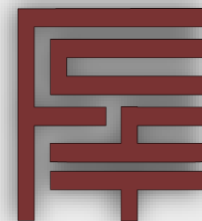


UNIVERSIDAD MAYOR DE SAN SIMÓN
FACULTAD DE CIENCIAS Y TECNOLOGÍA
CARRERA DE INGENIERÍA INFORMÁTICA



PROYECTO

PROGRAMACION Web Tiendita

Semestre 1-2020

Estudiante: Leonardo Dennis Roldán Fernández - Christian Pañuni Churata

Materia: Programación WEB

Docente: COSTAS JAUREGUI VLADIMIR ABEL

INTRODUCCIÓN

Para elaborar el proyecto aplicamos las siguientes herramientas Angular, NodeJS, JQuery, Bootstrap y Firebase. Basándonos en los requerimientos que debe cumplir el proyecto en nuestro caso escogimos el 3 Llamado llantucha, que consiste en crear una página web de ventas. Llegamos a hacer los siguientes avances.

Avance de Proyecto

Para empezar, Creamos un proyecto en angular para su correcto funcionamiento instalamos TypeScript, NodeJs, Luego instalamos Angular en su versión 7.

ANGULAR: es un *framework opensource* desarrollado por Google para facilitar la creación y programación de aplicaciones web de una sola página, las webs SPA (*Single Page Application*).

Angular separa completamente el frontend y el backend en la aplicación, evita escribir código repetitivo y mantiene todo más ordenado gracias a su patrón MVC (Modelo-Vista-Controlador) asegurando los desarrollos con rapidez, a la vez que posibilita modificaciones y actualizaciones.



TYPESCRIPT

Typescript es la solución a muchos de los problemas de JavaScript, está pensado para el desarrollo de aplicaciones robustas, implementando características en el lenguaje que nos permitan desarrollar herramientas más avanzadas para el desarrollo de aplicaciones.



Node.js: utiliza un modelo de entrada y salida sin bloqueo controlado por eventos que lo hace ligero y eficiente (con entrada nos referimos a solicitudes y con salida a respuestas). Puede referirse a cualquier operación, desde leer o escribir archivos de cualquier tipo hasta hacer una solicitud HTTP.

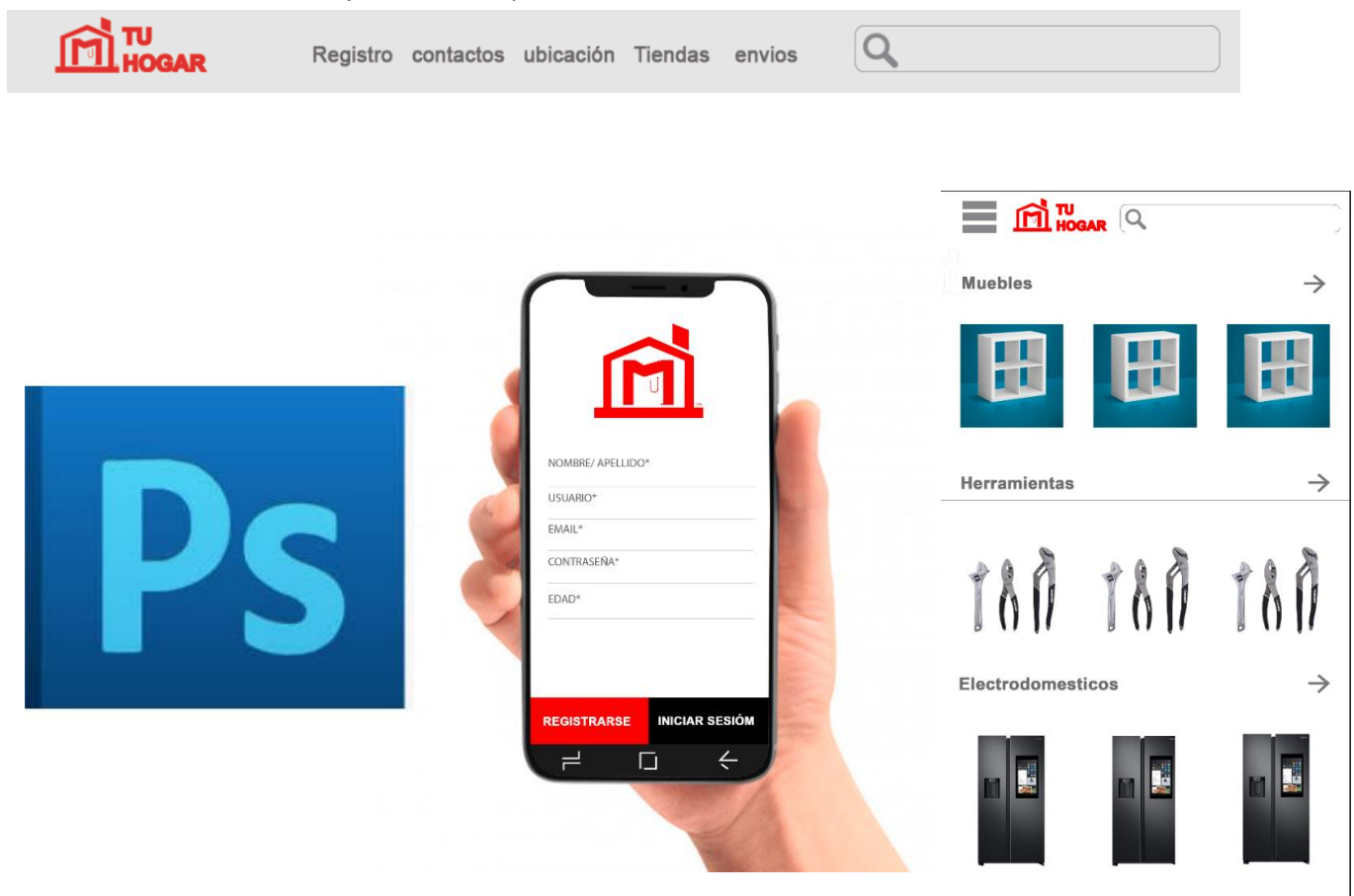


AVANCE 1

Después de crear el proyecto Estudiamos la manera de aprovechar la estructura basada en componentes de Angular y para ello nos tocó hacer varias pruebas ya que es la primera vez que usamos este entorno de programación y lenguajes. Y Para ello Estudiamos Todo el concepto de HTML, CSS y JavaScript. Apoyándonos muy bien de lo avanzado en clases.

Siguiendo con la descripción de avance, a medida que estudiamos lo anteriormente dicho, nos creamos una interface de usuario en Photoshop CS6 para empezar a programar la página. Teníamos como modelo las siguientes imágenes que poco a poco iba mejorando.

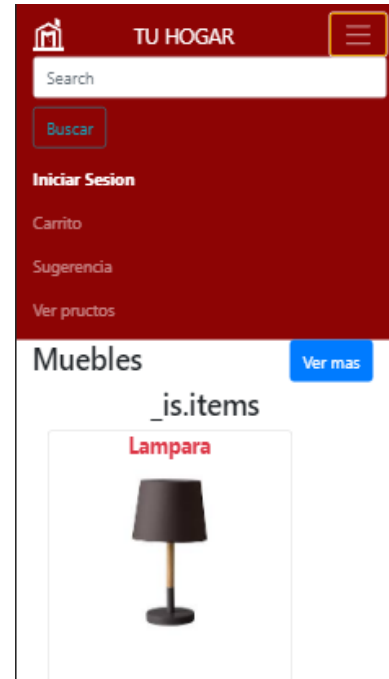
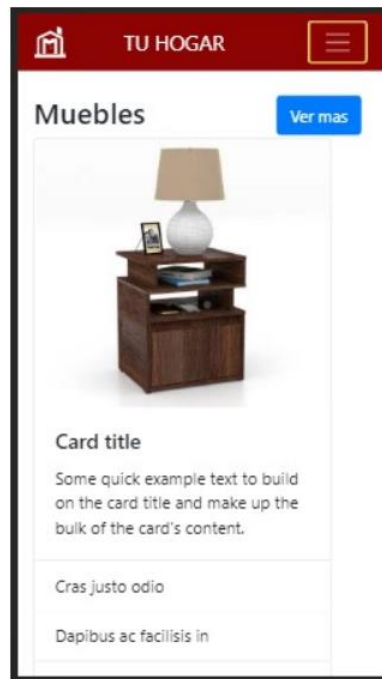
Luego se nos dio la instrucción que nos enfoquemos en un entorno Móvil. Y para ello creamos nuevos diseños. Intentamos asemejarnos lo más posible a ellos.



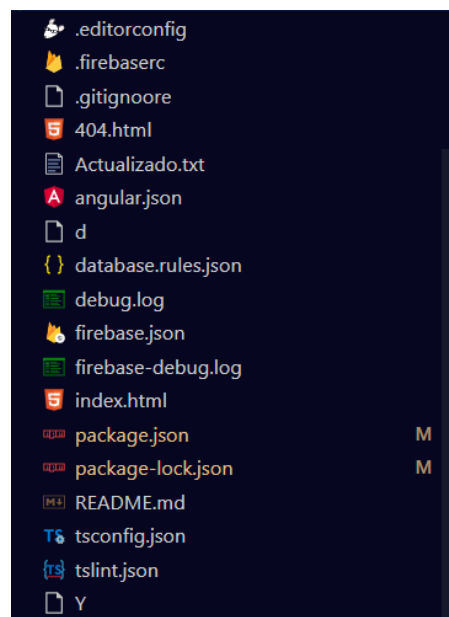
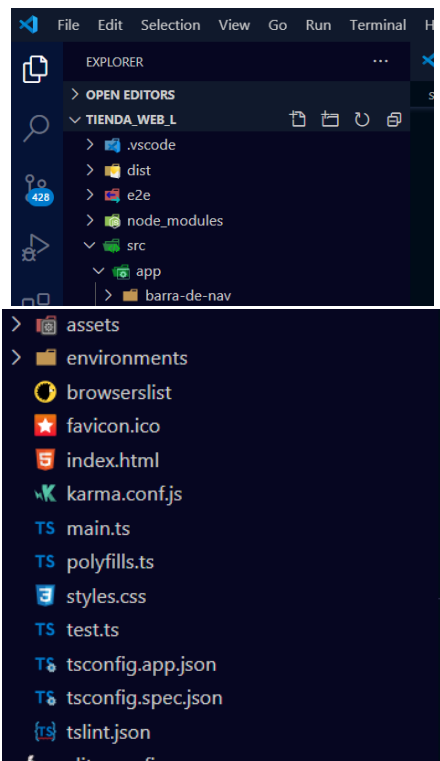
AVANCE 2

Y empezamos a aplicarlo por lo que teníamos los siguientes avances, ya en código implementado.

Integramos Bootstrap para que el diseño de nuestra pagina web sea mucho mas amigable.

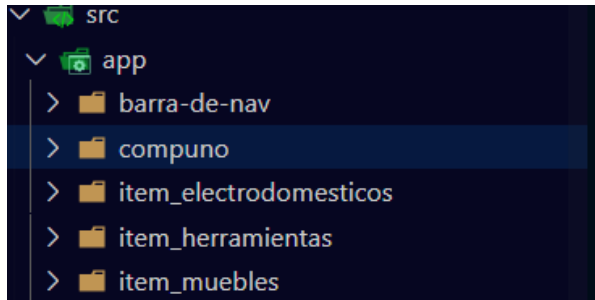


Aquí tenemos algunas capturas de la estructura de los archivos creados para que funciones correctamente.



AVANCE 3

Y poco a poco agregamos mas funcionalidad a medida que se nos daba la instrucción. Y mejorábamos cada vez un poco mas la inteface de usuario. En el siguiente avance agregamos nuevos componentes al proyecto, esto para tener categorías en nuestra aplicación. Tendremos la opción de Electrodomesticos, Herraminetas y Muebles.



Y lo añadimos para que se pueda ver en la pagina principal en la interface.

Los elementos solo estaban agregados por html, es por eso que la lampara se muestra en todas las categorías solo para hacer las pruebas.

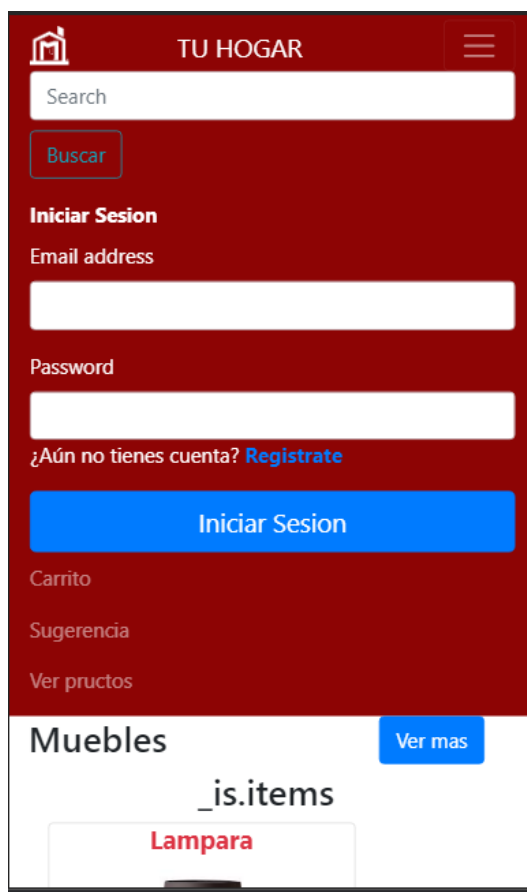
Y pensábamos agregar luego el componente para recolectar los datos de los elementos de una base de datos.

Al aplicar Bootstrap y seguir enfocándonos en un entorno móvil. Este ya implementaba clases para que la página sea responsiva, por lo que este ya tenia un buen diseño al expandir la pagina en un entorno de computadora de escritorio o Laptop.

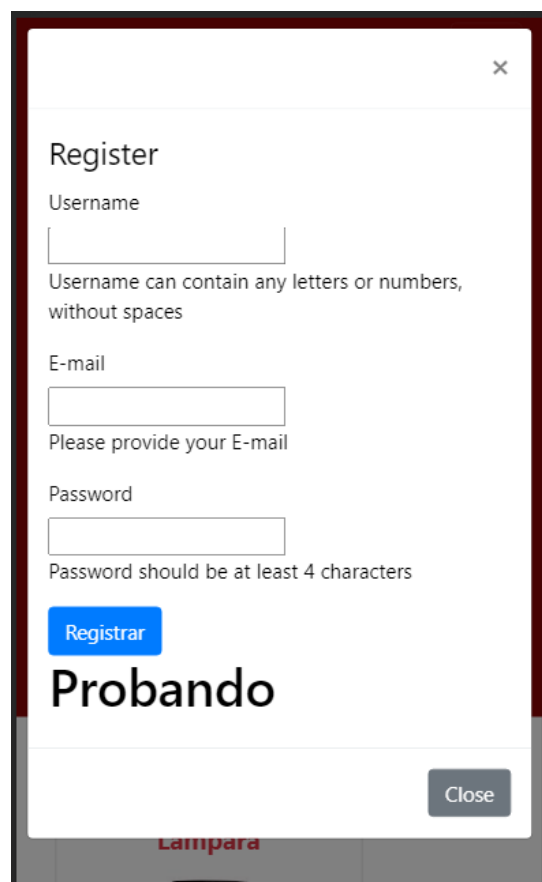


AVANCE 4

Trabajamos lo que era un poco más el diseño de nuestro Inicio de Sesión y El Registro. y lo hicimos de la siguiente manera, Como una pestaña deslizable que te mostraba los cuadros para introducir tus datos con su botón respectivo. Y al presionar el botón Registrarte saldra el siguiente formulario Register.



The home screen features a dark red header with a home icon, the text "TU HOGAR", and a hamburger menu icon. Below the header is a white search bar with a "Buscar" button. The main content area is white and contains a login section with the heading "Iniciar Sesión", an "Email address" label, an email input field, a "Password" label, a password input field, and a "¿Aún no tienes cuenta? [Registrate](#)" link. A large blue "Iniciar Sesión" button is positioned below the login fields. At the bottom of the main content area are links for "Carrito", "Sugerencia", and "Ver pructos". The footer is dark red and displays "Muebles" and "is.items" with a "Ver mas" button. A "Lampara" item is partially visible at the very bottom.



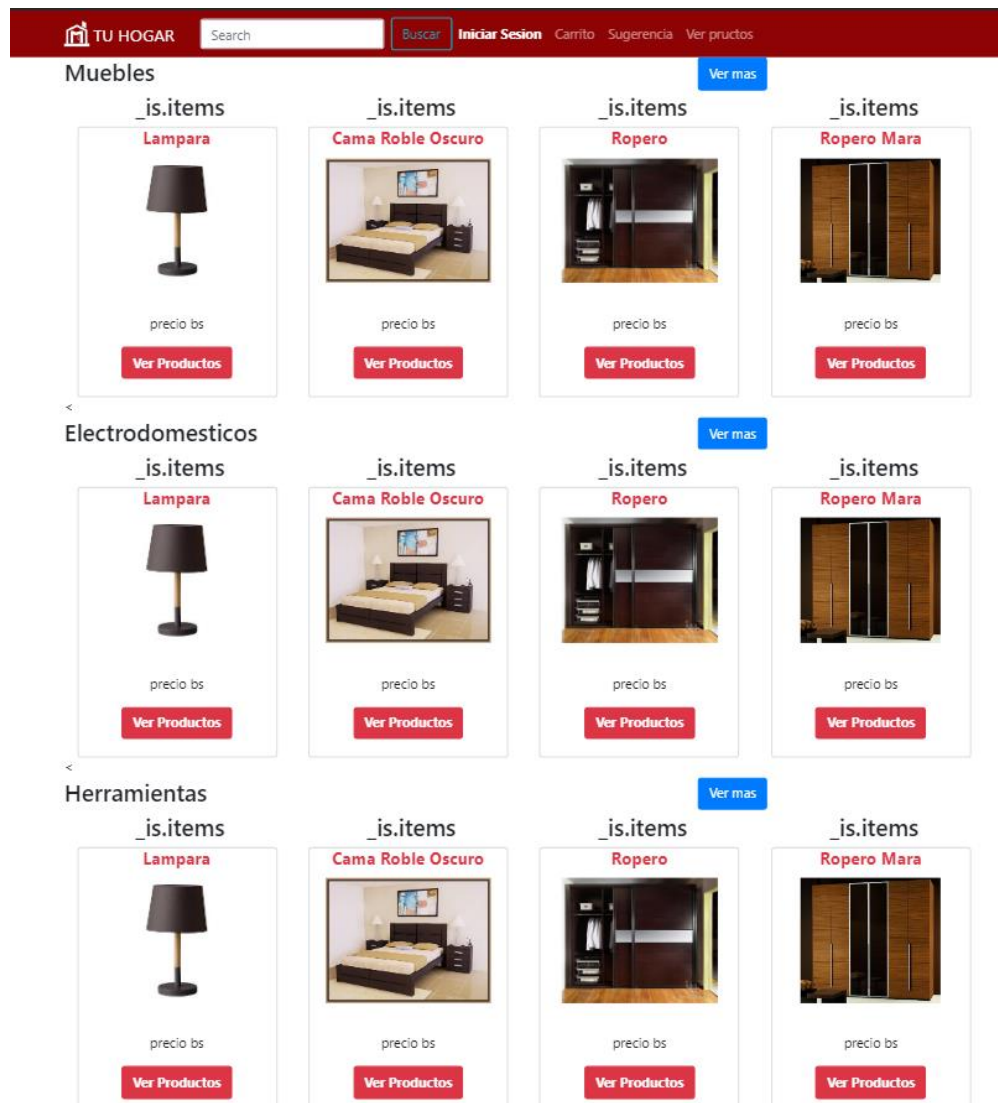
The register modal has a white background and a dark red border. It includes a close button (X) in the top right corner. The title "Register" is at the top. The form contains fields for "Username" (with a note: "Username can contain any letters or numbers, without spaces"), "E-mail" (with a note: "Please provide your E-mail"), and "Password" (with a note: "Password should be at least 4 characters"). A blue "Registrar" button is located below the password field. The word "Probando" is displayed in large black text at the bottom of the modal. A "Close" button is in the bottom right corner. The "Lampara" item from the home screen is visible behind the modal.

AVANCE 5

Añadimos una base de datos que elegimos que sería Firebase, para usar algo nuevo que implementar ya que este usaba una base de datos directamente en la web y podemos trabajar directamente ambos integrantes del Grupo con la misma base de datos.

Nos costó un poco que esto funcionara y algo de tiempo. Pero logramos conectarlos y recolectar información de la base de datos para mostrarlo en la interface de nuestra página principal.

Podemos ver que los elementos se repiten en las categorías esto es porque sigue usando la misma tabla esto porque seguíamos haciendo pruebas, pensábamos independizarlo luego.



AVANCE 6

Se nos pidió trabajar con Eventos Clic de la página, captura de datos en el login, añadir contadores para saber cuántas veces se hace clic en cada botón y terminar mostrarlo por consola.

Register

Username

Username can contain any letters or numbers, without spaces

E-mail

Please provide your E-mail

Password

Password should be at least 4 characters

[Registrar](#)

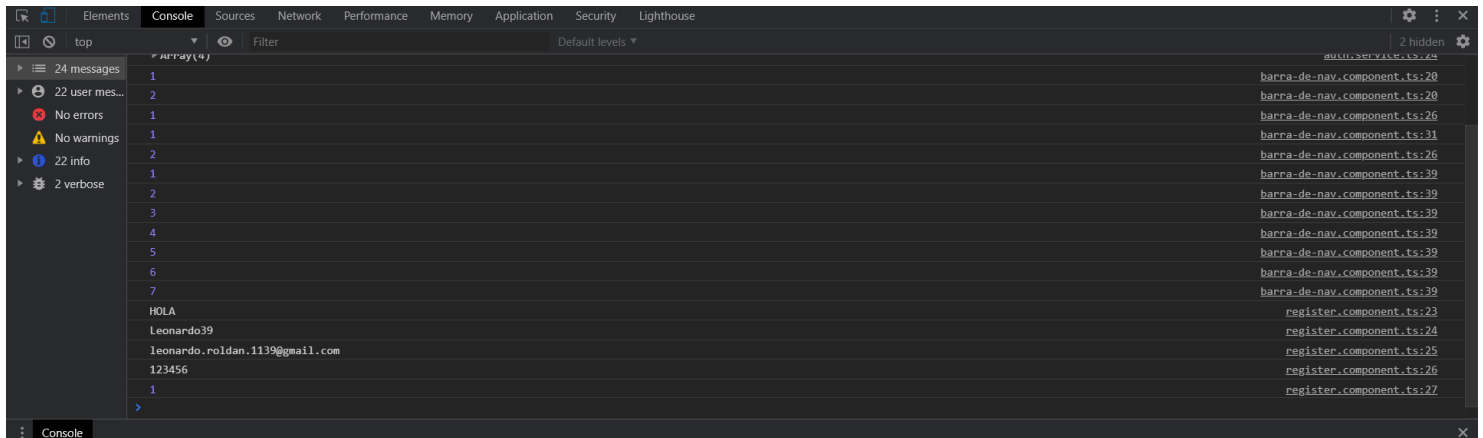
Probando

Leonardo39

leonardo.rolan.1139@gmail.com

123456

[Close](#)



Los números son los contadores de los clics que se hacen en diferentes botones en barra de navegación y los datos después de los números son los datos que se capturan en el componente registro que es nuestro formulario de registro.

AVANCE 7

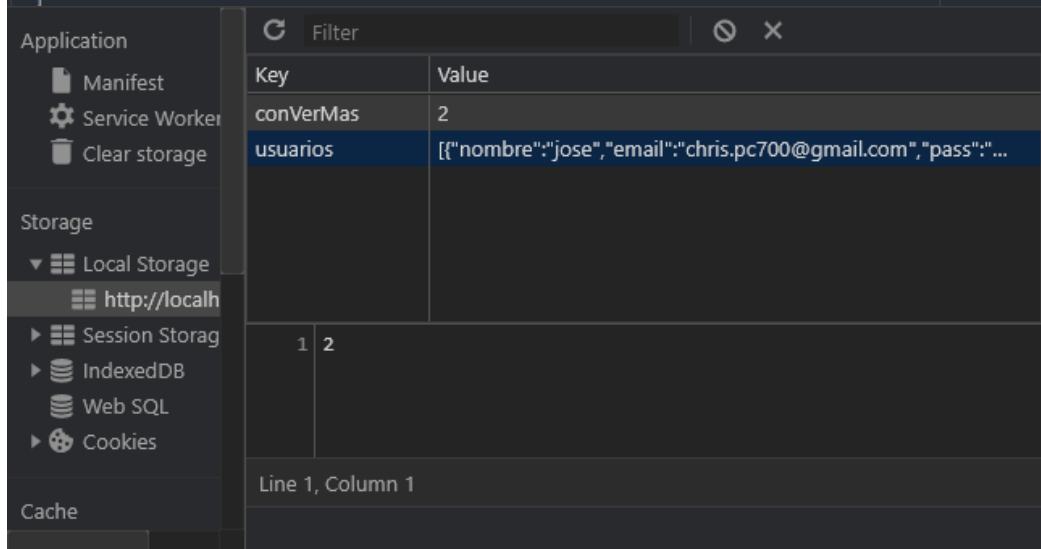
Se nos pidió guardar los datos capturados en una memoria local que almacenaba los datos en un plazo de tiempo mas largo. Usamos Local Storage.

LOCAL STORAGE

Hacemos uso de Local Storage para poder guardar nuestro usuario en este tipo de memoria. Vemos en las capturas como se guardan en la sección Local Storage y nuestro contador de clics para el botón (ver más), podemos hacer uso de esta información para mostrarla o crear nuevas funciones mas adelante.

Al invocar la función **addUsiario** se almacena la información del usuario en el localStorage como cadenas en una lista.

```
export class RegisterComponent implements OnInit {
  contador_de_clicks = 0;
  constructor(public user: UsuarioService) {
  }
  ngOnInit() {
  }
  addUsuario(newName: HTMLInputElement, newEmail: HTMLInputElement, newPass: HTMLInputElement){
    console.log("AGREGANDO USUARIO...");
    this.user.addUsuarios({
      nombre: newName.value,
      email: newEmail.value,
      pass: newPass.value,
      hide: true
    });
    console.log(this.user);
    newName.value="";
    newEmail.value="";
    newPass.value="";
    newName.focus();
  }
}
```



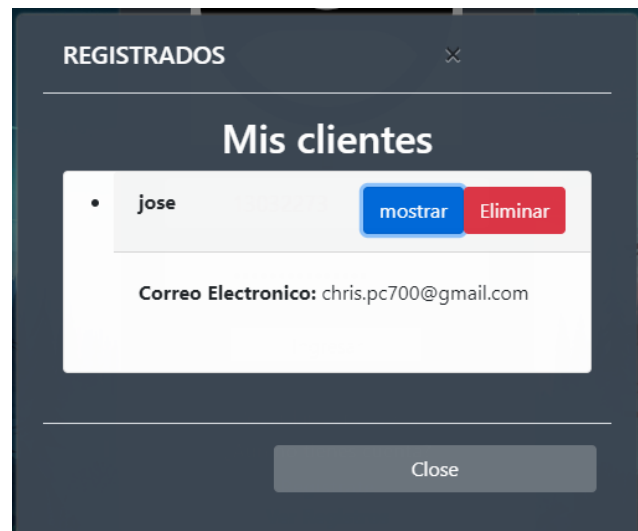
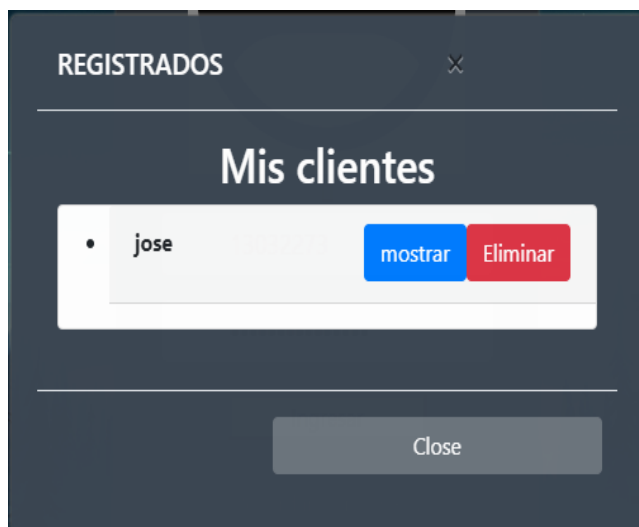
Key	Value
conVerMas	2
usuarios	[{"nombre":"jose","email":"chris.pc700@gmail.com","pass":"..."]

Para tener un poco más claro mostramos en consola los datos de usuario que se registraron en el local Storage.

Los datos son mostrados en un arreglo.

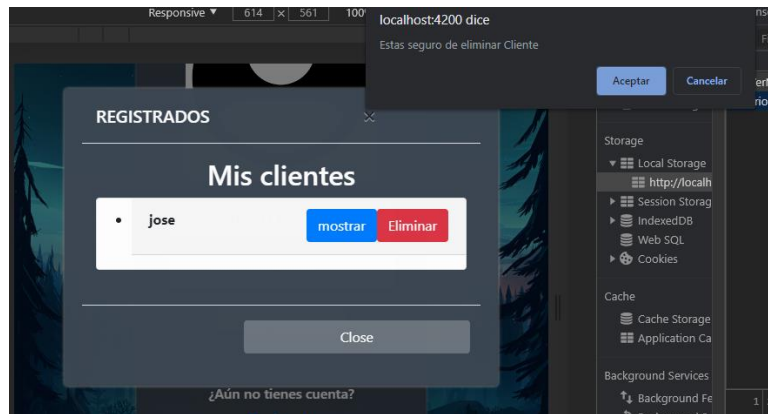
```
AGREGANDO USUARIO....                                register.component.ts:16
                                                         register.component.ts:23
▼ UsuarioService ⓘ
  ▼ usuarios: Array(1)
    ▶ 0: {nombre: "jose", email: "chris.pc700@gmail.com", pass: "1221221", hide...
        length: 1
    ▶ __proto__: Array(0)
    ▶ __proto__: Object
>
```

Mostrando en la interfaz tenemos como cliente Registrado, Y se puede realizar funciones de mostrar información del cliente(usuario) o Eliminar el cliente(usuario).

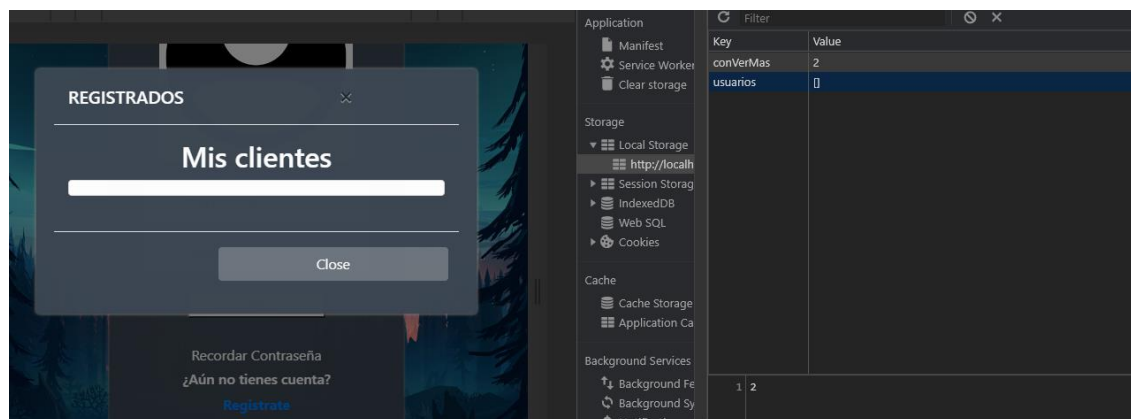


Al hacer una eliminación de Nuestro Cliente(usuario) nos aparece un mensaje de alerta para aceptar o cancelar.

Al hacer esta acción también tenemos que eliminar la información del localStorage, entonces realizamos a eliminar del localStorage y también el arreglo interno de usuarios para visualizar nuestra interfaz



Entonces en la siguiente captura vemos que si se elimina el usuario tanto en el local Storage y en la interface grafica.

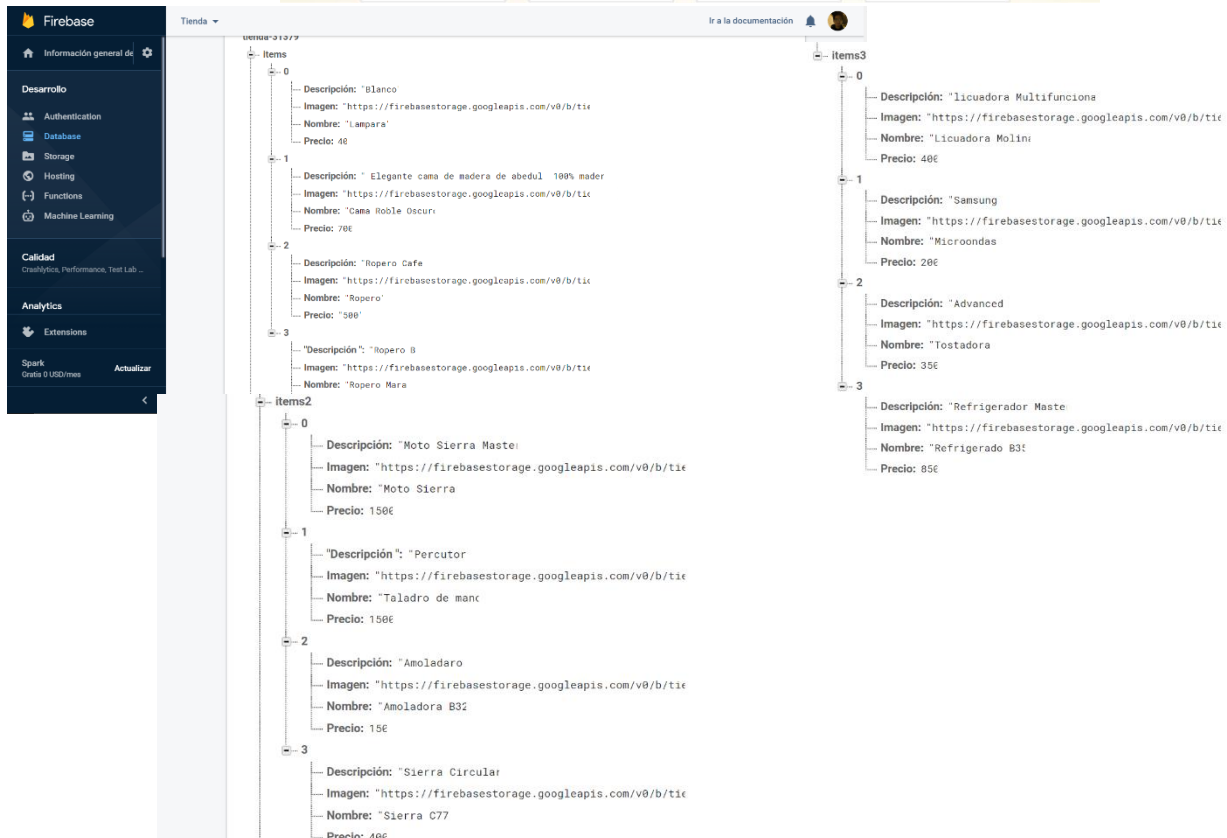
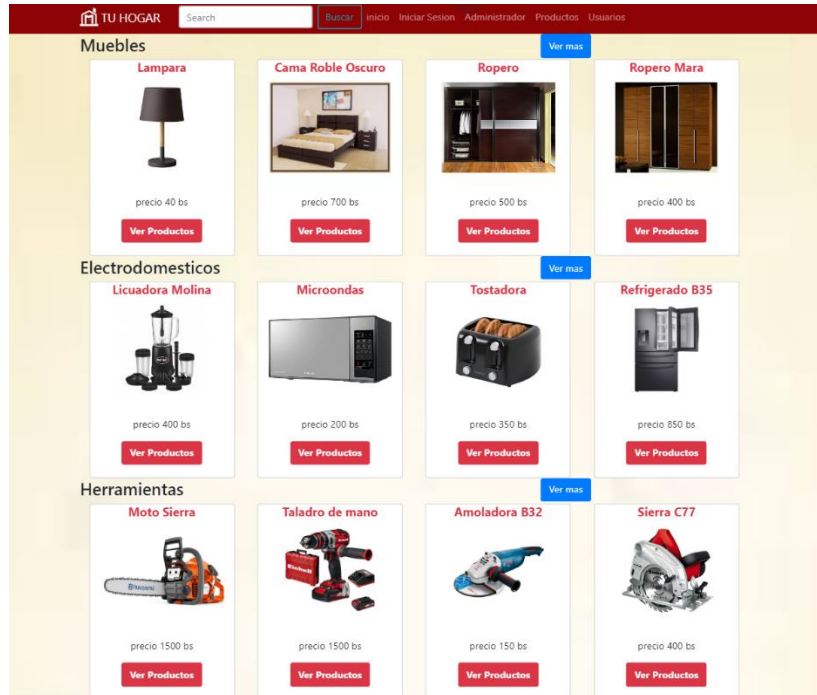


Otros Avances

- Mejora de la interface De inicio de sesión
- Se agrego imágenes de fondo para página principal

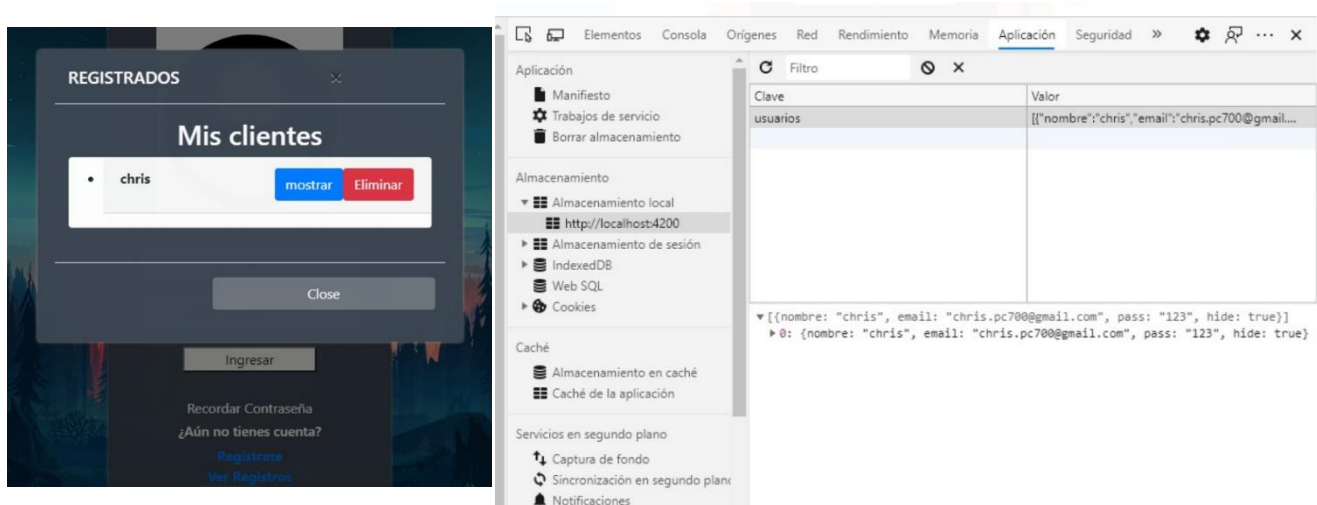
AVANCE 8

En este punto poblamos la base de datos de firebase para mostrar los diferentes ítems en sus categorías respectivas, Muebles, Electrodomésticos y Herramientas.

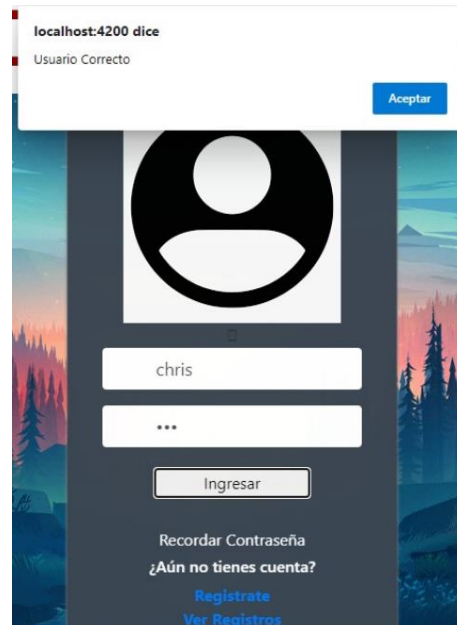
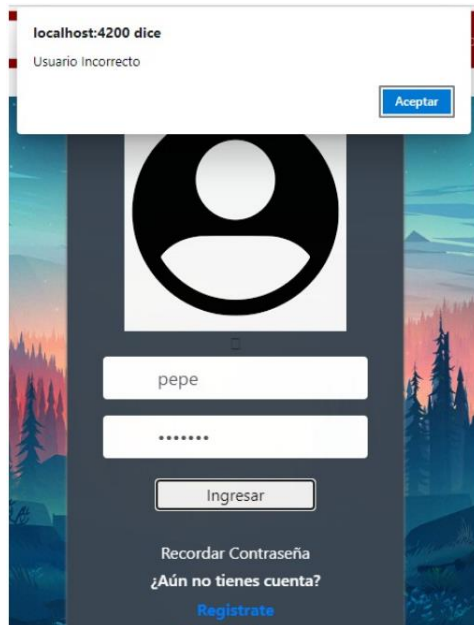


AVANCE 9

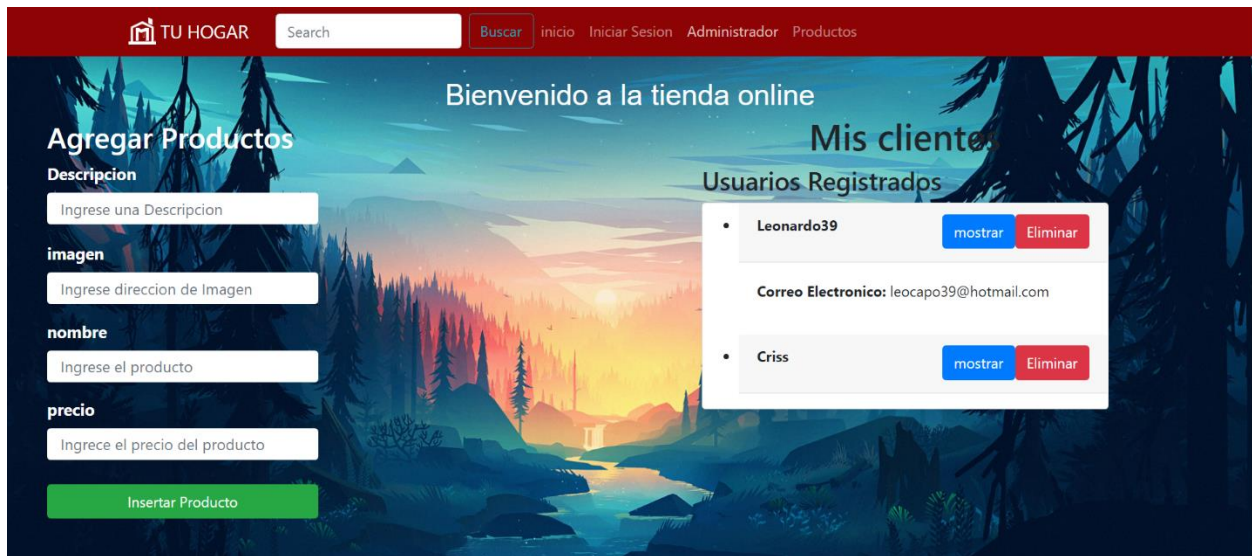
Ahora trabajamos con una autenticación de manera local usando usuarios registrados en el local Storage.



Probamos introduciendo Datos inválidos, luego con un usuario correcto y vemos como hace la validación correctamente.



AVANCE 10



Mejoramos algunos detalles de la interface grafica y tratamos de agregar nuevos Items a la base de datos de Firebase pero no tuvimos éxito, El código esta en el repositorio y comentado las partes que causan conflicto al intentar agregar esta funcionalidad. Nos falta un poco más de investigación.

Creamos un tipo de usuario administrador que también es identificado por la página. Puede probarlo con los siguientes datos. ID: admin PASS: pass.

Fuentes

<https://codigofacilito.com/articulos/typescript>

<https://openwebinars.net/blog/que-es-nodejs/>

<https://firebase.google.com/?authuser=0>

<https://www.youtube.com/watch?v=1B1U6zQXJTM>