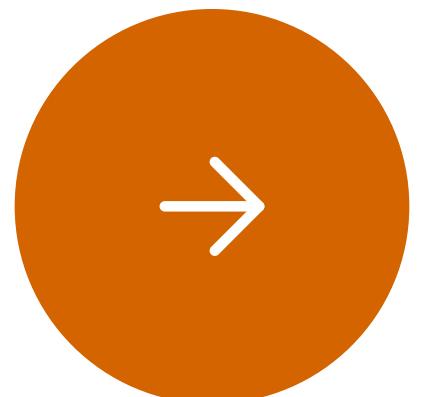


Bienvenidos al proyecto de TeacherApp 

# TFM

# Full Stack Dev

Trabajo realizado por el Grupo B: Cristina Valdés, Albert Altarriba, Jaume Comas, Jorge de Toro y Roberto Alonso.



01

## Equipo y organización

04

## La App

02

## Roadmap

05

## Documentación técnica

03

## Metodología

# Gestión de equipo

Para la elaboración del proyecto, el equipo se dividió en dos bloques: la parte FRONT y la parte BACKEND. El objetivo de esta división es que cada miembro tuviera sus tareas bien determinadas con unos tiempos ajustados al calendario y sacar el mejor rendimiento de cada uno de los miembros.

Se realizó una reunión previamente a la asignación de tareas con todos los miembros del equipo, en la que cada uno expuso sus ideas. Una vez finalizada, se asignaron todas las tareas a cada uno de sus miembros quedando la estructura de tal manera:

## Equipo y organización



### FRONT

- Jorge
- Albert
- Jaume
- Cristina



### BACKEND

- Roberto

# Roadmap

## 1º Sprint



### Front Desktop

Diseño UX/UI para toda la aplicación con parte pública y parte privada, primeros desarrollos en la primera aplicación.

## 2º Sprint



### API

Diseño de la API (todo basado en modo SQL) y estudio para la adecuación del FRONT.

## 3º Sprint



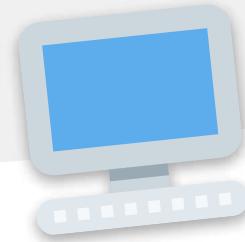
### Funcionalidad TypeScript con API

Equipo de FRONT, realiza todo el código necesario para meter funcionalidades a la APP (filtros, token, eventos). Adaptación responsive. Requisitos mínimos hechos y requisitos deseables acabados.



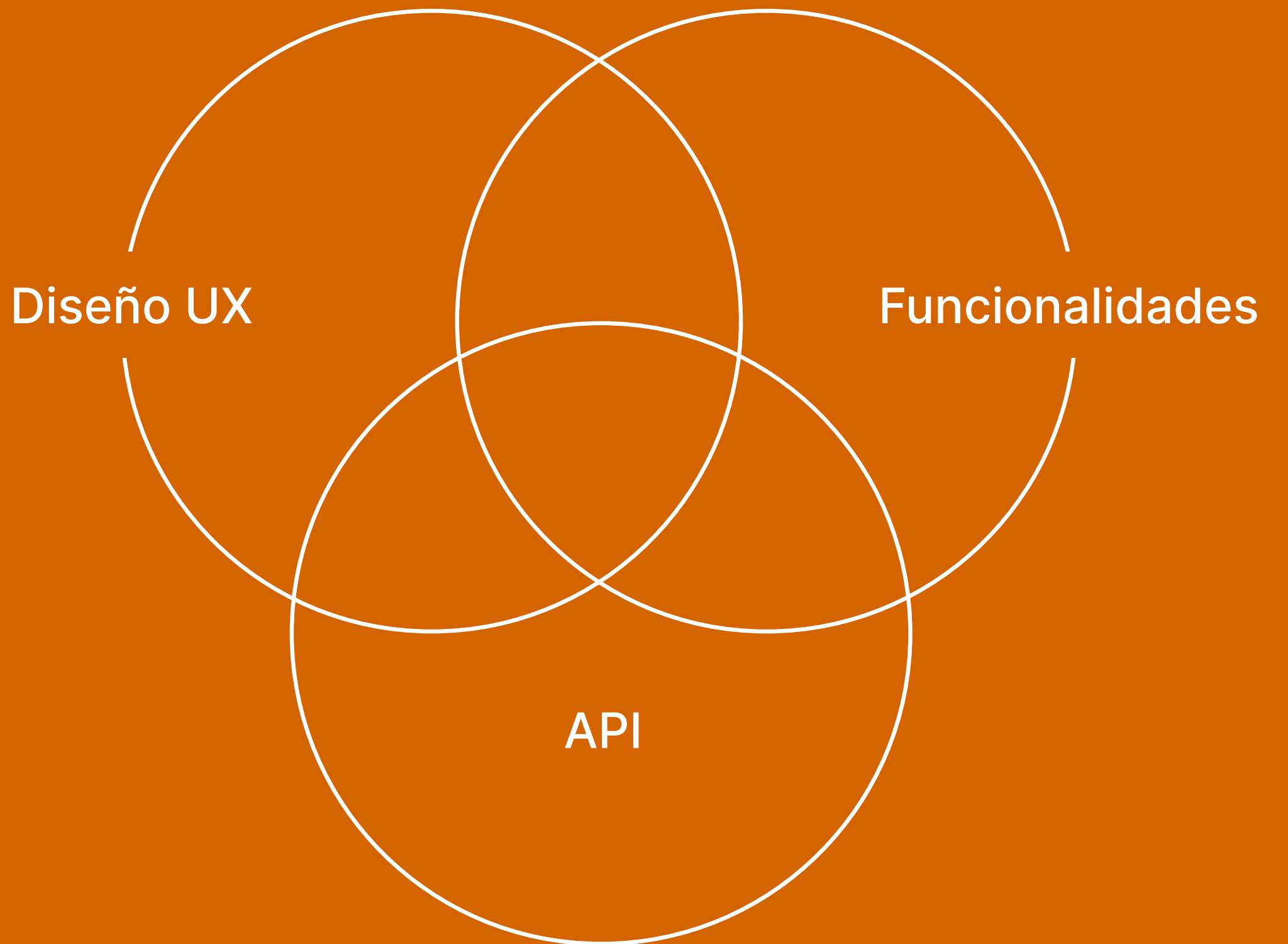
# Metodología Scrum

Hemos trabajado con unos sprints de aprox. 10 días en los que hemos definido unas tareas de manera independiente teniendo en cuenta las diferentes dependencias de código de cada uno de los miembros. Cada reunión se comentaba las diferentes problemáticas y posibles soluciones en el transcurso del desarrollo del proyecto. Hemos utilizado diferentes herramientas de comunicación como NOTION, FIGMA, TEAMS y WHATSAPP para hacer de manera transversal y transparente los objetivos marcados en cada sprint.



# La APP

Desde el principio hemos trabajado bajo estos principios para la realización del proyecto, consideramos de vital importancia la armonía entre estos tres campos.



# Nuestra home



- O1 Filtros para facilitar la búsqueda del profesor**
- O2 Visualización mapa interactivo**
- O3 Enlaces de registro y a mi cuenta**
- O4 Listado de profesores más cercanos**

# Formulario registro

01 Checkbox options:  
Profesor o alumno

02 Enlace a mi cuenta  
desde varios puntos

03 Datos informativos y  
de elección

04 Posibilidad de retorno  
home



# Mi cuenta

- O1 Entrar en mi cuenta y enlace a mi dashboard**
  
- O2 Enlace a formulario de registro**
  
- O3 Posibilidad de retorno home**



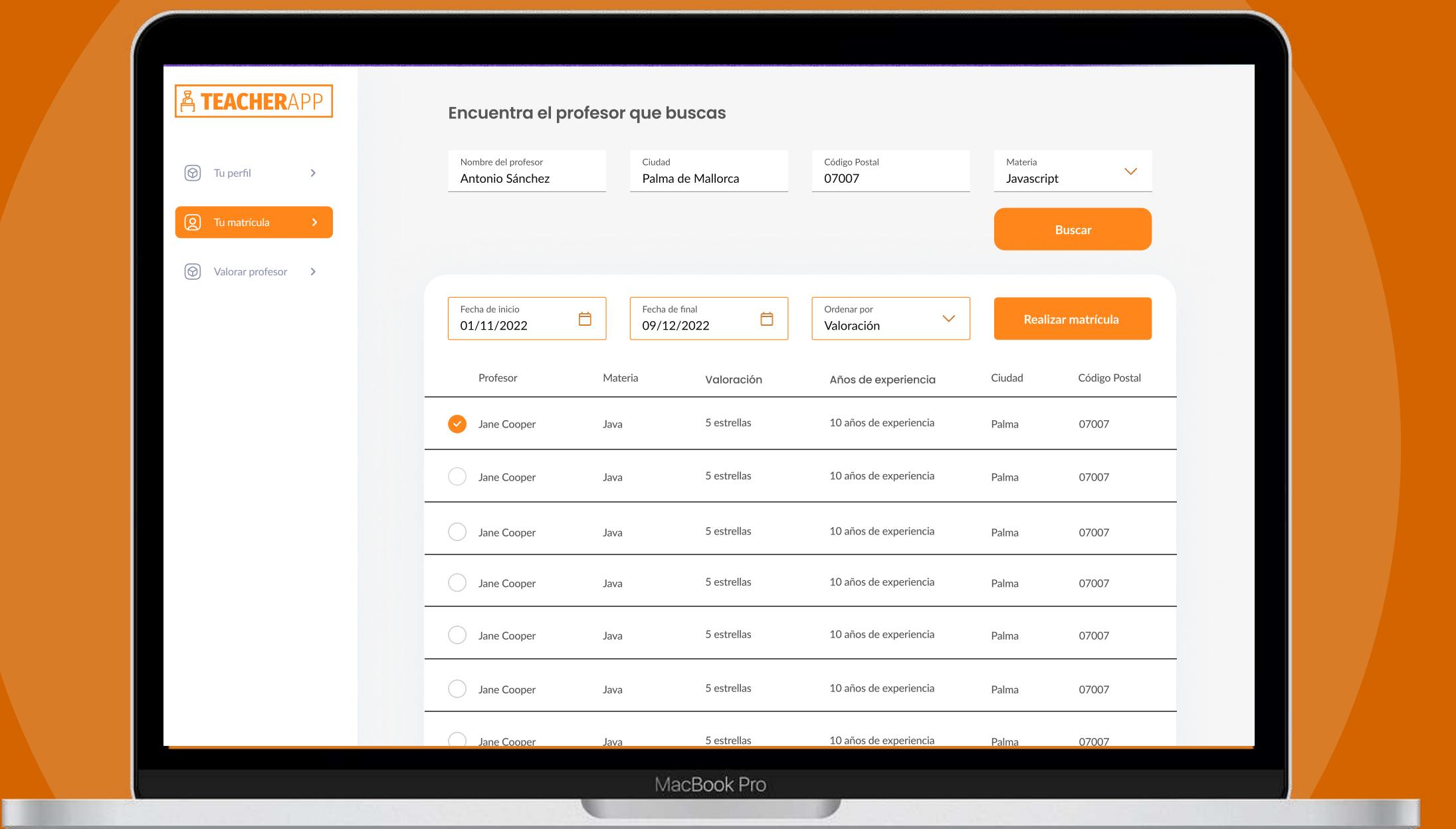
# Dashboard alumno

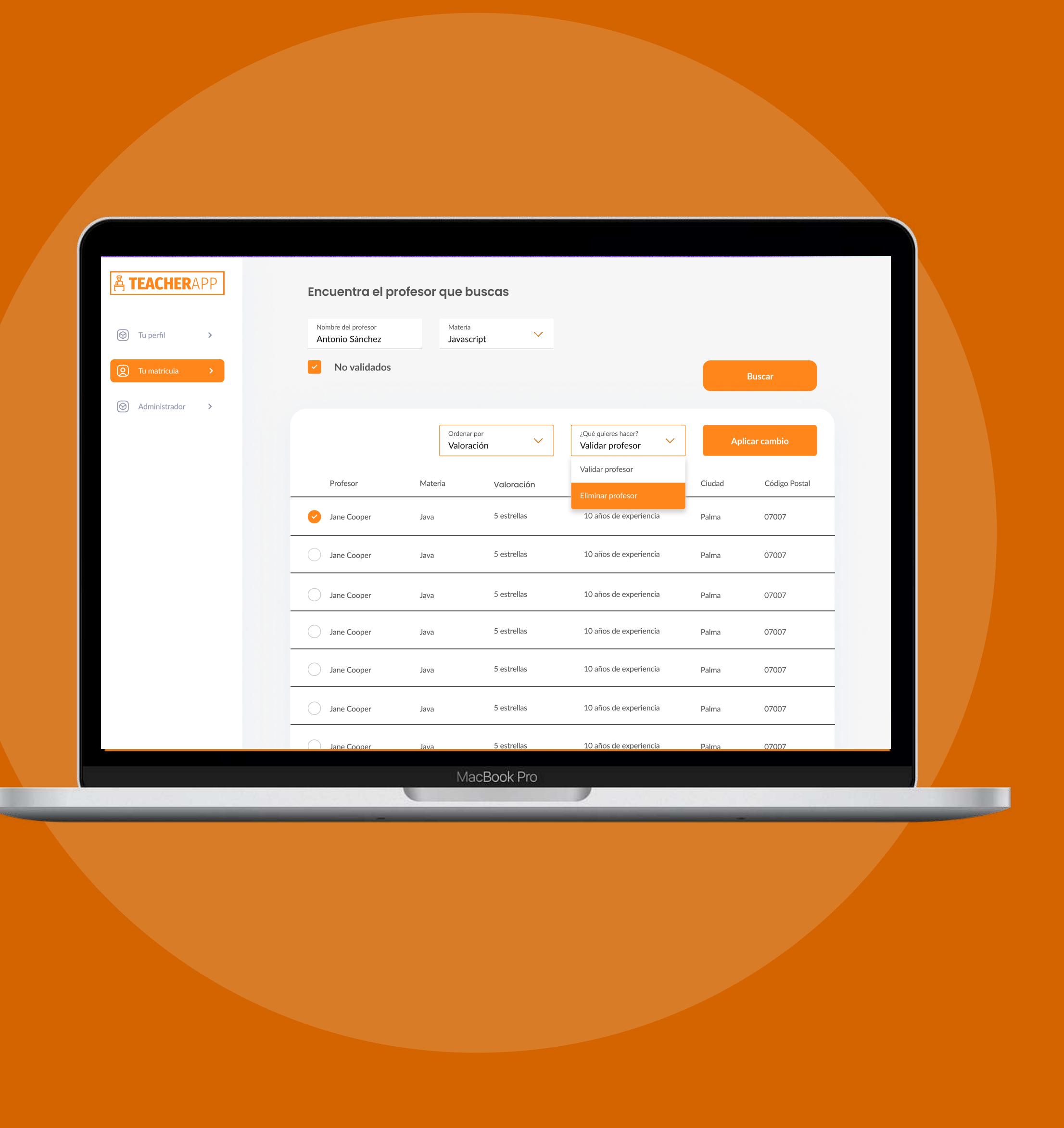
O1 Filtros para facilitar la búsqueda del profesor

O2 Listado de profesores y solicitar una matrícula

O3 Ver su perfil y modificar datos

O4 Realizar una valoración al profesor



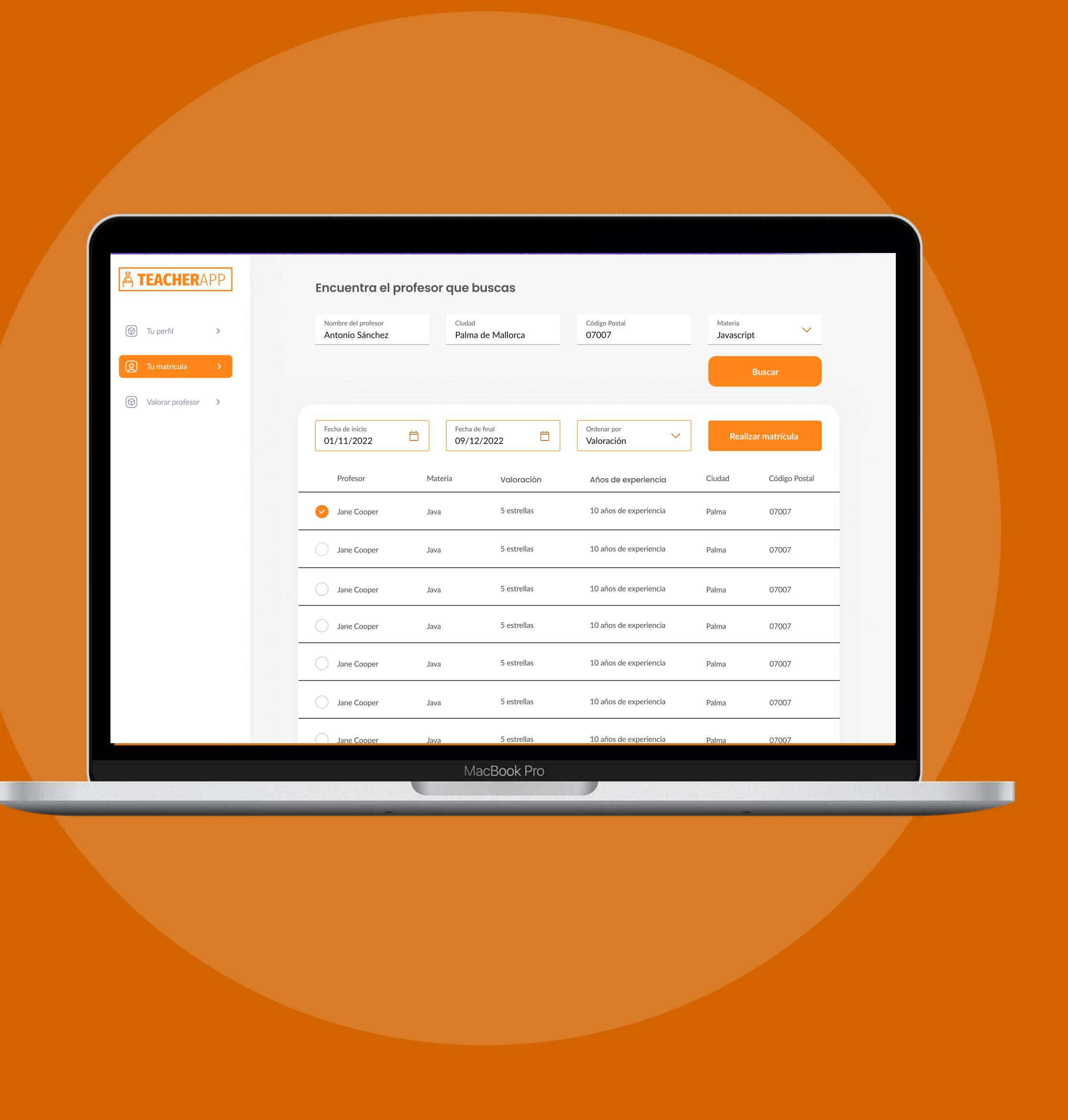


AppTeacher

# Dashboard administrador

O1 Filtros para facilitar la búsqueda del profesor

O2 Listado de profesores y validar matícula

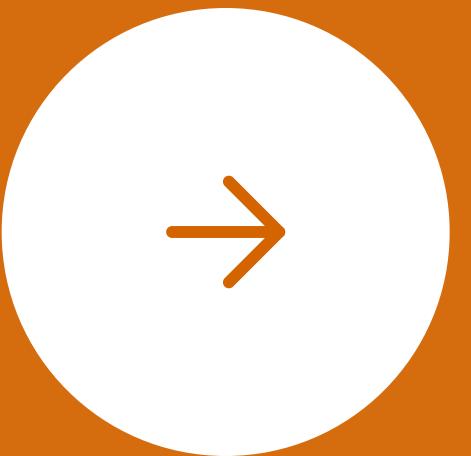


AppTeacher

# Dashboard profesor

- O1 Filtros para facilitar la búsqueda del alumno**
  
- O2 Listado de alumnos y aprobar o eliminar matrícula**
  
- O3 Ver su perfil y modificar datos**

# Documentación técnica



## 1. Inicio base de datos

Abrir el archivo modelo.mwb con MySQL Workbench para ver el modelo y hacer "Forward Engineer" para obtener la estructura de la base de datos o ejecutar el script estructura.sql directamente.

## 2. Instalación API

Con la terminal abrir la carpeta AppTFM/API y hacer npm install para instalar las dependencias. Hacer npm run dev para ejecutar la API.

## 4. Abrir la APP

Con otra terminal abrir la parte del front end en AppTFM/appTeacher y hacer npm install. NOTA: es posible que sea necesario tener que hacer npm install --force por problemas derivados de la versión de Angular y de la API de Google Maps.

**IMPORTANTE:** Para crear cualquier perfil de usuario, primero tienes que generar un user y a partir del user con el type asignado ya te crea el id correspondiente.

## 3. Creación user

Abrir la API con VSC, cambiar la contraseña de MySQL en API/db/dbConfig.js y en el archivo peticiones.rest hacer POST de User con user\_type = ADMIN, después hacer PUT de admin. NOTA: Recordar el mail y la contraseña para poder acceder a la área privada de admin

## 5. Correr la APP

Con ng serve -o ya tienes corriendo la app, ir con el navegador a la dirección <http://localhost:4200/>

## 6. Validaciones

Lo siguiente ya es registrarte o como alumno o como profesor y en caso de ser profesor el admin te tiene que validar y en caso del alumno solicitas matrícula en la matrícula que deseas.

# ¡Gracias!