

Tecnologias e Sistemas de Informação para a Web			
CURSO			
ANO LETIVO	DATA	HORA	DURAÇÃO
Serviços e Interfaces para a Cloud			3º ano
UNIDADE CURRICULAR			ANO
João Adriano Teixeira Ferreira			
DOCENTE			ÉPOCA

Observações:

Enunciado

1. Crie um Serviço Web que responda aos seguintes requisitos:

- Utilização de Node.JS com Express.JS;
- Implementação do Protocolo REST;
- O serviço web deve representar as seguintes entidades (models):

Estação Metereológica, composto por:

- nome (obrigatório);
- descricao (facultativo);
- latitude (obrigatório);
- longitude (obrigatório).

Registo Pluvial, composto por:

- id_estacao_metereologica (obrigatório);
- valor (obrigatório);
- data_hora (obrigatório).

- d. O serviço deve possuir as rotas que permitam responder as estas necessidades:
- Criar estação meteorológica;
 - Listar registos pluviais (id da estação deve ser passado na query do pedido).
- e. O serviço deve estar conectado a um Broker MQTT no HiveMQ e para tal deve:
- Se ainda não tiver criar conta no HiveMQ;
 - Criar um cluster no HiveMQ e configurar os respetivos acessos;
 - O serviço web NodeJS deve subscrever o tópico **/registos**;
 - Publicar no tópico **/registos**. A publicação deve ser feita através de uma rota adicional. O valor e o id da estação meteorológica devem ser passados como parâmetros do pedido.
 - Ao que receber um valor do campo subscrito inserir um novo registo na tabela Registos_Pluviais com o valor recebido.

- f. O serviço deve estar conectado a uma Base de Dados:

Pode utilizar Sequelize ou Mongoose:

Os nomes das tabelas/collections devem seguir o seguinte padrão: nome_numeroAluno
(ex: **Estacoes_Metereologicas_91000000** e **Registos_Pluviais_91000000**).

Acesso MySQL (Sequelize):

User: **joaoferr_tsiw**

Database: **joaoferr_tsiw**

Host: **www.joaofferreira.eu**

Dialect: **mysql**

Password: **GAa8xvmV3eKrVa8C**

String de Conexão (Mongoose):

mongodb+srv://tsiw:GAa8xvmV3eKrVa8C@cluster0.b0vmz.mongodb.net/TSIW?retryWrites=true&w=majority

- g. Criar o ficheiro **dockerfile** que permita transformar o serviço web criado num container.

2. Crie um segundo Serviço Web que responda aos seguintes requisitos:

- Implementação do Protocolo GraphQL;
- Utilização de NodeJS, Express.JS e Apollo Server Express;
- O serviço web deve representar as mesmas entidades descritas no ponto 1.c;
- O serviço web deve possuir uma query que permita pedir ao servidor informações sobre estações meteorológicas e registos pluviais associados;
- O serviço web deve possuir um mutator que permita criar uma estação meteorológica;

f. O serviço web deve estar conectado a uma base de dados estruturada pelo que deverão utilizar os acessos fornecidos no ponto 1.f (sequelize).

2. Submissão de teste

Deve submeter o teste no Moodle, no espaço de avaliação criado para o efeito. Deve comprimir todos os ficheiros do projeto num ficheiro .zip e deve excluir as pastas node_modules. O nome do ficheiro deve seguir o seguinte padrão: primeiroNome_UltimoNome_numeroAluno (ex: **João_Ferreira_91000000**).

Depois de submeter deve validar com o docente se a submissão foi bem sucedida.

Boa Sorte :)