

PEC 4

Frameworks: Introducció a Angular

UOC

Universitat Oberta
de Catalunya

Información relevante:

- Fecha límite de entrega: 28 de noviembre.
- Peso en la nota de FC: 15%.



Contenido

| | |
|--|----|
| Información docente | 3 |
| Presentación | 3 |
| Objetivos | 3 |
| Enunciado | 4 |
| Ejercicio 1 – Instalación y configuración de Angular CLI (1.5 puntos) | 4 |
| Ejercicio 2 – Hola Mundo con Angular (1.25 punto) | 6 |
| Ejercicio 3 – Primer componente en Angular (1.25 puntos) | 8 |
| Ejercicio 4 – Directivas de atributo y directivas estructurales (1.5 puntos) | 10 |
| Ejercicio 5 – Directivas en nuestro proyecto vinoteca (1.5 puntos) | 11 |
| Ejercicio 6 – Teoría de componentes. (1.5 puntos) | 13 |
| Ejercicio 7 – Componentes en nuestro proyecto vinoteca. (1.5 puntos) | 14 |
| Formato y fecha de entrega | 16 |

Información docente

Presentación

Esta práctica se centra en desarrollar una primera aplicación utilizando el framework Angular apoyándose de su herramienta CLI llamada Angular CLI.

Objetivos

Los objetivos que se desean lograr con el desarrollo de esta PEC son:

- Crear un proyecto en Angular **usando Angular CLI**
- Estructurar una aplicación web en **componentes**.
- **Comunicación de datos** entre los componentes de una aplicación.
- Utilizar el **patrón de arquitectura MVC** en el desarrollo de una aplicación web.
- Aprender a leer y comprender **la documentación oficial** de Angular.

Enunciado

Esta PEC contiene 7 ejercicios evaluables. Debéis entregar vuestra solución de los 7 ejercicios evaluables (ver el último apartado).



Debido a que las actividades están encadenadas (i.e. para hacer una se debe haber comprendido la anterior), **es altamente recomendable hacer las tareas y ejercicios en el orden en que aparecen en este enunciado.**

Antes de continuar debes:

Haber leído los recursos teóricos disponibles en el apartado “Contenidos y recursos” del aula de esta PEC

Crea una carpeta PEC4 e inicializa git en esa ruta. Al igual que en las anteriores PEC, debes crear un fichero README.md en la raíz de la carpeta que contenga que contendrá al menos esta información

- Login UOC
- Nombre y apellidos del alumno.
- Breve descripción de lo realizado en esta PEC, dificultades, mejoras realizadas, si hay que tener algo en cuenta a la hora de corregir/ejecutar la practica o cualquier aspecto que queráis destacar.

Ejercicio 1 – Instalación y configuración de Angular CLI (1.5 puntos)

Debes realizar los pasos necesarios para instalar Angular CLI en tu equipo. Para ello debes seguir los pasos explicados en la web oficial del proyecto: <https://angular.io/cli>

A continuación, crea una carpeta PEC4_Ej1, dentro crea un fichero Markdown que tenga como nombre PEC4_Ej1_respuestas_teoria.md y explica qué realiza cada una de las siguientes opciones de Angular CLI.

- `ng new`
- `ng generate`
 - `component`
 - `directive`
 - `enum`

- o guard
- o interface
- o pipe
- o service
- ng serve
- ng test
- ng version

Durante el desarrollo del ejercicio ejecuta los comandos `git` necesarios para añadir esta nueva carpeta, ficheros y los cambios que realices en ellos al repositorio local `git` de la PEC.

Ejercicio 2 – Hola Mundo con Angular (1.25 punto)

Crea una carpeta `PEC4_Ej2`, dentro crea un fichero Markdown que tenga como nombre `PEC4_Ej2_respuestas_teoria.md` y responde a cada uno de los siguientes puntos:

1. (0.75 puntos) ¿Qué comando debes utilizar para crear un nuevo proyecto Angular utilizando Angular CLI denominado `vinoteca`? (A las preguntas que te haga Angular CLI puedes contestar utilizando las opciones por defecto). Explica brevemente la estructura y ficheros que ha generado Angular CLI:

- Ficheros de configuración en la raíz del proyecto:

- `tsconfig.app.json`
- `angular.json`
- `package.json`
- `.editorconfig`,
- `.gitignore`
- ...

- Directorio `node_modules`

- Directorio `e2e`

- Directorio `src`:

- `index.html`
- `main.ts`
- `styles.css`
- `test.ts`
- `polyfills.ts`
- Directorio `environments`
- Directorio `assets`
- Directorio `app`
 - Ficheros `app.component.*`
 - Fichero `app.module.ts`

2. (0.25 puntos) Explica cada uno de los siguientes decoradores generados por Angular CLI, detallando cada una de las propiedades que se definen en:
 - `app.module.ts` - `@NgModule` (`declarations`, `imports`, `providers`, `bootstrap`).
 - `app.component.ts` - `@Component` (`selector`, `templateUrl`, `styleUrls`).

3. (0.25 puntos) ¿Es posible poder inyectar el template y los estilos en línea de un componente sin necesidad de especificarlos en `templateUrl`, `styleUrls`? ¿Es recomendable hacer esto?

Durante el desarrollo del ejercicio ejecuta los comandos `git` necesarios para añadir esta nueva carpeta, ficheros y los cambios que realices en ellos al repositorio local `git` de la PEC.

Ejercicio 3 – Primer componente en Angular (1.25 puntos)

Crea un nuevo proyecto Angular PEC4_Ej3 y realiza los siguientes apartados :

- Incluye un componente para mostrar un único vino (`wineitem`).
- El componente `wineitem` debería mostrar la información de un vino: nombre, precio, imagen del vino, si está en venta, la cantidad comprada e incluso la lista de maridaje a la que puede acompañar este vino. Puedes inicializar el componente con algunos valores por defecto. Utiliza la imagen que quieras como vino. La manera en la que se muestra la información de un vino es libre (pero te recomendamos la que te mostramos en la práctica).
- Debes construir los modelos (i.e. interfaces) necesarios `Wine` y `Food`. Los cuales estarán compuestos de los siguientes datos
 - `Wine`:
 - `name: string`
 - `imageUrl: string`
 - `price: number`
 - `isOnSale: boolean`
 - `quantityInCart: number`
 - `foodPairing: Food[]`
 - `Food`:
 - `name: string`
 - `kcal: number`
 - `vegan: boolean`
 - `gluten: boolean`

- Resalta el elemento entero en un color diferente (utilizando una condición lógica que permita decidir qué clase de CSS debería tener el componente) si el producto está en venta (es decir, si hay en stock).
- Añade botones para incrementar y decrementar la cantidad de unidades de vino que el usuario desea comprar. Esta cantidad deberá ser visible en la pantalla (y actualizada en tiempo real). Desactiva el botón de decrementar si la cantidad de vinos seleccionados es cero.

TPV Vinoteca!



TPV Vinoteca!



Durante el desarrollo del ejercicio ejecuta los comandos `git` necesarios para añadir esta nueva carpeta, ficheros y los cambios que realices en ellos al repositorio local `git` de la PEC.

La carpeta `node_modules`, no debe aparecer en el repositorio `git` local, ni tampoco entregarla en el zip final.

Ejercicio 4 – Directivas de atributo y directivas estructurales (1.5 puntos)

Crea una carpeta `PEC4_Ej4`, dentro crea un fichero Markdown que tenga como nombre `PEC4_Ej4_respuestas_teoría.md` y responde a cada uno de los siguientes puntos:

1. Explica qué son y cuándo se deberían utilizar las siguientes variables locales de la directiva estructural `NgFor`:
 - `index`
 - `even`
 - `odd`
 - `first`
 - `last`
2. ¿Para qué sirve la opción `trackBy` de la directiva estructural `NgFor`? Pon ejemplos de uso.
3. ¿Es posible ejecutar dos directivas estructurales simultáneamente? Explica la razón tanto si es si posible como si no lo es.

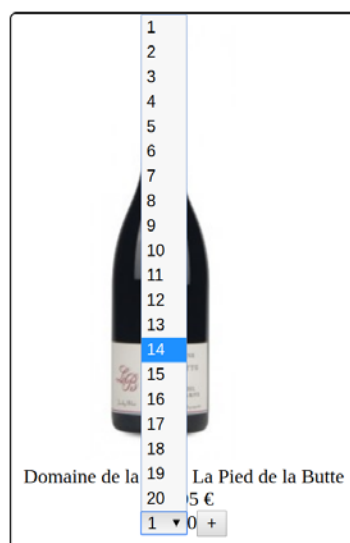
Durante el desarrollo del ejercicio ejecuta los comandos `git` necesarios para añadir esta nueva carpeta, ficheros y los cambios que realices en ellos al repositorio local `git` de la PEC.

Ejercicio 5 – Directivas en nuestro proyecto vinoteca (1.5 puntos)

Debes crear un nuevo proyecto Angular `PEC4_Ej5` (basado en el `PEC4_Ej3`) pero con las siguientes modificaciones:

- Haz uso de la directiva `ngClass` para cambiar el CSS dependiendo de si un vino se encuentra disponible para la venta o no.
- En lugar de desactivar el botón de decremento cuando el número de unidades que se desean comprar de un vino es cero, utiliza la directiva de estructura `ngIf` para mostrar/ocultar dicho botón.
- Agrega una lista desplegable que permita seleccionar entre 1 y 20 vinos (`select` y `option` de HTML) que se debe generar de manera automática en el template haciendo uso de la directiva estructural `NgFor`. No debes preocuparte por agregar los vinos ahora mismo al carrito puesto que eso se hará más adelante.

TPV Vinoteca!



Durante el desarrollo del ejercicio ejecuta los comandos `git` necesarios para añadir esta nueva carpeta, ficheros y los cambios que realices en ellos al repositorio local `git` de la PEC.

La carpeta `node_modules`, no debe aparecer en el repositorio `git` local, ni tampoco entregarla en el zip final.

Ejercicio 6 – Teoría de componentes. (1.5 puntos)

Crea una carpeta `PEC4_Ej6`, dentro crea un fichero Markdown que tenga como nombre `PEC4_Ej6_respuestas_teoría.md` y responde a las siguientes preguntas:

1. ¿Cuáles son los `style encapsulation` de los componentes? Pon un ejemplo de uso de cada uno de ellos.
2. ¿Qué es el `shadow DOM`?
3. ¿Qué es el `changeDetection`?
4. ¿Qué diferencias existen entre las estrategias `Default` y `OnPush`? ¿Cuándo debes usar una y otra? Ventajas e inconvenientes.
5. Explica con detalle el ciclo de vida de los componentes. Haz hincapié en cuándo se disparan los hooks `OnChanges`, `OnInit`, `AfterViewInit` y `OnDestroy`, puesto que son los más utilizados

Durante el desarrollo del ejercicio ejecuta los comandos `git` necesarios para añadir esta nueva carpeta, ficheros y los cambios que realices en ellos al repositorio local `git` de la PEC.

Ejercicio 7 – Componentes en nuestro proyecto vinoteca. (1.5 puntos)

Debes crear un nuevo proyecto Angular `PEC4_Ej7` (partiendo del `PEC4_Ej5`) pero con las siguientes modificaciones:

- Crear un componente denominado `winelist`. Inicializar un array de vinos. La aplicación debe comenzar mostrando una lista de vinos en lugar de un único vino. El template de `winelist` debe usar la directiva `ngFor` para crear un `wineitem` para cada uno de los vinos.
- Usar templates y estilos en línea en el componente `wineitem`. Generarlo utilizando Angular CLI.
- Cambiar el componente `wineitem` para que reciba un `Wine` como input. Además, debe emitir hacia el componente padre un objeto (i.e. interfaz) denominado `WineQuantityChange` que estará compuesto por el vino y la cantidad de unidades de ese vino que se han seleccionado.
- Modificar la lógica de incrementar y decrementar del componente `wineitem` y añadir dicha lógica a `winelist` eliminándola de `wineitem`. Utiliza `id` para encontrar el vino y cambiar su cantidad. Es decir, tienes que crear un nuevo campo en el modelo `Wine` denominado `id`.
- Se debe eliminar el `select` generado con `*ngFor` en el Ejercicio 5.
- Modificar el componente `wineitem` para que sea óptimo. Para ello modifica su estrategia de `ChangeDetection`.

TPV Vinoteca!



Durante el desarrollo del ejercicio ejecuta los comandos `git` necesarios para añadir esta nueva carpeta, ficheros y los cambios que realices en ellos al repositorio local `git` de la PEC.

La carpeta `node modules`, no debe aparecer en el repositorio `git` local, ni tampoco entregarla en el zip final.

Formato y fecha de entrega

Tienes que entregar un fichero *.zip, cuyo nombre tiene que seguir este patrón: loginUOC_PEC4.zip. Por ejemplo: dgarciaso_PEC4.zip. Este fichero comprimido tiene que incluir los siguientes elementos:

Una carpeta con nombre PEC4, y en su interior:

- Un fichero README.md en la raíz de la carpeta con la información indicada.
- La **carpeta oculta** .git con el contenido del repositorio local git
- Una carpeta PEC4_Ej1 con un fichero de texto PEC4_Ej1_respuestas_teoria.md para las respuestas del ejercicio 1.
- Una carpeta PEC4_Ej2 con un fichero de texto PEC4_Ej2_respuestas_teoria.md para las respuestas del ejercicio 2.
- Una carpeta PEC4_Ej3 con los ficheros resultado de haber realizado las tareas del ejercicio 3 (No olvides eliminar la carpeta node_modules)
- Una carpeta PEC4_Ej3 con un fichero de texto PEC4_Ej4_respuestas_teoria.md para las respuestas del ejercicio 4.
- Una carpeta PEC4_Ej5 con los ficheros resultado de haber realizado las tareas del ejercicio 5 (No olvides eliminar la carpeta node_modules)
- Una carpeta PEC4_Ej6 con un fichero de texto PEC4_Ej6_respuestas_teoria.md para las respuestas del ejercicio 6.
- Una carpeta PEC4_Ej7 con los ficheros resultado de haber realizado las tareas del Ejercicio 7 (No olvides eliminar la carpeta node_modules)

Penalizaciones

- Entrega en otro formato que no sea el especificado (ej. en zip): **-0.75 puntos**
- Comprimir archivos dentro del zip: **-1 puntos.**
- Para cada ejercicio/apartado donde no se respete la nomenclatura exacta de las carpetas o ficheros indicados (símbolos, minúsculas, mayúsculas, etc.): **-0.75 puntos.**
- La no entrega del repositorio local git **-3 puntos**
- Por cada carpeta node_modules entregada **-0.75 puntos**

El último día para entregar esta PEC es el **28 de noviembre de 2021** hasta las **23:59**.
Cualquier PEC entregada más tarde será penalizada.