

Trabalho prático

14 de novembro de 2024

Pretende-se desenvolver uma Plataforma de Gestão de Surtos Epidemiológicos e Recomendação de Saúde, composta por uma Web API a ser utilizada pela Organização Mundial de Saúde (OMS) e outra a ser utilizada por uma agência de viagens (SafeTourism). Cada Web API será constituída pelo *backend* desenvolvido em Node.js e pelo *frontend* desenvolvido em Angular. Para qualquer uma das partes, deve discutir com o cliente os requisitos funcionais.

Devem ser elaborados o modelo de domínio e as vistas lógicas (nível 1, 2 e 3) de análise e design;.

As aplicações desenvolvidas devem ser validadas e verificadas recorrendo a testes de integração usando o Postman.

Os processos de autenticação e autorização, em ambos os casos, devem ser implementados localmente utilizando uma base de dados relacional (ex: SQLite).

1 Desenvolvimento da API para a OMS

A OMS pretende disponibilizar uma Web API que permita gerir informação sobre surtos epidemiológicos e disponibilizar informação agregada em relatórios e recomendações aos seus utilizadores.

1.1 *Backend*

1. Desenvolva a Web API, em Node.js e com recurso à base de dados MongoDB, de forma a disponibilizar as operações CRUD (sempre que faça sentido), incluindo os recursos REST apresentados na Tabela 1 e tendo em conta as restrições definidas na Tabela 2.
2. A OMS tem dois tipos de utilizadores (administrador e funcionário). Implemente processos de autenticação e de autorização para os recursos definidos, tendo em conta que:
 - Apenas o administrador poderá fazer criar, editar e eliminar países e zonas
 - Apenas os funcionários poderão criar e editar vírus, surtos e recomendações. Não é possível eliminação.
 - Qualquer utilizador, autenticado ou não, poderá consultar a informação existente

1.2 *Frontend*

O frontend da Web API deve ser desenvolvido utilizando Angular.

Tabela 1: Recursos REST a definir para a API da OMS

Recurso REST	Descrição
POST: /api/paises/	Criar um país
POST: /api/zonas/	Cria uma zona geográfica
POST: /api/virus/	Criar um vírus
POST: /api/surtos/	Criar um surto epidemiológico
POST: /api/recomendacoes/	Criar uma nota de recomendação
GET: /api/surtos/pais/cp ¹	Obter os surtos ativos para o país
GET: /api/recomendacoes/cp ¹ /cs ²	Obter as recomendações válidas para o país e surto referidos
GET: /api/surtos/virus/cv ³	Obter informação sobre todos os surtos ativos associados ao vírus referido
GET: /api/surtos/virus/cv ³ /ocorridos	Obter informação todos os surtos ocorridos associados ao vírus referido
PUT: /api/surtos/cp ¹ /cv ³	Alterar a data de fim do surto
PUT: /api/recomendacoes/cr ⁴	Editar a nota de recomendação

¹ Código do país² Código do surto³ Código do vírus⁴ Código da recomendação

2 Desenvolvimento da Plataforma SafetyTourism

A empresa SafetyTourism.com pretende disponibilizar no mercado uma plataforma web que permita fornecer informações sobre o estado de segurança de saúde, nomeadamente de doenças infecciosas transmissíveis, de destinos turísticos, para que os clientes possam consultar a informação ao pesquisar um destino turístico com o intuito de reservar uma viagem. Com esse propósito, irá criar um sistema informático para suportar o negócio. Sobre cada destino turístico, os utilizadores poderão consultar relatórios epidemiológicos, acompanhar o estado de evolução de possíveis surtos epidemiológicos e recomendações da OMS.

Os utilizadores poderão pesquisar destinos turísticos e consultar os dados epidemiológicos dos mesmos. Para isso, a aplicação irá ligar-se à aplicação da OMS para consultar os dados, mostrando os surtos e/ou recomendações ativas (caso existam) ou que o destino é seguro (caso não haja surtos nem recomendações ativas). Caso o utilizador seja um cliente registado e com login ativo, para além desta informação, deverá ser-lhe apresentada informação sobre os pacotes disponíveis para o destino turístico (mesmo que não seja seguro) e deve ser-lhe permitido efetuar uma reserva (o que pode implicar atualização dos dados pessoais).

2.1 *Backend*

1. Desenvolva a Web API, em Node.js e com recurso à base de dados MongoDB, de forma a disponibilizar as operações CRUD (sempre que faça sentido), tendo em conta as restrições definidas na Tabela 3, e incluindo os recursos REST necessários para cumprir os seguintes requisitos funcionais:

- O administrador pode validar registos de utilizadores
- Um utilizador pode efetuar registo
- Um utilizador pode consultar destinos e respetivos dados epidemiológicos
- Um utilizador pode fazer login
- Um funcionário pode inserir/editar/eliminar destinos
- Um cliente pode visualizar pacotes para um destino
- Um cliente pode reservar um pacote
- Um cliente pode consultar o histórico das suas reservas

2.2 *Frontend*

O frontend da Web API deve ser desenvolvido utilizando Angular.

Tabela 2: Restrições às entidades presentes na Web API da OMS

Entidade	Identificado por	Restrições
Zona	<ul style="list-style-type: none"> • Código da zona geográfica • Nome da zona geográfica 	Não permite criar/editar se: <ul style="list-style-type: none"> • código da zona já existe • nome da zona já existe
		Não permite eliminar se: <ul style="list-style-type: none"> • tem países associados • tem surtos associados • tem recomendações associadas
País	<ul style="list-style-type: none"> • Código país (norma ISO 3166-1 alpha-2 - (https://www.rallybel.com/pt/links_iso_code2.html)) • Nome do país • Código da zona geográfica 	Um país pertence a uma zona geográfica
		Não permite criar/editar se: <ul style="list-style-type: none"> • zona não existe • código do país já existe • nome do país já existe
Vírus	<ul style="list-style-type: none"> • Código do vírus • Nome do vírus 	Não permite inserir/editar se: <ul style="list-style-type: none"> • código do vírus já existe • nome do vírus já existe
		Considera-se “ativo” um surto que ainda não tenha sido dado como extinto e “ocorrido” um surto que já o tenha sido
Surto	<ul style="list-style-type: none"> • Código do surto • Código do vírus associado ao surto • Código da zona geográfica • Data de deteção do surto • Data de fim do surto (se já tiver sido dado como extinto, ou null) 	Não permite inserir/editar se: <ul style="list-style-type: none"> • código do vírus não existe • código da zona geográfica não existe • código do surto já existe • Data de início ou fim é no futuro • Data de fim é anterior à data de início
		Considera-se “válida” uma recomendação cuja nota ainda está válida de acordo com a sua validade (em dias)
Recomendação	<ul style="list-style-type: none"> • Código da nota de recomendação • Código do surto • Data da nota • Validade (em dias) da nota 	Não permite inserir/editar se: <ul style="list-style-type: none"> • código do surto não existe • código da recomendação já existe

Tabela 3: Restrições às entidades presentes na Web SafeTourism

Entidade	Identificado por	Restrições
Destino turístico	<ul style="list-style-type: none"> • id • nome • país 	<p>Não permite inserir/editar se:</p> <ul style="list-style-type: none"> • há pacotes de viagem associados
Utilizador	<ul style="list-style-type: none"> • id • nome • email • password • tipo de utilizador • morada • telefone • NIF 	<p>Apenas id, nome, email, password, tipo de utilizador são obrigatórios para registo (cliente/funcionário). Restantes apenas obrigatórios para efetuar reserva.</p> <p>Não permite inserir/editar se:</p> <ul style="list-style-type: none"> • Email já existente • NIF já existente <p>Não permite eliminar, apenas desativar o utilizador</p> <p>Deve obedecer às regras estabelecidas pelo RGPD</p>
Pacote de viagem	<ul style="list-style-type: none"> • id • descrição • preço • destino (liga ao destino turístico) 	<p>Irá haver 3 pacotes por destino: voo, voo+hotel, hotel; sem possibilidade de escolher alternativa de diferentes voos e/ou hotéis.</p>
Reserva	<ul style="list-style-type: none"> • id • cliente (que efetua a reserva) • pacote (reservado) 	