



# BrokenHearts

傷ついた心

**Autor:** Eudys Rosario Villegas

**Ciclo Formativo:** DAW (Desarrollo de Aplicaciones Web)

**Curso:** 21/2022

**Profesor/Tutor:** Paco Ávila

## INTRODUCCIÓN

Broken Hearts es una Web la cual su principal funcionalidad es una tienda online y aparte de eso mostrar el trabajo de un fotógrafo, es decir, un portfolio. La idea de hacer este proyecto surgió por un amigo que tiene su propia “marca” con la suele hacer merchandising que después vende, además le gusta hacer fotos y publicarlas en redes sociales así que decidí simplificar esas dos cosas en un mismo proyecto.

En cuanto a la metodología usada me he basado un poco en Scrum, principalmente en el apartado de tener algo funcional que entregar cada cierto tiempo, siempre anotaba objetivos para hacer y terminar de la web en un tiempo aproximado.

## PLANTEAMIENTO DEL PROBLEMA:

El objetivo es hacer un e-commerce usando tecnologías que considero que son relativamente “nuevas” como React, NextJS y Firebase. El uso de estas tecnologías hace que la elaboración de este proyecto sea más fácil en ciertos puntos como por ejemplo el uso de Firebase para el Back-end, pero, la parte negativa de esto es que a veces no hay suficiente documentación oficial o en internet de otros usuarios para lo que uno quiere realizar.

Por ejemplo, en el caso de NextJS que es un framework de React que destaca por que es SSR (Service Side Rendering) así que hasta cierto punto no es necesario usar una tecnología adicional para crear un servidor porque Next lo “implementa”, la parte negativa es que muchas veces no puedes hacer lo que tienes en mente porque NextJS mezcla el Front-end con el lado de servidor lo que a veces genera problemas porque la información que se renderiza en el Front-end debe ser igual a la del servidor.

Los problemas o inconvenientes más llamativos encontrados durante el desarrollo del proyecto han sido:

- El uso de Firebase, una tecnología que empecé a utilizar porque se usó por un periodo muy corto de tiempo en la practicas y terminé usándolo y aprendiéndolo por mí mismo porque me parecía interesante. Su uso facilita mucho el usar

y crear un Back-end pero al igual que la mayoría de las tecnologías que he usado consume bastante tiempo aprenderlas.

- Como no he usado una tecnología específica para servidor como podría ser Express cada vez que necesitaba hacer alguna petición mediante el servidor tenía que usar la API que trae NextJS.
- NextJS y el uso de LocalStorage han sido un problema porque localstorage se guarda en el cliente y en este caso he usado en el lado de cliente Next que es cliente y servidor por lo que localstorage, a veces, da muchos problemas y errores.
- Lógica para añadir productos al carrito de la compra se complicó porque la información que llega al carrito de la compra la guardo en el localstorage que como ya he dicho antes da problemas y además guardar productos variables, es decir, con diferentes tallas y colores.
- Los tipados de TypeScript han generado mucha pérdida de tiempo, algunas veces lo que quería hacer no funcionaba porque no estaba usando el tipado correcto o porque no lo había descargado.
- El uso de Stripe como pasarela de pago, una plataforma completamente nueva para mí y difícil si quieres usarla únicamente para hacer el Checkout.

## **SOLUCIÓN DEL PROBLEMA:**

La gran mayoría de estos problemas se solucionaron con el tiempo porque lo que hacía falta eran horas de búsqueda y aprender a usar dicha tecnología fue el caso de Firebase, NextJS como framework, el uso de Tailwind en vez de usar CSS puro y TypeScript en cuanto a los otros problemas como el uso de Localstorage los errores que generaba se solucionaron diferenciando cuando estoy en el lado del cliente y cuando el de

servidor y también haciendo que ciertos componentes únicamente se rendericen en el lado del cliente para que el servidor y el cliente tengan la misma información.

Como base de datos he usado Firebase que funciona por llamadas a la base de datos parecido a las funciones Landa o a la API de NextJS y te cobran en función de dichas llamadas por lo que para un uso moderado es gratis.

La otra parte del Back-end ha sido desarrollada con NodeJS y la parte de SSR de NextJS.

En el Front-end Nextjs que es un Framework de React que lo he usado con TypeScript, Tailwind (Framework de CSS) lo que hace es quitar los estilos de los elementos para que uno lo cree como desee usando las clases de Tailwind que es CSS normal pero “comprimidas o acortadas”.

Como pasarela de pago he usado Stripe que te permite crear productos, formas de pago, hacer el checkout... en mi caso la he usado únicamente para hacer el checkout de los productos.

Algunas librerías usadas:

- Headless UI, para el uso de los Modales.
- React-responsive-carousel, librería de React para implementar un simple slider
- MobileNavbar, para el menú de hamburguesa en el modo móvil
- Nodemailer, servicio para enviar emails totalmente gratis hasta cierto punto.
- Formik, librería de React para la creación de formularios.
- Yup, librería para la validación de formularios.
- React-icons, una librería de iconos completamente gratis y muy fácil de usar para React.

En cuanto al editor de código he usado Visual Studio Code.

## **ANÁLISIS DE LA APLICACIÓN:**

El objetivo de la Web era poder mostrar el trabajo de fotógrafo hecho por una persona y a su vez poder vender merchandising y

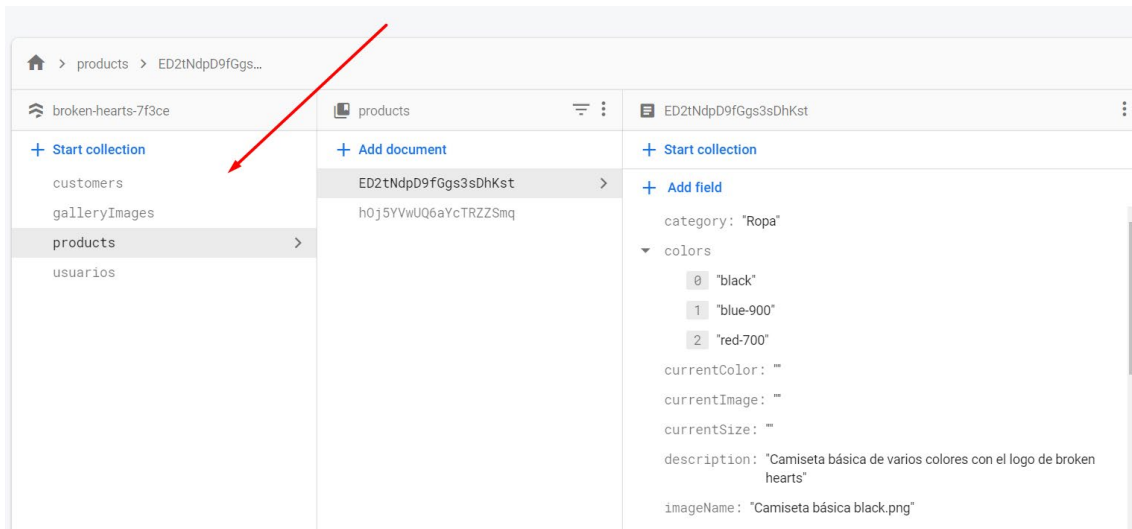
todo esto relacionado con la marca de dicha persona. Finalmente la estructura de las páginas de la web queda de la siguiente manera:

Página de inicio: contiene un header amplio con imágenes destacadas, una pequeña galería sin ningún tipo de interacción mas que un hover y un formulario de contacto principalmente para dudas.

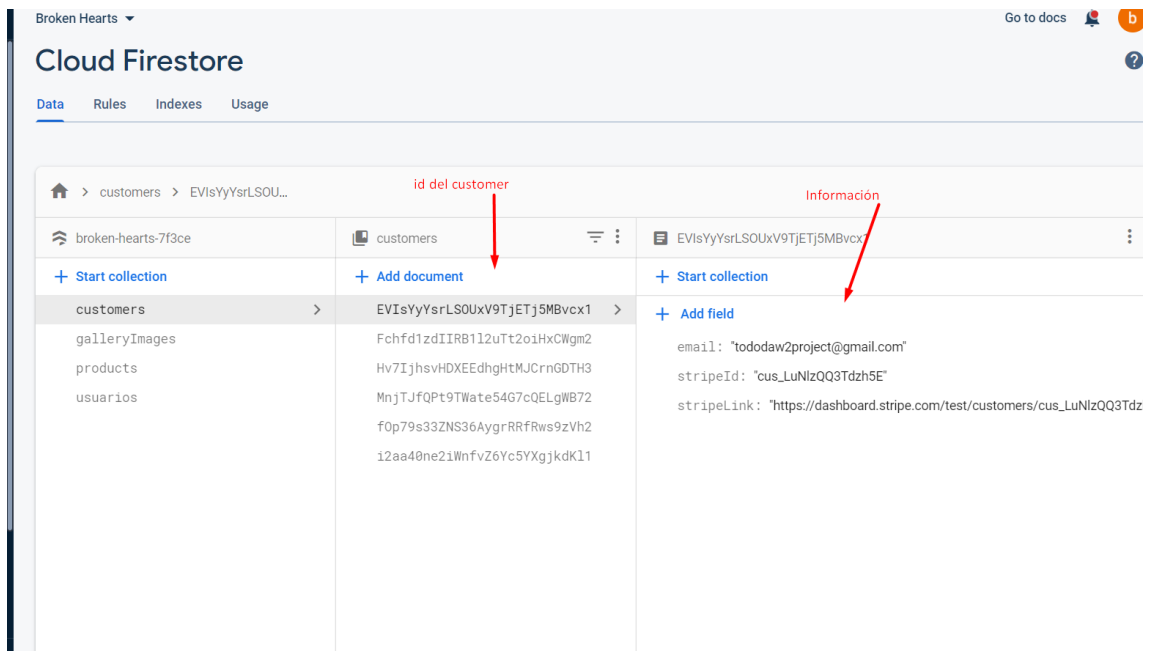
- **Página de fotografía:** es como un portfolio donde inicialmente hay un slider y mas a bajo una galería de imágenes que al hacer click se abre un modal para ver la imagen un poco mas grande, si eres admin puedes añadir o eliminar imágenes de la galería.
- **Tienda:** Una tienda básica con varios productos que puedes añadir al carrito, cuando haces click sobre un producto se abre un modal para ver una descripción y en caso de que exista elegir tallas y colores, si eres admin puedes añadir y eliminar productos.
- **Página de Cuenta:** te puedes registrar/iniciar sesión por Google o mediante email y contraseña, si eres un cliente puedes cambiar tu foto de perfil y ver los datos de tu cuenta, únicamente nombre, apellido, email, y eliminar tu cuenta. Si eres admin tienes ver los datos de todos los usuarios y eliminar sus cuentas.
- **Carrito:** aquí se ven los distintos productos que has añadido, el total de la compra y los gastos de envío.
- **Páginas legales:** las de políticas de privacidad y la de devoluciones.

## **DISEÑO DE LA APLICACIÓN:**

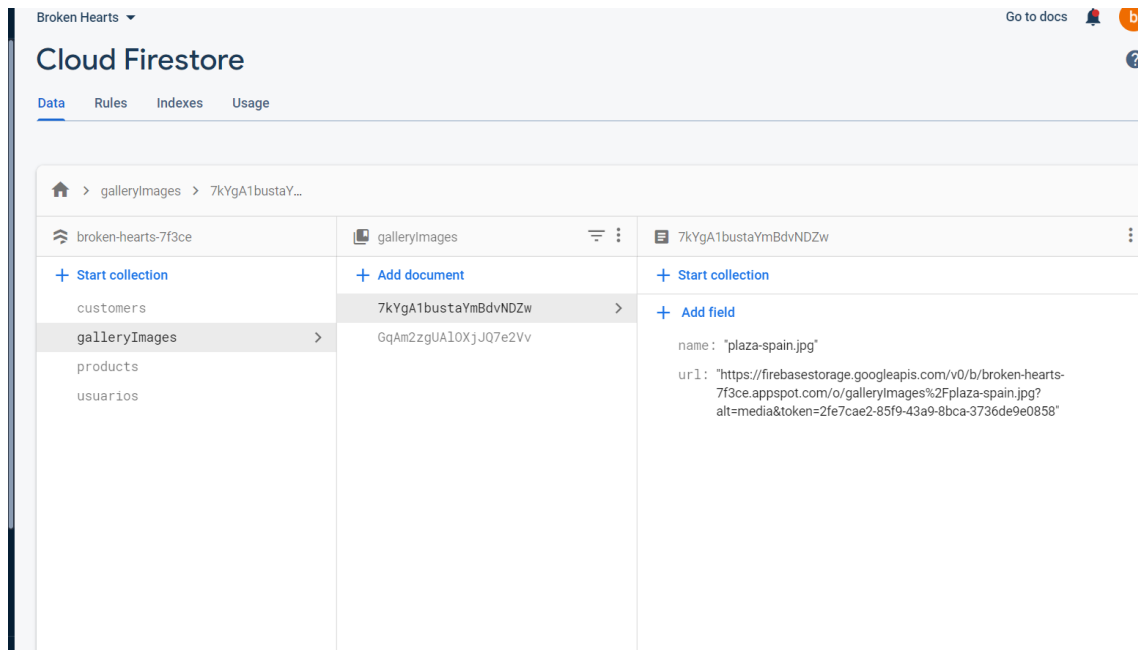
Las colecciones existentes son:



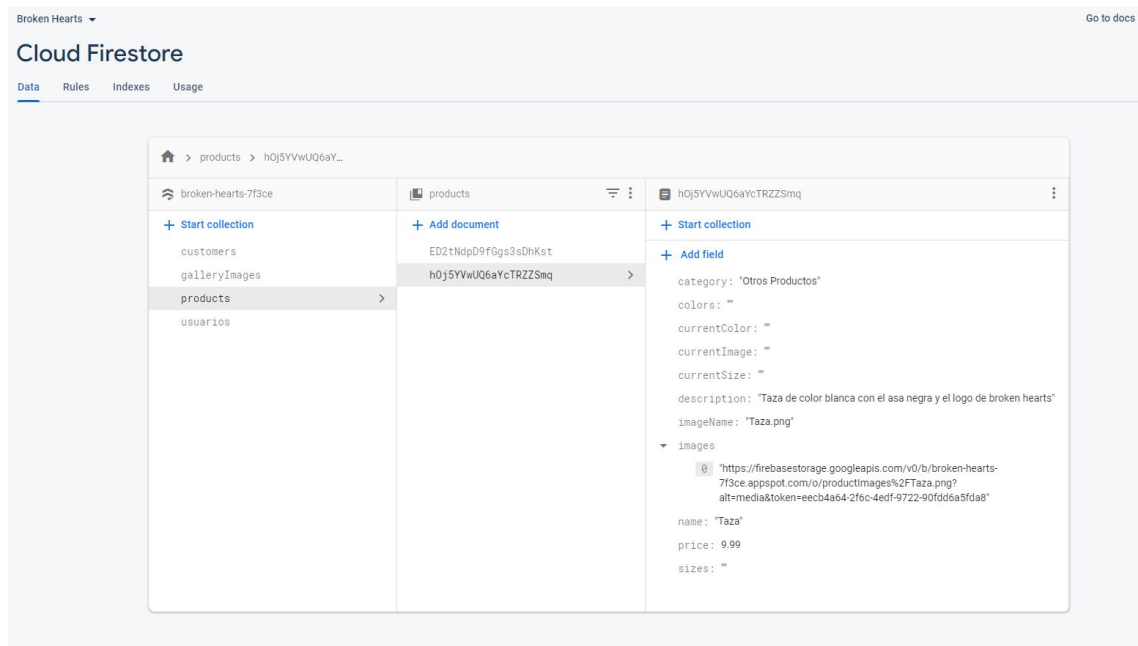
Customers, es una colección creada por Stripe cada vez que un usuario se registra se añade un nuevo documento a la colección.



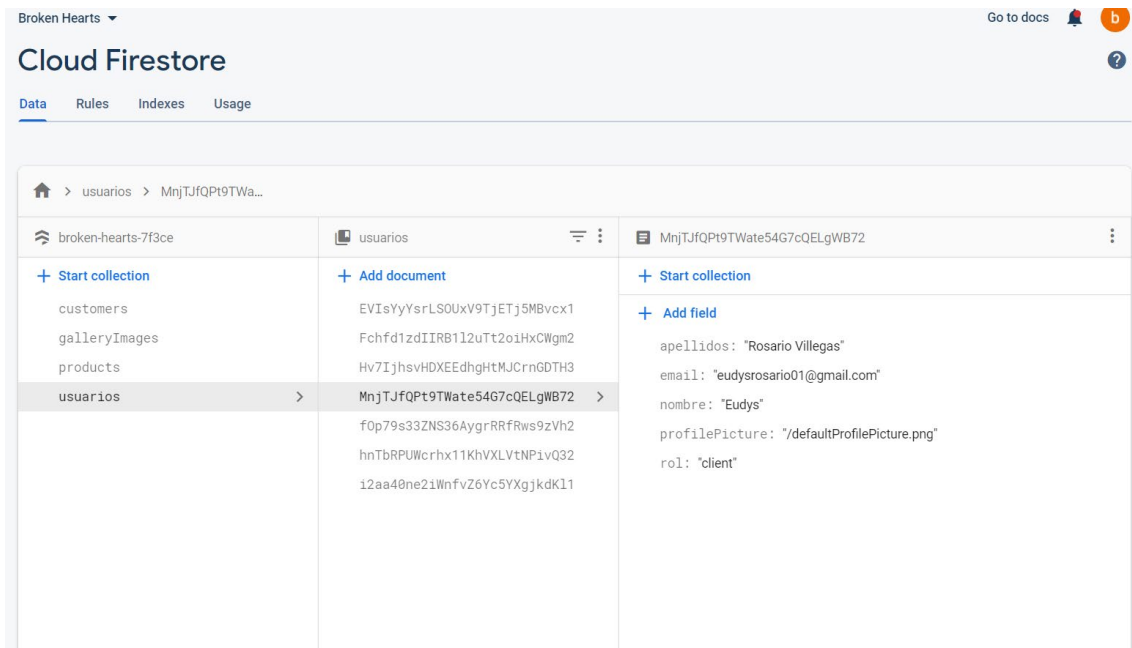
galleryImages, para guardar el nombre y url de las imágenes de la galería.



Products, donde se almacena toda la información de los productos de la tienda.



Usuarios, información básica de los usuarios registrados.



## MANUALES:

No hace falta nada especial para poder usar la web, el dominio principal es: <https://brokenhearts.es/> pero también se sube a otros dominios porque esta desplegada en vercel, dominios de vercel:

<https://broken-hearts-react-web.vercel.app/>

<https://broken-hearts-react-web-eudys1.vercel.app/>

<https://broken-hearts-react-web-git-main-eudys1.vercel.app/>

y en caso de que se quiera ver el código está disponible en github:

<https://github.com/eudys1/brokenHeartsReactWeb> instrucciones para usarlo en el readme.md del repositorio

La web únicamente tiene una cuenta con el rol de admin:

Email: [admin@gmail.com](mailto:admin@gmail.com)

Contraseña: 123456

Email para acceder a Firebase:

[brokenheartsjerezbrand@gmail.com](mailto:brokenheartsjerezbrand@gmail.com)

Contraseña: BrokenHearts123456



## GESTIÓN DEL PROYECTO:

El proyecto se empezó primero por el Front-end hasta cierto punto después, el Front-end y el Back-end se han desarrollado prácticamente al mismo tiempo en función de la necesidad al igual que los errores que iban surgiendo se solucionaban y después se continuaba excepto cuando tardaba mucho en solucionarlo. En cuanto a la documentación es lo último que he realizado.

## CONCLUSIONES:

Decidí hacer este proyecto porque me pareció buena idea y a demás porque supuse que habría mucha documentación sobre como hacer una tienda y una galería de fotos pero por el tipo de tecnologías que he usado en ciertas ocasiones me ha resultado un poco difícil encontrar información pero, finalmente esta terminado justo en el tiempo de entrega y el resultado final no está nada mal.

Como futuras mejoras trabajaría en un Dashboard más útil, editar usuarios, ver recibos, mejor optimización de la web y las imágenes, solucionar bugs, un sistema de reservas para sesiones de fotos...

## BIBLIOGRAFÍA:

Toda la información que he obtenido y usado por lo general la he encontrado en estas webs:

- <https://es.stackoverflow.com/>
- <https://www.youtube.com/> (muchos videos de diferentes canales)
- <https://nextjs.org/docs/getting-started>
- <https://stripe.com/docs>
- <https://firebase.google.com/docs/web/setup>
- <https://tailwindcss.com/docs/installation>

