



Documentação AutoStudy

Matheus Leandro

March 13, 2024

1 Introdução

1.1 Objetivo

Este documento tem como objetivo documentar todas as tecnologias, processos e decisões tomadas para o desenvolvimento de um simples fullstack project, como uma forma de auto estudo durante a primeira semana do NPI.

1.2 Descrição da aplicação

A aplicação será um local para o armazenamento de decks, onde os usuários poderão acessar uma lista contendo os decks já salvos, assim como criar novos decks e editar os já existentes.

2 Requisitos

2.1 Técnicas de elicitação

Por ser um projeto simples a única técnica utilizada pelos desenvolvedores foi a de brainstorm. Onde foi decidido as principais funcionalidades de acordo com as regras do projeto definidas pelo product owner.

2.2 Regras do projeto

- A aplicação deverá ser full-stack;
- A aplicação deverá ser capaz de realizar no mínimo um tipo de CRUD;
- O prazo de produção deverá ser de 3 semanas;
- Durante o desenvolvimento, deverá ser realizada dailys onde será compartilhado a evolução do desenvolvedor;

2.3 Requisitos funcionais

- O usuário deve ser capaz de acessar uma lista de decks;
- O usuário deve ser capaz de adicionar novos decks a lista de decks;
- O usuário deverá ser capaz de visualizar as cartas contidas em um deck;
- O usuário deverá ser capaz de editar as informações de decks já existentes;

2.4 Requisitos não funcionais

- A aplicação deve ter uma porcentagem de sucesso ao editar ou criar um deck de 95%;
- A aplicação deverá ter uma interface intuitiva;
- Os dados salvos do usuário devem permanecer após ele deixar a aplicação;

2.5 Tecnologias utilizadas

Foi definido que o desenvolvedor deverá criar uma aplicação full-stack. Portanto, para isso deverá usar:

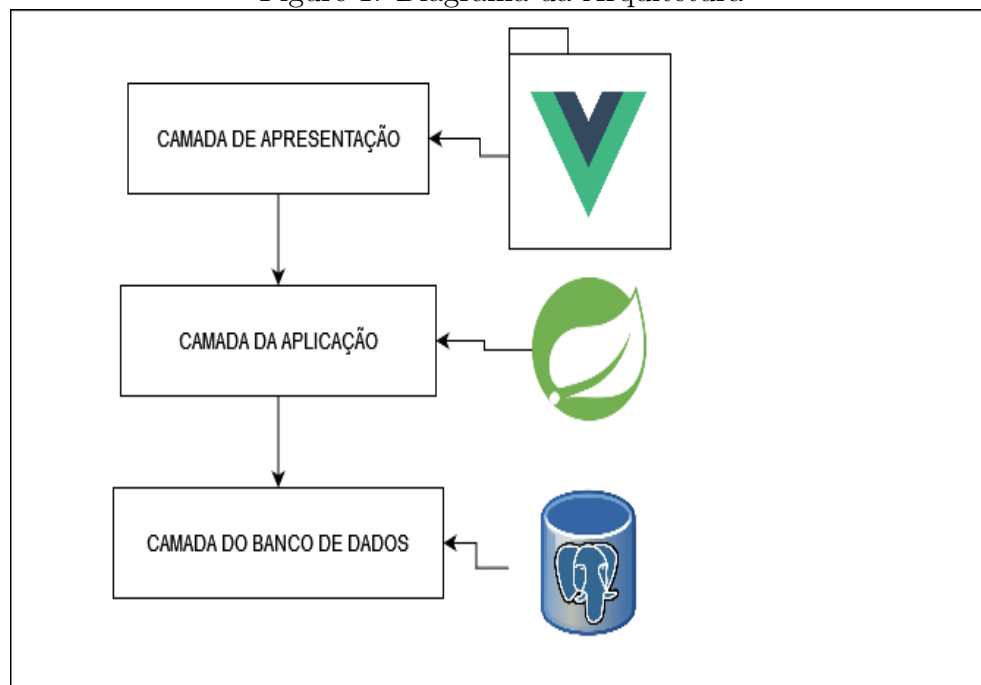
- Vue.js com Vuetify para o front;
- Spring boot para o back com as dependências de jpa assim como de drive para se conectar ao banco de dados;
- Postgresql e Pgadmin;
- De preferência rodando o banco em um docker.

3 Arquitetura

3.1 Arquitetura de Camadas

Foi-se escolhida essa arquitetura por ser a mais recomendada quando se trata de aplicações simples, ou até mesmo aplicações que começaram sem uma arquitetura definida. Portanto, assim como na arquitetura o planejamento de desenvolvimento é primeiro o front, depois o back focando no spring e por fim o banco de dados.

Figure 1: Diagrama da Arquitetura



4 Diagrama de Classes

4.1 Arquitetura de Camadas

A aplicação conterá duas entidades, deck e carta, cada deck terá várias cartas mas cada carta só estará em um deck, pois pra ser adicionada ao deck a carta

precisa ser criada pelo usuário e ela é removida do banco caso seja deletada de um deck, assim, não afetando outros decks.

Figure 2: Diagrama da Arquitetura

