

## Contents

<b>1</b>	<b>mas afinal, o que o simplesDB vai fazer?</b>	<b>1</b>
<b>2</b>	<b>Das funcionalidades de simplesDB</b>	<b>1</b>
2.1	definir um diretório para o armazenamento de currículos e fotos	1
2.2	criar um currículo a partir dos dados fornecidos . . . . .	2
2.3	salvar um currículo no banco e dados como um JSON . . . . .	2
2.4	carrega um currículo do banco de dados armazenado como JSON . . . . .	2
2.5	Verifica se um currículo existe no banco de dados (por CPF ou por ID) . . . . .	2
2.6	Editar os dados de um currículo já existente . . . . .	2
2.7	Deletar um currículo já existente . . . . .	2
2.8	Verifica se o CPF e senha fornecidos são válidos . . . . .	3

## 1 mas afinal, o que o simplesDB vai fazer?

- definir um diretório para o armazenamento de currículos e fotos.
- criar um currículo a partir dos dados fornecidos
- salvar um currículo no banco de dados como um JSON
- carrega um currículo do banco de dados armazenado como JSON
- Verifica se um currículo existe no banco de dados (por CPF ou por ID)
- Editar os dados de um currículo já existente
- Deletar um currículo já existente
- Verifica se o CPF e senha fornecidos são válidos

## 2 Das funcionalidades de simplesDB

### 2.1 definir um diretório para o armazenamento de currículos e fotos

as variáveis `DIR_DADOS` e `DIR_FOTOS` armazenam a localização dos diretórios de dados e fotos respectivamente.

Métodos: `setFotos(String dir)`, `setDados(String dir)`;

## **2.2 criar um currículo a partir dos dados fornecidos**

um método que chama o construtor de classe da classe currículo  
método: novoUsuario(<parametros>)

## **2.3 salvar um currículo no banco e dados como um JSON**

simplesDB usa a biblioteca GSON do google pra serializar um objeto do tipo currículo como um JSON. GSON foi escolhido devido a ser mais fácil de utilizar do que a utilização de JSON nativa do java.

Método: salvaCurriculo(Curriculo cur)

## **2.4 carrega um currículo do banco de dados armazenado como JSON**

Isso pode ser realizado por meio do CPF ou pelo ID de usuário. Caso o currículo exista no banco de dados, a biblioteca GSON é utilizada para ler o currículo e inicializar um objeto currículo, que será retornado por uma função.

Métodos: carregaCurriculoCpf(String cpf), carregaCurriculoId(String id)

## **2.5 Verifica se um currículo existe no banco de dados (por CPF ou por ID)**

as funções noDiscoCPF e noDiscoID são utilizadas para tal verificação, elas verificam isto por meio da classe file do java. Retornam true se existe, false caso contrário

Da pra usar as próprias funções carregaCurriculo. Se o currículo não existe elas retornam um null.

## **2.6 Editar os dados de um currículo já existente**

Caso um currículo exista, pode-se chamar uma função para editar seus atributos, o CPF e a senha não são editáveis no entanto.

Método: editaUsuario (<parametros>)

## **2.7 Deletar um currículo já existente**

Uma função que recebe o CPF e a senha, verifica se existe currículo associado ao CPF no banco de dados e então deleta ele caso a senha fornecida esteja correta.

Método: delUsuario(String cpf, String senha)

## **2.8 Verifica se o CPF e senha fornecidos são válidos**

Uma função que recebe CPF e Senha e retorna verdadeiro caso o currículo exista, falso caso o contrário

Método: `autenticaUsuario(String cpf, String senha)`