Introdução ao problema:

Desenvolver uma aplicação para atender as necessidades expostas no problema. Deve ser criada uma web api e uma interface web. Deve haver uma tela de login que irá redirecionar o usuário para as demais funcionalidades.

Você deve fornecer provas suficientes usando testes unitários de que a sua solução é completa por, no mínimo, o que indica que ele funciona corretamente contra os dados de teste fornecidos.

Problema: Reserva de salas para reunião.

Uma empresa chamada Meet Group deseja automatizar o processo de reservas de salas de reunião, que atualmente é feito pela secretária da empresa. A Meet Group tem 12 salas de reuniões disponíveis onde:

● Salas 1 a 5 contam com computador, tem lugar para 10 pessoas, acesso à internet, tv e webcam para vídeo conferências;

● Salas 6 e 7 contam com lugares para 10 pessoas e acesso à internet;

● Salas 8, 9 e 10 contam com computador, acesso à internet, tv e webcam para vídeo conferências e tem lugar para 3 pessoas;

● Salas 11 e 12 contam com lugares para 20 pessoas;

Desenvolva um programa para ajudar os colaboradores da empresa a encontrar a melhor sala de reuniões para suas necessidades e que esteja disponível. A entrada para o programa será, a data de início, horário de início, data do fim da reunião, horário final da reunião, quantidade de pessoas que irão participar da reunião, se precisa de acesso à internet e se precisa de TV e webcam;

Algumas regras:

● Implementar autenticação que deve seguir o padrão **JWT** e o token a ser recebido deve estar no formato **Bearer** ou utilizar das sessões com contexto de banco;

● Somente pessoas logadas devem conseguir realizar as requisições no web service;

● As reuniões devem ser agendadas com no mínimo um dia de antecedência;

● As reuniões devem ser agendadas com no máximo 40 dias de antecedência;

● As reuniões devem ser agendadas apenas para os dias úteis;

● Reuniões não podem durar mais que 8 horas;

● Não se pode agendar reuniões conflitantes de lugar e hora;

Caso não exista nenhuma sala que atenda às necessidades do colaborador, seu programa deverá sugerir 3 datas e locais possíveis de se agendar.

Cada agendamento deverá ser salvo em um banco de dados para armazenar os dados. Se utilizar .Net core pode ser utilizado as migrations para gerar o banco de dados.

O projeto deve ser documentado utilizando Swagger

Extras:

Os itens abaixo não são obrigatórios porém desejáveis.

● Teste de integração da API em linguagem de sua preferência.

● Cobertura de testes utilizando Sonarqube.

● Utilização do Docker (Encapsular por completo a aplicação com suas configurações)

A introdução para o formato:

<DATA_INICIO>; <HORA_INICIO>; <DATA_FIM>; <HORA_FIM>; <QUANTIDADE_PESSOAS>; <ACESSO_INTERNET>; <TV_WEBCAM>

Input 1:

26-02-2018; 10:00; 23-02-2018; 12:00; 10; Sim; Sim OUTPUT 1: Sala 1

Input 2:

26-02-2018; 10:00; 23-02-2018; 12:00; 10; Sim; Sim OUTPUT 2: Sala 2

Input 3:

26-02-2018; 10:00; 23-02-2018; 12:00; 10; Sim; Não OUTPUT 3: Sala 6

Tecnologias permitidas:

● .Net core 2.1 ou superior

● .Net framework 4.5 ou superior

● **Banco de dados:** postgres 10 ou superior (informar versão utilizada)

● **ORM:** NHibernate ou Entity

● JWT

● Visual Studio 2017 ou superior

● **Front:** Vue, React, AngularJS

● IIS 7 ou superior

● **Documentação:** Swagger

● Docker

Obs: qualquer tecnologia utilizada além das permitidas não será avaliada.

Premissas do projeto:

Deve estar pronto para rodar apenas apertando F5 no Visual Studio e nada mais. Exceção vai para os pacotes do Nuget, que poderão ser baixados na hora da compilação (não nos envie-os).

Será avaliado:

- Clareza e qualidade do código;
- Arquitetura e organização do projeto;
- Segurança da API, autenticação e senhas salvas no banco;
- Boas práticas da Linguagem/Framework;
- Capricho e empenho na resolução do problema;

Obs.: Caso não consiga finalizar todos os requisitos nos envie até onde concluir a atividade para realizarmos a avaliação.