

Relatório do Projeto Final de Programação Orientada a Objetos

Eduardo Garcia Mendonça
Hector Fernando Tarasiuk
Jair Eugenio Ferreira
João Pedro Beltrão Cortes

Índice

1	Um site de geração de currículos	1
2	As partes do projeto	2
2.1	frontend	2
2.2	backend	2
2.3	simplesDB	3
2.3.1	sdb	4
2.3.2	aux	4
2.3.3	Curriculo	4
2.3.4	Recebendo	4
2.3.5	Enviando	4

1 Um site de geração de currículos

link do projeto no github: <https://github.com/garcia-ufg/trabalhofinaltudojunto>

O projeto desenvolvido foi um site de geração de currículos. O site recebe dados do usuário como o nome, CPF, telefone, endereço, email, habilidades, formação acadêmica, experiência profissional e gera um currículo correspondente.

O projeto tem três partes: O frontend, um aplicativo REACT que é a interface que um usuário utilizaria, o backend, uma API RESTful feita com spring-boot que lidaria com requests GET e POST feitos pelo frontend e a biblioteca simplesDB, que cuida do banco de dados de currículos gerados.

2 As partes do projeto



Além deste documento de documentação, todas as partes do projeto tem documentação própria disponível em seus respectivos diretórios. A documentação de cada parte inclui a especificação de requisitos da parte, uma descrição de suas funcionalidades, casos de uso e a documentação de seu código fonte.

Os arquivos de documentação de cada parte são nomeados:

- frontend: frontend_funcionalidades.pdf
- backend: backend_funcionalidades.pdf
- simplesDB: simplesDB_funcionalidades.pdf

2.1 frontend

Um aplicativo react, no código fonte no github, está na pasta "frontend". para rodar o frontend, execute os comandos:

```
npm install  
npm run dev
```

A tela principal do front end é um formulário onde um usuário preenche seus dados, depois de feito isso, confirmar leva pra uma prévia do currículo gerado, nesta prévia é possível imprimir um pdf que seguiria o modelo do currículo.

O frontend não está completo e não está integrado com o backend devido a dificuldades no desenvolvimento com javascript e react.

2.2 backend

Um programa em java feito utilizando o framework spring-boot. o backend é uma API RESTful que providencia os seguintes endpoints:

- /nov (POST): recebe dados de um currículo para gravar no disco.
- /cur (GET): envia os dados de um currículo para o frontend.

- /edit (POST): edita um currículo já existente com novos dados recebidos.
- /del (POST): deleta um currículo no banco de dados caso a autenticação fornecida pelo frontend esteja correta.
- /auth (POST): autentica um usuário.
- /verU (GET): verifica se um usuário já existe no banco de dados.

Na falta da integração entre backend e frontend, o funcionamento correto do backend foi testado manualmente por meio do comando curl. Pelo curl é possível mandar de forma manual requests POST e GET para uma API e averiguar o resultado.

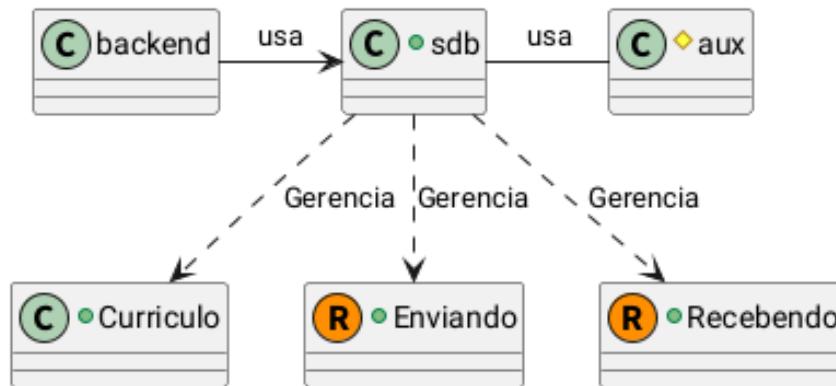
O arquivo curriculoBackend.jar na raiz do diretório é o executável do backend. para rodar, execute o comando na raiz do diretório:

```
java -jar curriculoBackend.jar
```

Para informações sobre como compilar o backend, ver sua documentação detalhada no diretório backend.

2.3 simplesDB

Uma biblioteca feita em java para gerenciar um banco de dados de currículos. A parte mais complexa do código, utiliza o padrão de projeto facade para abstrair toda a complexidade de ler e escrever arquivos no disco e gerenciar os objetos do tipo Curriculo. É composta pelas seguintes classes: sdb, aux, Curriculo. Possui os seguintes records: Enviando, Recebendo. Segue agora uma breve explicação de seu funcionamento. Para uma documentação detalhada, sistemática e extensiva das classes, métodos, atributos e funcionamento, ver o seu arquivo de documentação própria no diretório simplesDB.



2.3.1 sdb

sdb é a interface de interação com o banco de dados. Providencia métodos para converter entre objetos do tipo Curriculo e JSON. currículos são salvos no disco como arquivos JSON, são nomeados por um hash calculado a partir do CPF. Acessar um currículo consiste em calcular seu hash a partir do CPF e então abrir o arquivo no disco com o nome correspondente.

2.3.2 aux

aux define uma série de funções auxiliares utilizadas por sdb e Curriculo.

2.3.3 Curriculo

Define os atributos e métodos de um objeto do tipo currículo.

2.3.4 Recebendo

este record define os dados que o backend esperaria receber do frontend.

2.3.5 Enviando

este record define os dados que o frontend esperaria receber do frontend.