
Documentação de Projeto

para o sistema

Trainer Card Maker

Versão 1.1

Projeto de sistema elaborado pelo(s) aluno(s)
-Victor Lucas Tornelli
-Guilherme Rodrigues
como parte da disciplina **Projeto de Software**.

29/05/2025

Tabela de Conteúdo

1. Introdução	1
2. Modelos de Usuário e Requisitos	1
2.1 Descrição de Atores	1
2.2 Modelo de Casos de Uso e Histórias de Usuários	1
2.3 Diagrama de Sequência do Sistema e Contrato de Operações	1
3. Modelos de Projeto	1
3.1 Arquitetura	1
3.2 Diagrama de Componentes e Implantação.	2
3.3 Diagrama de Classes	2
3.4 Diagramas de Sequência	2
3.5 Diagramas de Comunicação	2
3.6 Diagramas de Estados	2
4. Modelos de Dados	2

Histórico de Revisões

Nome	Data	Razões para Mudança	Versão
Documentação	26/05	Criação da primeira versão	1.0
Rev_1	29/05	Adicionado diagrama Entidade Relacionamento na sessão modelagem de dados	1.1

1. Introdução

Este documento agrega: 1) a elaboração e revisão de modelos de domínio e 2) modelos de projeto para o sistema **Trainer Card Maker**. A referência principal para a descrição geral do problema, domínio e requisitos do sistema é o documento de especificação que descreve a visão de domínio do sistema.

2. Modelos de Usuário e Requisitos

2.1 Descrição de Atores

Usuário Comum: Pessoa que acessa o sistema para criar, visualizar ou editar cartões de treinador Pokémon com base em informações personalizadas como nome, avatar, time de Pokémon, insígnias, etc.

2.2 Modelo de Casos de Uso

Casos de Uso (IDs UC)

UC-01: Criar Cartão

O usuário insere suas informações e gera um cartão de treinador.

UC-02: Atualizar Cartão

O sistema permite atualizar dados de um cartão salvo.

UC-03: Carregar Cartão

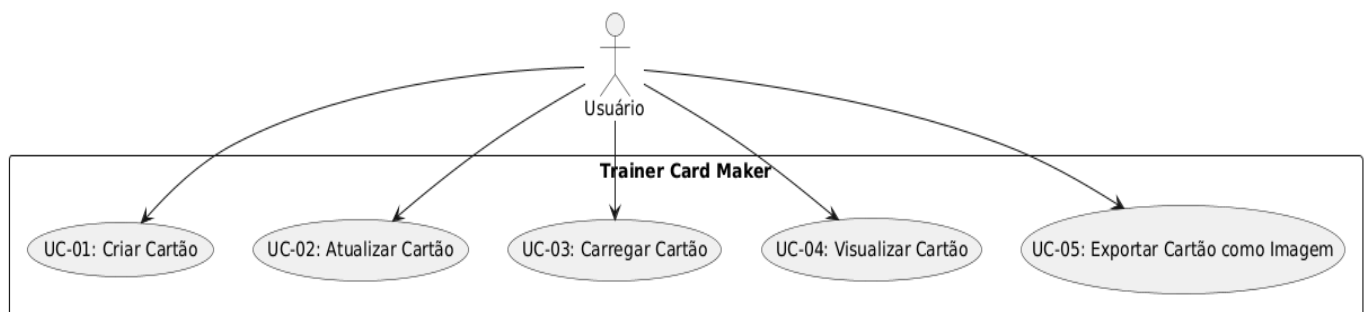
Permite carregar os dados do cartão para uso futuro.

UC-04: Visualizar Cartão

O sistema gera a visualização com animações personalizadas do cartão carregado.

UC-05: Exportar Cartão como Imagem

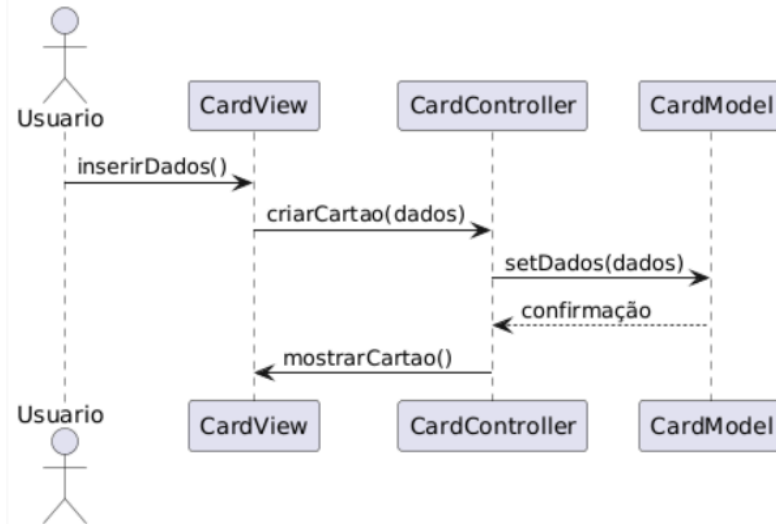
O sistema gera uma imagem PNG do cartão criado.



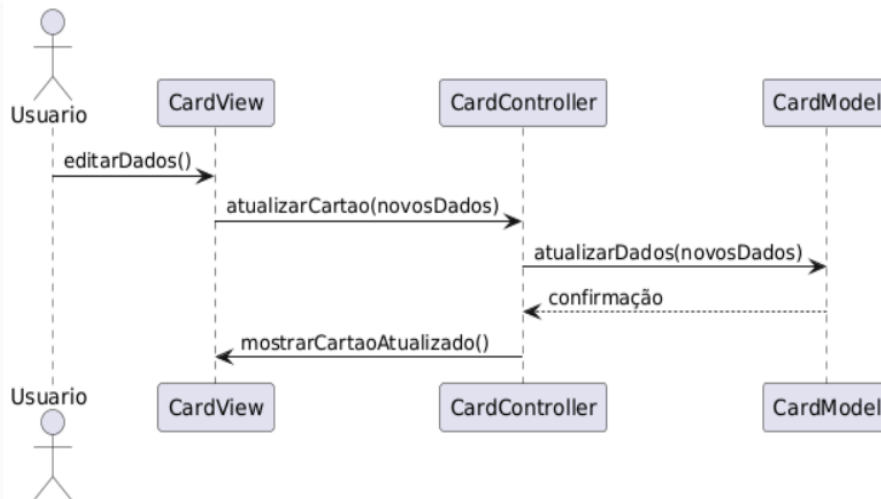
2.3 Diagrama de Sequência do Sistema

Nesta subseção é apresentado o diagrama de sequência do sistema de pelo menos, 3 Casos de Uso ou Histórias de Usuário descritos na Seção 2.3.

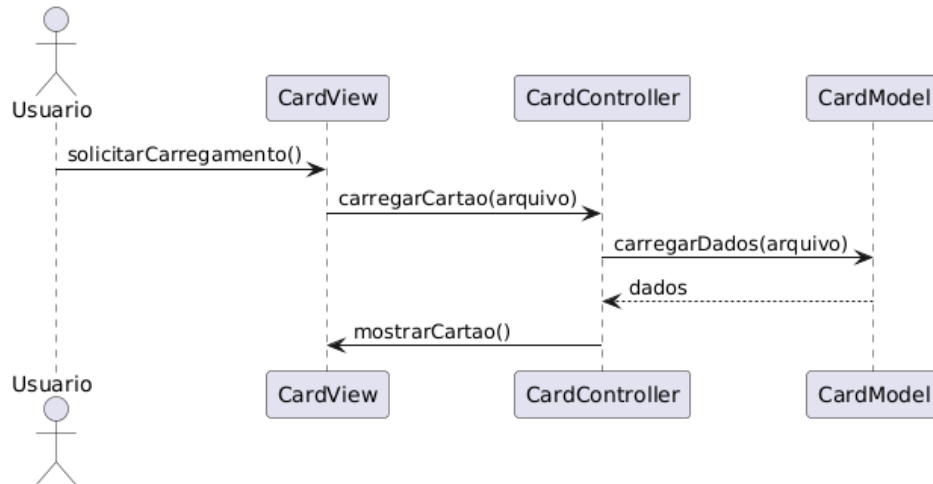
Contrato	Criar Cartão
Operação	criarCartao(dados)
Referências cruzadas	UC-01
Pré-condições	O usuário forneceu nome, imagem e time de Pokémon.
Pós-condições	Um novo cartão é criado e exibido na interface.



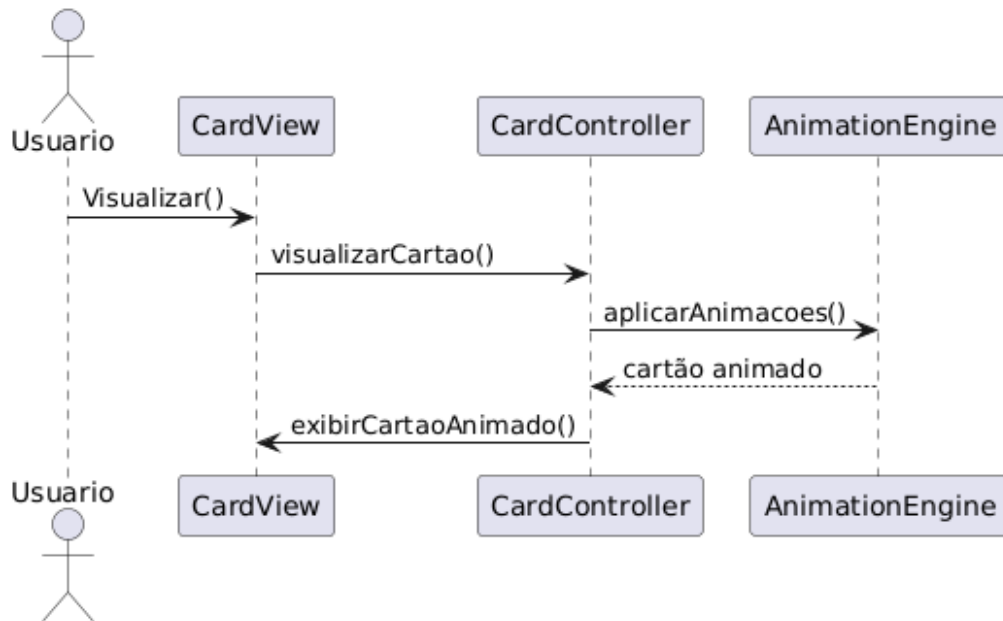
Contrato	Atualizar Cartão
Operação	atualizarCartao(novosDados)
Referências cruzadas	UC-02
Pré-condições	Um cartão previamente criado ou carregado está em memória.
Pós-condições	O cartão é atualizado com os novos dados.



Contrato	Carregar Cartão
Operação	carregarCartao(arquivo)
Referências cruzadas	UC-03
Pré-condições	O usuário selecionou um arquivo de cartão previamente salvo.
Pós-condições	Os dados do cartão são carregados e exibidos.



Contrato	Visualizar Cartão
Operação	visualizarCartao()
Referências cruzadas	UC-01
Pré-condições	O cartão está carregado e contém dados válidos.
Pós-condições	O cartão é exibido com animações personalizadas.



3. Modelos de Projeto

3.1 Arquitetura

O sistema segue uma arquitetura baseada no padrão **MVC (Model-View-Controller)**:

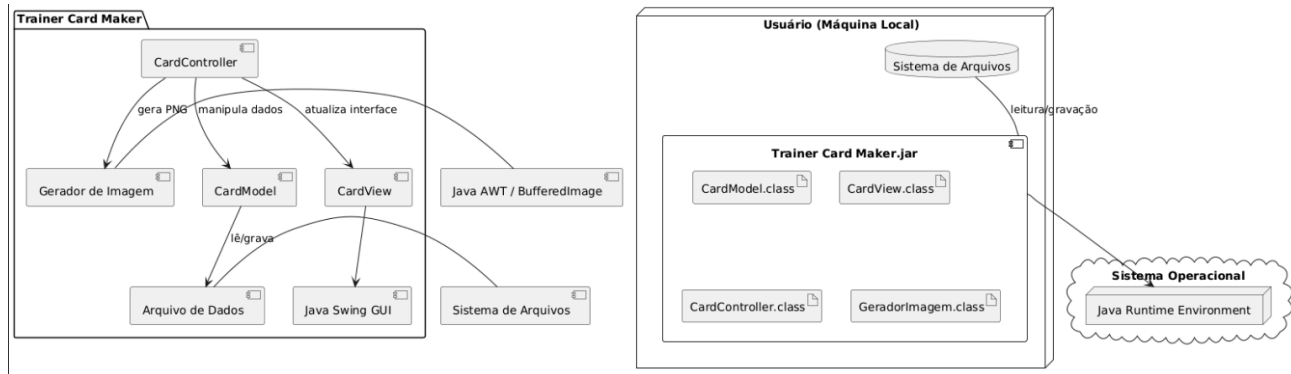
- **Model**: representa os dados do cartão.
- **View**: interface gráfica feita em Java Swing.
- **Controller**: manipula eventos, atualiza a View e o Model.

3.2

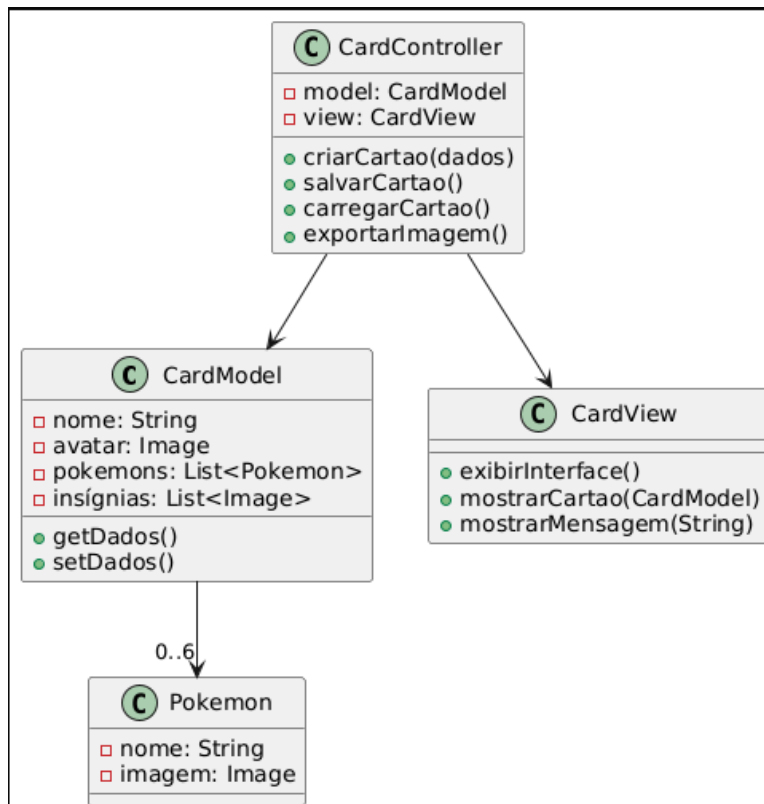
```
## Estrutura do Projeto
~~~

PokemonTrainerCard/
|-- build.xml
|-- manifest.mf
|-- nbproject/           # Configurações do NetBeans
|-- src/
|   |-- pokemontrainercard/
|       |-- Models
|       |-- Views (.forms)
|       |-- Controllers
|
|-- Database
|-- imagens/, pokesprites/, trainers/   # Recursos visuais
|
~~~
```

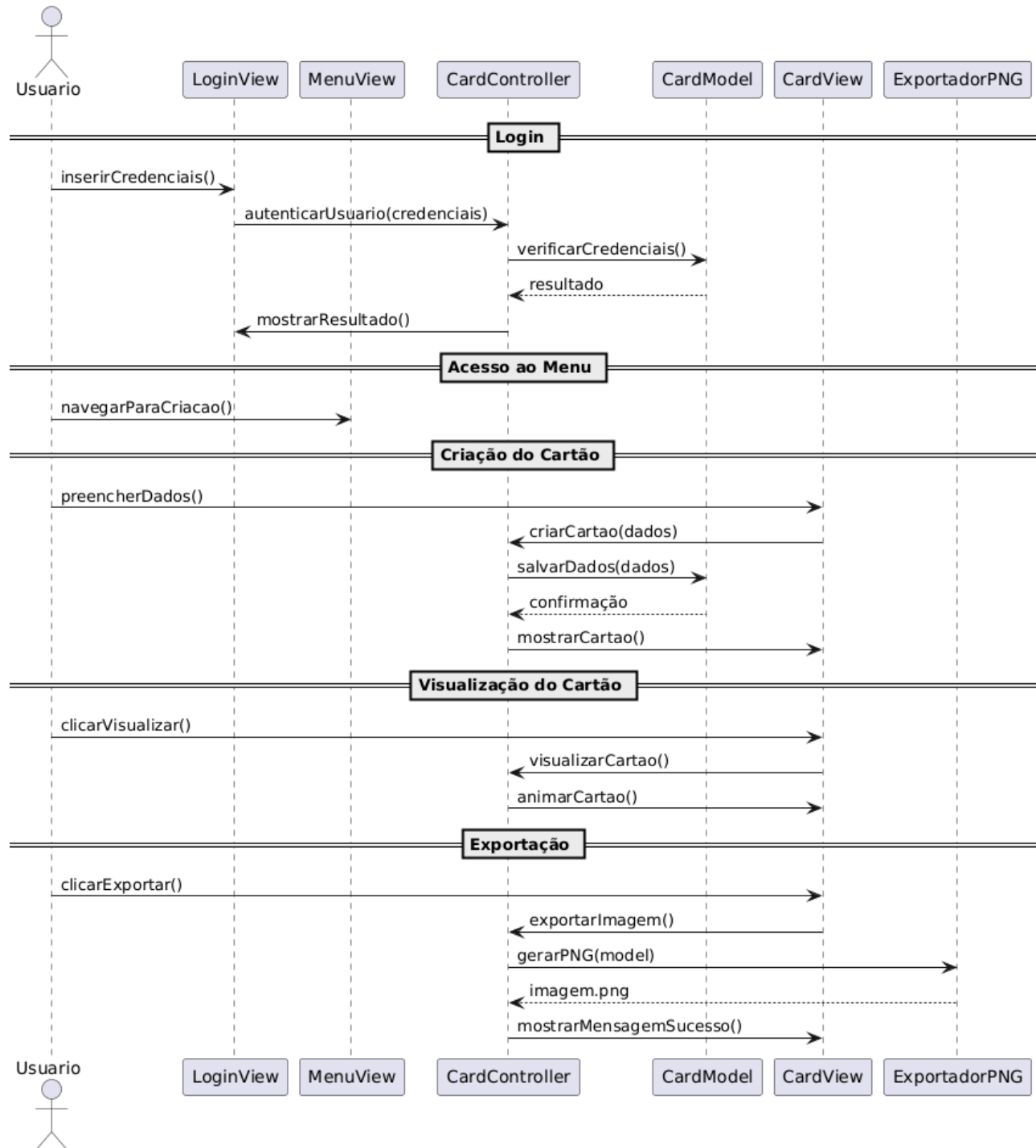
3.3 Diagrama de Componentes e Implantação.



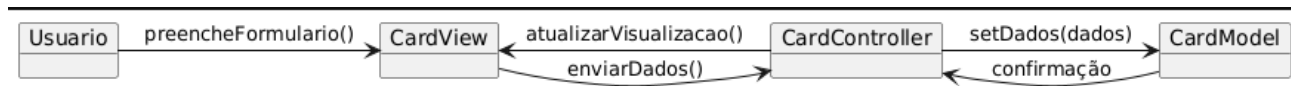
3.4 Diagrama de Classes



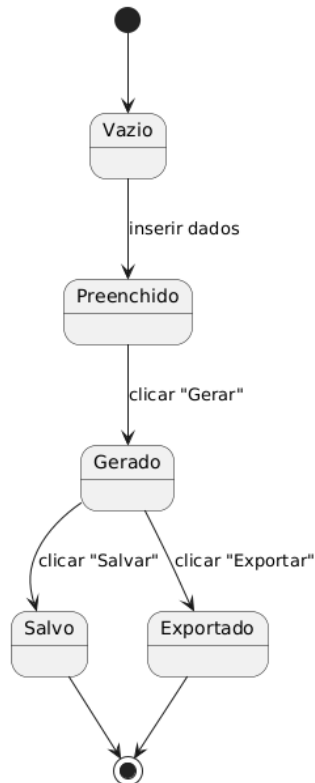
3.5 Diagrama de Sequencia



3.6 Diagramas de Comunicação

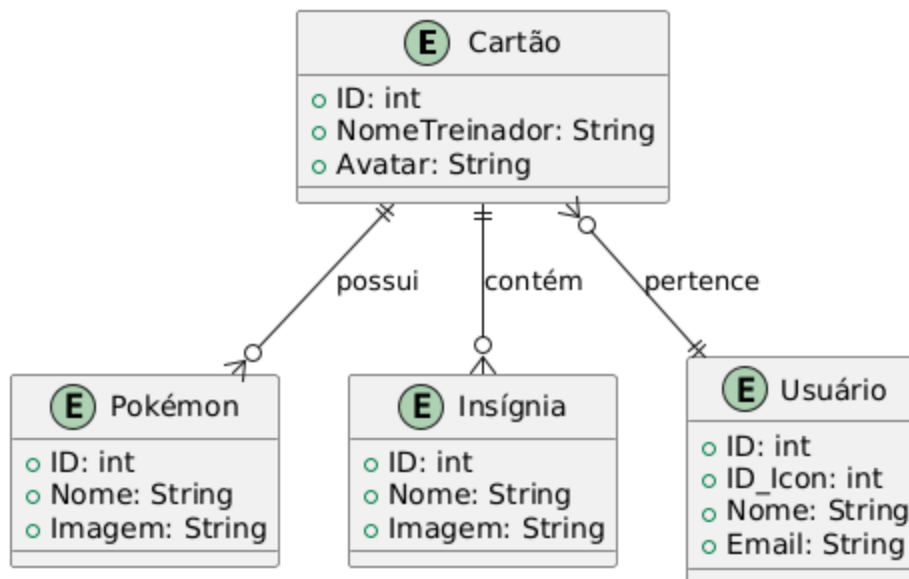


3.7 Diagramas de Estados



4. Modelos de Dados

O sistema *Trainer Card Maker* não utiliza um banco de dados relacional tradicional, mas armazena os dados em **estruturas de objetos** e arquivos locais, atualmente utilizando SQLite para armazenamento. A estrutura de dados do sistema representa um modelo de cartão de treinador que pode ser exportado como imagem.



Entidades principais:

- **Cartão:** Entidade principal. Representa o cartão de treinador.
- **Usuário:** Dados inseridos manualmente (nome, avatar).
- **Pokémon:** Até 6 Pokémon associados ao cartão.
- **Insígnias:** Lista de insígnias coletadas pelo treinador.

Relacionamentos:

- Um **Cartão** pertence a um **Usuário**.
- Um **Cartão** contém até 6 **Pokémon**.
- Um **Cartão** pode conter várias **Insígnias**.

Modelo de Armazenamento

- Os dados do cartão podem ser **serializados em arquivos locais** ou estruturados como objetos em memória.
- O sistema utiliza objetos Java para representar as entidades, que são manipuladas por controladores e exibidas via interface gráfica.