

FINAL CHALLENGE

MIND
HUB.

ÍNDICE

- **1** Gestión
- **2** Fechas
- **3** Producto
- **4** Requerimientos y Tecnologías

> Gestión

Los tiempos son auto-gestionados por cada grupo. Esto incluye:

- organización de equipos de trabajo
- organización de tareas para cumplir con objetivos parciales (sprints)
- uso de plataformas (discord, meet, zoom, trello, jira)
- dailies/reportes diarios
- retros/conclusiones

Las consultas se deben dirigir de equipo a líder y de líder a líderes.

Recuerden que esto es un desafío y debe ser 100%

original/ingenioso y autónomo/independiente para que ustedes se den cuenta qué tan preparados están para trabajar como desarrolladores o cuánto tienen que seguir investigando/estudiando.

> Fechas

Los líderes deberán informar diariamente los avances que van logrando y los inconvenientes/resoluciones que van surgiendo durante el desafío.

Realizaremos dos puntos de control, **(checkpoint y simulacro)**, para ver los avances de las apps web y mobile.

- **Serán miércoles 11/10 y miércoles 18/10**

Realizaremos la presentación de las app (web y mobile) de cada grupo el **jueves 19/10**. En la misma deben hablar todos y no debe durar más de 5 minutos.

La presentación tiene que ser dinámica y efectiva y además con **cámara**.

> Producto

- Debe ser un **MARKETPLACE/E-COMMERCE**
- La app debe ser full responsive y debe estar en inglés.
- La app debe incluir una característica “extra” que la diferencie de otros e-commerces del mismo rubro.
- El producto debe incluir mínimamente las interfaces de:
HOME - REGISTER - LOGIN - STORE- DETAILS - CART -
ADMIN PANEL

> Producto

El cliente espera una versión mobile 100% pensada para el usuario "comprador". Por lo que la presentación deberá mostrar todas las vistas necesarias para que un usuario registrado pueda:

- buscar un producto (mínimo 2 filtros)
- ver sus detalles (puede ser en la misma página o en un modal)
- sumarlo a un carrito
- gestionar el pago
- re-dirigirlo hacia alguna interfaz donde se le informe que la compra se realizó con éxito

Un usuario no registrado solo puede ver la página de inicio, los formularios de registro/ingreso y los productos.

> Producto

El cliente espera que la versión web tenga las mismas funcionalidades que la versión mobile pero tiene muy en claro que la versión web va a ser para que su empresa **gestione/administre los productos**, por lo que en la presentación quiere ver:

- vistas para crear productos nuevos
- vistas para carga de stock
- vistas para gestionar los envíos

> Producto

Otros cohorts han hecho tiendas como:

- venta de vinos, gin, bebidas
- venta de pizzas, hamburguesas, comestibles,
- venta de artículos de decoración, plantas, ropa,
- venta de NFT, cursos,
- alquiler de libros, de películas, de canchas,
- alquiler de salas, de habitaciones de hoteles, etc.

Incluso hubo un proyecto multi-store donde el administrador podía crear comercios de distintos rubros.

Pueden mostrar la app en vivo o grabar un video para mostrar el día de la presentación.

Requerimientos

(frontend)

Estudiar los lineamientos del paradigma **mobile-first** y comenzar con el diseño (al menos bocetos) de la app mobile.

De forma **opcional** estudiar cómo generar un **apk** de la aplicación y generar un **qr** que le permita al cliente probar la app mobile.



Requerimientos

(frontend)

Luego de confirmar los productos/servicios, el cliente pagará a través de una **PASARELA DE PAGOS** que debe estar integrada como mínimo con una forma de pago.

Adicionalmente debe enviar un **MAIL DE RESUMEN** al usuario con lo que compró.



Requerimientos

(frontend)

Crear los modelos, controladores, rutas, validaciones, protecciones necesarias para su app.

Sistema de **login/signup** con roles (al menos admin y user).

El backend deberá ser funcional a la app mobile y a la app web.

Se deberá realizar la mayor cantidad de tests posibles, **mínimo 2 test por cada endpoint** de la app (éxito y error).



Requerimientos (frontend)

Para el backend deben trabajar con Node/Express, MongoDB como base de datos, Joi como librería de validación y Passport/JWT como librerías de autenticación.

Para el frontend de la web-app deben trabajar con React y para la mobile-app React Native.

Pueden implementar las librerías que quieran.

Documentar la API con swagger/postman/apidoc y en ambos repositorios definir un .md (markdown) para describir el proyecto

La app debe estar **hosteada en su totalidad**, tanto backend como frontend.



¡MUCHAS GRACIAS!

!!
U.