

2º DAW – Angular – Recuperación – 22 de Mayo de 2023

Para comenzar:

1. Descargad el archivo DAW2-Angular-Recuperacion.zip de Google Classroom.
2. Extraed el contenido del zip y en la terminal de linux entrad en la carpeta.
3. Ejecudad: "npm install".
4. Añadid la carpeta a VS Code y lanzad la aplicación.
5. No debe de haber nada más en el workspace de VS Code.

La aplicación se compone de los siguientes módulos (además del módulo principal app.module.ts):

- CompartidoModule – /compartido/**compartido.module.ts**:
 - Contiene el componente **BarraLateralComponent** que nos muestra el menú de la barra lateral.
- AgenciaviajesModule – /agenciaviajes/**agenciaviajes.module.ts**. Contiene los siguientes componentes:
 - **BuscarViajeComponent**: Muestra la lista de viajes cuya duración en días es mayor o igual al número que se introduce en el input.
 - **MostrarViajesComponent**: Muestra todos los viajes.
 - **NuevoViajeroComponent**: Contiene un formulario para registrar nuevos viajeros.
 - **ReservarComponent**: Muestra el viaje que se indica en la URL y tiene la opción de reservarlo para los viajeros indicando el ID del viajero. Muestra una lista con los viajeros para saber el ID de cada uno.
 - **TablaComponent**: Se encarga de mostrar la información sobre viajes en formato tabla.

Además también están presentes los siguientes archivos:

- Servicio – /agenciaviajes/services/**datos.service.ts**: Gestiona los datos de la aplicación: los viajes, viajeros, etc...
- Routing – **app-routing.module.ts**: Tiene la configuración de rutas de la aplicación.
- Interfaces - /agenciaviajes/interfaces/**agenciaviajes.interfaces.ts**

Se pide realizar lo siguiente:

Ejercicio 1: (1.5 puntos)

Configurar el routing para la aplicación usando el archivo *app-routing.module.ts*. Hay que configurar una ruta para cada opción del menú lateral y relacionarla con su correspondiente componente.

- buscar => Componente buscarViaje
- viajes => Componente mostrarViajes
- nuevoviajero => Componente nuevoViajero

Cualquier otra ruta debe redirigirse a la URL 'buscar'.

Se accederá a las rutas a través del menú lateral. Los componentes se deben mostrar en la pantalla principal. El header y la barra lateral serán siempre visibles.

Se debe resaltar la opción del menú lateral cuya URL esté activa.

Ejercicio 2: (2.5 puntos)

Mostrar la información de los viajes en Buscar Viaje y en Viajes:

- En *Viajes* se deben mostrar todos los viajes.
- En *Buscar Viaje* hay que configurar el Input para que reciba un número y cuando se presione Enter muestre los viajes con número de días mayor o igual al valor del Input. Si el valor del Input está vacío, no debe ocurrir nada al presionar el Enter.
- En ambos casos hay que usar el componente tabla para mostrarlos. Hay que configurar éste componente para que se muestre la lista de viajes.
- El campo plazas disponibles es el resultado de restar los viajeros que ya han reservado a las plazas totales del viaje.
- Hay interfaces ya declaradas, hay que usarlas en todos los sitios donde proceda.

Ejercicio 3: (3.5 puntos)

Añadir al módulo routing la ruta *reservar/:id* y asociarla con el componente *Reservar*.

Accederemos a esa ruta haciendo click en la tabla de los viajes (tanto en Buscar Viaje como en Viajes), en la columna Reservar de cada viaje.

El componente ya está creado y configurado para mostrar el viaje y poder reservarlo para algún viajero.

Hay también métodos creados en el servicio que se pueden usar.

Hay que conseguir lo siguiente:

- Que se muestren los detalles del viaje en el que se ha hecho click con el ratón, redirigiendo a la URL correcta. Ejemplo: `/reservar/3`
- Añadir una comprobación para asegurarse de que el ID pertenece a un viajero existente. Si no es así, se debe mostrar un error que diga "El ID no pertenece a ningún viajero". El resto de comprobaciones y el método reservar ya están configurados.
- Actualmente se muestran la lista de todos los viajeros en la parte de abajo. Haz los cambios necesarios para mostrar la lista de los viajeros que han reservado ese viaje.

Ejercicio 4: (2.5 puntos)

Usando los Formularios Reactivos de Angular configura el componente de *nuevo-viajero* para obtener y guardar los datos de un nuevo viajero.

Hay que configurar validaciones

- Los Input de texto no pueden estar vacíos.
- Los Input de números no pueden estar vacíos y deben ser mayor o igual que 0.

Los campos mostrarán un error si no cumplen las validaciones.

Al pulsar el botón Guardar los valores de los campos se deben recopilar y añadirlos a la propiedad viajeros. Y hay que limpiar el formulario una vez que se ha guardado con éxito.

Se debe generar un nuevo ID para el viajero, este nuevo ID se obtendrá sumándole 1 al ID más alto de los viajeros.

Si hay errores en el formulario al pulsar Guardar, deben mostrarse para cada campo.