



ESGTS - ESCOLA SUPERIOR DE GESTÃO E TECNOLOGIA DE SANTARÉM

Pokémon App - Relatório Final

Relatório elaborado para a disciplina de
Base de Dados II - VFX

Preparado para:

Prof^a. Vânia Bastos

TSPSI - Tecnologias e Programação de Sistemas de Informação

Janeiro 2024

GRUPO

About Us

Home



Trabalho de Grupo

TPSI - VFX 2023/2024

Base de Dados II

Elementos do Grupo

João Pedro Costa - 220001295

Pedro Marques - 220000988

Rodrigo Passos - 220000051

Alexandre Guerreiro - 210100515

Objetivos da aplicação:

- > Gerir pokémons
 - criar
 - listar
- mostrar tipos associados
 - pesquisar
 - editar
- desativar / reativar



Índice

1. Introdução.....	4
2. Arquitetura da Aplicação.....	5
3. Diagramas.....	7
3.1 DER – Diagrama de Entidade-Relacionamento	7
3.2 Diagrama de Estados	8
4. Funcionalidades.....	8
5. Conclusão	10
6. Referências	11

Lista de Figuras

Figura 1 - Diagrama de Entidade e Relacionamento.....	7
---	---



1. Introdução

Este relatório visa apresentar o projeto de desenvolvimento de uma aplicação com o intuito de gerir Pokémons. O principal objetivo da mesma é oferecer uma plataforma intuitiva e eficiente para que os utilizadores possam explorar e manipular dados relacionados aos Pokémons. Através de consultas, edições e exclusões, proporcionamos uma experiência completa e interativa aos utilizadores interessados neste vasto mundo de criaturas fictícias.

A criação desta aplicação demonstra a aplicação prática de tecnologias modernas, como Spring Boot, Java, Javascript, HTML, Bootstrap, CSS e MySQL no desenvolvimento de sistemas web. Além disso, o projeto proporciona uma valiosa oportunidade para explorar conceitos fundamentais, como arquitetura de software, modelagem de base de dados e usabilidade, aprimorando ainda mais o conhecimento prático dos envolvidos.

O relatório está estruturado de forma a fornecer uma visão abrangente do projeto. Iniciaremos com uma análise detalhada da arquitetura da aplicação, destacando o uso do modelo MVC (*Model-View-Controller*) para promover uma organização eficaz e modular. Em seguida, apresentaremos o Diagrama de Entidade-Relacionamento (DER) para ilustrar a estrutura da base de dados, seguido pelo Diagrama de Estados, que será explorado para compreender o ciclo de vida das informações relacionadas aos Pokémons.

Dessa forma, procuramos não apenas documentar o desenvolvimento da aplicação, mas também compartilhar uma jornada rica em aprendizado, inovação e aplicação prática de conhecimentos essenciais no vasto campo de desenvolvimento de software.



2. Arquitetura da Aplicação

A arquitetura da aplicação é uma parte fundamental no desenvolvimento, proporcionando uma estrutura organizada para a implementação das funcionalidades. A nossa aplicação adota a estrutura MVC (*Model-View-Controller*), um padrão arquitetónico amplamente utilizado para o desenvolvimento de software web.

A aplicação segue a organização em três camadas principais do modelo MVC, desta forma, mantemos os métodos que lidam com as interações dos utilizadores (*front-end*) separados dos métodos que lidam com os acessos à base de dados (*back-end*).

A nossa aplicação foi construída de acordo com seguinte estrutura:

PokémonApp (*Model*)

A camada model é responsável por encapsular a lógica de negócio e o acesso aos dados. Na nossa aplicação, os modelos são representados pelas entidades “Pokémon”, “Pokémon_Type” e “Type”.

Cada entidade reflete aspetos específicos do domínio, como as características dos Pokémon, a associação entre Pokémon, Tipos e Level, e a definição dos próprios Tipos.

PokémonApp (*View*)

A camada de visão concentra-se na apresentação da interface do utilizador. Basicamente é a estrutura visual da aplicação, permitindo que os utilizadores interajam, realizem consultas, façam edições e exclusões, entre outras ações. Ao examinar o nosso código, essas interações são implementadas por meio das classes conhecidas como “rotas”.

As classes de rotas desempenham um papel crucial ao reencaminhar e controlar as solicitações dos utilizadores. Elas atuam como guias, ao direcionar as solicitações para os controladores apropriados na camada de Controlo (*Controller*). Dessa forma, garantimos uma clara separação de responsabilidades, facilitando a manutenção e expansão da aplicação.

Esta abordagem modular e estruturada permite que a camada de Visão funcione como um ponto central para a interação do utilizador, mantendo uma organização eficiente no código da PokémonApp.



PokémonApp (Controller)

A camada de Controlador desempenha um papel central na nossa aplicação, sendo responsável por gerir as interações entre a camada de Visão (*View*) e a camada de Model (*Model*). As principais funcionalidades são a monitorização de solicitações. As principais funcionalidades incluem:

Roteamento de Solicitações: Métodos anotados com `@GetMapping` e `@PostMapping` definem as rotas da aplicação, respondendo às solicitações dos utilizadores. Cada rota está associada a um método específico no controlador.

Processamento de Dados: Os métodos do controlador processam dados da camada de Visão. No método `handleEditForm`, por exemplo, os dados do formulário de edição são processados antes de interagir com a camada de Modelo.

Atualização da Visão: Após a interação com o modelo, os métodos do controlador atualizam a camada de Visão, garantindo que as alterações sejam refletidas na interface do utilizador. No método `listAllPokemons`, a lista de Pokémons é adicionada ao modelo para ser exibida na visão "listPokemons".

Gestão de Fluxo: Os métodos do controlador gerem o fluxo de controlo da aplicação, decidindo como lidar com solicitações específicas. O método `showEditForm`, por exemplo, determina se exibir o formulário de edição ou uma mensagem de alerta, dependendo da existência do Pokémon.



3. Diagramas

Na construção da nossa aplicação PokémonApp, a visualização e compreensão das relações entre entidades e o comportamento dinâmico de objetos foram fulcrais. Para atender essas necessidades, utilizamos dois tipos essenciais de diagramas: o Diagrama de Entidade e Relacionamento (DER) e o Diagrama de Estados.

3.1 DER – Diagrama de Entidade-Relacionamento

Os Diagramas de Entidade e Relacionamento (DER) são ferramentas fundamentais na modelação de bases de dados relacionais. Eles oferecem uma visão estruturada e visual das entidades que compõem um sistema, destacando as suas interações e dependências.

Destacamos através da imagem abaixo a estrutura da nossa base de dados:

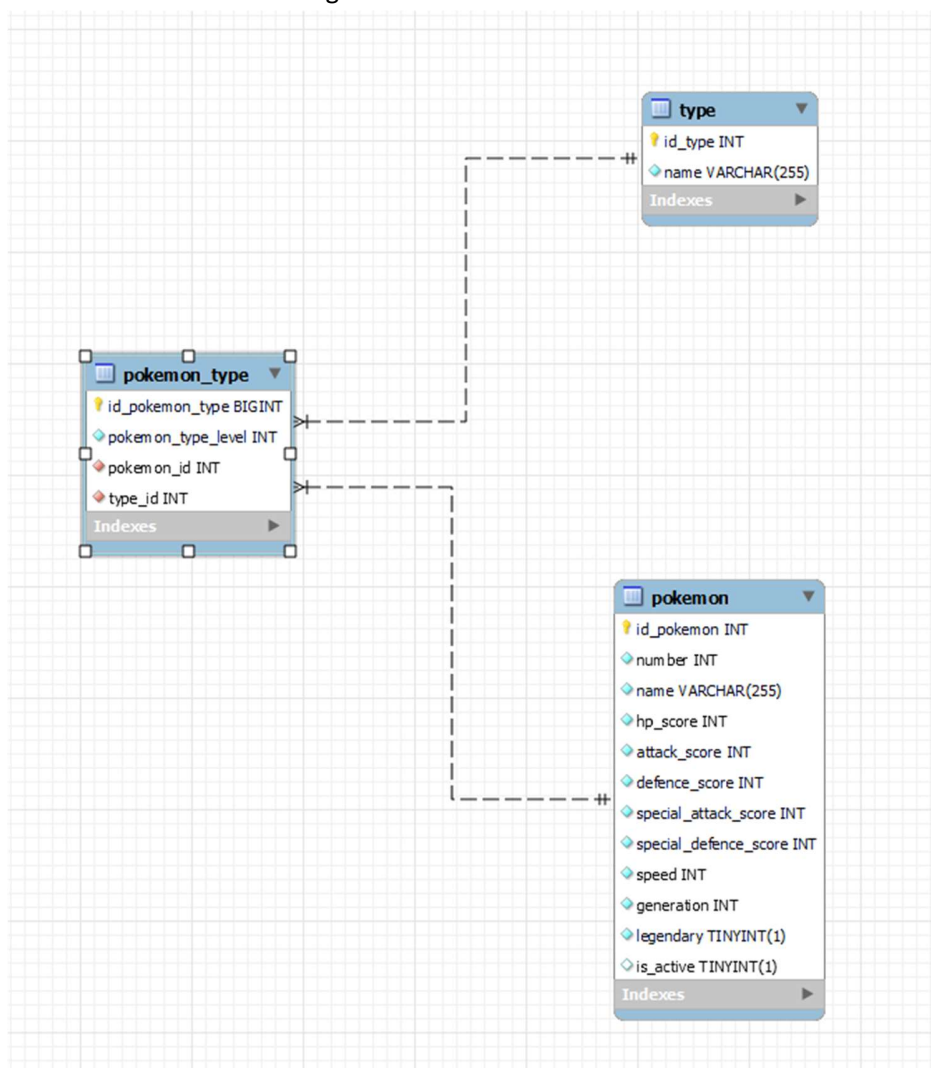


Figura 1 - Diagrama de Entidade e Relacionamento



3.2 Diagrama de Estados

O Diagrama de Estados não apenas facilita a compreensão do ciclo de vida dos Pokémons na PokémonApp, mas também serve como guia fundamental para o desenvolvimento de funcionalidades relacionadas. Ao explorar as nuances de cada estado e evento, revelamos a complexidade e a beleza por trás de como essas criaturas interagem em nosso ambiente digital.

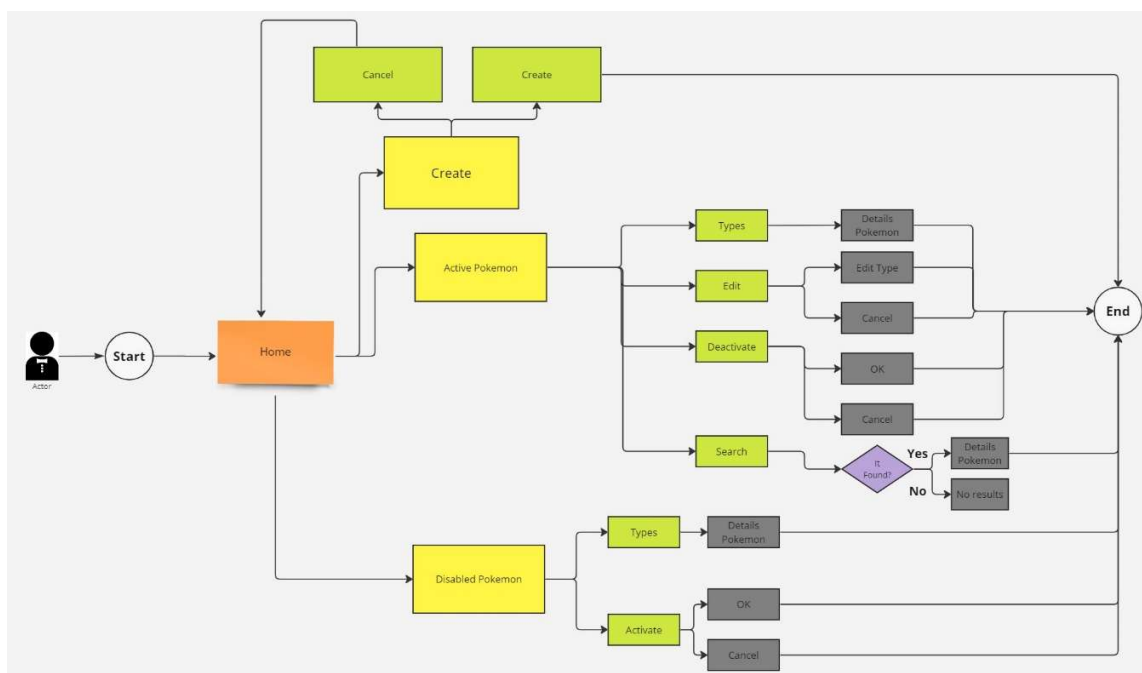


Figura 2 - Diagrama de Estados

4. Funcionalidades

A PokémonApp foi desenvolvida para oferecer aos utilizadores uma experiência completa e interativa na gestão de informações sobre Pokémons. As funcionalidades principais abrangem:

Listagem de Pokémons: Os utilizadores podem explorar uma lista abrangente de todos os Pokémons registados na aplicação, proporcionando uma visão geral e acessível de todas as criaturas. Existem duas opções de listagem, a dos Pokémons que estão ativos e a listagem em que os Pokémons estão desativados.

Criação de Novos Pokémon: A capacidade de adicionar novos Pokémon à aplicação é uma funcionalidade fundamental. Os utilizadores podem contribuir para a expansão da base de dados,



inserindo informações sobre novas criaturas e as suas características específicas, obedecendo, é claro, à estrutura criada.

Edição de Informações: A funcionalidade de edição permite aos utilizadores atualizarem as informações existentes sobre um Pokémon. Isso inclui a alteração de atributos como nome, habilidades, etc. A flexibilidade de editar dados existentes garante uma aplicação dinâmica e adaptável.

Desativação (Não Exclusão) de Pokémon: Em vez de excluir completamente um Pokémon, optamos por oferecer a opção de desativar Pokémon. Essa funcionalidade preserva o histórico de dados, mantendo os registos antigos intactos. Esta abordagem é valiosa para a integridade dos dados e controlo ao longo do tempo.



5. Conclusão

O desenvolvimento desta aplicação proporcionou ao grupo um mix de sensações, já que estas pequenas criaturas fizeram parte em algum momento das nossas vidas, seja na nossa infância ou na nossa vida adulta (através dos filhos). O facto da aplicação permitir a listagem, criação, edição e desativação cuidadosa de Pokémons, não atende apenas às necessidades funcionais, mas transcende para uma experiência, onde cada interação é uma celebração do universo Pokémon.

A aplicação não é apenas um código, é um convite para explorar, criar e preservar um mundo onde cada funcionalidade é um elo na corrente de experiências. Ao listar, criar, editar e desativar Pokémons, a PokémonApp cria um portal para a imaginação, a curiosidade e a comunidade. Através do processo de criação foi possível vincular o conteúdo lúdico a todo o conceito estrutural e de desenvolvimento de uma aplicação, cuja estrutura seja MVC.



6. Referências

<https://pt.wikipedia.org/wiki/MVC>

