**Entrega 2 Semana 5**

Fredy Andres Quevedo

Duvan Mauricio Rodríguez Penagos

Yonathan Alexis Montilla Combita

Politécnico Gran Colombiano

Frontend

Olarte John

Grupo B01

27 de septiembre de 2022

Contenido

**Contenido**

[**Introducción** 3](#_Toc115026617)

[**Registro de usuario** 4](#_Toc115026618)

[**About Us:** 6](#_Toc115026619)

# **Introducción**

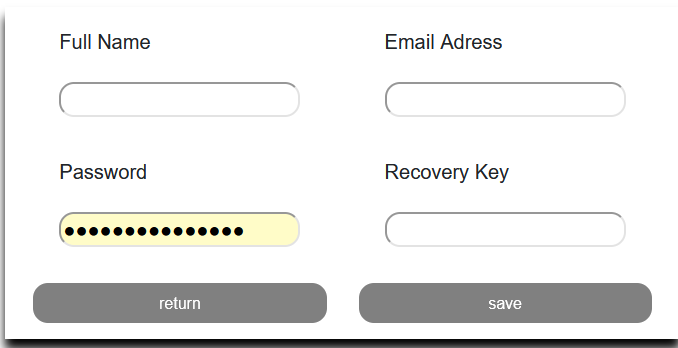
El presente documento tiene como objetivo la presentación del código que mejora visualmente la página de badstore.

# **Registro de usuario**

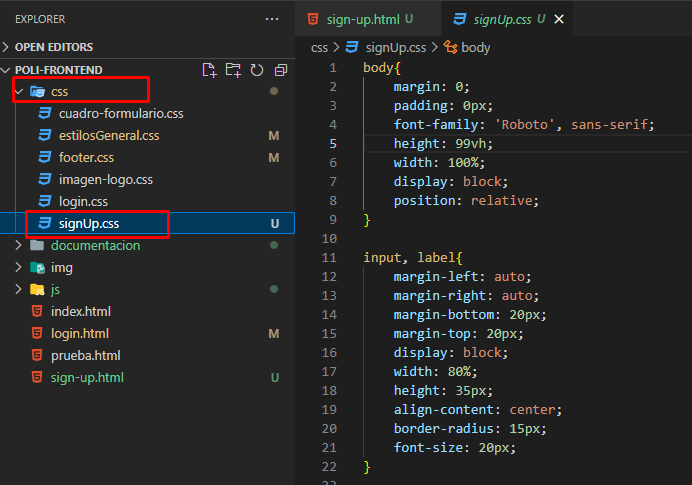
El código del html para esta vista se basa en un conjunto de DIV anidado que permuten acomodar los input en un formato tipo tabla, es decir con filas y columnas.



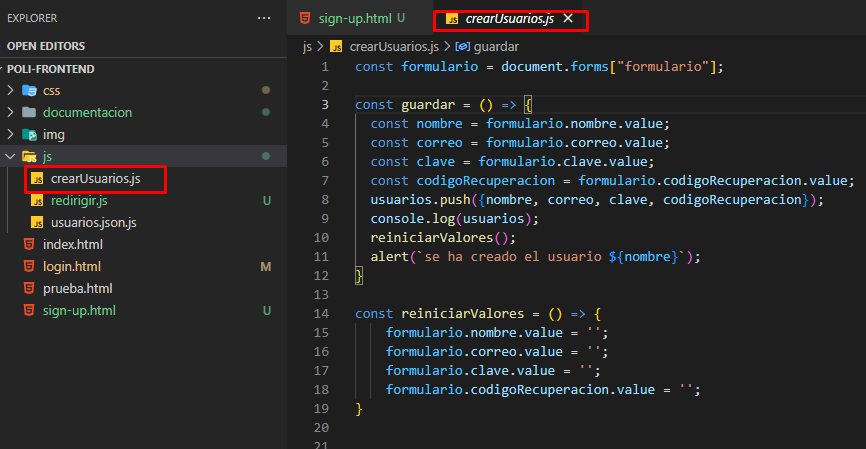
Resultado.



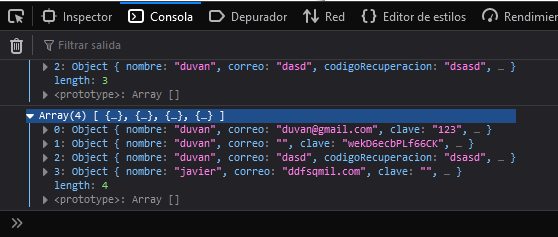
Como se puede observar se generan 3 filas y dos columnas con los datos correspondientes, para el diseño se creó su archivo independiente el cual da los diseños de input con borde ovalado, sombreado a la caja y algunos hover para dar animaciones a los botones.



Para dar un funcionamiento que simule el registro, se uso javascript para capturar información del formulario y rellenar un arreglo de usuarios



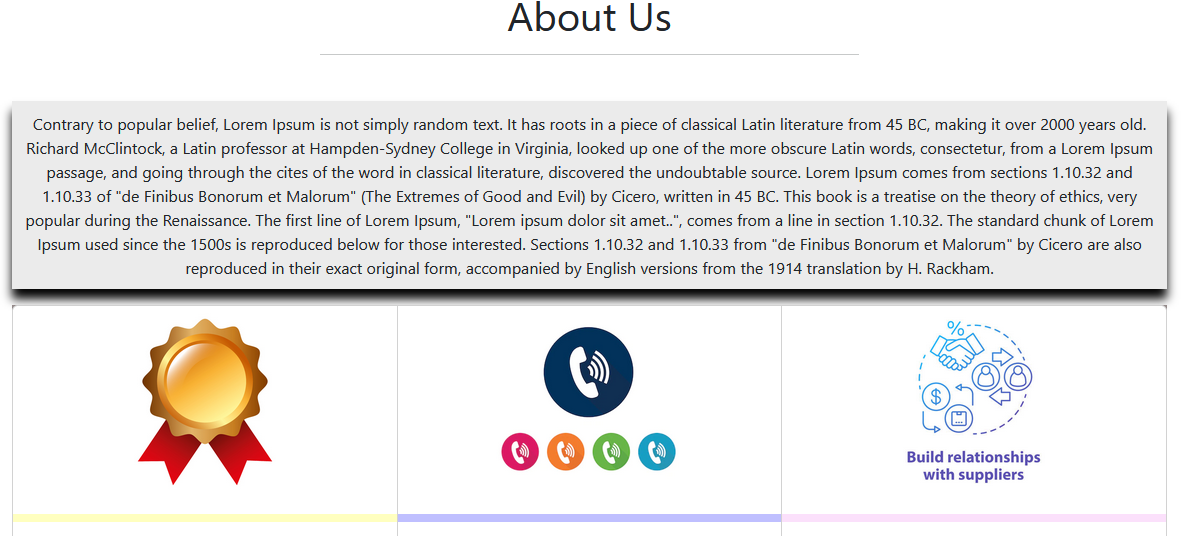
La variable a la que se le hace push “usuarios”, proviene de la clase independiente “usuarios.json.js”, la cual almacena en memoria un listado de usuarios, estos se pueden ver reflejados en el log, por defecto solo hay uno precargado, sin embargo cuando vamos guardando más usuarios el arreglo se va cargando



El diseño de esta vista a comparación del anterior es más sencillo de manejar debido a su independencia del resto de módulos, también contiene un diseño de un formulario sencillamente lógico con sus botones de guardar y regresar claramente visibles.

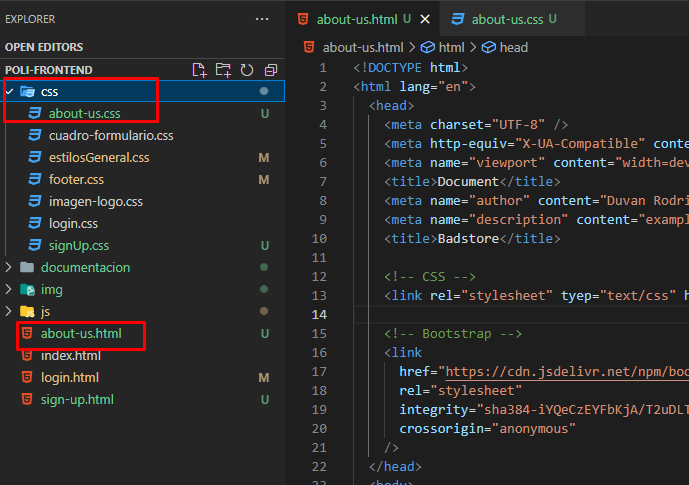
# **About Us:**

Se realiza un diseño que sea más amigable con el usuario y de fácil visualización para que tengan acceso a la información más relevante de la página.



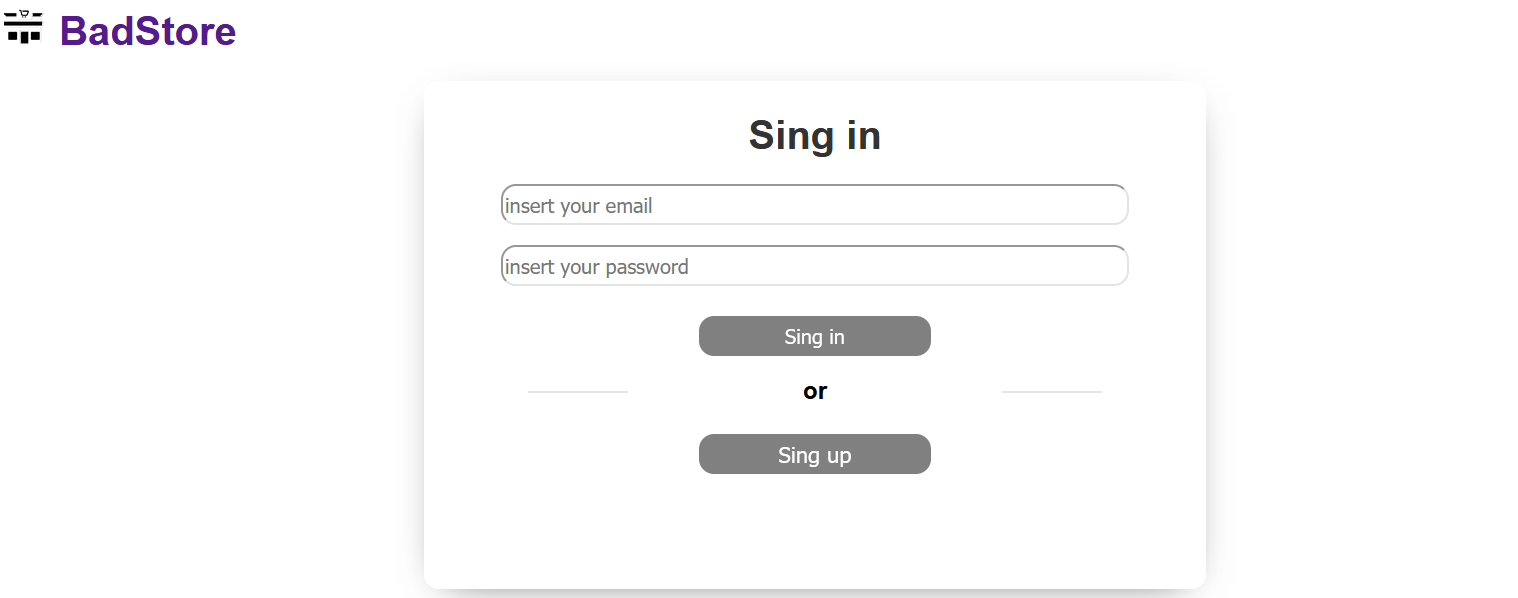
Para este desarrollo se usaron las cards de bootstrap más css personalizado, cada carta quiere decir la información a la que tendrá acceso.

El código usado se dejó de manera independiente, es decir que tiene su propio css, para este caso que solo era diseño visual no se utilizó javascript.



**Login de usuario**

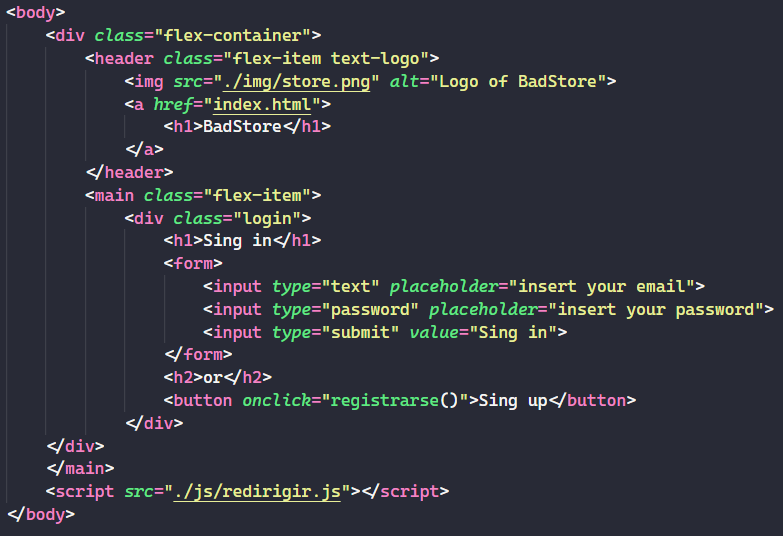
En el caso del Login se permite la visualización en los distintos tipos de pantalla



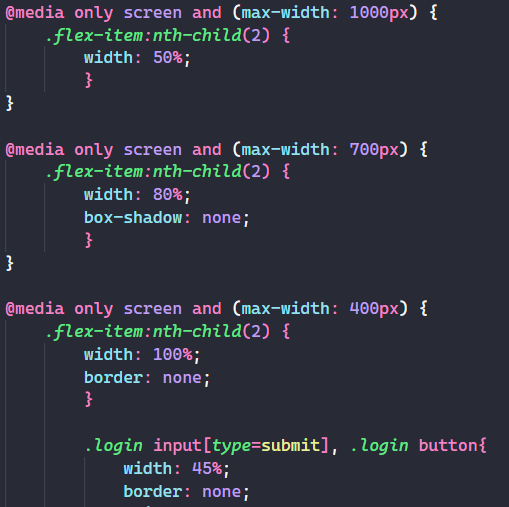
Interfaz de usuario gráfica

Descripción generada automáticamente

Se programo para que se ajuste a los distintos tipos de pantalla manteniendo el diseño de los mockups realizados con anterioridad:

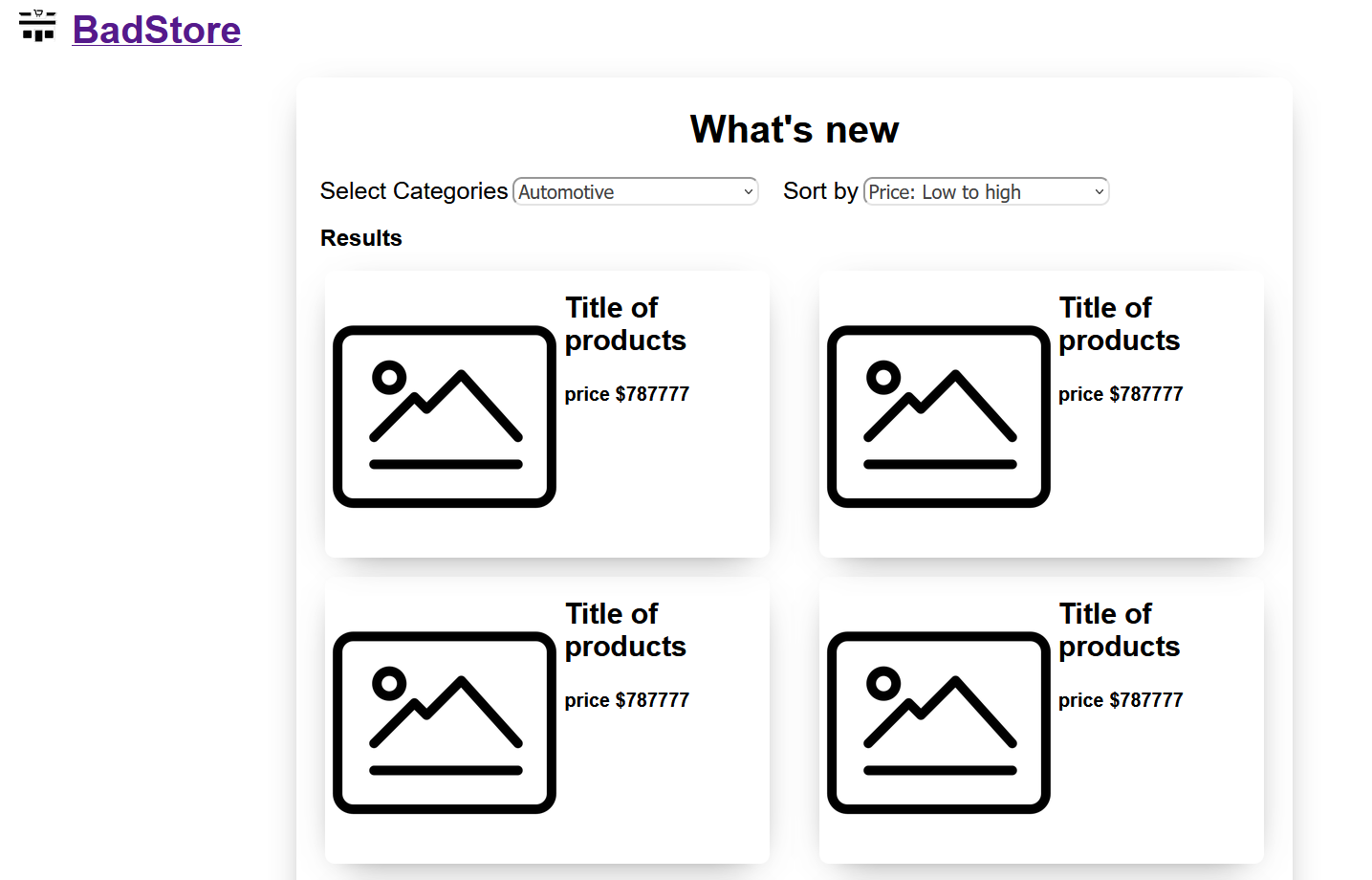


Se manejo html5 para su contenido y mediante css se realizaron los ajuste de diseño y compatibilidad con las demás pantallas mediante media querys y ordenando todo mediante carpetas en la cual las img y css están separadas para poder ser reutilizadas.



**What’s New**

Para la codificación de los What’s new se considero el uso de grid en css para permitir a futuro rellenar cada producto desde DB y que automáticamente se acomode de igual forma se maneja un diseño responsive para las diferentes pantallas:



Como podemos observar en el diseño para dispositivos con una resolución superior a 800px

Se mostrara este diseño en el cual se contempló que mediante css grid que cada uno de los productos este encapsulado en un div en el cual se aplican las configuraciones del grid



El cual permite que cualquier nuevo producto cumpla con este estándar y sea responsive sin importar la pantalla

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Como podemos ver para adaptar a cada pantalla es solo modificar el grid de la clase parent en la cual se contemplar los 3 item que este tiene, en caso de necesitar una nueva adaptación gracias a esta implementación el cambio es relativamente sencillo.

