

Documentação do projeto Banco de Vagas.

Esta é uma amostra descritiva do projeto contendo o seu corpo técnico resumido para o funcionamento, manutenção e flexibilização do serviço.

SENAI SÃO PAULO DE INFORMÁTICA

2020

AUTOR

New Solution Group

PROFESSORES

Priscila Medeiro

Caíque Zanetti

TURMA

Técnico em Desenvolvimento de Sistemas (manhã)

Antes de prosseguir...

As informações contidas neste documento se referem a parte lógica da aplicação.

Todos os processos essenciais para o funcionamento do sistema estarão aqui expostos.

O objetivo deste documento é facilitar o trabalho do receptor, aquele que atua no tratamento dos dados e interage diretamente com a plataforma.

Neste sistema, não há qualquer tipo de interface e/ou visual, todo o processo se limita às variadas requisições web apenas (tracionamento de dados para exibição, edição e cadastro).

Os dados são armazenados pelo banco, a plataforma é responsável somente pelas funcionalidades, tirando o armazenamento de alguns arquivos estáticos em pastas (imagens).

Todos os tipos de resposta do protocolo HTTP estão padronizados, assim como os parâmetros de retorno para o usuário (não todos).

A plataforma foi feita em conjunto pelo grupo, e posteriormente revisada para o uso.

Este documento se manterá atualizado conforme a aplicação for sendo visionada, seja com incrementações ou correções.

Sumário

Equipe	4
Introdução	5
Metodologias.....	5
Tecnologias utilizadas.....	6
Sobre as requisições da API.....	7
Dicionário das rotas	8
LOGIN	9
CANDIDATO.....	10
EMPRESA	13
ADMINISTRADOR	16

Equipe

Ana Laura Feltrim

Eduardo Luiz do Prado

Gabriel de Souza Moreira

Lucas Simplício de Oliveira

Luan de Sousa Barbosa

Introdução

A plataforma do **BANCO DE VAGAS DO SENAI** é uma solução interna desenvolvida pela equipe **NEW SOLUTION GROUP**, da turma de Técnico em Desenvolvimento de Sistemas, no período da manhã, escola **SENAI DE INFORMÁTICA, SANTA CECÍLIA**.

A premissa foi de fazer uma plataforma de divulgação de vagas para os alunos do SENAI, inclui também um sistema de monitoramento do prazo de estágios ativos.

A solução se manteve realista ao que foi pedido. Todo o processo de desenvolvimento da plataforma ocorreu sob a orientação e mentoria dos professores. Todo o planejamento ficou a par da própria equipe.

Metodologias

No processo de desenvolvimento, foi preferência da equipe utilizar a metodologia ágil Scrum e Cambam. A plataforma utilizada para a organização foi o **Microsoft AZURE DEVOPS**.

As ferramentas utilizadas para o versionamento de código são o GIT, e o sistema de repositórios internos da plataforma **AZURE DEVOPS** para o armazenamento.

Tecnologias utilizadas

Ferramentas do ambiente de desenvolvimento

Visual Studio 2019

Visual Studio Code

Microsoft SQL Server Management

Arquitetura

API - (Repositórios, Controladores e Modelo) – Migração Back End – Database.

Frameworks

ASP .NET Core API – Versão 3.1 (C#) – Framework de desenvolvimento de aplicações.

Entity Framework Core (ORM) – Desenho de mapa relacional do banco de dados.

Banco de dados

Microsoft SQL Server 2017

DOCUMENTAÇÃO INTERNA

Swagger (API)

Sobre as requisições da API

- As requisições entre as pontas de acesso se limitam ao GET, PUT e POST.
- Alguns retornos possuem mensagens de erro padronizadas e parâmetros fixos.
- Não há uso do protocolo HTTPS, todas as requisições deverão de ser feitas pelo protocolo padrão HTTP.
- Houve o uso de formulários para a inserção de arquivos estáticos para a API.
- Todas as requisições estão assíncronas.
- Todas as requisições estão pré-configuradas em JSON.
- Uso de autenticação JWTBearer. O token válido possui trinta (30) minutos de duração.
- Uso do Swagger, com todas as requisições de controle devidamente documentadas, o endereço é: <http://servidor/swagger>.

Dicionário das rotas

*todas as requisições estão padronizadas em **JSON**.

*todas as requisições estão assíncronas.

[**“HTTP://localhost:5000”**] – Endereço local da aplicação

[**GET**] – Requisição de busca

[**POST**] – Requisição de inserção

[**PUT**] – Requisição de alteração

[**Authorize**] [Cargo] – Autorização designada a uma rota

[**Anônimo**] – Qualquer usuário acessa a rota

[**Body**] – Se a requisição vem do corpo (objeto)

[**FormData**] – Se a requisição vem através de um form/data (objeto)

[**Url**] – Se a requisição vem através do endereço de URL (parâmetro)

[**200**] – Ok

[**201**] - Created

[**400**] - BadRequest

[**403**] – Forbidden

[**404**] - NotFound

Parâmetros de retorno padrão

{ msgerro = “mensagem de erro” }

{ msgsucesso = “mensagem de sucesso” }

LOGIN

Login do usuário

[Anônimo] [POST] [“HTTP://localhost:5000/login”] [Body]

String Email

String Senha

[200] – { token de acesso }

[404] – Usuário não encontrado

Observação:

O token contém três tipos de usuários, são eles:

“Administrador”,

“Candidato”,

“Empresa”

O token identifica o tipo de usuário e coloca no atributo ROLE do token.

Os três atributos do token.

IdUsuario,

ROLE,

Email

CANDIDATO

Cadastrar Candidato

[Anônimo] [POST] [“HTTP://localhost:5000/candidato”] [Body]

string Email

string Senha

string NomeAluno

DateTime? DataNascimento (pode ser nulo)

string Cpf

[201] – Cadastro realizado com sucesso

[403] – Email existente

Observação:

O currículo é cadastrado automaticamente.

Candidatar à vaga

[Authorize] [Candidato] [POST] [“HTTP://localhost:5000/candidatura”] [Body]

Int idvaga

[200] – Vaga aplicada com sucesso

[403] – Você já se candidatou à essa vaga

Minhas candidaturas

[Authorize] [Candidato] [GET] [“HTTP://localhost:5000/candidato/MinhasCandidaturas”]

[200]

Observações: Contém (IdCandidatoNavigation, IdVagaNavigation)

Exibir currículo

[Authorize] [Candidato] [GET] [“HTTP://localhost:5000/candidato/curriculo”]

[200]

Observações: Contém (IdCandidatoNavigation)

Atualizar currículo

[Authorize] [Candidato] [PUT] [“HTTP://localhost:5000/candidato/curriculo”] [Body]

bool Cursando

string? Linguas

string? Descricao

string? Escolaridade

string? CursosFormacoes

string? PalavraChave

[200] – Currículo editado com sucesso.

Editar informações

[Authorize] [Candidato] [PUT] [“HTTP://localhost:5000/candidato”] [Body]

string? NomeAluno

DateTime? DataNascimento

string? Cpf

[200] – Editado com sucesso.

Editar foto

[Authorize] [Candidato] [PUT] [“HTTP://localhost:5000/candidato/alterarFoto”] [FormData]

IFormFile(“foto”)

[200] – Imagem de perfil alterada.

[400] – Ocorreu um erro na hora de atualizar, tente novamente mais tarde.

Listar vagas

[Anônimo] [GET] [“HTTP://localhost:5000/vaga”]

[200]

Observações: Contém (IdTipoVagaNavigation, IdEmpresaNavigation)

Buscar vaga

[Anônimo] [GET] [“HTTP://localhost:5000/vaga/pesquisa/{string titulo}”] [Url]

[200]

Listar vagas pelo tipo de contrato

[Authorize] [Candidato] [GET] [“HTTP://localhost:5000/vaga/tipoContrato/{int id}”] [Url]

[200]

Listar vagas pelo tipo de contrato

[Authorize] [Candidato] [GET] [“HTTP://localhost:5000/vaga/tipoCurso/{int id}”] [Url]

[200]

Vaga por id

[Authorize] [Candidato] [GET] [“HTTP://localhost:5000/vaga/{int id}”] [Url]

[200]

[404]

EMPRESA

Cadastrar empresa

[Anônimo] [POST] [“[HTTP://localhost:5000/empresa](http://localhost:5000/empresa)”] [Body]

string Email

string Senha

string RazaoSocial

string NomeFantasia

string Cnae

string Cnpj

[200] – Cadastro realizado com sucesso

[403] – Email existente

[403] – CNPJ já está cadastrado

Ver perfil

[Authorize] [Empresa] [GET] [“[HTTP://localhost:5000/empresa/perfil](http://localhost:5000/empresa/perfil)”]

[200]

Editar perfil

[Authorize] [Empresa] [PUT] [“[HTTP://localhost:5000/empresa/perfil](http://localhost:5000/empresa/perfil)”] [Body]

string RazaoSocial

string NomeFantasia

string Cnae

string Cnpj

[200] – Editado com sucesso.

Cadastrar vaga

[Authorize] [Empresa] [POST] [“HTTP://localhost:5000/empresa/cadastrarVaga”] [Body]

string Titulo

string Descricao

string PalavraChave

string Endereco

int Salario

int IdTipoCurso

int IdTipoVaga

[201] – Vaga criada com sucesso

Desativar vaga

[Authorize] [Empresa] [PATCH] [“HTTP://localhost:5000/empresa/vagas/desativar/{int id}”] [Url]

[200]

[400]

Listar vagas da empresa

[Authorize] [Empresa] [GET] [“HTTP://localhost:5000/empresa/vagas”] [Url]

[200]

Listar candidatos da vaga

[Authorize] [Empresa] [GET] [“HTTP://localhost:5000/empresa/vagas/{int id}”] [Url]

[200]

[403] – Não é permitido listar candidatos da vaga de outra empresa

Escolher um candidato da vaga

[Authorize] [Empresa] [POST] [“HTTP://localhost:5000/empresa/escolherCandidato/{int id}”] [Url]

[200] – O candidato foi escolhido com sucesso, foi enviado um E-Mail para ele.

[400] – Candidatura inválida

[403] – O candidato em questão já foi escolhido

Editar foto

[Authorize] [Empresa] [PUT] [“HTTP://localhost:5000/empresa/perfil/imagem”] [FormData]

IFormFile(“foto”)

[200] – Editado com sucesso.

[400] – Ocorreu um erro ao atualizar, tente novamente mais tarde.

Banco de talentos (pesquisar candidatos e visualizar perfil)

[Anônimo] [GET] [“HTTP://localhost:5000/BancoDeTalentos/{string busca}”] [Url]

[200]

ADMINISTRADOR

Listar estágios

[Authorize] [Administrador] [GET] [“HTTP://localhost:5000/administrador/EstagiosComPrazos”]

[200]

Estágio por id

[Authorize] [Administrador] [GET] [“HTTP://localhost:5000/administrador/EstagiosComPrazos/{int id}”]
[Url]

[200]

[400] – A vaga de estágio não foi encontrada

Cadastrar administrador

[Authorize] [Administrador] [POST] [“HTTP://localhost:5000/administrador/cadastrarAdministrador”]
[Body]

string Email

string Senha

[201] – Um administrador foi cadastrado com sucesso no sistema

[400] – Já temos um usuário cadastrado com este E-Mail no sistema

Adicionar prazo de 3 meses estagio

[Authorize] [Administrador] [POST] [“HTTP://localhost:5000/administrador/ProrrogarPrazo/{int id}”]
[Url]

[200] – Um novo prazo de 3 meses foi adicionado com sucesso

[403] – A vaga de estágio não foi encontrada

[403] – Ocorreu um erro, tente novamente mais tarde

