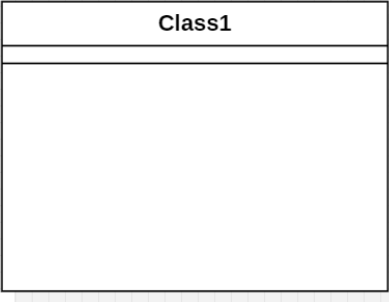
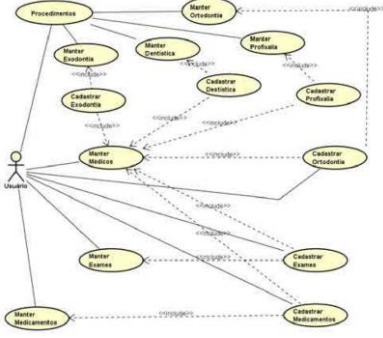
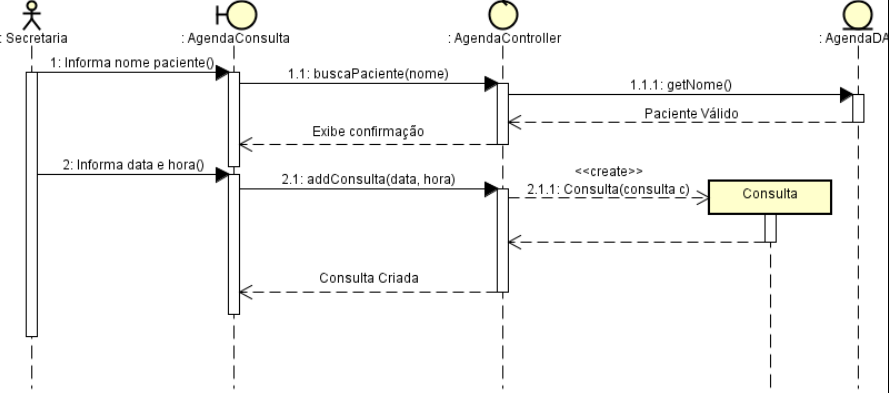


Dicionário do Javeiro FullStack	
Termo	O que é? Para que serve? Como fazer?
CRUD	É acrônimo que significa inserir, deletar, atualizar e pesquisar no banco de dados
Insomnia	É uma ferramenta para testar a API - Inseriu? -O retorno foi o correto: 200, 201 ou 204
Xampp	Ferramenta utilizada por programadores PHP No treinamento é necessário iniciar o xampp (mysql)
StarUML	<div><p>Diagrama de Classe</p><p>Importante para modelar sistemas. (criar modelos)</p><p>Diagrama de atividade Papitada: Documento de Requisitos RF (vai fazer isso) e RNF (vai se comportar assim) Regra de Negócio RN</p></div>

	Primeiro documento é Documento de Visão Geral do Sistema
HTTP	Protocolo – regras de utilização é uma forma de comunicação entre os computadores em rede
Application.properties	<pre> 1 spring.datasource.url=jdbc:mysql://localhost/vollmed_api 2 spring.datasource.username=root 3 spring.datasource.password='' 4 5 spring.jpa.show-sql=true 6 spring.jpa.properties.hibernate.format_sql=true 7 8 server.error.include-stacktrace=never 9 10 api.security.token.secret=\${JWT_SECRET:12345678} </pre>
Excluir	<p>204 No Content</p> <pre> @DeleteMapping("/{id}") @Transactional public ResponseEntity excluir(@PathVariable Long id) { var medico = repository.getReferenceById(id); medico.excluir(); // Resposta: 204 No Content - requisição foi processada e sem conteúdo para retornar // exercício teste return ResponseEntity.noContent().build(); } </pre>
Listar	<pre> @GetMapping public ResponseEntity<Page<DadosListagemMedico>> listar(@PageableDefault(size = 10, so var page = repository.findAllByAtivoTrue(paginacao).map(DadosListagemMedico::new); return ResponseEntity.ok(page); } </pre> <p>Retorno 200 (sucesso)</p>
Atualizar	
Cadastrar	

Detalhar	<pre> 1 package med.voll.api.domain.medico; 2 3 import med.voll.api.domain.endereco.Endereco; 4 5 public record DadosDetalhamentoMedico(Long id, 6 String nome, 7 String email, 8 String crm, 9 String telefone, 10 Especialidade especialidade, 11 Endereco endereco) { 12 13 public DadosDetalhamentoMedico(Medico medico) { 14 this(medico.getId(), medico.getNome(), 15 medico.getEmail(), medico.getCrm(), 16 medico.getTelefone(), 17 medico.getEspecialidade(), 18 medico.getEndereco()); 19 } 20 } 21 </pre>
Put	<pre> @PostMapping @Transactional public ResponseEntity atualizar(@RequestBody @Valid DadosAtualizacaoMedico dados) { var medico = repository.getReferenceById(dados.id()); medico.atualizarInformacoes(dados); return ResponseEntity.ok(new DadosDetalhamentoMedico(medico)); } </pre>
Cadastrar	<pre> @PostMapping @Transactional public ResponseEntity cadastrar(@RequestBody @Valid DadosCadastroMedico dados, UriComponentsBuilder uriBuilder) { var medico = new Medico(dados); repository.save(medico); var uri = uriBuilder.path("/medicos/{id}").buildAndExpand(medico.getId()).toUri(); return ResponseEntity.created(uri).body(new DadosDetalhamentoMedico(medico)); } </pre> <p>201</p> <ul style="list-style-type: none"> • <code>var medico = new Medico(dados);</code> :: O código cria um novo objeto Medico, passando os dados que foram recebidos na requisição. • <code>repository.save(medico);</code> :: O <code>repository</code> é uma interface que lida com a comunicação com o banco de dados. O método <code>save</code> salva o objeto Medico no banco. O Spring gerencia essa operação automaticamente. • <code>var uri = uriBuilder.path("/medicos/{id}").buildAndExpand(medico.getId()).toUri();</code> :: Aqui, o UriComponentsBuilder constrói a URL para o recurso que acabou de ser criado. Ele pega o ID do

	<p>novos médicos e os insere no caminho <code>/medicos/{id}</code>. O resultado é uma URL completa, como por exemplo <code>/medicos/123</code>.</p> <ul style="list-style-type: none">• <code>return ResponseEntity.created(uri).body(new DadosDetalhamentoMedico(medico));</code> <p>Por fim, o método retorna uma resposta HTTP completa. O <code>ResponseEntity.created(uri)</code> define o status 201 (Created), informando que a operação de criação foi bem-sucedida.</p> <p>Ele também inclui o cabeçalho Location com a URL do novo recurso, o que é uma boa prática para APIs REST. O método <code>.body(...)</code> insere o corpo da resposta, que contém os dados detalhados do médico que foi criado.</p>
--	---