

Práctica Grupal: Single Page Application

1. Crear una carpeta para el proyecto, abrir una ventana de comandos y ubicarse dentro de la carpeta. Crear un proyecto de Angular con el comando (el comando tarda algo de tiempo en finalizar):

```
$ ng new singlepageapp
```

2. Validar la creación del proyecto, visualizándolo en el navegador.
3. Abrir la carpeta “singlepageapp” con Atom.
4. En una ventana de comandos, ubicados en la carpeta del proyecto, crear un componente para el encabezado utilizando Angular CLI, con el siguiente comando:

```
\SinglePageApplication\singlepageapp>ng g c componentes/comunes/encabezado
```

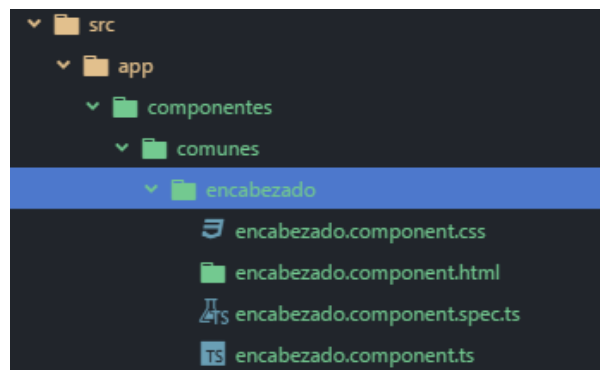
g = Generate

c = Component

Ruta donde se generará el componente: “componentes/comunes/”.

Nombre del componente: “encabezado”

Se deben generar los archivos necesarios para comenzar a programar nuestro componente:



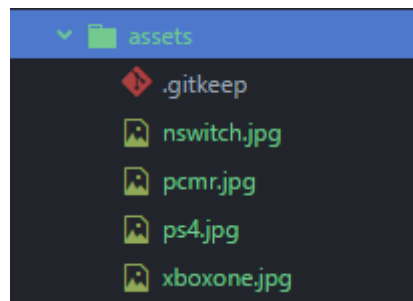
El archivo “app.module.ts” se actualiza de forma automática importando y declarando el componente creado:

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

import { AppComponent } from './app.component';
import { EncabezadoComponent } from './componentes/comunes/encabezado/encabezado.component';

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

5. Agregar soporte de bootstrap a la aplicación.
6. Copiar las imágenes proporcionadas a la carpeta “assets”:

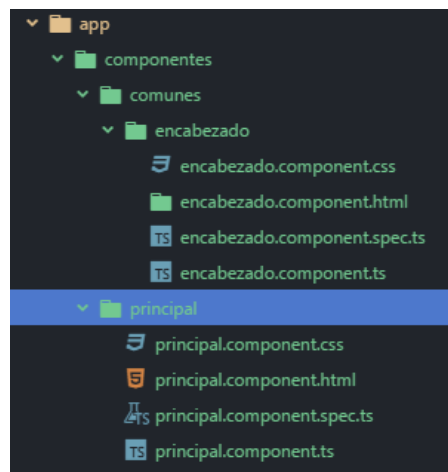


7. Colocar el siguiente código en el archivo “encabezado.component.html”:

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarTogglerDemo02" aria-controls="navbarTogglerDemo01" aria-expanded="false"
aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarSupportedContent">
    <ul class="navbar-nav mr-auto mt-2 mt-lg-0">
      <li class="nav-item active">
        <a class="nav-link" href="#">Principal <span class="sr-only">(current)</span></a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Consolas</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Acerca De</a>
      </li>
    </ul>
    <form class="form-inline my-2 my-lg-0">
      <input class="form-control mr-sm-2" type="search" placeholder="Search">
      <button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search</button>
    </form>
  </div>
</nav>
```

8. Crear un componente para la página principal utilizando Angular CLI, directamente dentro de la carpeta “componentes”, con el siguiente comando:

```
SinglePageApplication\singlepageapp>ng g c componentes/principal
```



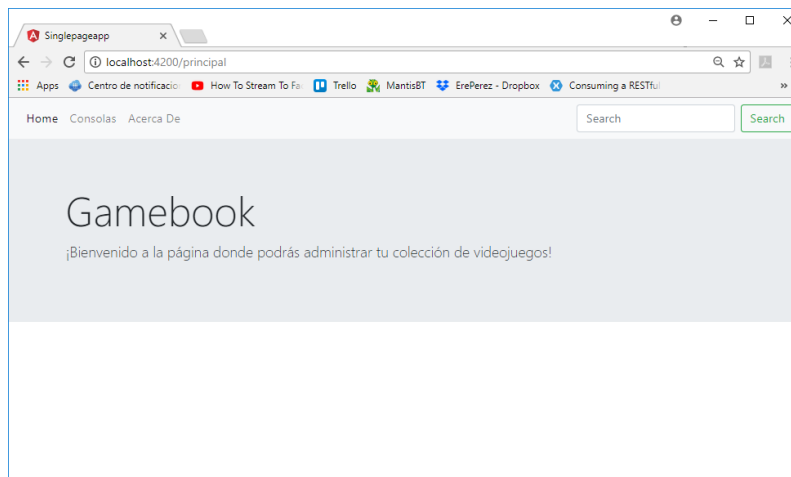
9. Colocar el siguiente código en el archivo “principal.component.html”:

```
<div class="jumbotron jumbotron-fluid">
  <div class="container">
    <h1 class="display-4">Gamebook</h1>
    <p class="lead">¡Bienvenido a la página donde podrás administrar tu colección de videojuegos!</p>
  </div>
</div>
```

10. Colocar el siguiente código en el archivo “app.component.html”:

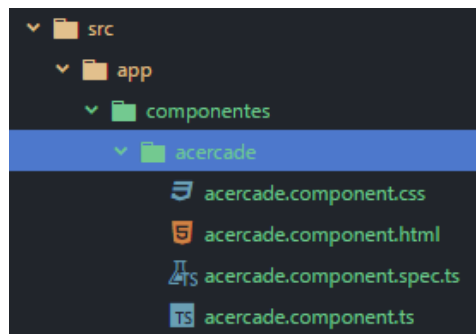
```
<app-encabezado></app-encabezado>
<app-principal></app-principal>
```

11. Validar los cambios en el navegador:



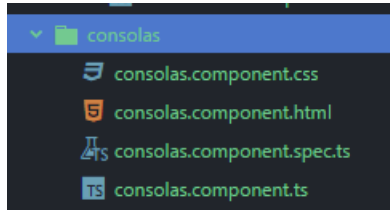
12. Crear un componente para el menú “Acerca de” utilizando Angular CLI, directamente dentro de la carpeta “componentes”, con el siguiente comando:

```
SinglePageApplication\singlepageapp>ng g c componentes/acercade
```

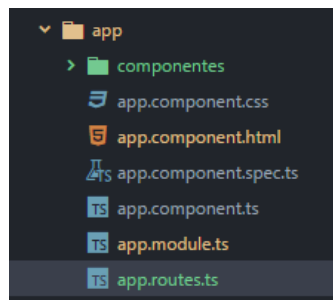


13. Crear un componente para la página de consolas utilizando Angular CLI, directamente dentro de la carpeta “componentes”, con el siguiente comando:

```
SinglePageApplication\singlepageapp>ng g c componentes/consolas
```



14. Dentro de la carpeta “app” del proyecto, crear un archivo llamado “app.routes.ts”:



15. Coloque el siguiente código en el archivo “app.routes.ts”:

```
import { RouterModule, Routes } from '@angular/router';

//Se tienen que importar todas las rutas que vamos a utilizar
import { PrincipalComponent } from './componentes/principal/principal.component';
import { AcercadeComponent } from './componentes/acercade/acercade.component';
import { ConsolasComponent } from './componentes/consolas/consolas.component';

const APP_ROUTES: Routes = [

  //Se inicializan todas las rutas que vamos a utilizar.
  { path: 'principal', component: PrincipalComponent },
  { path: 'acercade', component: AcercadeComponent },
  { path: 'consolas', component: ConsolasComponent },
  { path: '**', pathMatch: 'full', redirectTo: 'principal' }

];

//Se tiene que importar APP_ROUTING en el archivo app.module.ts como un import,
//y dentro del arreglo de imports[]
export const APP_ROUTING = RouterModule.forRoot(APP_ROUTES);
```

16. Importar las rutas en el archivo “app.module.ts”:

```
import { AppComponent } from './app.component';
import { EncabezadoComponent } from './componentes/comunes/encabezado/encabezado.component';
import { PrincipalComponent } from './componentes/principal/principal.component';
import { AcercadeComponent } from './componentes/acercade/acercade.component';
import { ConsolasComponent } from './componentes/consolas/consolas.component';
import { APP_ROUTING } from './app.routes';

@NgModule({
  declarations: [
    AppComponent,
    EncabezadoComponent,
    PrincipalComponent,
    AcercadeComponent,
    ConsolasComponent
  ],
  imports: [
    BrowserModule,
    APP_ROUTING
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

17. Modificar el código html del componente “encabezado” de tal forma que ahora se utilice “routerLink” para definir qué página debe abrir cada liga. Se debe reemplazar la propiedad “href” por “[routerLink]”:

```
<li class="nav-item active">
  <a class="nav-link" [routerLink]="['principal']">principal <span class="sr-only">(current)</span></a>
</li>
<li class="nav-item">
  <a class="nav-link" [routerLink]="['consolas']">Consolas</a>
</li>
<li class="nav-item">
  <a class="nav-link" [routerLink]="['acercade']">Acercade</a>
</li>
```

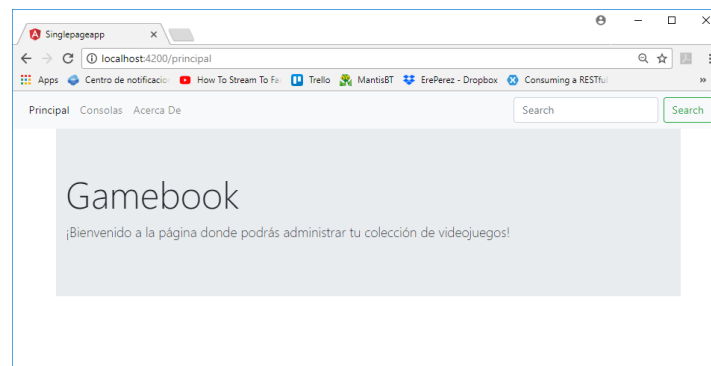
18. Colocamos el siguiente código en el archivo “app.component.html”:

```
<app-encabezado></app-encabezado>

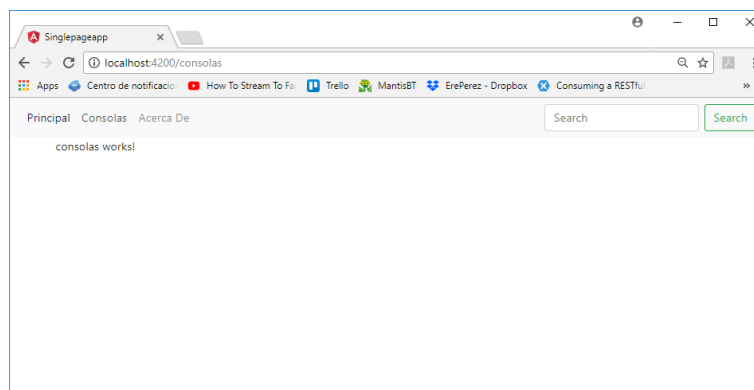
<div class="container">
  <!-- Le indicamos a Angular que aquí queremos mostrar las rutas que definamos. -->
  <router-outlet></router-outlet>
</div>
```

19. Validar la funcionalidad. Cada liga debe verse de la siguiente forma:

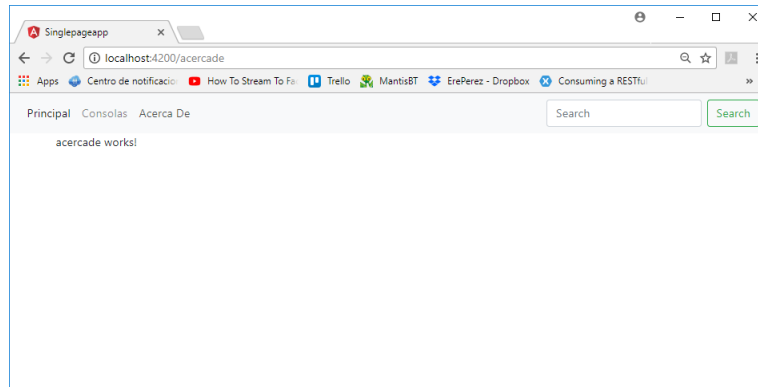
- Principal:



- Consolas:

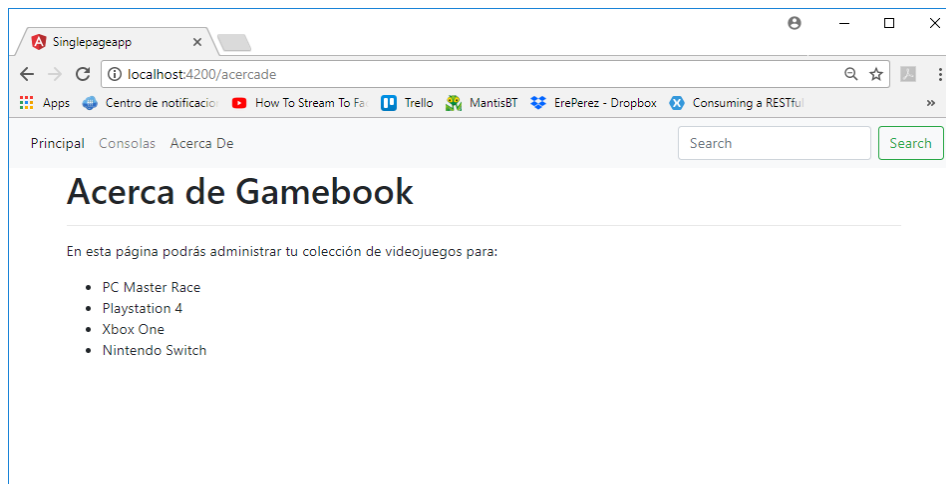


- Acerca De:



20. Cambiar el código del archivo “acercade.component.html” por el siguiente:

```
<h1>Acerca de Gamebook</h1>
<hr>
<p>
  En esta página podrás administrar tu colección de videojuegos para:
</p>
<ul>
  <li>PC Master Race</li>
  <li>Playstation 4</li>
  <li>Xbox One</li>
  <li>Nintendo Switch</li>
</ul>
```



21. Colocar el siguiente código en el archivo “consolas.component.html”:

```
<h1>Mis Consolas</h1>
<hr>
```



```
<div class="row">
  <div class="card" style="width: 18rem;">
    
    <div class="card-body">
      <h5 class="card-title">PC Master Race</h5>
      <p class="card-text">Juegos para PC</p>
      <a href="#" class="btn btn-primary">Información</a>
    </div>
  </div>
  <div class="card" style="width: 18rem;">
    
    <div class="card-body">
      <h5 class="card-title">Playstation 4</h5>
      <p class="card-text">Juegos para PS4</p>
      <a href="#" class="btn btn-primary">Información</a>
    </div>
  </div>
  <div class="card" style="width: 18rem;">
    
    <div class="card-body">
      <h5 class="card-title">Xbox One</h5>
      <p class="card-text">Juegos para Xbox One</p>
      <a href="#" class="btn btn-primary">Información</a>
    </div>
  </div>
  <div class="card" style="width: 18rem;">
    
    <div class="card-body">
      <h5 class="card-title">Nintendo Switch</h5>
      <p class="card-text">Juegos para Nintendo Switch</p>
      <a href="#" class="btn btn-primary">Información</a>
    </div>
  </div>
</div>
```

22. Validar los cambios en el navegador.

