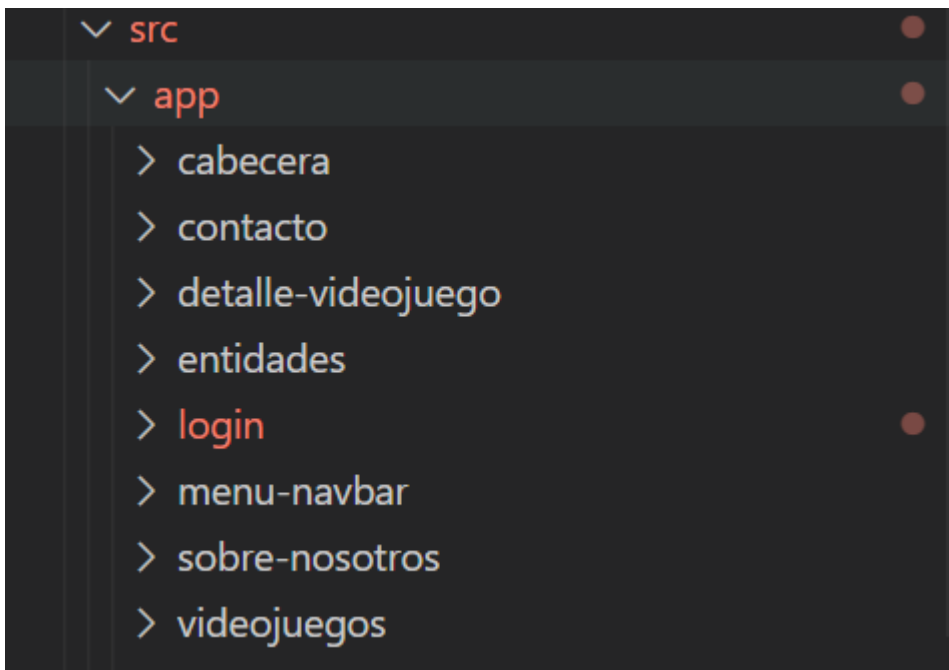


Para la realización de la tarea creamos los siguientes módulos que se encuentran dentro de la carpeta app.



Dentro de los módulos creamos los siguientes componentes:

1. Un archivo html donde vienen estructuradas las páginas.
2. Un archivo css donde definimos los estilos.
3. Un archivo ts donde definimos la funcionalidad.

Para la realización de la tarea, hemos utilizado la consola de comandos con los respectivos comandos de creación de la con

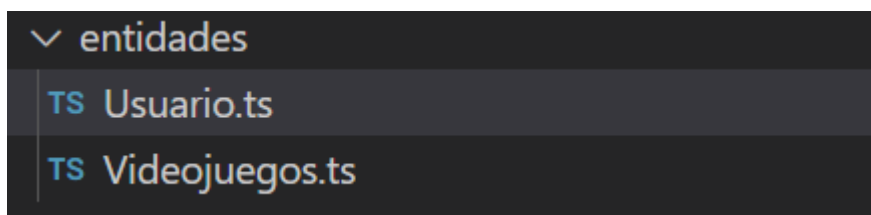
- Dentro de los módulos de **cabecera** y **contacto** hemos puesto información de la página.
- En la parte de **detalle-videojuego**, dentro del componente HTML hemos creado una tabla con los atributos que nos pedía la actividad.

```
<h2>Detalle del videojuego:</h2>
<div class="card">
  
  <table>
    <tr>
      <th>ID</th>
      <td>{{identificador}}</td>
    </tr>
    <tr>
      <th>NOMBRE</th>
      <td>{{titulo}}</td>
    </tr>
    <tr>
      <th>COMPAÑÍA</th>
      <td>{{compania}}</td>
    </tr>
    <tr>
      <th>VALORACIÓN MEDIA</th>
      <td>{{valorMedio}}</td>
    </tr>
  </table>
</div> /.card
```

Dentro del componente TS definimos los parámetros que recibimos en la lista.

```
export class DetalleVideojuegoComponent implements OnInit {
  //estos son los datos que recogemos en la lista
  identificador : number = 0
  titulo : string = ""
  compania : string = ""
  imagen : string = ""
  valorMedio : number = 0
}
```

- **Entidades** hemos creado dos componentes TS Usuario y Videojuegos



Clase Usuario añadimos el constructor con parámetros y el método toString.  
Clase Videojuegos añadimos únicamente el constructor con parámetros.

- **Login** en el componente TS hemos creado un usuario en el constructor.

```
constructor(private router:Router) {
  let usuario: Usuario = new Usuario("user1", "1111")
  this.listaUsuarios.push(usuario);
}
```

Hemos definido el método para verificar el usuario.

```
public verificarUsuario(){
  for(let i = 0; i < this.listaUsuarios.length; i++){
    if(this.username === this.listaUsuarios[i].username && this.password === this.listaUsuarios[i].password) {
      this.router.navigate(['/videojuegos', this.username]);
      break
    }
  }
  else {
    alert('Usuario no existe.')
    this.username = ""
    this.password = ""
    break
  }
}
```

Este método comprueba si el usuario y la contraseña están registrados. Si el usuario introducido es estrictamente igual a algún usuario de los que están registrados y la contraseña introducida es correcta, permite el acceso.

En caso de que alguno de los datos introducidos sea erróneo no permite el acceso y muestra el mensaje alert.

- **Menú-navbar** es un menú superior con las distintas opciones (Videojuegos, Contacto, Sobre nosotros).
- **Sobre-nosotros** contiene información sobre la web Vandal.
- **Videojuegos** en el archivo TS creamos una lista de cinco videojuegos.

```
export class VideojuegosComponent implements OnInit {  
  
  listaVideojuegos : Videojuego[] = [] // Lista de videojuegos vacía  
  videojuego : Videojuego | null = null // Videojuego apunta a null  
  
  username:string= ""  
  
  constructor(route:ActivatedRoute) {  
  
    let videojuego : Videojuego = new Videojuego( 1,"The Legend of Zelda: Breath of the Wild", "Nintendo", "/assets/img/legend-zelda.jpg", 9.8)  
    this.listaVideojuegos.push(videojuego);  
    videojuego = new Videojuego( 2,"Persona 5 Royal", "ATLUS", "/assets/img/persona-5-royal.jpg", 9.1)  
    this.listaVideojuegos.push(videojuego);  
    videojuego = new Videojuego( 3,"Red Dead Redemption 2", "RockStar", "/assets/img/red-dead-redemption.jpg", 8.8)  
    this.listaVideojuegos.push(videojuego);  
    videojuego = new Videojuego( 4,"Call of Duty: Modern Warfare 2", "Infinity Ward", "/assets/img/call-of-duty.jpg", 7.4)  
    this.listaVideojuegos.push(videojuego);  
    videojuego = new Videojuego( 5,"Elden Ring", "FromSoftware", "/assets/img/elden-ring.jpg", 9.4)  
    this.listaVideojuegos.push(videojuego);  
  
    this.username=route.snapshot.params["username"]  
  }  
  
  ngOnInit(): void {  
  }  
}
```

Esta aplicación ha sido realizada con el IDE VSCode y comprobada en tiempo real a través del plugin Live Server. Aunque también se ha comprobado que pueda accederse a ella a través del archivo HTML. La actividad se ha ido realizando a lo largo de jornadas los 4 integrantes del equipo juntos.

GitHub: <https://github.com/addryx/Desarrollo-web-entorno-cliente-group/tree/main/ae3-angular>

Este proyecto está realizado por:

- Lierni Beltrán
- Adrián Fernández
- Marcos Sandá
- Isaac Calderón