A close up of a logo

Description automatically generated

**Relatório**

**Comunicação de Dados**

**Alunos**

João Apresentação (21152)

Pedro Simões (21140)

Augusto Pereira (21136)

**Docente**

Miguel Lopes

**Licenciatura Engenharia Sistemas Informáticos**

Barcelos, junho de 2022

Indice

[2. Resumo 1](#_Toc105080821)

[3. Objetivos **Erro! Marcador não definido.**](#_Toc105080822)

[4. Base de dados 2](#_Toc105080823)

[5. Back-End 3](#_Toc105080824)

[6. Front-End 21](#_Toc105080825)

[7. Bibliografia 21](#_Toc105080826)

[8. Conclusão 21](#_Toc105080827)

# Índice de ilustrações

**Não foi encontrada nenhuma entrada do índice de ilustrações.**

# Resumo

Em suma, o trabalho prático da cadeira de Comunicação de Dados teve como propósito a montagem de uma API capaz de correr serviços web com determinadas funcionalidades. Era pretendido a implementação para o problema do escalonamento FJSSP (flexible job shop problem).

De uma forma mais descritiva, a existência de várias simulações, constituídas por Jobs, que por sua vez constituídos por operações, e cada operação será desempenhada por uma máquina com uma determinada duração. O escalonamento tem por objetivo encontrar um plano de produção mais eficiente ao nível de duração.

Esta aplicação oferece de uma forma geral os seguintes serviços:

* Gestão de Simulações;
* Tabela de Produção;
* Plano de Produção;
  + Automatizado;
  + Manual;
* Gestão de Utilizadores;

Este programa oferece a possibilidade de construção de simulações, á qual poderá se fazer download, simular manualmente e automaticamente, além de muitas outras funcionalidades.

Este terá duas categorias de permissões:

* Funcionário;
* Administrador;

O Administrador tem todas as funcionalidades iguais ás do Funcionário, mas este ainda pode gerir os funcionários e as máquinas.

Para a resolução da mesma recorreu-se ao uso:

* linguagem C# para o Back-End realizado no Visual Studio 19;
* Html, Css e TypeScript para o Front-End elaborado em Visual Studio Code;
* Base de dados em Microsoft SQL Server.
* Testes de request realizados em Postman;

# Base de dados

Para construção da base de dados, tal como o referido acima, construído em Microsoft SQL Server levou a seguinte estrutura:

Uma imagem com mesa

Descrição gerada automaticamente

Dentre as 3 tabelas existe a do utilizador e da máquina que são independentes da existência de dados na tabela das conexões. A tabela das conexões representa cada vez que um utilizador cria uma simulação com respetivas associações. Esta tabela será usada para a grande maioria das funcionalidades e apresentações.

# Back-End

Para a criação do Back-End realizado em Visual Studio 19, foi criado uma ligação á base de dados através de

Este contem Controllers e os Models.

## Models

Os models são criados através da linkagem com a base de dados e contem todos os atributos e respetiva tabela:

## Controllers

Os Controllers são criados para se poder realizar os requests e poderem ser executados para o Front-End. Desta forma seguem-se os Controllers criados e suas respetivas funcionalidades:

## Utilizador

A função que se segue, realiza um Get para devolder a lista dos utilizadores através da base de dados, e esta só será autorizada para um user que tenha token de administrador.

Uma imagem com texto

Descrição gerada automaticamente

Esta ainda usa uma função auxiliar para esconder passwords:

Uma imagem com texto

Descrição gerada automaticamente

Temos uma funcionalidade Get, para buscar um utilizador através da token apanhada no Front-End através do login realizado. Este verifica através do seu mail e este é retornado.

Uma imagem com texto

Descrição gerada automaticamente

Para quando se necessitar dos dados de um utilizador em especifico, é retornado através de um Get que leva por parâmetro o id do user em questão.

Uma imagem com texto

Descrição gerada automaticamente

Para o login do utilizador é feito um Post para autenticar este através das credenciais inseridas. Este verifica o utilizador, e no caso deste ser verificado, é gerada a token de sessão dependendo da sua rule.

Uma imagem com texto

Descrição gerada automaticamente

A funções auxiliares para a token e verificação da conta são as seguintes:

Uma imagem com texto

Descrição gerada automaticamente

Uma imagem com texto

Descrição gerada automaticamente

Uma imagem com texto

Descrição gerada automaticamente

Para o utilizador ser criado é usado um POST do objeto Utilizador passado por parâmetro. Todos os seus dados são armazenados, a sua password é criptografada, e uma rule é lhe atribuído. No fim o utilizador é armazenado na lista.

Uma imagem com texto

Descrição gerada automaticamente

No caso de os dados do utilizador serem alterados é realizado Patch. É enviado por parâmetro o id do utilizador e um objeto com os novos dados a serem alterados. Este recebe uma verificação de autorizações, os dados são alterados e atualizados.

Uma imagem com texto

Descrição gerada automaticamente

Para o caso de se alterar a password, a função que se segue cumpre. Este recebe o id do utilizador e o objeto utilizador que contem a nova password. A nova password é criptografa e armazenada.

Uma imagem com texto

Descrição gerada automaticamente

Para do utilizador querer querer remover a sua conta, este terá o estado da sua conta alterado para “Inativo”, ou seja, este terá a sua conta arquivada. É feito uma verificação da sua autorização, pois este so poderá remover a sua própria conta.

Uma imagem com texto

Descrição gerada automaticamente

Outro delete existente é da remoção de um utilizador através do admin. Este terá autorização para remover os seus funcionários.

Uma imagem com texto

Descrição gerada automaticamente

## Máquina

De forma a conseguir listar as máquinas todos com estado “Ativo” é utilizado um Get:

Uma imagem com texto

Descrição gerada automaticamente

Para a listagem de uma máquina em especifico é passado por parâmetro o seu id:

Uma imagem com texto

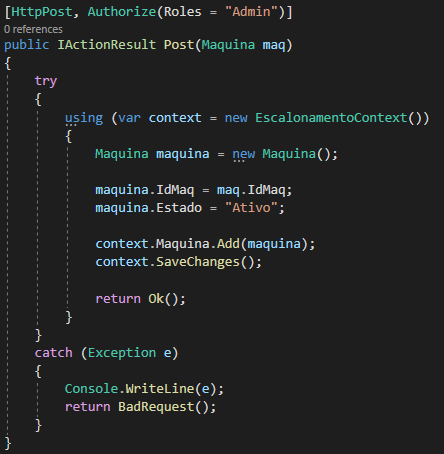
Descrição gerada automaticamente

Para a obtenção de uma máquina e sua duração em uma operação, de um job, de uma simulação é enviado um objeto do tipo Conexao que contem os ids do user, simulação, job e operação. Pesquisando depois na base de dados este retorna a conexão que contem a máquina e a duração a serem apresentadas ao utilizador.

Uma imagem com texto

Descrição gerada automaticamente

Inserção de uma máquina pelo Admin na base de dados:



O método que se segue atualiza os dados de uma máquina, mais especificamente o seu estado:

Uma imagem com texto

Descrição gerada automaticamente

Para a remoção de uma máquina pelo Admin, este envia por parâmetro o id da máquina e altera o seu estado para “Inativo”. No caso desta ser desativada, todas as conexões que a contenham, o id maquina e a duração serão convertidas a null. Mais tarde o user pode verificar que tal ocorreu e substituir por máquinas e durações novas.

Uma imagem com texto

Descrição gerada automaticamente

## Conexao

Para buscar as conexões temos os seguintes métodos:

Este retorna para o Admin as conexões cujo estado sejam verdade (por outras palavras ativa para uso).

Uma imagem com texto

Descrição gerada automaticamente

Este retorna todas as conexões falsas (ou inativas) para o Admin:

Uma imagem com texto

Descrição gerada automaticamente

Para o caso do utilizador querer ver as suas conexões, este através do seu id, consegue retornar todas que lhe sejam pertencidas. O admin também as pode consultar:

Uma imagem com texto

Descrição gerada automaticamente

No caso de se querer obter os dados de uma simulação pertencente a um utilizador, esta função envia os dados dessa simulação em especifico:

Uma imagem com texto

Descrição gerada automaticamente

Na criação de uma nova conexão é feito um Post dos dados passados por um objeto do tipo Conexão:

Uma imagem com texto

Descrição gerada automaticamente

Para o caso de o utilizador querer alterar o id de máquina e duração de uma conexão em especifico, este terá de enviar todos os ids de user, sim, job e operação que o identifiquem e os novos valores a substituir:

Uma imagem com texto

Descrição gerada automaticamente

Para o caso de remover uma simulação de uma conexão, este recebe por parâmetros id do user e id da simulação e onde estes forem encontrados na base de dados é lhes atribuído o estado falso (ou inativo).

Para o Admin:

Uma imagem com texto

Descrição gerada automaticamente

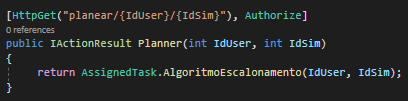
Para Admin e Funcionario:

Uma imagem com texto

Descrição gerada automaticamente

Para a criação do plano de produção, são realizados os seguintes métodos:

Para automático, este chama uma função que representa o algoritmo (Encontrado no AssignedTak.cs) por trás do escalonamento e este retorna um objeto que será utilizado no Front-End:



Para o manual:

# Front-End

# Bibliografia

# Conclusão