

ENUNCIADO PEC 1

Programación reactiva: Formularios reactivos

Desarrollo front-end avanzado

Máster Universitario en Desarrollo de sitios y aplicaciones web

UOC

Contenido

- Introducción
- Objetivos
- Formato y fecha de entrega
- Enunciado
- Puntuación



Introducción

Esta primera práctica nos servirá para construir una aplicación web completa a la cual le iremos ampliando funcionalidades y herramientas progresivamente durante el curso. En líneas generales, en esta primera práctica deberemos implementar la navegación de nuestra aplicación, la comunicación con el backend (simulada) y la implementación de varios formularios reactivos.

Objetivos

Los objetivos que se deben conseguir con esta práctica son:

- Implementar la base de la aplicación global que desarrollaremos, haciendo uso de todo lo aprendido a lo largo del máster.
- Recordar conceptos fundamentales de Angular como componentes, routing, servicios, módulos, comunicación entre componentes, ...
- Implementar la navegación de toda la aplicación.
- Implementar la comunicación con el backend (**simulado**).
- Crear formularios reactivos complejos de manera reutilizable.
- Implementar una parte pública y una parte privada en la aplicación.
- Implementar alguna funcionalidad transversal a toda la aplicación.
- Introducción a la gestión del localStorage (**opcional**).

Formato y fecha de entrega

Se entregará todo el proyecto comprimido en formato **.zip** sin incluir el directorio **"node_modules"**. La fecha de entrega se mostrará en el aula de la asignatura.

A nivel de documentación, se podría entregar un pequeño documento donde se indiquen las decisiones de diseño más relevantes si es necesario, y si alguna parte de la práctica no se ha implementado o no funciona también se podría indicar. Finalmente, estaría bien añadir un pequeño *checklist* a modo de juego de pruebas para indicar que se ha probado cada funcionalidad y que ésta funciona.

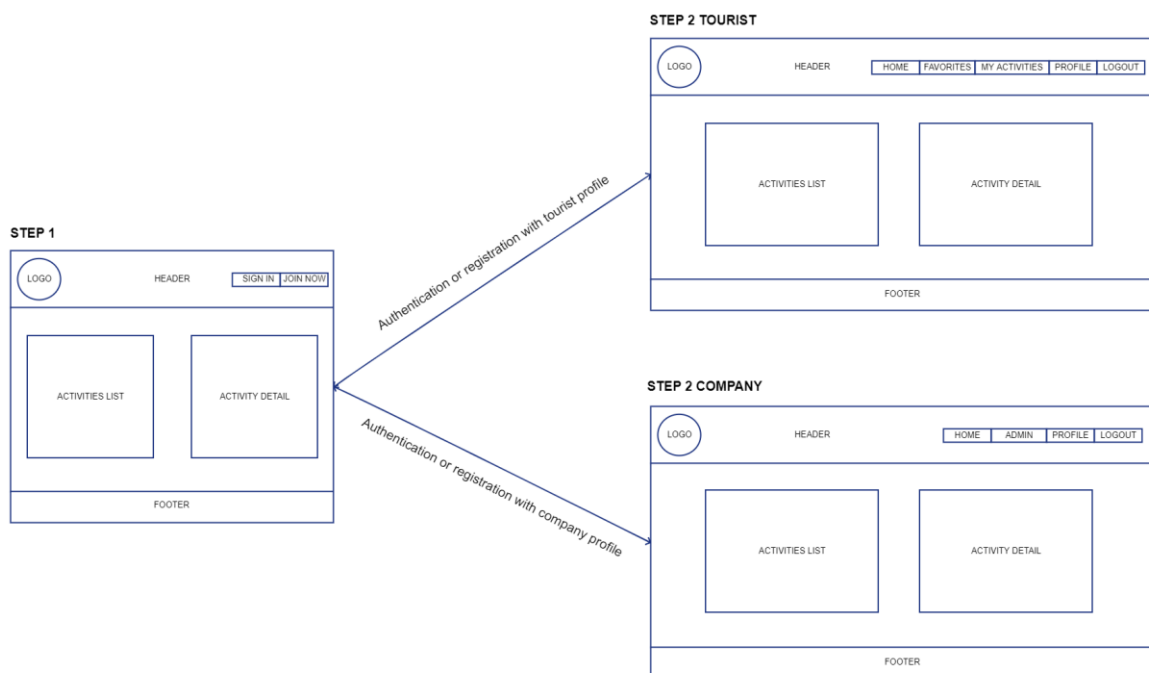
La estructura de directorios que se propone dentro del proyecto para facilitar la corrección es la siguiente:

- **Shared:** aquí tendremos todos los servicios, modelos, componentes comunes, ... por ejemplo, dentro de **shared** podríamos tener los directorios **directives**, **inmemory-db**, **models** y **services**.

- **Views:** aquí tendremos las vistas que interactuarán con el usuario. Por ejemplo, dentro de **views** podríamos tener los componentes **activity-list**, **activity-detail**, **profile**, ...

Enunciado

A continuación, mostramos la visión inicial de la aplicación e iremos explicándola poco a poco para tener claros todos los pasos que deberemos ir desarrollando:



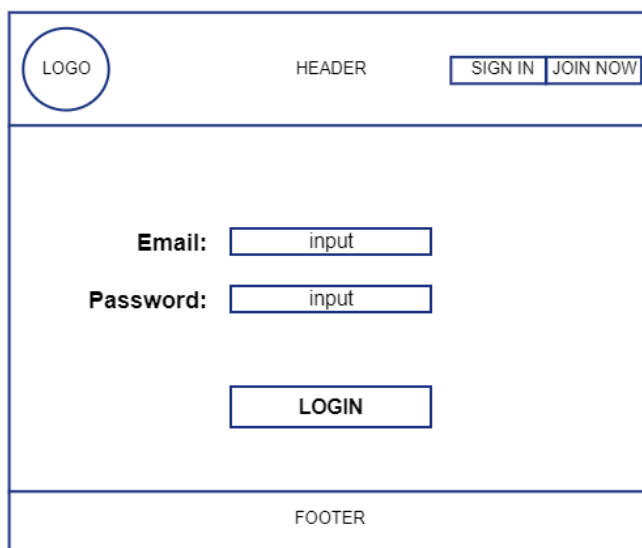
La pantalla **step 1** sería nuestra **home**, la pantalla inicial. Es la vista pública que podrían ver todos los usuarios. En esta pantalla podemos observar que tenemos tres secciones diferenciadas, secciones que se deberán mantener en toda la plataforma, ya veremos posteriormente que irán cambiando dinámicamente su contenido:

- **Header:** en la cabecera tendremos en la parte izquierda el logotipo (puede ser por ejemplo el logotipo de la universidad), en el centro un nombre cualquiera que sería el título de la plataforma y en la parte derecha las opciones para autenticarse o registrarse. Estas opciones cambiarán en función de si estamos autenticados o no y en función del perfil autenticado.
- **Sección principal:** la sección central de la pantalla será donde tendremos el flujo de navegación y es donde veremos las diferentes vistas. Inicialmente, al no estar autenticados a la plataforma, veremos la parte pública, que será una vista tipo lista-detalle de las actividades turísticas dadas de alta al sistema. A la parte izquierda tendremos el listado de actividades, y al hacer clic en cada una de ellas, veremos su detalle a la parte derecha con la información ampliada.
- **Footer:** esta será una sección que simplemente contendrá algún texto y la anualidad en la que nos encontramos.

Autenticación a la plataforma:

Suponemos que visitamos la aplicación web sin estar autenticados, por lo tanto, veríamos la parte pública, y a la cabecera de la página principal tendríamos las acciones para autenticarnos (**sign in**) y para registrarnos (**join now**).

Suponemos que queremos **autenticarnos** a la plataforma para realizar las gestiones pertinentes. Haríamos clic en la opción de autenticarnos (**sign in**) y en la sección central de la vista (entre la cabecera y el pie) nos aparecerá el **formulario reactivo de autenticación**:



Las especificaciones de este formulario reactivo son las siguientes:

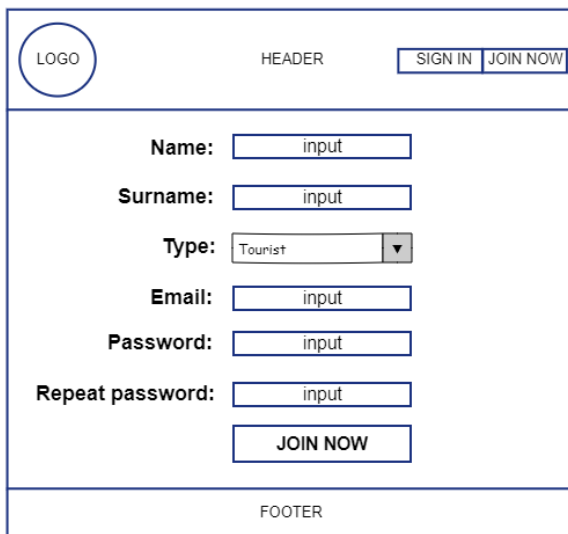
- **email:** campo obligatorio, se deberá validar que tiene un formato de correo electrónico válido.
- **password:** campo obligatorio.
- Botón **login:** se habilitará si se cumplen las validaciones anteriores.
- Acción **login:** al hacer clic al botón de **login** se hará una petición al backend, del cual podremos tener dos respuestas:
 - Si el usuario no existe (porque no coinciden el correo y/o la contraseña) se indicará con un mensaje en rojo arriba del botón de **login** y no podremos acceder.
 - Si el usuario sí que existe y por lo tanto, nos podemos autenticar, en función del perfil podremos acceder a unas funcionalidades o otras, concretamente:
 - Perfil **tourist:** tendrá habilitadas las opciones de la cabecera **home**, **favorites**, **my activities**, **profile** y **logout**.

- Perfil **company**: tendrá habilitadas las opciones de la cabecera **home**, **admin**, **profile** y **logout**.
- En la sección central, nos redirigirá a la vista de la **home** donde nos aparecerá la vista lista-detalle con el listado de actividades a la izquierda, y el detalle de la actividad seleccionada a la derecha. La particularidad es que a la vista detalle, al estar autenticados al sistema, si somos **perfil tourist**, tendremos las opciones de apuntarnos a la actividad o de guardarla como favorita. Si somos **perfil company** no tendremos disponibles estas acciones.

Registro a la plataforma:

De manera semejante al caso anterior, suponemos que visitamos la aplicación web sin estar autenticados, por lo tanto, veríamos la parte pública, y en la cabecera de la página principal tendríamos las acciones para autenticarnos (**sign in**) y para registrarnos (**join now**).

Suponemos que queremos **registrarnos** en la plataforma para empezar a interactuar con ésta, para gestionar actividades turísticas. Haríamos clic en la opción de registrarnos (**join now**) y en la sección central de la vista (entre la cabecera y el pie) nos aparecerá el **formulario reactivo de registro**:



LOGO

HEADER

SIGN IN JOIN NOW

Name:

Surname:

Type: Tourist ▼

Email:

Password:

Repeat password:

JOIN NOW

FOOTER

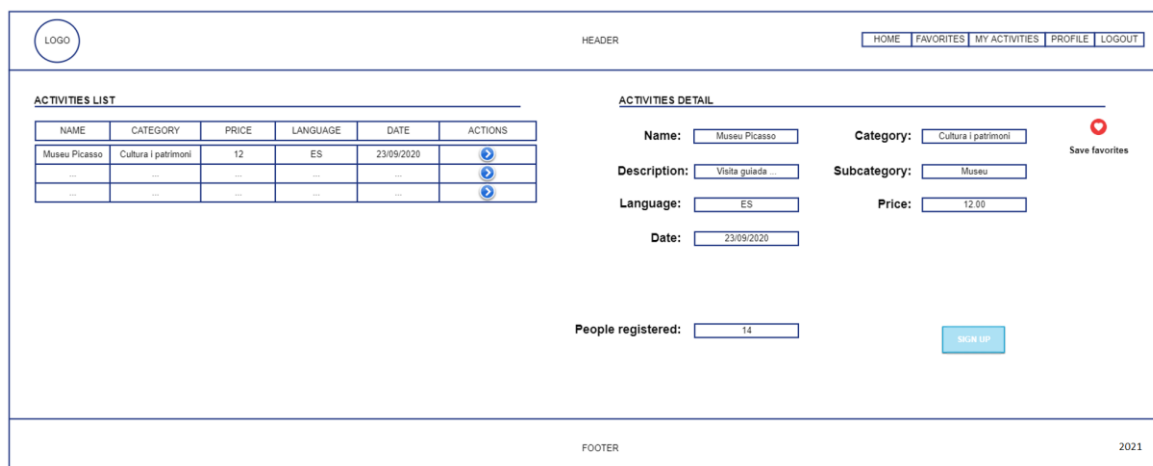
Las especificaciones de este formulario reactivo son las siguientes:

- **name**: campo obligatorio, deberá tener entre 3 y 55 caracteres, sin espacios en blanco al principio ni al final ni caracteres especiales.
- **surname**: campo opcional, si está informado deberá tener entre 3 y 55 caracteres, sin espacios en blanco al principio ni al final ni caracteres especiales.
- **type**: desplegable con las opciones **tourist** y **company**. Se deberá seleccionar una de las dos opciones.

- **email:** campo obligatorio, se deberá validar que tiene un formato de correo electrónico válido.
- **password:** campo obligatorio, deberá tener como mínimo 8 caracteres.
- **repeat password:** campo obligatorio, deberá tener como mínimo 8 caracteres.
 - Se deberá validar que el contenido de los dos campos anteriores **password** y **repeat_password** son exactamente iguales.
- Botón **join now:** se habilitará si se cumplen las validaciones anteriores.
- Acción **join now:** al hacer clic al botón de **join now** se hará una petición al backend, del cual podremos tener dos respuestas:
 - Validaremos que el correo electrónico del usuario que se quiere registrar no exista en el sistema, en caso de que ya exista se indicará con un mensaje en rojo arriba del botón de **join now** y no se podrá acceder.
 - Si el correo electrónico del usuario no existe en el sistema y por lo tanto, nos podemos registrar, de la misma manera que en la función de autenticación anterior, tendremos que en función del perfil podremos acceder a unas funcionalidades u otras, concretamente:
 - Perfil **tourist:** tendrá habilitadas las opciones de la cabecera **home**, **favorites**, **my activities**, **profile** y **logout**.
 - Perfil **company:** tendrá habilitadas las opciones de la cabecera **home**, **admin**, **profile** y **logout**.
 - Igual que en el proceso de autenticación → En la sección central, nos redirigirá a la vista de la **home** donde nos aparecerá la vista lista-detalle con el listado de actividades a la izquierda, y el detalle de la actividad seleccionada a la derecha. La particularidad es que a la vista detalle, al estar autenticados al sistema, si somos **perfil tourist**, tendremos las opciones de apuntarnos a la actividad o de guardarla como favorita. Si somos **perfil company** no tendremos disponibles estas acciones.

Punto de menú HOME:

La pantalla principal **HOME** en la sección central mostrará el listado de actividades dadas de alta al sistema. Se mostrará en formato lista en la parte izquierda y cada fila tendrá un botón que al hacerle clic nos mostrará el detalle de la actividad en la parte derecha de la sección. A continuación, mostramos un prototipo que sólo pretende dar una imagen visual de lo que se quiere implementar:



The prototype shows a web interface for activity management. It includes a header with a logo, navigation links (HOME, FAVORITES, MY ACTIVITIES, PROFILE, LOGOUT), and a footer with the year 2021. The main content area is divided into two sections: 'ACTIVITIES LIST' and 'ACTIVITIES DETAIL'.

ACTIVITIES LIST: A table with columns: NAME, CATEGORY, PRICE, LANGUAGE, DATE, and ACTIONS. The first row shows 'Museu Picasso' in the 'Cultura i patrimoni' category, priced at 12, in Spanish (ES), dated 23/09/2020, with a blue circular action button.

ACTIVITIES DETAIL: A form showing details for 'Museu Picasso'. Fields include: Name (Museu Picasso), Category (Cultura i patrimoni), Description (Visita guiada...), Subcategory (Museu), Language (ES), Price (12.00), and Date (23/09/2020). There is a 'Save favorites' button with a heart icon and a 'SIGN UP' button. A 'People registered' field shows the number 14.

Como comentábamos, al hacer clic en cada fila del listado de actividades podremos ver toda la información ampliada en el componente de la derecha.

Aquí tenemos que destacar unas casuísticas muy importantes para tener en cuenta a la hora de implementar esta pantalla. Lo explicamos a continuación:

- Si no estamos autenticados en el sistema, es decir, mostramos la vista pública, cuando accedemos a esta pantalla **home** podremos ver el listado de actividades y si accedemos a su detalle podremos ver toda la información, pero no tendremos los botones de acción **save favorites** ni **sign up**.
- Si estamos autenticados en la plataforma:
 - Si somos perfil **tourist** al detalle de la actividad podremos realizar las acciones **save favorites** y **sign up** (posteriormente explicaremos cómo funcionan).
 - Si somos perfil **company** no tendremos disponibles estos dos botones de acción, por lo tanto, es como si visualizásemos la misma página pública.
- Toda la información de la actividad la da de alta su propietario, el usuario determinado de perfil **company** (esto lo veremos cuando expliquemos la pantalla **admin** de este perfil).
- El único campo que será calculado en tiempo de ejecución será el campo **People registered**. Este campo mostrará el número de turistas inscritos en una actividad determinada.

- Acción **sign up**: como hemos dicho, si el usuario esta autenticado y es perfil **tourist** tendrá este botón habilitado y si le hace clic quedará registrado en la actividad. Esto lo que implicará será los siguientes dos aspectos a tener en cuenta:
 - Cuando el usuario vaya al punto de menú **my activities**, podrá visualizar esta actividad a la cual acaba de inscribirse.
 - El contador **people registered** deberá incrementarse en una unidad.
 - Solo nos tiene que dejar apuntar una vez.
- Acción **save favorites**: como hemos dicho, si el usuario esta autenticado y es perfil **tourist** tendrá este botón habilitado y si le hace clic guardará la actividad como favorita. Esto implicará que cuando el usuario vaya al punto **favorites** podrá visualizar esta actividad. Esta es una parte opcional que comentaremos posteriormente.

Punto de menú LOGOUT:

Si un usuario autenticado en el sistema pulsa la opción **logout**, se saldrá de la aplicación y se redireccionará a la **home** (pantalla principal pública). En la cabecera deberán quedar visibles las acciones de autenticarse y registrarse.

FUNCIONALIDADES DEL PERFIL TOURIST

Punto de menú PROFILE:

Tanto un perfil **tourist** como un perfil **company** tendrán una opción de menú para poder personalizar sus datos de perfil (punto de menú **profile**). Esta opción sólo será accesible cuando el usuario esté autenticado en el sistema.

Cuando el usuario haga clic en esta opción, en la sección central de la vista nos aparecerá todo lo necesario para gestionar estos datos, a continuación, mostramos un prototipo para dar una idea de lo que buscamos:



HEADER

HOME
FAVORITES
MY ACTIVITIES
PROFILE
LOGOUT

PROFILE

Name:

Surname:

Birth date:

Phone:

Nationality:

NIF:

About me:

UPDATE PROFILE
INFORMATION

EDUCATION

TYPE	LEVEL	NAME	UNIVERSITY	FINISH DATE	ACTIONS	
Universidad	Postgrado	Psicología	UOC	23/09/2020	UPDATE	DELETE
...	UPDATE	DELETE
...	UPDATE	DELETE

NEW EDUCATION

FOOTER

Podemos ver dos secciones diferenciadas.

La primera sección **profile** es un bloque de información que se podrá actualizar mediante el botón **Update profile information**. Si le hacemos clic, nos redireccionará al propio formulario (componente), podremos editar los datos y guardar. Al guardar nos devolverá a la página de perfil donde estábamos con los datos actualizados. (pensemos que queremos practicar con los formularios reactivos, en principio es más sencillo haciendo esta navegación que editar los datos utilizando ventanas modales que serían más elegantes, pero para esta práctica, sin ventanas modales sería suficiente).

Por otra parte, en la segunda sección, **education**, tendremos que gestionar un **CRUD**. Aquí la idea será que podremos añadir tantas entidades como necesitemos, y para cada una, podremos editarla o eliminarla. De manera semejante al caso anterior, cuando creamos o editemos una entidad (sea una formación o sea un idioma) nos redirigirá al mismo formulario (componente) para hacer las acciones pertinentes. La opción eliminar nos pedirá confirmación, y si le decimos que sí, eliminará la entidad y no aparecerá al listado.

Las especificaciones de estos formularios reactivos son las siguientes:

- Sección **profile**:
 - **name**: campo obligatorio, deberá tener entre 3 y 55 caracteres, sin espacios en blanco al principio ni al final ni caracteres especiales.
 - **surname**: campo opcional, deberá tener entre 3 y 55 caracteres, sin espacios en blanco al principio ni al final ni caracteres especiales.
 - **Birth date**: campo opcional, si está informado deberá tener el formato DD/MM/YYYY. (Más adelante introduciremos el *DatePicker* o aplicaremos alguna validación utilizando librerías de terceros como *MomentJS*).
 - **phone**: campo numérico opcional.
 - **nationality**: desplegable con las opciones **ES**, **FR**, **IT** y **PT**.
 - **NIF**: Suponemos que el usuario escoge la opción ES del desplegable anterior y en este campo introduce un NIF español. Validaremos que este NIF sea correcto. Debemos validar que tiene un formato correcto y que la letra del NIF se corresponde con su numeración (https://es.wikipedia.org/wiki/N%C3%BAmero_de_identificaci%C3%B3n_fiscal) Solo deberemos validar este caso.
 - **about me**: campo de texto opcional.
- Sección **education**:
 - CRUD donde deberemos mantener los siguientes datos:
 - **type**: desplegable con las opciones **título universitario** y **ciclo formativo**, se deberá escoger alguna.
 - **level**: desplegable condicionado al anterior:
 - si el tipo escogido en el campo **type** es título universitario, al desplegable **level** deberán aparecer las opciones **grado**, **diplomado**, **licenciado**, **ingeniero**, **máster** y **doctorado**.
 - Si el tipo escogido en el campo **type** es ciclo formativo, al desplegable **level** deberán aparecer las opciones **grado superior** y **grado medio**.

- **name:** campo obligatorio entre 3 y 55 caracteres.
- **university:** campo obligatorio entre 3 y 55 caracteres.
- **finish date:** campo opcional con formato DD/MM/YYYY.

Punto de menú MY ACTIVITIES:

Se mostrará la misma pantalla que el listado de actividades lista-detalle de la **home** pero sólo se visualizarán las actividades a las que se haya inscrito el usuario de perfil **tourist** autenticado. **En el detalle de las actividades deberemos tener la opción de cancelar la inscripción.**

Si se realiza la cancelación, esta actividad ya no debería aparecernos en esta pantalla de **my activities** y al contador de **people registered** debería de restarse una unidad.

FUNCIONALIDADES DEL PERFIL COMPANY

Punto de menú PROFILE:

Esta pantalla será exactamente igual que la del perfil **tourist** explicada anteriormente, con la diferencia que el perfil **company** en la primera sección **profile** también tendrá los siguientes campos:

- **company name:** campo obligatorio entre 3 y 255 caracteres, sin espacios en blanco inicialmente.
- **company description:** campo de texto opcional.
- **CIF:** campo libre obligatorio.

Punto de menú ADMIN:

Cuando un usuario de perfil **company** esté autenticado en el sistema tendrá disponible esta opción **admin** en la cabecera. Al hacer clic nos aparecerá en la sección central una vista la cual nos permitirá hacer el mantenimiento (CRUD) de todos los datos de la entidad **activity** que gestione el usuario concreto.

De la misma manera que por ejemplo en la sección **education** de la pantalla **profile**, tendremos una tabla con la lista de actividades dadas de alta por parte del usuario. En cada fila tendremos los diferentes campos de la actividad y la columna de la derecha tendrá las acciones editar y eliminar. Debajo de la tabla tendremos el botón de añadir actividad.

Para hacer el mantenimiento de una actividad deberemos trabajar con formularios reactivos teniendo en cuenta los siguientes campos:

- **name:** campo obligatorio entre 3 y 55 caracteres.
- **category:** Este campo nos indicará a que categoría pertenece la actividad turística determinada. El campo será un desplegable con las opciones **cultura y patrimonio**, **enoturismo** y **playas**. Deberemos tener alguna opción seleccionada para poder guardar.
- **subcategory:** Este campo nos permitirá definir un poco más una actividad turística ya que cada categoría tiene unas cuantas subcategorías. Deberemos tener alguna opción seleccionada para poder guardar. De esta manera, este campo será un desplegable condicionado a la selección del desplegable **category**, concretamente, si al camp **category** hay seleccionada la categoría:
 - **cultura y patrimonio:** en el desplegable **subcategory** deberán aparecer las opciones **concierto**, **espectáculo**, **excursión**, **festivales**, **visita guiada**, **museo** y **monumento**.
 - **enoturismo:** en el desplegable **subcategory** deberán aparecer las opciones **bodega**, **cata de productos**, **excursión**, **museo del vino** y **visita guiada**.
 - **playa:** en el desplegable **subcategory** deberán aparecer las opciones **actividad náutica**, **cala**, **concierto**, **excusión** y **taller**.
- **description:** campo de texto opcional.
- **language:** desplegable con las opciones **inglés**, **catalán**, **español**, **francés** y **alemán**.
- **date:** campo opcional con formato DD/MM/YYYY.
- **price:** campo obligatorio, decimal, no puede ser negativo.

IMPLEMENTACIÓN OPCIONAL

Punto de menú FAVORITES perfil TOURIST:

Se mostrará la misma pantalla que el listado de actividades lista-detalle de la **home** pero solo se visualizarán las actividades en las que haya marcado como favoritas el usuario de perfil **tourist** autenticado. **En el detalle de las actividades deberemos tener la opción de dejar de tenerla como favorita.**

En caso de realizar la acción de quitar la actividad de favoritas, esta ya no nos debería aparecer en esta pantalla de **favorites**.

Aquí la diferencia respecto a la funcionalidad de **my activities** está en que, en lugar de guardar y recuperar la información mediante el backend, lo haremos mediante el **localStorage** del navegador.

La tarea está en investigar cómo podríamos guardar y recuperar la información mediante el localStorage del navegador.

BACKEND SIMULADO

Debido a que nos llevaría demasiado tiempo implementar un backend para poder utilizarlo con nuestro front-end que estamos desarrollando, lo que haremos será implementar un backend simulado.

Nosotros, des del front-end, haremos las llamadas exactamente igual que si apuntáramos a un backend real, con la url determinada, pero esta url internamente apuntará a una backend que trabaja en memoria y nos permitirá trabajar de manera transparente.

Para desarrollar este backend en memoria utilizaremos el paquete **angular-in-memory-web-api**. Es muy importante que instaléis este paquete y lo configuréis correctamente para que podáis avanzar con la práctica.

Os recomiendo hacer lo siguiente, ir a la página oficial:

<https://angular.io/tutorial>

Y realizar el manual **tour of heroes**, pero en lugar de utilizar la entidad **Hero**, vosotros trabajad con la entidad **User** o **Activity**. Si seguís el manual veréis que muy rápidamente tendréis un ejemplo de vista lista-detalle que podremos aplicar en nuestra práctica y posteriormente en el punto 6 del manual:

<https://angular.io/tutorial/toh-pt6>

Explican como utilizar el paquete **angular-in-memory-web-api** con los ejemplos CRUD completos sobre la entidad **Hero**, para nosotros la entidad **User** o **Activity**.

Seguid este manual y aseguraros tener un CRUD completo y funcionando y a partir de aquí seguid con la práctica.

También hay que comentar que los datos que aparecerán en los diferentes desplegable de la aplicación se pueden obtener directamente de un fichero del tipo **mock-data.ts** propio.

Finalmente, hay que comentar que en esta primera práctica no debéis preocuparos por el aspecto visual de la aplicación, trabajaremos específicamente en una práctica posterior el tema de los estilos con **angular material**.

Puntuación

A continuación, mostramos cuanto puntúan cada uno de los apartados de la práctica para obtener la nota final de la misma. Hay que tener en cuenta que la práctica se puntúa sobre 10, y tenemos un punto extra si implementamos la funcionalidad de mis favoritos. Este punto nos ayudaría a subir nota si tenemos algún otro apartado con menos nota.

- Configurar navegación de las pantallas de la aplicación [**0,5 puntos**]
- Configurar paquete **angular-in-memory-web-api** con todos los objetos necesarios para el correcto funcionamiento de la aplicación [**1 punto**]
- Autenticación a la plataforma [**1 punto**]
- Registro a la plataforma [**0,5 puntos**]
- Funcionalidad *logout* [**0,25 puntos**]
- Pantalla *profile* turista [**1,50 puntos**]
- Pantalla *profile* empresa [**0,5 puntos**]
- Pantalla lista-detalle actividades (pantalla principal) [**1,25 puntos**]
- Pantalla mis actividades [**1 punto**]
- Pantalla ADMIN perfil empresa [**1,25 puntos**]
- Funcionalidades: [**1 punto**]
 - Inscribir/anular inscripción de una actividad
 - Contadores del número de inscritos a una actividad
- Diferentes puntos de menú en función del perfil del usuario autenticado [**0,25 puntos**]
- Funcionalidad mis favoritos con *localStorage* [**1 punto extra**]