

Instituto Politécnico de Viseu
Escola Superior de tecnologia e Gestão de Viseu
Departamento de Informática



Licenciatura em tecnologias e design de multimédia

Relatório de Aplicações para a Internet III

Beatriz Mendes - 22608

Filipa Mesquita - 23198

João Nascimento - 23958

Michael Nunes - 23208

2023/2024

Introdução:

Neste relatório, será descrito o processo de desenvolvimento de uma API para Gestão de Tarefas em Node.js. O objetivo principal deste trabalho prático foi criar uma API RESTful que permitisse aos utilizadores realizar operações essenciais na gestão de tarefas, como criar, atualizar, eliminar e listar tarefas de forma eficiente.

Começamos por definir os nossos diagramas de contexto e contentores, nos quais especificámos o comportamento da API. Posteriormente, criamos as estruturas JSON que representam a forma como iremos receber os dados. Para validar esses dados, desenvolvemos esquemas, estabelecendo regras que incluem limitações de caracteres, padrões para e-mails, números de telefone, datas e enumerações, detalhando de forma precisa os dados desejados.

Continuamos com a elaboração da documentação da API em OpenAPI, utilizando o formato YAML. Neste processo, integramos todos os esquemas e definimos os endpoints. A partir deste ponto, exportámos o YAML no servidor Node.js, codificámos os endpoints e iniciámos a ligação à base de dados, neste caso, MongoDB.

Desta forma garantimos que todos os passos propostos tivessem sido alcançados com sucesso.

Diagramas de Contexto e Containers

Nos diagramas de contexto e containers definimos como seria a arquitetura da nossa api, no nosso caso, tínhamos que usar node.js como servidor, a partir daqui definimos como seria toda a comunicação entre cliente, servidor e base de dados, sendo a comunicação do cliente efetuada com solicitações HTTP, e devolvida em formato json, por fim o servidor comunica com a base de dados, MogoDB, para armazenar e solicitar os dados.

Descrição dos endpoints

Foram criados endpoints para relacionar Tarefas, Users e TarefasPadrao.

TarefasPadrao foi um schema criado para listar tarefas comuns a todos os utilizadores, para evitar que os mesmo tenham que repetir tarefas que realizam todos os dias, estas tarefa são predefinidas pela aplicação não havendo a possibilidade de criar novas por parte do utilizador.

Grupo de Endpoints Tarefas

	URL	SCHEMA
GET	/tarefas/{userId}	listaDeTarefasDeTarefas
GET	/tarefas/{userId}/{prioridade}	listaDeTarefasPorPrioridade
GET	/tarefas/{userId}/{dataVencimento}	listaDeTarefasDataVencimento
GET	/tarefas/{userId}/{estado}	listaDeTarefasEstado
GET	/tarefas/{tarefald}	tarefa
PATCH	/tarefas/{userId}/{estado}	tarefa
POST	/tarefas	starefa
DELETE	/tarefas/{tarefald}	- - - - -
PUT	/tarefas/{tarefald}	tarefa

Nestes endpoints temos funcionalidades de criar, listar(filtrar), editar e eliminar tarefas.

Grupo de Endpoints Tarefas Padrão

	URL	SCHEMA
GET	/tarefaspadrao	schemas/ listaTarefaPadraos
DELETE	/tarefaspadrao/{tarefald}	schemas/ tarefaPadraos
PATCH	/tarefaspadrao/{tarefald}/{dataVencimento}	schemas/ tarefaPadraos
POST	/tarefaspadrao/{userId}/{tarefald}	schemas/ tarefaPadraosUser

Nestes endpoints temos funcionalidades de associar uma tarefa padrão a um utilizador, eliminar uma tarefa padrão, atualizar a data de uma tarefa padrão, e listar todas as tarefas padrão existentes.

Grupo de Endpoints User

	URL	SCHEMA
POST	/User	User
GET	/User/login	Login
GET	/User/logout	- - - - -
GET	/User/{email}	User
DELETE	/User/{email}	- - - - -
PATCH	/User/{email}	- - - - -

Nestes endpoints temos funcionalidades de criar um utilizador, eliminar, atualizar, listar um utilizador pelo seu email, assim como fazer login e logout.

Códigos de resposta utilizados:

"200": Quando o pedido é feito corretamente
 "201": Quando o pedido é feito corretamente e cria alguma coisa
 "400": Quando os parâmetros são inválidos
 "401": Quando as credenciais são invalidas
 "404": Quando não encontra um parâmetro
 "500": Quando ah um erro do servidor

Headers utilizados:

Accept	application/json
Content-Type:	application/json

Task Manager

1.0 OAS 3.0

Gere a tua vida com TODO

Servers

https://virtserver.swaggerhub.com/JOAOPNASCIMENTO6/ToDo/1.0 - SwaggerHub API Auto Mocking

Authorize

Tarefa

GET	/tarefas/{userId}	Lista todas as tarefas do user logado.	
GET	/tarefas/{userId}/{prioridade}	Lista todas as tarefas por prioridade	
GET	/tarefas/{userId}/{dataVencimento}	Lista todas as tarefas por data de vencimento	
GET	/tarefas/{userId}/{estado}	Lista todas as tarefas por estado	
PATCH	/tarefas/{userId}/{estado}	Atualizar o estado de uma tarefa	
POST	/tarefas	Criar uma nova tarefa	
DELETE	/tarefas/{tarefaId}	Excluir uma tarefa existente	
GET	/tarefas/{tarefaId}	Obter informações sobre uma tarefa	
PUT	/tarefas/{tarefaId}	Atualizar uma tarefa existente	

TarefaPadrao

GET	/tarefaspadrao	listar tarefas padrao	
DELETE	/tarefaspadrao/{tarefaId}	Eliminar uma tarefa padrao por ID	
PATCH	/tarefaspadrao/{tarefaId}/{dataVencimento}	Atualizar a data de vencimento de uma tarefa por ID	
POST	/tarefaspadrao/{userId}/{tarefaId}	Associar uma tarefa padrão a um usuário	

User

POST	/User	Crear User	
GET	/User/login	User Login	
GET	/User/logout	User Logout	
GET	/User/{email}	Procurar User apartir do email	
DELETE	/User/{email}	Eliminar um user apartir do email	
PATCH	/User/{email}	Atualiza o email do usuário	

Schemas

tarefa	>
listaTarefasPadrao	>
tarefasPadrao	>
User	>
listaDeTarefasDataVencimento	>
listaDeTarefasEstado	>
listaDeTarefasPorPrioridade	>
tarefaPadraosUser	>
Login	>
listaDeTarefas	>

Conclusão:

Finalmente, a criação desta API RESTful foi realizada de maneira adequada.

As maiores dificuldades enfrentadas ao longo de todo o processo foram a implementação da autenticação e a ligação à base de dados. Embora tenhamos conseguido estabelecer a ligação com a base de dados, enfrentamos desafios ao utilizá-la corretamente. Certamente, será necessário aprofundar o conhecimento neste tópico para compreender como implementá-lo corretamente.

Em resumo, o desenvolvimento desta API RESTful serviu tanto para consolidar nossos conhecimentos quanto para aprender novas abordagens na criação de APIs.