

# Trabalho #2

## API Mashup

## 1 Trabalho #2

Data de Entrega: 26 de junho de 2025

---

### 1.1 1. Objetivo

Desenvolver uma aplicação web que:

- Consoma e integre dados de pelo menos 2 APIs externas, com o **servidor** a efetuar todas as requisições (server-side).
- Inclua um sistema de autenticação de utilizadores baseado em **Express Sessions** ou **Passport-local**.
- Utilize base de dados **MongoDB** para armazenar informação dos utilizadores (por exemplo histórico de pesquisas)

### 1.2 2. Funcionalidades

#### 1. Autenticação (Server)

- Registo de utilizador (username + password)
- Início de sessão com sessões Express ou Passport-local
- Proteção de rotas para apenas utilizadores autenticados

#### 2. Mashup de APIs

- O utilizador, após login, introduz um termo de pesquisa (e.g., nome de cidade, artista, palavra)

- O servidor consome até duas APIs REST externas e retorna dados integrados ao cliente
- **Exemplos de APIs:**
  - **OpenWeatherMap:** clima da cidade (`/weather?q={city}`)
  - **RestCountries:** informações do país (`/alpha/{code}`)
  - **Wikipedia REST API:** resumo de artigos (`/page/summary/{title}`)
  - **Pixabay** ou **Unsplash:** imagens livres de royalties
  - **NewsAPI** ou **GNews API:** notícias relacionadas
  - **TMDB API:** informação e posters de filmes

### 3. Persistência em MongoDB

- Guardar credenciais (idealmente hash das passwords)
- Histórico de pesquisas por utilizador

## 1.3 3. Tecnologias

- **Frontend:** HTML, CSS (ou Tailwind/Bootstrap), JavaScript
- **Backend:** Node.js (v12+), Express
  - Autenticação: **express-session** ou **passport-local**
  - Chamadas a APIs feitas no servidor com `fetch` API (alternativamente com **Axios**, ou **node-fetch** em versões mais antigas) usando **async/await**
- **Base de Dados:** MongoDB (Atlas ou local)

## 1.4 4. APIs Externas (sugeridas)

- **OpenWeatherMap** (clima e geocoding)
- **RestCountries** (bandeiras, capitais, moedas)
- **Wikipedia REST API** (enciclopédia)
- **Pixabay / Unsplash** (imagens)
- **NewsAPI / GNews API** (notícias)
- **Exchange Rates API** (câmbio de moedas)
- **DictionaryAPI** (definições, sinónimos)
- **TMDB API** (filmes, trailers)

**Nota:** Registem-se nas plataformas e obtenham as chaves necessárias. Todas as requisições a estas APIs devem ser feitas pelo servidor, protegendo as suas credenciais. As API Keys não devem ficar expostas no código.

## 1.5 5. Regras & Avaliação

1. **Grupos:** 2 elementos por grupo.
  2. **GitHubClassroom:** Repositório privado com acesso ao utilizador **pedromoreira-estg**.
  3. **Build & Install:** Incluir script para instalar dependências e iniciar a aplicação.
  4. **Documentação (README.md):** Incluir:
    - Identificação dos elementos do grupo
    - Tecnologias e APIs utilizadas
    - Instruções de instalação e configuração das chaves e do MongoDB
    - Comandos para executar localmente
    - Link de deployment (**render.com** ou equivalente)
  5. **Deployment:** Aplicação operacional online (ex.: render.com).
  6. **Entrega em Moodle:** Cópia do README.md e **.zip** **link** do repositório.
- 

Boa sorte!