



Tema: Aplicação para Criar Equipa Favorita de Pokémons “myPokémon Crew”

Objetivo: Desenvolver uma aplicação que consuma a PokéAPI, permita consultar os Pokémons e criar equipas personalizadas de favoritos, armazenando tudo numa base de dados e que tenha uma UI intuitiva e que permita gerir e visualizar as características dos Pokémons favoritos do utilizador.

No final, deves partilhar o código num repositório GitHub

Contents

1	Requisitos Funcionais	3
2	Arquitetura da Aplicação	4
3	Possíveis Tecnologias a Usar	5
4	Requisitos Técnicos.....	7
5	Segurança e Proteção de Dados dos Utilizadores	7

1 Requisitos Funcionais

A aplicação deverá contemplar as seguintes funcionalidades, organizadas por fases, a serem implementadas de forma sequencial e progressiva.

Não é obrigatório concluir todas as fases.

No entanto, é importante que cada fase que seja concluída esteja minimamente funcional. É preferível completar uma fase anterior garantindo as funcionalidades mínimas do que avançar para várias fases de forma incompleta.

**O caminho é dividido em 3 fases,
tal como a jornada de um verdadeiro mestre Pokémons**

Fase 1: Descobrir Pokémons

Como Treinador, precisas de conhecer os Pokémons.

- **Listagem dos Pokémon**

- A aplicação deve consumir a PokéAPI: <https://pokeapi.co/>.
- Deve ser possível fazer pesquisa pelo nome.
- Apresentar lista de todos os Pokémons correspondentes à pesquisa.
- Cada Pokémon aparece num card, com informações básicas (ex. nome, tipo, peso, altura, imagem, estatísticas principais).
- Ao clicar/selecionar um Pokemon, deve mostrar detalhes de cada Pokémon.

- **Detalhe de cada Pokémon**

- A aplicação deve consumir a PokéAPI: <https://pokeapi.co/>.
- Apresentar as características do Pokémon.

É como abrir a tua Pokédex e começar a explorar o mundo Pokémon.

Fase 2: Construir a tua Equipa Favorita

Um Treinador precisa de uma equipa.

- **Criação de Equipa**

- O utilizador pode selecionar até **6 Pokémon** para compor a sua equipa favorita.

- A seleção pode ser feita na página de listagem dos Pokémons e/ou na página de detalhe do Pokémon.
 - A estrela  no card permite-te adicionar ou remover Pokémons da tua equipa.
 - Deve ser possível adicionar ou remover um Pokémon da equipa.
 - Os dados da equipa e a referência aos Pokémons selecionados deve ficar guardados na base de dados da aplicação.
 - Na tua página inicial, terás sempre os Pokémons favoritos em destaque.
- **Área dedicada à gestão da equipa favorita**
 - Informação da equipa.
 - Listagem dos Pokémons da equipa.
 - Possibilidade de remover Pokémons.

Agora tens a tua própria “box” de Pokémons, sempre pronta para a batalha!

Fase 3: O Campo de Batalha

É hora de treinar e lutar!

- **Batalha**
 - Desenvolverás um simulador de batalha onde dois Pokémons se enfrentam.
 - O utilizador pode selecionar dois Pokémons para iniciar uma batalha.
 - A simulação da batalha deve considerar atributos principais (HP, ataque, defesa, velocidade) para determinar ordem e dano. Qualquer conjunto de regras que inventes serve.
 - O combate deve ser apresentado de forma visual e interativa (ex.: barras de energia, log de eventos, efeitos de ataque).
 - A aplicação deve guardar o resultado da batalha.

Aqui mostras a tua criatividade: como tornar uma batalha Pokémon emocionante!

2 Arquitetura da Aplicação

A imagem seguinte ilustra os componentes principais do sistema.

O utilizador usa a aplicação através de um browser

A aplicação que vais desenvolver

A API que vais consumir
<https://pokeapi.co/>



Figura 1: Componentes principais do sistema

Tem em conta as várias camadas da aplicação:

- **Camada de Dados**

- Base de dados com tabelas tais como:
 1. Utilizadores (id, username, email, password, ...)
 2. Equipas (id, utilizador_id, nome_equipa, ...)
 3. Pokemons (id, pokeapi_id, equipa_id, ...)
 4. Batalhas (id, utilizador_id, pokemon_a_id, pokemon_b_id, pokemon_vencedor_id, ...)
- Se quiseres podes não guardar os dados dos Pokémons na base de dados e consultar a API sempre que necessário.

- **Camada de Negócio (Lógica da Aplicação)**

- Funções para consumir a API.
- Funções para inserir e consultar dados da base.
- Funções para gerir equipas (adicionar/remover Pokémon).

- **Camada de Apresentação (Interface)**

- Listagem dos Pokémons.
- Detalhes de cada Pokémon.
- Gestão da equipa favorita.

3 Possíveis Tecnologias a Usar

Podes escolher os componentes da stack tecnológica que quiseres, mas sugerimos que consideres um stack de desenvolvimento de aplicações web com base nas tecnologias seguintes.

Componentes de stacks tecnológicas:

- **Backend:**
 - Python (eg. Flask, Django, FastAPI)
 - Java (eg. Spring Boot)
 - Node.js (eg. Express, NestJs)
- **Base de Dados:**
 - SQLite (simples)
 - MySQL/PostgreSQL
 - MongoDB
- **Frontend:**
 - React
 - Angular
 - Vue
- **Repositório:** Github

Exemplos de stacks tecnológicas

Se não estás à vontade com nenhuma das opções, e quiseres usar um dos stacks comuns (opcional), sugerimos que comeces com os **tutoriais** da tabela seguinte para te familiarizares.

No final partilha o código, resultante do que fizeres neste tutorial, num repositório GitHub para podermos analisar.

Stack	Seguir o procedimento
Django	https://docs.djangoproject.com/en/5.2/intro/tutorial01/ https://docs.djangoproject.com/en/5.2/intro/tutorial02/ https://docs.djangoproject.com/en/5.2/intro/tutorial03/ https://docs.djangoproject.com/en/5.2/intro/tutorial04/
Python	

NestJs	https://docs.nestjs.com/first-steps https://docs.nestjs.com/techniques/mvc
React	
FastAPI	https://testdriven.io/blog/fastapi-react/
Pyhon	

4 Requisitos Técnicos

- A aplicação é uma solução web acessível através de navegadores web.
- A interface gráfica deve ser construída em HTML/CSS/JavaScript.
- A comunicação entre o navegador e o backend deve ser feito por meio de requisições HTTP (RESTful).
- O acesso à PokéAPI deve ser feito via requisições HTTP (RESTful).
- Submete o resultado do teu trabalho num repositório teu no GitHub.

5 Segurança e Proteção de Dados dos Utilizadores

A aplicação deve garantir a privacidade e segurança das informações pessoais dos utilizadores e das suas equipas de Pokémon favoritas.

- Implementar **sistema de login** com credenciais (nome de utilizador e palavra-passe).
- As palavras-passe devem ser armazenadas de forma segura, usando técnicas como **hashing** (ex.: **bcrypt**, **Argon2**).
- Apenas utilizadores autenticados podem criar e gerir equipas.