

## **CONDICIONES MVP SPRINT 3 - MODULO MERN**

Se sigue utilizando el mismo repositorio que para el sprint 2, y debe ser pusheado por última vez el **lunes 26/04 a las 08:45 hs AM**. A las 08:50 hs realizaré el último pulleo y el estado evaluable del repositorio será el de ese momento. Se sugiere realizar pusheos diarios para incorporar Git a la rutina de trabajo e ir identificando problemas y errores en el uso de la herramienta.

Durante el día 26/04 se procederá a la corrección de cada trabajo y al explained code correspondiente. Para obtener un MVP se deben cumplimentar todas las consignas aquí especificadas y además saber explicar parte del código de forma concisa, adecuada y coherente. Incumplir alguno de los dos criterios evaluatorios implica un NO-MVP.

Se recuerda que la interface de toda la aplicación de Mytinerary debe estar en idioma inglés (verificar el uso correcto de palabras, expresiones, etc; evitar traducciones literales no verificadas). El producto debe seguir siendo **full responsive** y mantener en todas las resoluciones una experiencia de usuario intuitiva, ágil y accesible.

Tanto en back como en front recordar las premisas **DRY Don't repeat yourself** y **KIS Keep it simple**. apuntar a un código moderno, legible, mantenible y bien estructurado. Atención nombres (semántica) de variables, funciones, componentes, etc. Utilizar mayúsculas donde corresponde.

Los objetivos y condiciones de este sprint son:

A nivel **frontend**: al hacer click en una ciudad de "Cities" debe abrirse un componente que reconozca que se pidió por determinada ciudad (el que actualmente ya está funcionando y muestra under construction). Ese componente debe renderizar los **itinerarios que corresponden a esa ciudad**, y de cada itinerario mostrar:

- Foto y nombre de la persona que lo publicó.
- Price: es un valor de 1 a 5 (numérico) que se representa visualmente con billetitos (1 billetito, 2 billetitos, 3 billetitos, etc). Duration: indica la cantidad de horas que dura el itinerario. Y likes (por defecto 0). Además debe mostrar 3 o 4 "hashtags" temáticos. Debe haber un botón "view more" que, por el momento, "abra" un espacio donde irán las activities y los comments, aunque por ahora deberá decir "under construction". Atención! tener en cuenta que si bien todavía no hay que mostrar activities ni comments, recordar que comments debería ser una propiedad del modelo de itinerario... tenerlo en cuenta para no tener que agregarlo después y que deban volver a cargar info. Activities debe ser un modelo aparte.

Se deben cargar 4 itinerarios (de la lista de ciudades cargar 2 itinerarios en la primera ciudad y 2 en la segunda, el resto de las ciudades por ahora quedan sin itinerarios, DEBE HABER UN MENSAJE **DINAMICO**, SIMPATICO Y

AGRADABLE QUE INFORME QUE "NO HAY ITINERARIOS AUN PARA ESTA CIUDAD").

Obviamente, no se puede hardcodear NADA (todo dinámico desde la BD).

Se debe aplicar REDUX al frontend y cargar en el store toda aquella información que sea pertinente que esté ahí. Todo pedido AJAX debe ser hecho a través de actions/reducers y toda la funcionalidad del filtro también debe ser hecha a través de redux (por más que, al ser de necesidad de un solo componente - Cities-, no sería estrictamente necesario en un proyecto real).

A nivel backend:

### **HACER TRABAJO INDIVIDUAL/GRUPAL DE MODELADO DE BD**

En base a este trabajo hay que hacer:

- Modelo de Itinerario con sus campos.
- Controladores para: consultar (GET) todos los itinerarios, consultar itinerarios de una ciudad en particular, consultar un itinerario en particular (por id), cargar un nuevo itinerario, modificar un itinerario, borrar un itinerario. Los controladores que no se usen via Frontend deben estar igual (y ser chequeados con Insomnia o Postman que funcionen ok).
- Atención a la **relación entre documentos** en Mongo, así es como deben establecerse las "pertenencias" de un itinerario a una ciudad (y más adelante, de un paquete de activities a un itinerario).