

Apartados

Disponer de una interfaz web que liste (con paginado a ser posible), las issues de cualquier repositorio de github.

Para la web he utilizado el framework CSS de Material, algo simple, he utilizado los módulos de botones, Lista, Input y ToolBar.

```
MatButtonModule, MatListModule, MatInputModule, MatToolbarModule
```

La interfaz deberá solicitar la URL del repositorio, y listar las issues, con toda la información que consideres interesante incluir de cada issue.

He considerado que para mostrar una lista lo más limpia posible, lo mejor sería mostrar el mensaje del problema y la url a este mismo en github. A posteriori, se podría lanzar un evento dentro de la misma web que nos diese un desglose del problema en concreto al hacer clic sobre él.

El proyecto ha sido arrancado usando angular-cli, se usa angular 6.

Descripción de la solución

Dentro de la estructura creada, he hecho uso del apartado app para crear mi aplicación.

Lo primero, después de estudiar la documentación de la API REST de git hub, fue importar los módulos HttpClient para poder trabajar con ello.

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { BrowserAnimationsModule } from '@angular/platform-browser/animations';

import { AppComponent } from './app.component';
import { HttpClientModule } from '@angular/common/http';
import { MaterialModule } from './material.module';
import { FormsModule } from '@angular/forms';

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule,
    HttpClientModule,
    BrowserAnimationsModule,
    MaterialModule,
    FormsModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

A continuación, realice una interfaz básica con dos cajas input y un botón el cual no devolvía nada.

Siguiente paso, fue crear la función que hace la llamada al servicio y recupera el json. Con este json se lo paso al template html mediante un binding, y haciendo uso de *ngFOR recorro todo el array json mostrando los datos que creo que son más relevantes. También habilito un link a la url para poder ver en detalle el problema.

Una vez escrita dicha funcionalidad, actualice la aplicación para que hiciese uso del framework CSS Material.

Añadí las dependencias al proyecto, y para instanciar los componentes, en vez de hacerlo dentro de la propia app, he preferido hacerlo en un módulo aparte, por si en un futuro fuesen más, tener un código más limpio.

El módulo lo he llamado material.module.ts en él se pueden ver los diferentes módulos que he utilizado y exportado.

Captura de la aplicación mostrando el listado de problemas:

