

**Unidade curricular:** Sistemas Móveis Interativos

**Curso:** CTESP Desenvolvimento Web e Multimédia

**Ano letivo:** 2024/25

## **Trabalho prático – Gestão de viagens**

### **Âmbito**

O objetivo do trabalho prático desta unidade curricular é desenvolverem uma aplicação em que os utilizadores possam planear e registar as suas viagens. Para isso devem organizarem-se em grupos de 3 elementos (máx.).

### **Tecnologias obrigatórias**

Ionic + Angular + Formulários

### **Antes de começarem**

É obrigatório darem um nome ao vosso projeto bem como um logotipo.

É obrigatório desenharem todos os *mockups* relevantes (balsamiq, figma ou outra ferramenta de *mockups*).

Devem utilizar uma plataforma de gestão e planeamento do projeto (trello, goodday ou outra ferramenta de gestão e planeamento de projetos).

É obrigatório todo o código estar no GitHub.

É obrigatório que todos os dados sejam dinâmicos.

### **Funcionalidades obrigatórias:**

- Criar, listar, editar e/ou remover viagens:
  - Associar dados relevantes como:
    - Descrição, tipo, estado, localização, datas de início e fim, etc...;
    - Permitir escrever comentários.
  - Associar locais às viagens:
    - Associar dados relevantes como descrição, tipo, estado, localização, datas de início e fim, etc...;

- Permitir escrever comentários.
- Em cada HTTP Request terem um loading (loadingController);
- Sistema de notificações (toastController);
- Tratamento de erros (alertController).

#### Funcionalidades opcionais:

- Deploy do projeto para o vercel ou outra plataforma;
- Pesquisa, filtros e ordenação (deve ser feito diretamente na aplicação);
- Integração com google maps ou outro tipo de mapa:
  - Escolha do local através do mapa;
  - Mostrar todos os locais de uma viagem no mapa;
  - Rota de uma viagem com os locais.
- Bónus
  - Desenvolverem a vossa própria API;
  - *Intro sliders*;
  - Registo de conta e controlo de sessões;
  - Multi-idioma;
  - Partilhar viagens com outros utilizadores;
  - Funcionalidades adicionais, integrações com *terceiros*, bem como tudo o que considerarem ser pertinente para o desenvolvimento do trabalho prático.

#### Avaliação e entrega

Será avaliada a qualidade do código fonte e a originalidade da abordagem ao âmbito do projeto. **Em casos de plágio, os trabalhos serão anulados.** O trabalho deve ser entregue no moodle até às 23:59 do dia 12 de janeiro de 2025.

Todos os alunos devem entregar um ficheiro .txt com os links do repositório do GitHub e da plataforma de gestão de projetos, bem como todo o código fonte (exceto a pasta node\_modules). **Em casos de não entrega, o trabalho não será considerado para avaliação do aluno e o grupo será penalizado.**

Para uma avaliação justa, individualmente deve ser entregue a seguinte grelha de avaliação de cada elemento do grupo, incluindo o próprio:

Nome do aluno:

Nº do aluno:

Nome de utilizador:

Avaliação final do trabalho prático (0-20):

Nome	Avaliação (0-20)	Apreciação

Boa sorte,

O docente: Marcelo Antunes Fernandes

## Documentação

### Base de dados

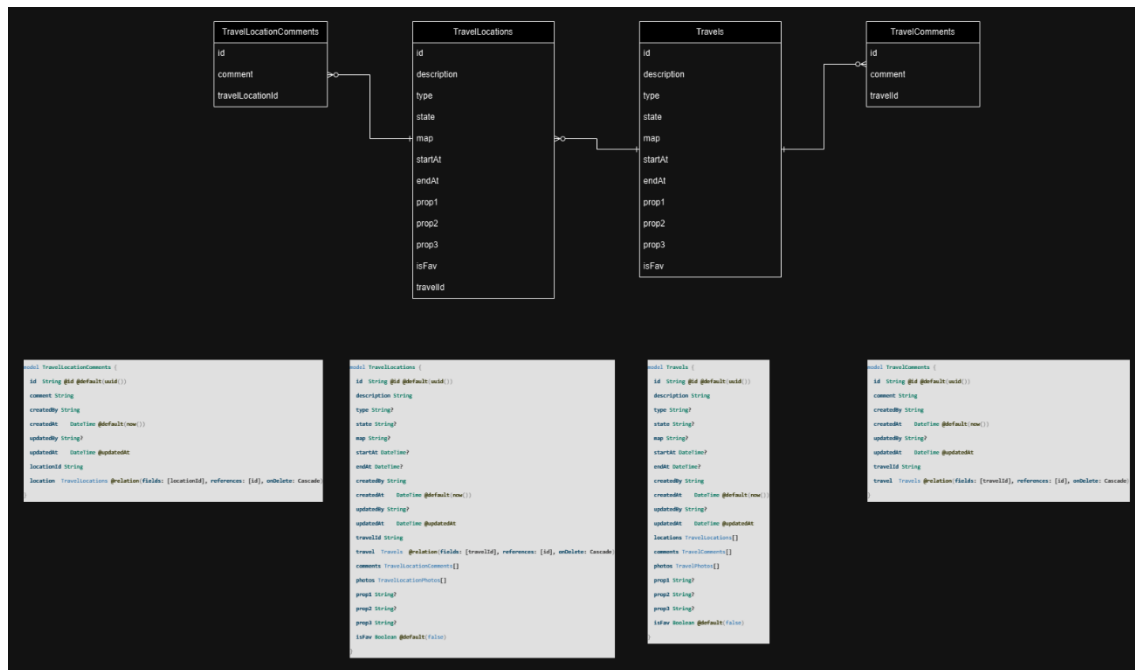


Figura 1 - Relações e tipos de dados

A base de dados foi estruturada de forma que 1 viagem possa ter várias localizações e vários comentários. E cada localização possa também ter vários comentários.

Foram colocadas 3 colunas (prop1, prop2, prop3) nas tabelas de viagens de locais de viagem para que possam utilizar no que bem entenderem.

Os deletes são *cascade*, ou seja, ao eliminarem uma viagem todas os outros dados associados são eliminados também automaticamente.

Na Web API nas viagens foram feitos os *includes* de locais, comentários e comentários em locais.

Os dados são ordenados *desc* pelo campo startAt, se for null pelo createdAt.

A autenticação é *Basic Auth* e devem utilizar as vossas credenciais de acesso fornecidas nas aulas.

## Viagens

GET: <https://mobile-api-one.vercel.app/api/travels>

GET: <https://mobile-api-one.vercel.app/api/travels/:id>

POST: <https://mobile-api-one.vercel.app/api/travels>

```
data: {  
  description,  
  type,  
  state,  
  startAt: startAt ? new Date(startAt) : null,  
  endAt: endAt ? new Date(endAt) : null,  
  createdBy,  
  prop1,  
  prop2,  
  prop3,  
  isFav,  
},
```

PUT: <https://mobile-api-one.vercel.app/api/travels/:id>

```
data: {  
  description,  
  type,  
  state,  
  startAt: startAt ? new Date(startAt) : null,  
  endAt: endAt ? new Date(endAt) : null,  
  updatedBy: user,  
  prop1,  
  prop2,  
  prop3,  
  isFav,  
},
```

DELETE: <https://mobile-api-one.vercel.app/api/travels/:id>

## Locais

GET: <https://mobile-api-one.vercel.app/api/travels/:id/locations>

GET: <https://mobile-api-one.vercel.app/api/travels/:id/locations/:id>

POST: <https://mobile-api-one.vercel.app/travels/locations>

```
data: {  
  travelId,  
  description,  
  type,  
  state,  
  map,  
  startAt,  
  endAt,  
  createdBy,  
  prop1,  
  prop2,  
  prop3,  
},
```

PUT: <https://mobile-api-one.vercel.app/travels/locations/:id>

```
data: {  
  description,  
  type,  
  state,  
  map,  
  startAt,  
  endAt,  
  updatedBy: user,  
  prop1,  
  prop2,  
  prop3,  
},
```

DELETE: <https://mobile-api-one.vercel.app/travels/locations/:id>

### Comentários de viagens

POST: <https://mobile-api-one.vercel.app/travels/comments>

```
data: {  
    travelId,  
    comment  
},
```

DELETE: <https://mobile-api-one.vercel.app/travels/comments/:id>

### Comentários de locais de viagens

POST: <https://mobile-api-one.vercel.app/travels/locations/comments>

```
data: {  
    locationId,  
    comment  
},
```

DELETE: <https://mobile-api-one.vercel.app/travels/locations/comments/:id>