

CONDICIONES MVP SPRINT 4 - MODULO MERN

Se sigue utilizando el mismo repositorio que para el sprint 3, y debe ser pusheado por última vez el **lunes 03/05 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 03/05 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**: se deben crear los componentes de Sign Up, de Sign In y el componente (o solo action) de Sign Out. Atención a los nombres!

En el sign up el usuario debe poder completar un formulario con, como mínimo, estos campos: nombre, apellido, email, password, URL de la foto y país (en un select). También debe estar la opción de crear cuenta con Google.

Debe haber un call to action para que, si el usuario YA TIENE UNA CUENTA, se vaya a loguear.

En el sign in el usuario debe poder ingresar su mail y su password y loguearse. También debe estar la opción de Google. El log out debe, por supuesto, desloguear al usuario de la app.

Debe haber un call to action para que, si el usuario NO TIENE UNA CUENTA, se la vaya a crear.

Cada cambio de estado (logueado-deslogueado) debe cambiar las opciones disponibles en los menús y el estado se debe perdurar a través de localStorage para no perder el estado de logueado en caso de hacer F5. También, al cambiar de estado, deben cambiar aquellas páginas que el usuario tiene disponibles para ver (aunque en el caso de Mytinerary, un usuario no logueado puede hacer y ver todo, excepto comentar y likear).

Deben hacer una validación mínima via frontend antes de enviar los datos al backend.

Mostrar mensajes de error personalizados, claros, atractivos y amigables.

A nivel **backend**: se deben crear los endpoints y controladores de crear cuenta y loguear usuario. En el caso de crear cuenta la request debe pasar antes por un validador hecho con **Joy** (no exagerar en las condiciones de validación pero tampoco obviarlas ni poner 1 o 2). Atención al mail (debe ser unico y con formato de mail) y la password. Los paises pueden estar hardcodeados en un array o alimentarse de alguna api de paises. Como minimo debe haber 8 paises para elegir y la creación de "options" debe ser **dinámica**. Al loguearse, el backend debe enviar al frontend un token (usando JWT y Passport). **No hay roles**, se puede agregar como un extra pero no lo sugiero.

Atención al tema password en Google... ver cuales pueden ser las opciones disponibles para solventar este tema.