

ETAPA 1 - Especificação de requisitos, Modelagem de hierarquia de classes (2,5 pontos de um total 10 de pontos)

Integrantes: Pedro Stein, Samuel, Thiago e Vitor

Verbo HTTP	Descrição da Funcionalidade	Path	Dados de envio em JSON (Body)	Dados de Resposta em JSON (Body)	Responsável
GET	Recupera a lista de alunos que utilizam o sensor.	wellbeing/alunos/listar	-	<pre>{ "nome": "Marcelo", "endereco": "Planejada Vista", "prontuario": "BP248376-X", "email": "marcelinho.paulo@gmail.com", "cpf": "734.474.247-02" }, { "nome": "Vinicius", "endereco": "Rua y", "prontuario": "BP3333", "email": "vinicius@gmail.com", "cpf": "444.333.111-15" }</pre>	Thiago
GET	Recupera um aluno que utiliza o sensor.	wellbeing/alunos/{nome}	-	<pre>{ "nome": "Marcelo", "endereco": "Planejada Vista", "prontuario": "BP248376-X", "email": "marcelinho.paulo@gmail.com", "cpf": "734.474.247-02" }</pre>	Vitor
GET	Recupera a lista com as leituras de um dia específico dos alunos.	wellbeing/leituras/{dia}	-	<pre>[{ "dataEntrada": "2023-08-23", "valor": 200.0, "descricao": "null", "hora": "12:00:10" }, { "dataEntrada": "2023-08-23", "valor": 100.0, "descricao": "primeiro", "hora": "18:17:40" }]</pre>	Samuel
GET	Recupera a lista com as leituras de um dia com uma hora específica do aluno	wellbeing/leituras/{dia}/{hora}	<pre>const [leituras] = await db.query(`SELECT * FROM leituras WHERE dia = ? AND hora = ?`, [dia, hora]); if (leituras.length === 0) { return res.status(404).json({ message: 'Leitura não encontrada' }); } return res.json(leituras);</pre>	<pre>{ "valor": "200", "hora": "12:00" },</pre>	Stein
Post	Adicionar um aluno a lista de alunos.	wellbeing/alunos/cadastrar	<pre>{ "nome": "Henrique", "endereco": "Planejada 4", "prontuario": "BP248376-X", "email": "henriquepro@gmail.com", "cpf": "734.474.247-02" }</pre>	-	Thiago
Post	Adicionar um sensor a lista de Sensores.	wellbeing/cadastrar/sensores	<pre>{ "codigo": "BP51441", "preco": 120.00, "nome": "Bio-222", "tipoSensor": "Cardiovascular", "dataEntrada": "24/05/2023", "garantia": "24/05/2025" }</pre>	-	Vitor
Post	Adicionar uma leitura a lista de leituras.	wellbeing/leituras/cadastrar	<pre>{ "dataEntrada": "2023-08-20", "valor": "200", "descricao": "Ok", "hora": "10:30:15" }</pre>	-	Samuel

Post	Adicionar o relatório do Aluno na lista de Relatórios.	wellbeing/relatorios/cadastrar	{ "email": "matheus@gmail.com" },	<pre>// RelatorioController - Rel public class RelatorioController { private List<String> listaRelatoriosAlunos = new ArrayList<>(); @PostMapping("/wellbeing/relatorios") public void adicionarRelatorio(@RequestBody Map<String, String> request, String email = request.get("email"); listaRelatoriosAlunos.add(email); }</pre> <p>(jdk dando problema no meu computador, então eu coloquei a imagem do desenvolvimento do código)</p>	Stein
Delete	Deletar um aluno da lista	wellbeing/alunos/deletar/{nome}	-	<pre>{ "nome": "Lucas", "endereco": "Planejada Vista", "prontuario": "BP248376-X", "email": "marcelinho.paulo@gmail.com", "cpf": "734.474.247-02" }</pre>	Thiago
Delete	Deletar sensor	wellbeing/deletar/sensor/{nome}	-	<pre>{ "codigo": "BP47295", "preco": 90.0, "nome": "Sensor X", "tipoSensor": "Cardiovascular", "dataEntrada": "02-08", "garantia": "3 meses" }</pre>	Vitor
Delete	Deletar leitura de um dia	wellbeing/leitura/deletar/{dia}	-	<pre>{ "dataEntrada": "2023-08-23", "valor": 200.0, "descricao": "null", "hora": "12:00:10", "localDateTime": null }</pre>	Samuel
Delete	Deletar um relatório do Aluno.	wellbeing/relatorios/deletar/{descricao}	-	<pre>@PostMapping("/wellbeing/relatorios") public void adicionarRelatorio(@RequestBody Map<String, String> request, String email = request.get("email"); listaRelatoriosAlunos.add(email); }</pre>	Stein

Stein: Get, Post e Delete

Samuel: Get, Post e Delete

Thiago: Get, Post e Delete

Vitor: Get, Post e Delete