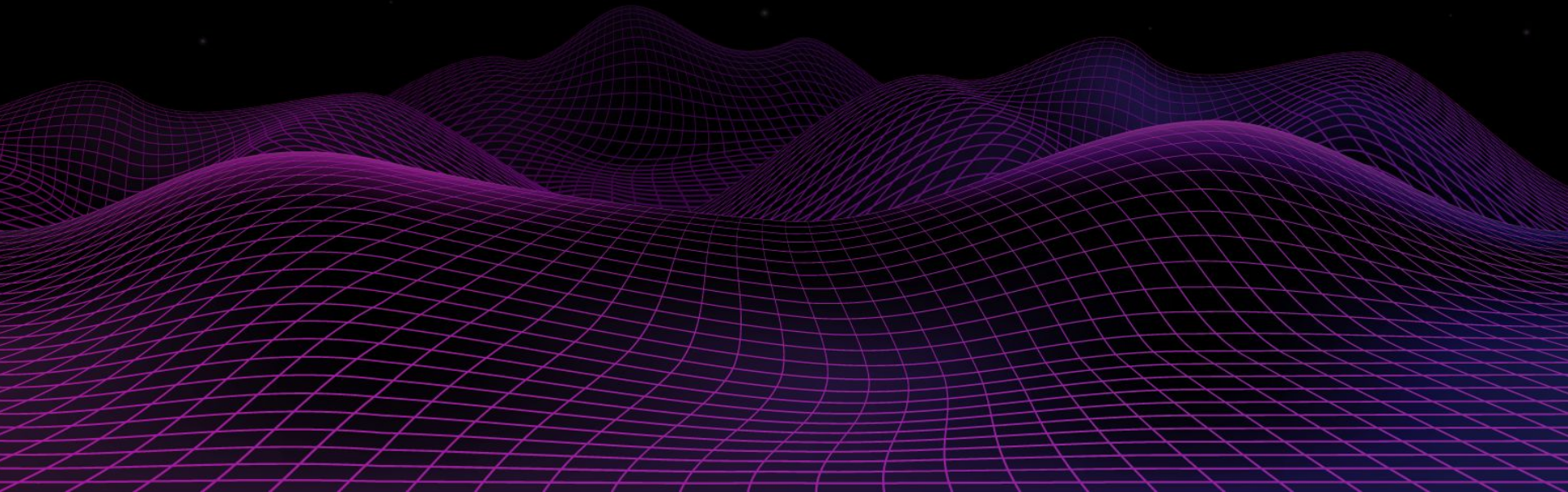


# BIENVENIDOS

**MIND  
HUB.**

# Sprint 3

## Condiciones MVP



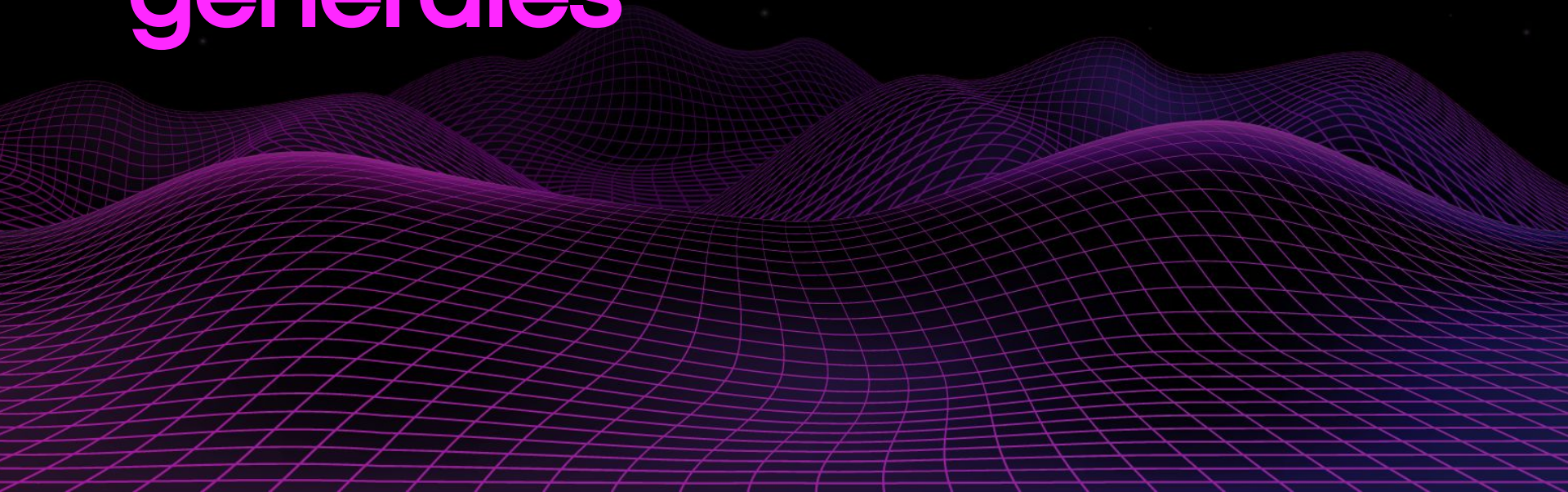
# ÍNDICE

———— ■ **1** Pautas generales

———— ■ **2** Requisitos mínimos



# Pautas generales



## > Pautas generales

- El objetivo de este sprint es desarrollar el componente (página) City (Detalle de una ciudad).
- Se sigue utilizando el mismo repositorio que para el sprint 2.
- 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:

- 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 o algún ícono alusivo (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".

## Más aclaraciones:

- ¡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.
- DEBE HABER UN MENSAJE DINÁMICO, SIMPÁTICO 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).



# Requisitos mínimos

A nivel backend:

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)
  - Crear un nuevo itinerario
  - Modificar un itinerario
  - Borrar un itinerario
- Los controladores que no se usen vía Frontend deben estar igual (y ser chequeados con Insomnia o Postman que funcionen ok).
- Atención a la relación entre documentos en **MongoDB**, así es como deben establecerse las "pertenencias" de un itinerario a una ciudad (y más adelante, de un paquete de actividades a un itinerario).

¡MUCHAS GRACIAS!

!!  
U.