

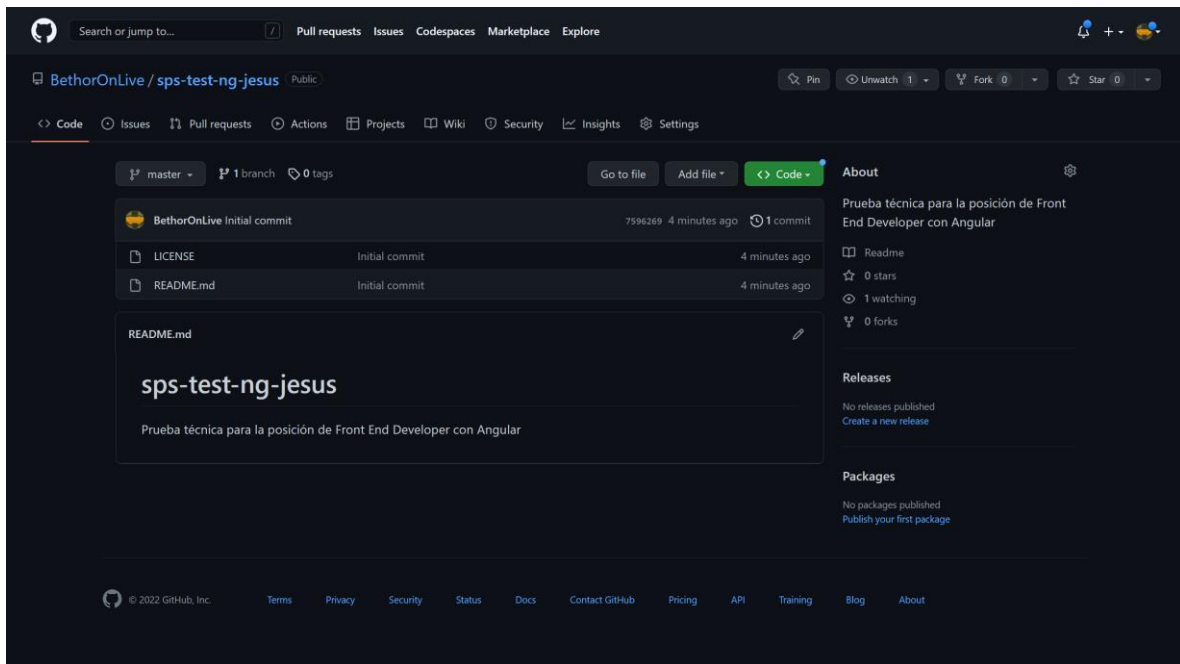


Front End Developer Test

Angular

12/12/2022

Se crea un repositorio público en GitHub para subir el proyecto.



Se clona el repositorio de GitHub a local.

```
computadora@DESKTOP-8H2VSGO MINGW64 ~  
$ git clone https://github.com/BethorOnLive/sps-test-ng-jesus.git  
Cloning into 'sps-test-ng-jesus'..  
remote: Enumerating objects: 4, done.  
remote: Counting objects: 100% (4/4), done.  
remote: Compressing objects: 100% (4/4), done.  
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0  
Receiving objects: 100% (4/4), done.
```

Se crea un nuevo proyecto Angular.

```
MINGW64/c:/Users/computadora/sps-test-ng-jesus  
computadora@DESKTOP-8H2VSGO MINGW64 ~/sps-test-ng-jesus (master)  
$ ng new shopee-store  
CREATE shopee-store/angular.json (2952 bytes)  
CREATE shopee-store/package.json (1043 bytes)  
CREATE shopee-store/README.md (1065 bytes)  
CREATE shopee-store/tsconfig.json (863 bytes)  
CREATE shopee-store/.editorconfig (274 bytes)  
CREATE shopee-store/.gitignore (548 bytes)  
CREATE shopee-store/.browserslistrc (600 bytes)  
CREATE shopee-store/karma.conf.js (1429 bytes)
```

Se hace el primer commit.

```
computadora@DESKTOP-8H2VSGO MINGW64 ~/sps-test-ng-jesus/shopee-store (master)
$ git commit -am "Initial commit"
[master c6fca50] Initial commit
28 files changed, 21359 insertions(+)
create mode 100644 shopee-store/.browserslistrc
create mode 100644 shopee-store/.editorconfig
create mode 100644 shopee-store/.gitignore
create mode 100644 shopee-store/.vscode/extensions.json
create mode 100644 shopee-store/.vscode/launch.json
```

```
computadora@DESKTOP-8H2VSGO MINGW64 ~/sps-test-ng-jesus/shopee-store (master)
$ git pull origin master
From https://github.com/BethorOnLive/sps-test-ng-jesus
* branch      master      -> FETCH_HEAD
Already up to date.

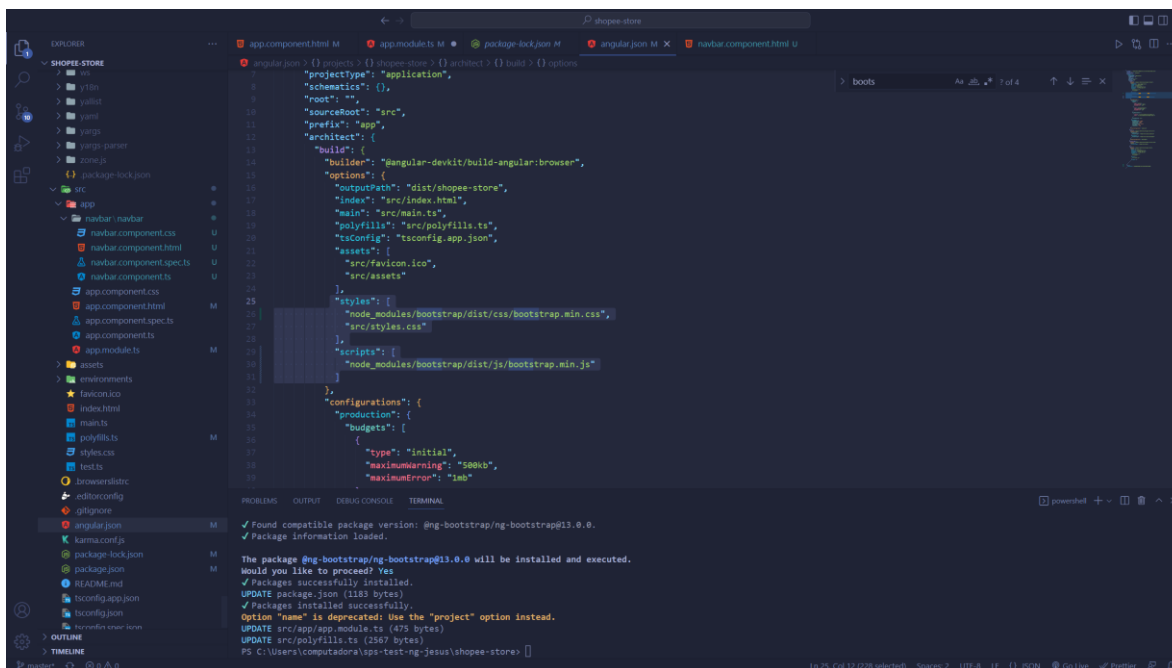
computadora@DESKTOP-8H2VSGO MINGW64 ~/sps-test-ng-jesus/shopee-store (master)
$ git push origin master
Enumerating objects: 36, done.
Counting objects: 100% (36/36), done.
```

Se corre el proyecto en local host.

The screenshot shows the Angular CLI web interface. At the top, a blue bar contains the Angular logo, the word 'Welcome', and social media icons for Twitter and YouTube. Below this, a red circular icon with a white rocket ship is next to a blue button that says 'shopee-store app is running!'. The main area is divided into two sections: 'Resources' and 'Next Steps'. The 'Resources' section has a heading 'Here are some links to help you get started:' and five buttons: 'Learn Angular >', '< CLI Documentation >', 'Angular Material >', 'Angular Blog >', and 'Angular DevTools >'. The 'Next Steps' section has a heading 'What do you want to do next with your app?' and four buttons: '+ New Component', '+ Angular Material', '+ Add PWA Support', and '+ Add Dependency'. Below these buttons are two more buttons: '+ Run and Watch Tests' and '+ Build for Production'. At the bottom, there is a 'Love Angular' button and a 'Give our repo a star.' button with a star icon. In the foreground, a terminal window is open, showing the output of the 'ng serve' command. The terminal output includes the following text: 'ng serve --o', 'Generating browser application bundles (phase: setup)...', 'Browser application bundle generation complete.', a table of initial chunk files, 'Initial Total | 2.34 Mb', 'Build at: 2022-12-11T16:15:05.323Z - Hash: 8015c9c0b01f59fe - Time: 12663ms', and 'Compiled successfully.'.

Initial Chunk Files	Names	Raw Size
vendor.js	vendor	1.77 Mb
polyfills.js	polyfills	328.06 kb
styles.css, styles.js	styles	210.09 kb
main.js	main	48.05 kb
runtime.js	runtime	6.52 kb

Se instala ng-bootstrap para los componentes UI.



<https://ng-bootstrap.github.io/#/getting-started>

Se cargan algunas fuentes para darle estilo a la tipografía.



<https://fonts.google.com/specimen/Basic?query=basic>

Se instala Font Awesome para usar iconos.

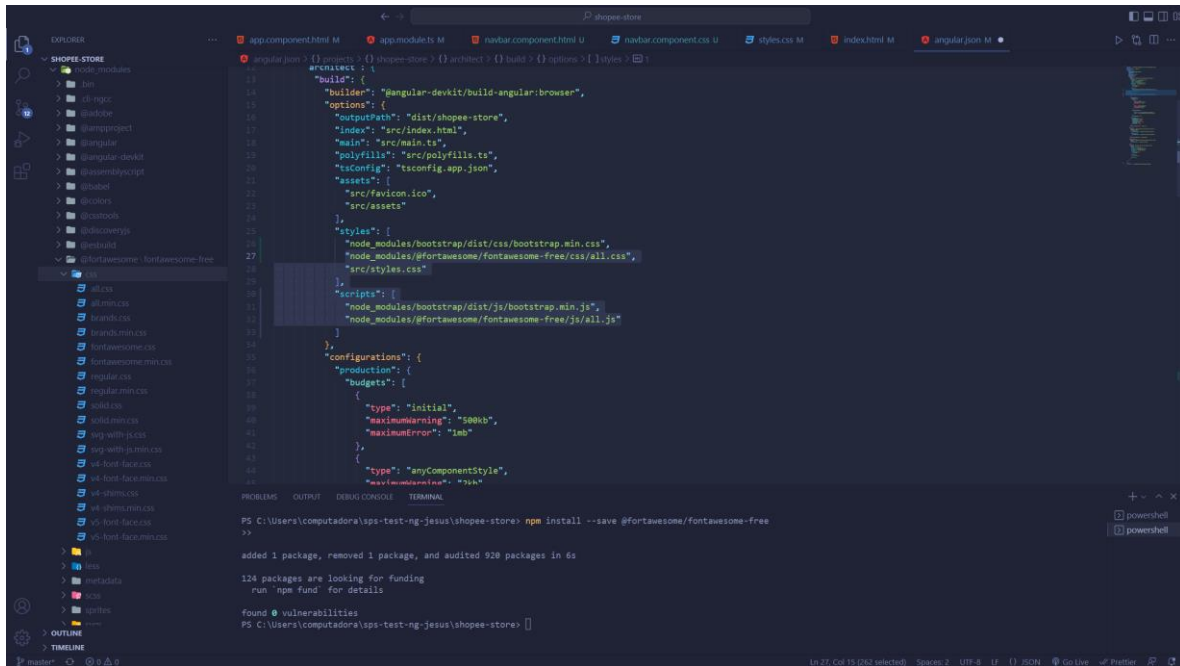
The Free Package

If you need just the Free icons, we have a slimmed-down free-only package for that. Install with this command:

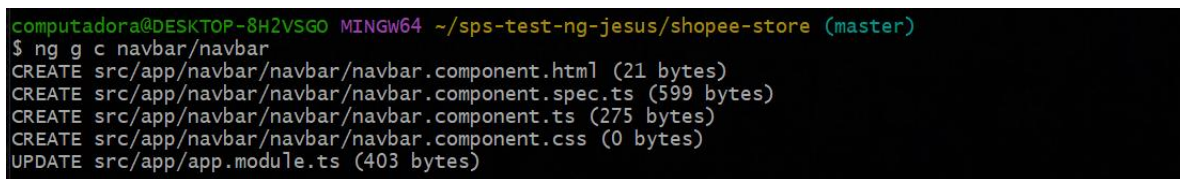
npm yarn

```
npm install --save @fortawesome/fontawesome-free
```

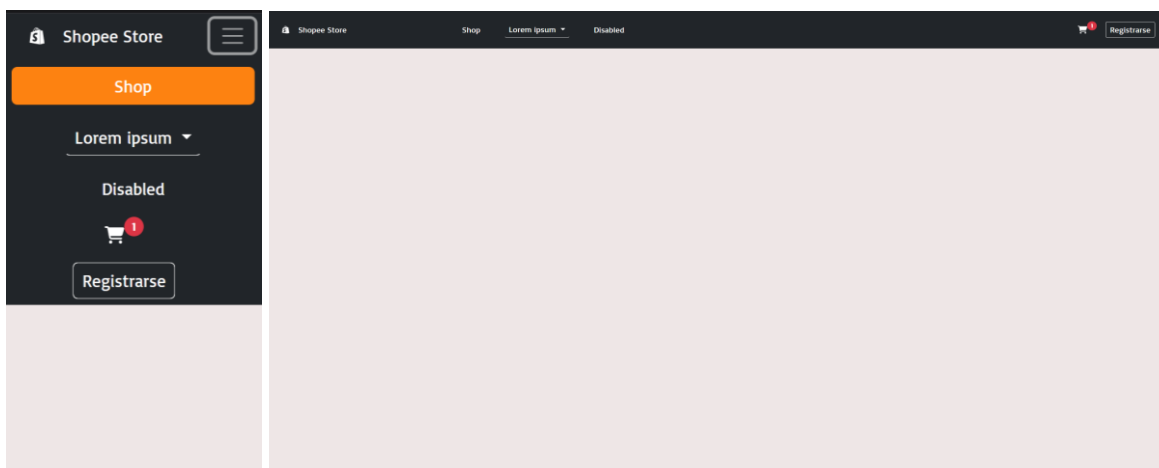




Se crea una carpeta y un componente para el navbar.



Y se agrega una navbar de los componentes de Bootstrap y posteriormente se personaliza con estilos CSS propios y se optimiza el responsive para visualizar en dispositivos móviles.



Se agrega un nuevo componente llamado “register” el cual sustituye el botón “Registrarse” en el navbar.

```
computadora@DESKTOP-8H2VSGO MINGW64 ~/sps-test-ng-jesus/shopee-store (master)
$ ng g c register
CREATE src/app/register/register.component.html (23 bytes)
CREATE src/app/register/register.component.spec.ts (613 bytes)
CREATE src/app/register/register.component.ts (283 bytes)
CREATE src/app/register/register.component.css (0 bytes)
UPDATE src/app/app.module.ts (564 bytes)
```

Al dar click en el botón se abre un modal tamaño lg con la ayuda del componente de bootstrap para modales, el cual se importa en el componente donde se va a utilizar:

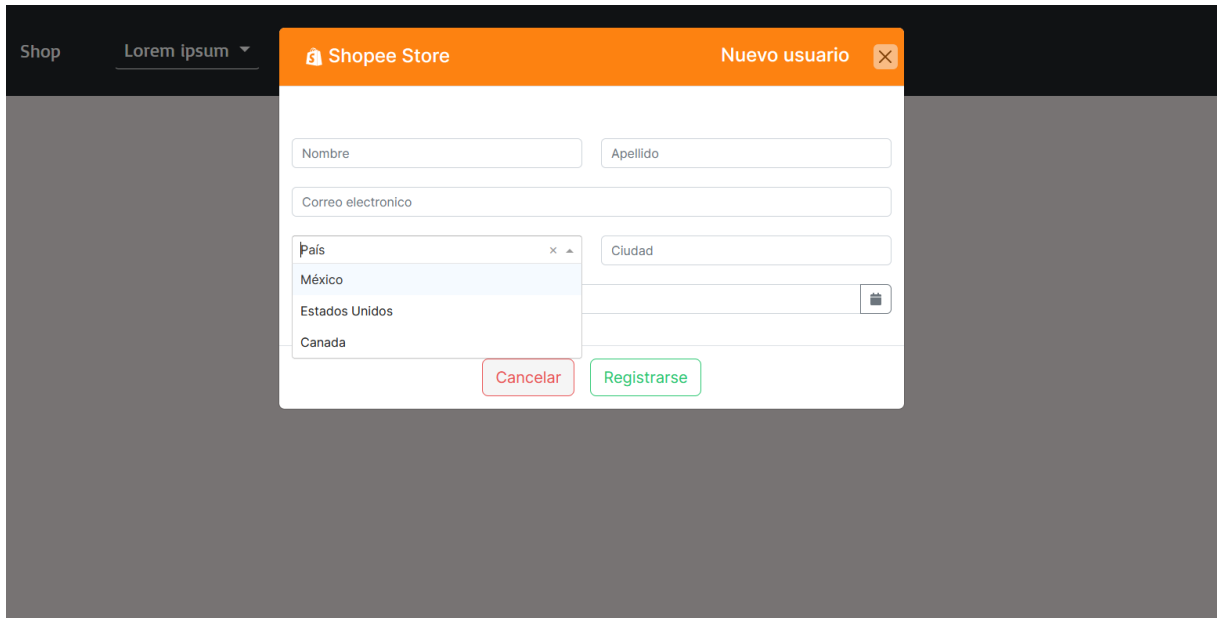
```
1 import { NgbModal } from '@ng-bootstrap/ng-bootstrap';
```

El objetivo del formulario es simular el registro de los datos de un usuario nuevo, haciendo uso de **formularios reactivos**, los campos se validan y devuelven un mensaje de error en caso de no cumplir con los requerimientos.

The screenshot shows a web application with a dark header. On the left, there's a sidebar with a dropdown menu labeled 'Lorem ipsum'. The main content area features a modal window with an orange header bar containing the text 'Shopee Store' and 'Nuevo usuario' with a close button. The modal body contains a registration form with the following fields: 'Nombre' (Name), 'Apellido' (Last Name), 'Correo electronico' (Email), 'Pais' (Country), 'Ciudad' (City), and a date field labeled 'yyyy-mm-dd' with a calendar icon. At the bottom of the modal, there are two buttons: 'Cancelar' (Cancel) in a red-outlined box and 'Registrarse' (Register) in a green-outlined box.

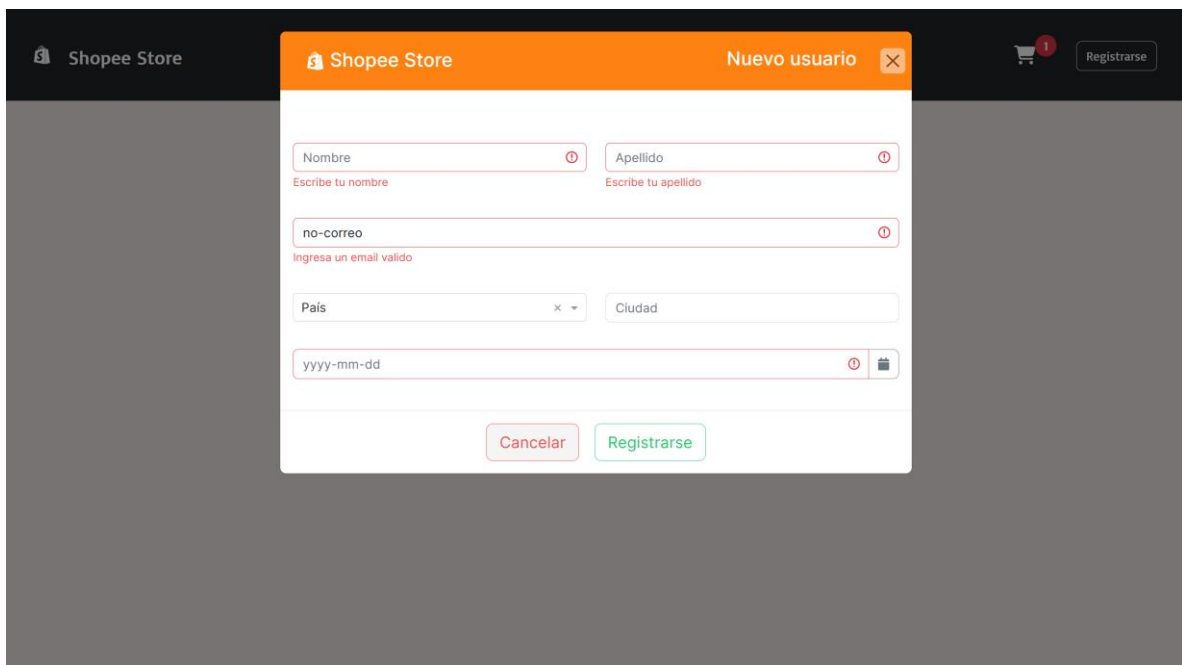
Se agregó un ng-select en campo países.

<https://github.com/ng-select/ng-select>



The screenshot shows a modal window titled 'Nuevo usuario' for the 'Shopee Store'. It contains a registration form with the following fields: 'Nombre', 'Apellido', 'Correo electronico', 'País' (a dropdown menu with 'México', 'Estados Unidos', and 'Canada' selected), and 'Ciudad'. There are also 'Cancelar' and 'Registrarse' buttons at the bottom of the form.

Después se personalizan estilos para los mensajes de validación del formulario que son disparados a través de una condicional **ngClass** y **class.active**.

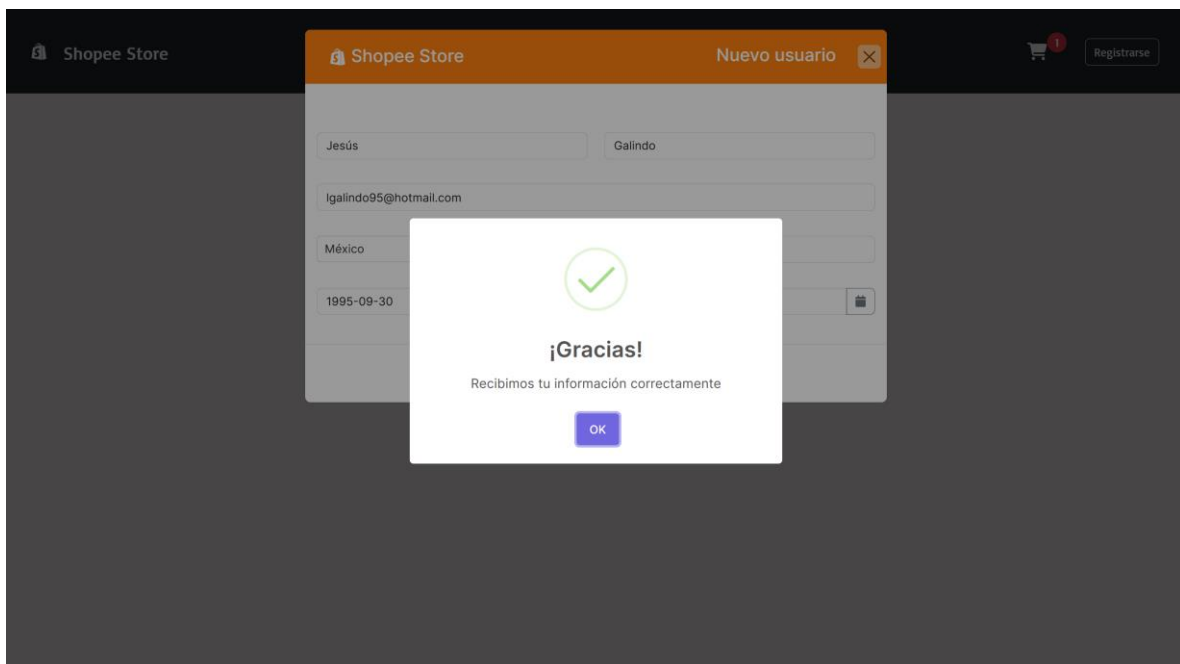
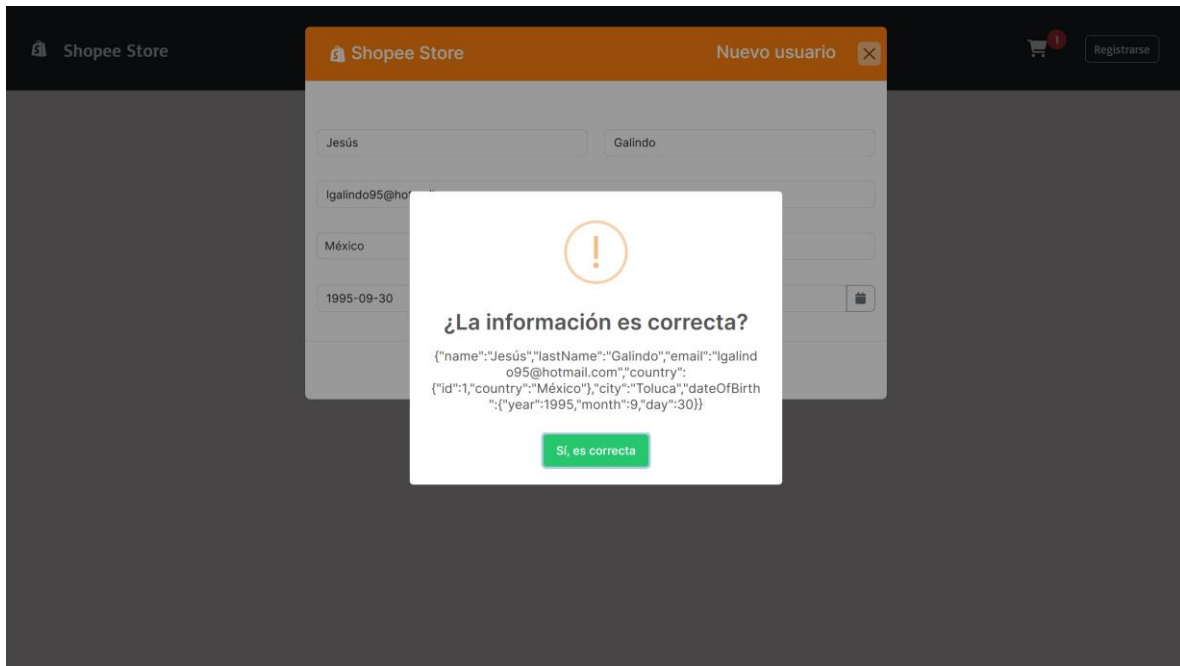


The screenshot shows the same 'Nuevo usuario' modal window, but now with validation messages. The 'Nombre' field has a red border and the message 'Escribe tu nombre'. The 'Apellido' field has a red border and the message 'Escribe tu apellido'. The 'Correo electronico' field has a red border and the message 'Ingresa un email valido'. The 'País' field is now a standard dropdown menu. There are also 'Cancelar' and 'Registrarse' buttons at the bottom of the form.

Para esta simulación se utiliza una alerta de confirmación de envío de datos, mostrando los mismos en formato **JSON** a manera de demostración del request al hacer submit.

Para lo anterior se requiere alguna librería como Sweet Alert:

<https://github.com/sweetalert2/ngx-sweetalert2>



Se agrega un componente nuevo que contendrá el banner de fondo, la barra de búsqueda y un select para hacer filtros.

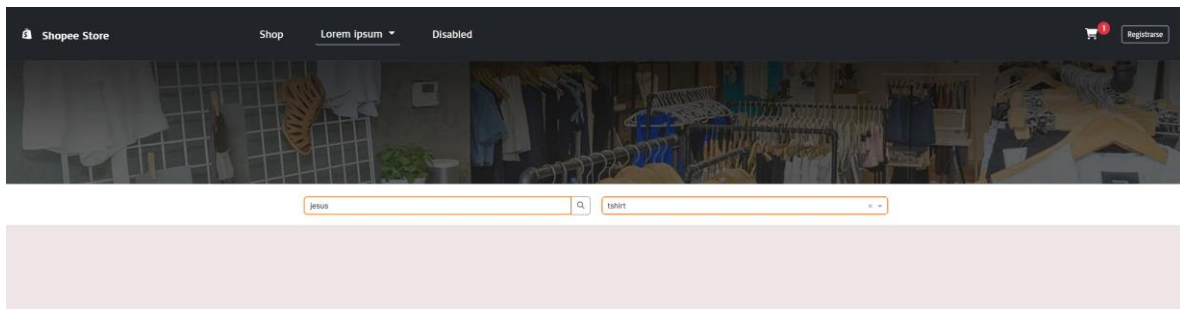
Para esta barra de búsqueda se agregaron controles reactivos uno para la búsqueda libre y otro para un select que filtrará por tipo de producto. Cabe mencionar que hasta este punto aun no se hacen los filtros puesto que no se han agregado servicios, sin embargo, se dejan los controles listos.

```
1 import { Component, OnInit, ViewEncapsulation } from '@angular/core';
2 import { FormControl } from '@angular/forms';
3 import { debounceTime } from 'rxjs/operators';
```

debounceTime es un operador de **RxJS** parecido al método `setTimeout()` que nos va ayudar a devolver la respuesta de los controles en un intervalo de tiempo determinado.

```
1 searchProduct = new FormControl();
2 searchCategory = new FormControl();
```

```
1 ngOnInit(): void {
2
3     this.searchProduct.valueChanges
4     .pipe(debounceTime(500))
5     .subscribe(value => {
6         console.log('searchProduct has changed:', value)
7     });
8
9     this.searchCategory.valueChanges
10    .pipe(debounceTime(500))
11    .subscribe(value => {
12        console.log('searchCategory has changed:', value)
13    });
14 }
```

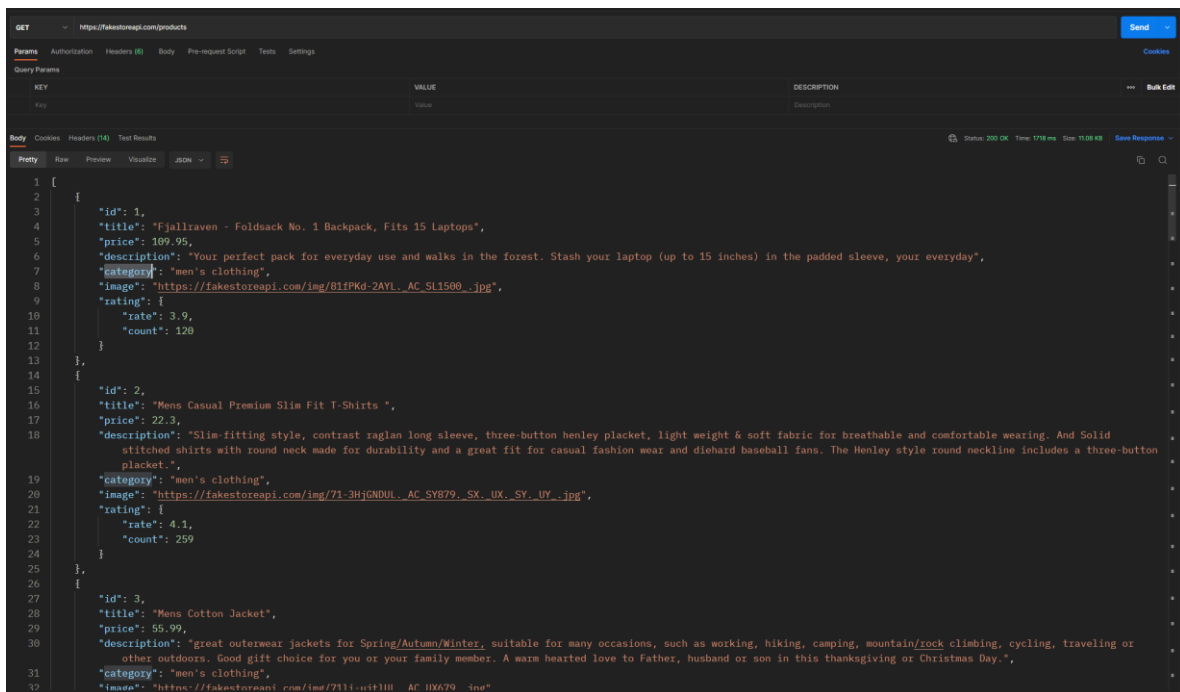


Para la sección principal donde se mostrarán las tarjetas se crea un nuevo componente y un servicio para empezar a integrar el consumo del api, para este proyecto se utiliza la siguiente:

<https://fakestoreapi.com/>

Es necesario agregar HttpClientModule en nuestro modulo.

Probamos en Postman:



```
import { HttpClientModule } from '@angular/common/http';
```

Importar nuestro servicio:

```
import { StoreServiceService } from '../main-content/store-service.service';
```

y agregarlo en providers:

```
providers: [HttpClientModule, StoreServiceService],
```

```
1 import { Component, OnInit } from '@angular/core';
2 import { StoreServiceService } from './store-service.service'
3
4 @Component({
5   selector: 'app-main-content',
6   templateUrl: './main-content.component.html',
7   styleUrls: ['./main-content.component.css']
8 })
9 export class MainContentComponent implements OnInit {
10
11   constructor(
12     private _api: StoreServiceService
13   ) { }
14
15   ngOnInit(): void {
16   }
17
18 }
19
```

Finalmente despues de integrar servicios y estilar un poco las cards el resultado es:

