Z8Z/ReactV2?type=design&node-id=0-1&mode=design&t=USTibKKsWV5pQ7Np-0>

Figma-Diagrama:

<https://www.figma.com/file/xbEkhhwliZvD1LMtnUjJDY/Diagrama-de-Flujo?type=whiteboard&t=USTibKKsWV5pQ7Np-0>

INTEGRANTES GRUPO 1

Nombre	Apellido
Monica Irene	Crichton
Fani Mariel	Casco
Antonela	Borgogno
Horacio	Porto
Hugo Antonio	Segura
Maria Antonieta	Fasce
Yasmin Alexandra	Pyrih
Agustin Gerardo	Haag
Sofía	Gandulfo
Alejo	Piñeiro