

Condiciones MVP – Sprint 2 – Módulo MERN

Se sigue utilizando el mismo repositorio que para el sprint 1, y debe ser pusheado por última vez el **LUNES 29/11 A LAS 08:45 HS AM**. A las 08:50 hs se realizará 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 **LUNES 29/11** 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 correcto uso 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.

El objetivo y las condiciones de este sprint son:

- **A nivel frontend:**
 - Desarrollar el componente Cities. El mismo deberá fetchear de forma dinámica a la API, obtener todas las ciudades (nombre + foto) y a través de un input posibilitar al usuario a que filtre aquellas ciudades que le interesen.
 - Las ciudades deben ser 15.
 - El filtro debe funcionar de la siguiente manera: si el usuario ingresa "bue" deben aparecerle las ciudades que comienzan con "bue" (indistintamente si pone mayúsculas o minúsculas o una combinación). Si borra la e, deben aparecerle ciudades que comiencen con "bu". No debe aparecer "Paris" por ejemplo si el usuario pone "ri", solo ciudades que comiencen con "ri". Incluso si la persona pone espacios en blanco al comienzo y/o

al final de lo que busca debe reconocerlo. Si no hubiera ciudades con las letras buscadas, la app debe informarlo de forma agradable y con buena UX.

- Al hacer click en una ciudad me debe dirigir a un componente reutilizable (obviamente no un componente por ciudad) que me indique en que ciudad ingresé (a través de texto y foto) y tenga la leyenda "Under construction" (que es donde irán los itinerarios y las actividades).
- Debe haber un botón para retornar a Cities (además del link del menu del header).

- **A nivel backend:**

- Se debe crear el servidor, la base de datos con la colección cities, el modelo de City que incluya como minimo nombre de ciudad, foto (link a web o ruta/nombre archivo local) y el pais al que pertenece. Si quieren agregar más propiedades para dar otras funcionalidades, adelante.
- Se deben crear también el router y los controladores necesarios para:
 - cargar una ciudad,
 - obtener todas las ciudades,
 - obtener una ciudad específica por su ID,
 - modificar una ciudad
 - y borrar una ciudad (aunque los controladores de modificar, cargar y borrar no tengan en el frontend como ser utilizados, pero deben estar disponibles para usarse desde, por ejemplo, Insomnia/Postman).

TANTO EN BACK COMO EN FRONT RECORDAR LAS PREMISAS DRY y KIS (Don't repeat yourself y Keep it simple). Apuntar a un código moderno, legible, mantenible y bien estructurado. Atención nombre (semántica) de variables, funciones, componentes, etc.