

# DESARROLLO WEB EN ENTORNO CLIENTE

## PRÁCTICA 3

GRADO SUPERIOR DE DESARROLLO DE APLICACIONES WEB

UNIR FP - CURSO 2023-2024

Grupo 15

Alfredo Hernández- Rovati Bisso, Daniel Gómez Pérez,  
David Cenicerros Aransay, y Diego García Rodríguez.

## 1. ORGANIZACIÓN DE LA TAREA

La tarea se puede consultar en GitHub en el siguiente enlace: <https://github.com/alfredd/AE1-DWEC/tree/main/lista-videojuegos> en la carpeta lista-videojuegos.

El trabajo se ha desarrollado a través de Telegram y GitHub para fomentar la colaboración y la interacción entre los miembros del equipo y esto nos ha permitido trabajar de una forma ágil y organizada.

Dentro del equipo, Diego y Alfredo han trabajado centrados en el primer ejercicio y Daniel y David se han centrado en el requerimiento 2 y la elaboración de la memoria final de la tarea para la explicación de las actividades. Así mismo mutuamente hemos revisado los avances de cada uno de ellos para garantizar el buen desarrollo.

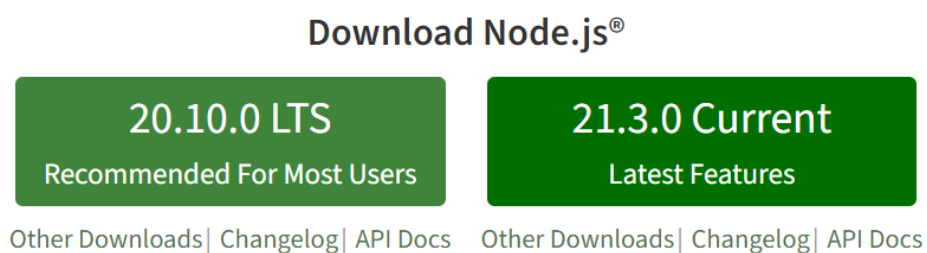
En el repositorio podrás encontrar 2 carpetas, una para cada requerimiento de la tarea con sus archivos correspondientes. Además, se podrá encontrar este informe en formato PDF para facilitar la revisión de la tarea y su correcto funcionamiento.

## 2.-DESARROLLO

Para poder realizar esta tarea, el primer paso a realizar, es la instalación tanto de angular como de node.js para poder comenzar a desarrollar el proyecto en sí.

Para esto, lo ideal es realizarlo a través de comandos.

Una vez hemos descargado node.js desde su página web (lo ideal es descargar la versión LTS)



For information about supported releases, see the [release schedule](#).

debemos ejecutar una serie de comandos para proceder a la instalación de forma correcta.

Debemos actualizar el NPM para obtener las versiones más actuales de Node, de forma que utilizaremos el siguiente comando **"npm install -g npm@latest"** con el que se nos instalará la última versión existente. Tras esto, debemos borrar la caché de npm, por si tuviéramos algún paquete de npm guardado en la caché. De esta forma, evitamos que nos pueda dar problemas, por lo que ejecutaremos el siguiente comando **"npm clean - - force"**. Un paso recomendable para poder trabajar más cómodamente (ya que suponemos que angular se descarga de una página segura, y trabajaremos con páginas seguras) debemos desinstalar los sistemas de auditorías a través del siguiente comando **"npm set audit false"**. En caso de tener alguna versión antigua de Angular CLI, deberíamos desinstalar a través del siguiente comando **"npm uninstall -g angular -cli"** aunque en caso de no haber instalado nunca el angular cli, este comando no sería necesario.

Tras borrar esto, debemos proceder a la descarga a través del siguiente comando: **"npm install -g @angular/cli@latest"** con el que se nos instalará la última versión de angular.

Ahora, ya podremos crear nuestro proyecto. Lo ideal es hacerlo a través de comandos, y hacerlo en el directorio que deseemos. En caso de trabajar con windows, navegaremos a través del comando **"cd"** para llegar a la carpeta en la que deseamos crear el proyecto.

Una vez situados en el workspace, debemos ejecutar el comando **"ng new"** para crear el proyecto de Angular. Nos preguntará si queremos configurar el Angular routing, cosa que deberemos hacer dependiendo del proyecto que sea, así como que hoja de estilos usaremos, en este caso será CSS.

Una vez tenemos el proyecto creado, podemos confirmar que funciona ejecutándolo a través del comando **"ng serve"**. Si queremos trabajar con el proyecto arrancado, este comando siempre tiene que estar en ejecución, ya que de no estarlo, el proyecto se pararía.

Para trabajar con los componentes, nos debemos situar en la carpeta src, en donde podremos crear los componentes que necesitemos a través del comando **"ng new component nombreComponente"** así como los servicios que necesitemos utilizar **"ng new service nombre"**. Es importante tener en cuenta, que en el caso de los servicios debemos crear una carpeta en la que guardarlos, ya que a diferencia de los componentes, los servicios no crean una carpeta de forma automática al ejecutar el comando.

## 2. DESARROLLO DEL PROYECTO EN ANGULAR

La primera tarea cuenta con el siguiente enunciado:

Para la creación del proyecto se ha usado la CLI y se ha nombrado el proyecto como lista-videojuegos.

Una vez creado el proyecto se han definido los componentes.

### COMPONENTES:

El proyecto tiene 3 componentes principales. Para generarlos se ha utilizado el comando:

*ng generate component XXX* con los siguientes componentes.

1. Videojuegos
2. Detalle
3. Login

A continuación, se describen cada uno de los html vinculados a ellos de forma más detallada:

#### **Videojuegos:**

En el archivo html, podemos encontrar una tabla que genera para cada elemento del array listaVideojuegos (situado en el videojuegos.service.ts) una fila con su id, su título, obtenido mediante un routerlink vinculando el id y por último la valoración media. Para ello se utiliza una directiva estructural como ngfor para iterar el array.

#### **Detalle:**

El componente detalle va a ofrecer más información al usuario sobre el videojuego seleccionado. A través de una directiva estructural tipo ngIf valida si se encuentra el videojuego. Si el videojuego no se encuentra, lanza un mensaje avisándose. Si se encuentra dentro del listado de videojuegos, muestra en un div, el nombre, la compañía, la valoración y una imagen asociada al mismo

#### **Login:**

En el archivo html podemos encontrar el formulario para iniciar sesión. Se ha trabajado para ofrecer una experiencia completa con la posibilidad de que el usuario elija si quiere que su contraseña sea visible o no.

En este caso debemos hacer uso del TWB para poder trabajar en la validación con campos como ngModel

```
<div class='form-group password-group'>
  <label for="contrasena">Introduce la contraseña</label>
  <div class="password-input-container">
    <input [type]="mostrarContrasena ? 'text' : 'password'" id="contrasena" name="contrasena" [(ngModel)]="contrasena" required>
    <button type="button" class="toggle-password" (click)="mostrarContrasena = !mostrarContrasena">
      <img [src]="mostrarContrasena ? 'assets/images/ojo-abierto.png' : 'assets/images/ojo-cerrado.jpeg'" alt="Toggle Password">
    </button>
  </div>
</div>
```

Primero se pide el usuario como un campo requerido y a continuación al contraseña con un toggle para ocultar o mostrar el texto.

Por último, si da error, se lanza un aviso en rojo mostrando en pantalla el problema.

## SERVICIOS:

Dentro de la aplicación se encuentran configurados dos servicios:

1. Videojuegos
2. Datosperfil

Estos sirven como repositorio de datos para crear los elementos de los que obtiene la información la app.

### 1. Videojuego:

En este archivo generamos y inicializamos los listados de videjuegos y su información correspondiente.

En primer lugar, importamos tanto el Injectable como la entidad videojuego.

A continuación, tras la etiqueta de injectable, exportamos la clase VideojuegosService. En primer lugar generamos dos variables privadas para comprobar si la lista está inicializada o no y para almacenar la lista.

Generamos un constructor vacío e incluimos algunas funciones para insertar la lista de videojuegos en el objeto, obtener un videojuego por id y por último inicializar la lista con la información almacenada y devolverla como objeto.

### 2. Datosperfil

Con esta servicio validamos la autenticación de usuarios a través de un sistema de usuario y contraseña. En la función de login validamos que la Info aportada cumple con el requerimiento.

Se desarrollar también una función de logout para vaciar el usuario actual y una función de getUsuarioActual para tener la Info de quien está logado.

#### ENTIDADES:

La entidad que encontramos en la aplicación es videojuego.ts en esta entidad formamos la clase Videojuego con un id numérico autoincremental y con un constructor con la siguiente estructura:

```
1 export class Videojuego{
2     private _id: number;
3     static idAuto = 0;
4
5     constructor(public titulo: string, public compania: string, public imagen: string, public valoracionMedia: number, public descripcion: string)
6     {
7         this._id = ++Videojuego.idAuto;
8         this.titulo = titulo;
9         this.compania = compania;
10        this.imagen = imagen;
11        this.valoracionMedia = valoracionMedia;
12        this.descripcion = descripcion;
13    }
14
15    public get id(): number{
16        return this._id;
17    }
18 }
19 }
```

Se añade también un getter para obtener el id.

#### ROUTING:



Dentro de app.routes.ts metemos los componentes y las rutas que tenemos después de importarlos.

```
1 import { Routes } from '@angular/router';
2 import { VideojuegosComponent } from '../componentes/videojuegos/videojuegos.component';
3 import { DetalleComponent } from '../componentes/detalle/detalle/detalle.component';
4 import { LoginComponent } from '../componentes/login/login/login.component';
5
6 export const routes: Routes = [
7     {path: "videojuegos", component: VideojuegosComponent},
8     {path: "videojuego/:id", component: DetalleComponent},
9     {path: 'login', component: LoginComponent},
10    {path: '', redirectTo: 'login', pathMatch: 'full'}
11 ];
```

En el app.component.html metemos el elemento router-outlet

```
lista-videojuegos > src > app > <> app.component.html > router-outlet
Go to component
1
2 <router-outlet></router-outlet>
3
```


**RESULTADO:**

Página de login con usuario y contraseña. Se implementa botón (  /  ) para cambiar entre vista de contraseña y texto normal para ver la contraseña que se introduce:

Introduce el usuario

usuario

Introduce la contraseña

contrasena 

Iniciar sesión

Se muestra el listado de videojuegos con sus características principales:

**Usuario:** usuario

**Contraseña:** contrasena

1	<a href="#">The Legend Of Zelda: Breath Of The Wild</a>	Valoración: 9.5
2	<a href="#">Devil May Cry</a>	Valoración: 9.4
3	<a href="#">Three Fourths Home</a>	Valoración: 7.4
4	<a href="#">The Talos Principle</a>	Valoración: 8.5

Al hacer clic en uno de los títulos se muestra el detalle del videojuego seleccionado, el título, la compañía, la valoración, una imagen del juego y una pequeña descripción del juego. Finalmente, un botón Volver que retorna a la página anterior.

## The Legend Of Zelda: Breath Of The Wild

**Compañía:** Nintendo

**Valoración:** 9.5



**Descripción:** The Legend of Zelda es una serie de videojuegos de acción-aventura creada por los diseñadores japoneses Shigeru Miyamoto y Takashi Tezuka, desarrollada por Nintendo, empresa que también se encarga de su distribución internacional. Su trama por lo general describe las heroicas aventuras del joven guerrero Link, que debe enfrentarse a peligros y resolver acertijos para ayudar a la Princesa Zelda a derrotar a Ganondorf y salvar su hogar, el reino de Hyrule.

Volver