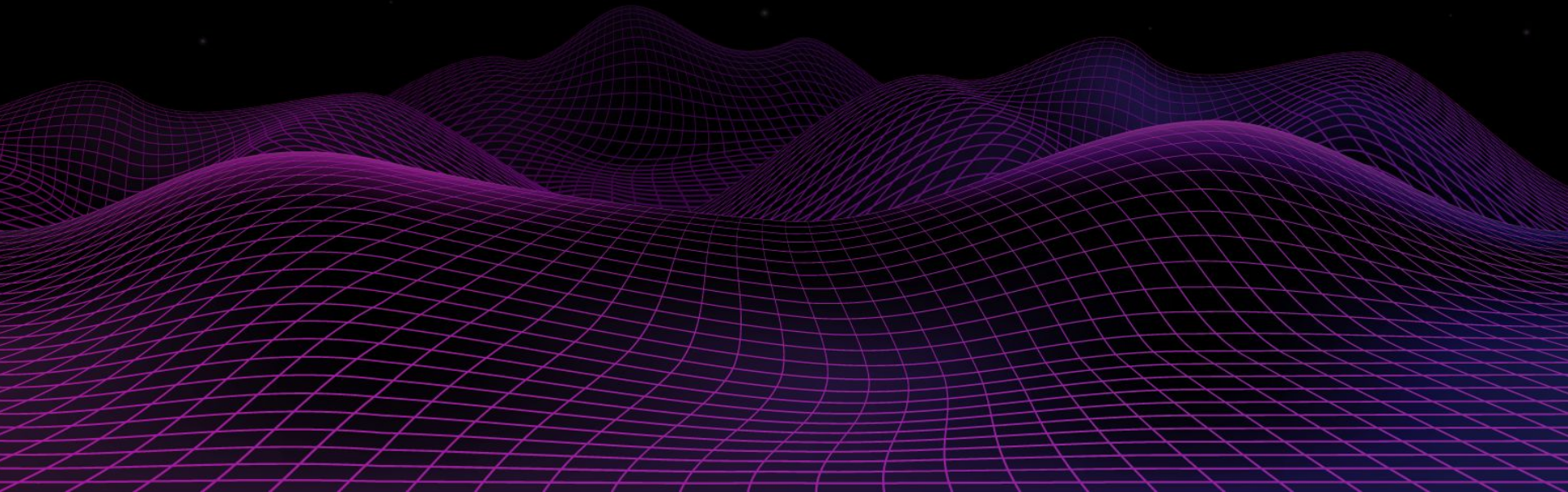


# BIENVENIDOS

**MIND  
HUB.**

# Sprint 4

## Condiciones MVP



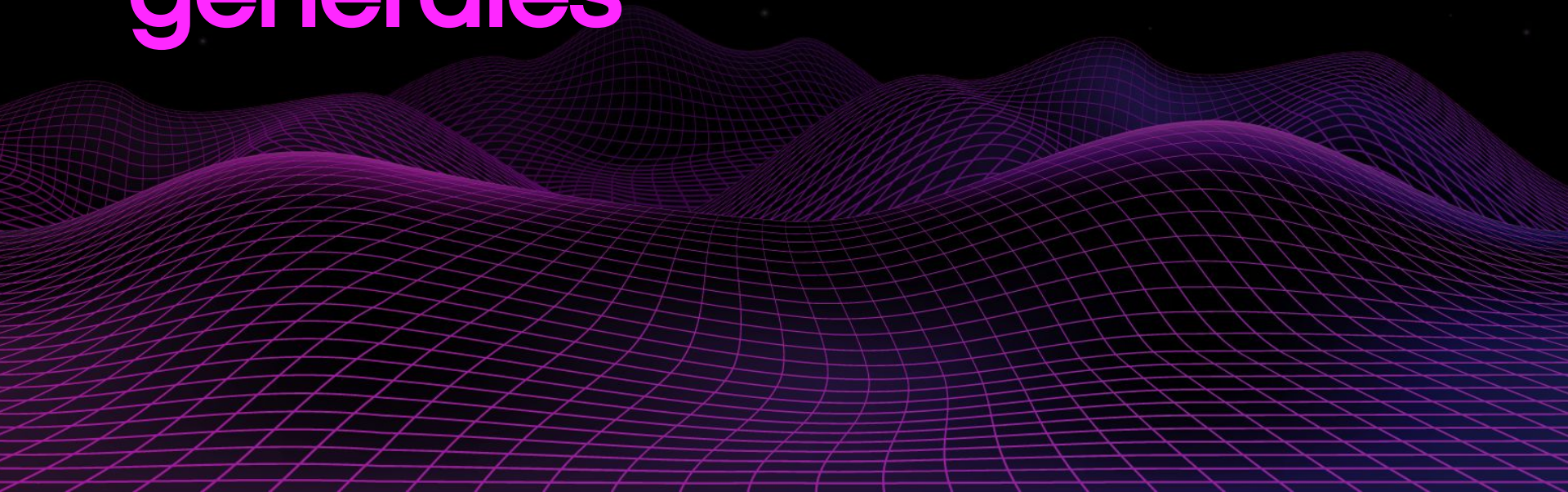
# ÍNDICE

————■ **1** Pautas generales

————■ **2** Requisitos mínimos



# Pautas generales



## > Pautas generales

- El objetivo de este sprint es desarrollar los componentes (páginas) Sign Up y Sign In.
- Se sigue utilizando el mismo repositorio que para el sprint 3.
- El repositorio debe ser pusheado antes del horario de entrega, los mentores realizarán el último pull 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 se procederá a la corrección de cada trabajo. Para obtener un MVP se deben cumplimentar todas las consignas aquí especificadas. Incumplir algunos de los criterios evaluatorios implica un NO-MVP.
- La interfaz 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 código (nombre de funciones, variables, comentarios, etc) puede estar en inglés o en español (**evitar la mezcla**).
- El no respeto a los requisitos del MVP por falta de lectura, falta de cross-check (o hecho de manera deficiente), etc. los hará incurrir en un NO-MVP.

# Requisitos mínimos

# Requisitos mínimos

A nivel frontend:

- Se deben crear los componentes de **Sign Up**, **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 logout debe, por supuesto, desloguear al usuario de la app.

## Más aclaraciones:

- 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, pero para ejercitar protección de rutas de frontend, no permitamos el acceso a signup/login a usuarios logueados).
- Deben hacer una validación mínima via frontend antes de enviar los datos al backend.
- **Mostrar mensajes de error personalizados, claros, atractivos y amigables.**



# Requisitos mínimos

A nivel backend:

En base a este trabajo hay que hacer:

- Se deben crear los endpoints y controladores de crear cuenta y loguear usuario y por supuesto, el modelo.
- En el caso de crear cuenta la request debe pasar antes por un validador hecho con **Joi** (no exagerar en las condiciones de validación pero tampoco obviarlas ni poner 1 o 2).
- Atención al mail (debe ser único ya que actúa como **USERNAME** y debe tener formato de **email**) y la **password**.
- Los países pueden estar hardcodedos en un array o alimentarse de alguna api de países. Como mínimo debe haber 8 países 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.

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

¡MUCHAS GRACIAS!

!!  
U.