## Ejercicio: Sistema de reserva de localidades para eventos

Se quiere implementar un cliente web para gestionar reservas de localidades de eventos con distintas sesiones y disponibilidades. Para ello, el sistema debe contar con una vista principal que contenga un listado de los eventos disponibles en este momento (cartelera) y una vista detalle (ficha evento) que contenga un listado de las sesiones disponibles para ese evento y un carrito de la compra.

Como layout genérico solamente se reserva un espacio en la parte superior de la app posibilitando la inclusión de un header en un futuro.

La cartelera deberá cumplir con las siguientes especificaciones:

* Events grid:
  + Small screens: single column
  + Large/médium screens: 2 columns
* Event card content:
  + Title (link to details)
  + Subtitle (link to details)
  + Image (dummy) (link to details)
  + Place
  + Start date
  + End date
  + Short description (show only on medium/large screens)
  + Action button (link to details)
* Order by end date, ASC

\* Véanse imágenes adjuntas *catalog-mobile.png y catalog-desktop.png* para una idea general de lo que se pide para esta sección.

En la ficha evento se quiere mostrar:

* listado de sesiones con controles para añadir o eliminar localidades al carrito de la compra
* carrito de la compra mostrando el listado de las sesiones seleccionadas por el usuario
* Botón de volver a cartelera de eventos

El listado de sesiones deberá cumplir con las siguientes especificaciones:

* Sessions list:
  + 1 session item per row
* Session item content:
  + session date
  + session availability
  + number of selected locations
  + - / + buttons
* Order by date, ASC
* Prevent selection of more locations than available
* Prevent negative values in number of selected locations
* Content in a block that fits
  + 100% of total container width for small screens
  + 50% of total container width for medium/large screens
* If there’s no content for some event ID, show a text saying “EVENT INFO NOT FOUND” on the place of this block.

El shopping cart deberá cumplir con las siguientes especificaciones:

* Listado de las localidades seleccionadas en los distintos eventos
  + Agrupadas por evento, mostrándose bajo el título del mismo
  + Solo se mostrará el título de los eventos para los que se haya seleccionado alguna localidad
  + Debajo del título del evento se mostrará el listado de sesiones para las que se ha seleccionado alguna localidad, una por fila
  + Cada sesión se identificará por su fecha, que deberá mostrarse junto con el numero de localidades seleccionadas para esa sesión y, al lado, un icono ‘trash’ para eliminar las localidades de dicha sesión del carrito de la compra (de una en una, decrementandolas hasta que no quede ninguna)

\* Véase imagen adjunta *sessions-desktop.png* para una idea general de lo que se pide para esta sección.

El ejercicio debe desarrollarse bajo los siguientes criterios:

* Uso de los datos contenidos en los ficheros JSON como mock para las llamadas a “getEvents” [cartelera] y “getEventInfo” [ficha evento] (cargando el contenido mediante HTTPClient para emular llamadas a un servidor real)
* Uso de la imagen de muestra *simple-image.jpg* para el listado de eventos y el icono *trash.png* para el carrito de la compra. Ambas proporcionadas junto con el ejercicio.
* Enrutado de estados mediante @angular/router
* Se permite usar cualquier librería CSS de layout existente así como el módulo @angular/flex-layout si se prefiere en lugar de los CSS.
* Responsive design
* Shopping-cart como componente reusable a nivel del proyecto
* Diseño de una arquitectura lo más escalable y desacoplada posible tanto a nivel de filesystem como lógico.
* Arquitectura basada en el modelo de programación reactiva con Angular (observables/OnPush communication strategy)