



Programação Distribuída

Trabalho Prático: Meta 2

Docente: José Manuel Meireles Marinho

Bruno Cardoso – 2018018880
Gonçalo Pereira – 2019139650
Gonçalo Leite - 2018014643

Coimbra, 27 de novembro de 2023

Índice

Conteúdo

1. Introdução2
2. Endpoints3

1. Introdução

Este trabalho surge no âmbito da Unidade Curricular de Programação Distribuída, no 1º Semestre do ano letivo de 2023/2024. A sua implementação foi feita recorrendo à linguagem Java.

Este trabalho tem como foco a integração de um serviço Web do tipo Rest (API REST) no servidor principal concebido na meta 1. Este processo envolve a criação de um projeto Spring Boot do tipo Maven, no qual o código-fonte do servidor da primeira meta é integrado, adicionando a lógica associada à API REST desejada, incluindo controladores, autenticação e outros elementos relevantes. Utilizando também a lógica de acesso à base de dados já existente (DataBaseHelper) e o desenvolvimento de uma aplicação cliente fazendo uso de todas as funcionalidades proporcionadas pela API REST, substituindo o uso de sockets TCP por meio de pedidos HTTP/API REST.

2. Endpoints

Os endpoints usados nesta meta foram :

Metodo	URL	Operação	Descrição	Corpo do pedido	Resposta do pedido
POST	/login	Create	Este método lida com requisições de login, autenticando o utilizador	String	String
GET	/events	Read	Este método mostra todos os eventos criados podendo ser aplicados diversos tipos de critérios/filtros (período, nome ...)	Nada	Lista de eventos

POST	/events	Create	Este método adiciona um novo evento caracterizado pelo nome, local, data de realização, hora de início e hora de fim	String	Evento
DELETE	/events/{name}	Delete	Este método elimina um evento já criado através do seu nome	String	Evento
GET	/events/{name}/attendance	Read	Este método mostra um evento já criado com uma presença registrada de acordo com o seu ID	Nada	Lista de eventos com presenças

PUT	/events/{name}/generateCode	Update	Este método gera um código para um evento já criado através do seu ID com indicação do tempo de validade	String	Nada
POST	/register	Create	Este método adiciona um novo utilizador sendo caracterizado pelo nome, número de identificação, endereço de email (unico) e uma password	String	Utilizador

GET	<i>/profile</i>	Read	Este método mostra um utilizador registado através do seu ID	Nada	Utilizador
POST	/profile/attendance	Create	Este método cria uma presença no evento associado ao código	Código do evento	Lista de eventos
GET	/profile/attendance	READ	Este método devolve as presenças de um utilizador	String	Presença

