



Nome: Eduardo de Melo Albanese RM 561790

Nome: Nathan Gonçalves Pereira Mendes RM 564666

Nome: Guilherme de Andrade Martini RM 566087

Descrição objetiva e concisa da solução proposta

O objetivo deste projeto foi desenvolver o back-end completo da aplicação "Memoriza Saúde". A solução foi implementada como uma API Restful robusta, utilizando Java, o framework Spring Boot e persistência de dados em um banco Oracle.

O escopo do projeto inclui a criação de um modelo de dados para gerenciar Usuários, Consultas, Lembretes, Metas e outras entidades; a implementação de uma arquitetura em camadas (Model-Repository-Service-Controller); o desenvolvimento de um CRUD funcional e de validações (regras de negócio) para todas as entidades principais; e a exposição de "Endpoints completos e funcionais" prontos para serem consumidos por uma aplicação front-end.

Destaque das funcionalidades implementadas

A API "Memoriza Saúde" implementa as seguintes funcionalidades:

- Gerenciamento de Usuários:

CRUD completo (Create, Read, Update, Delete) para usuários.

Validação de e-mail único no cadastro.

- Controle de Acesso:

Endpoint de POST /api/login para validação de credenciais.

Regra de Negócio (Validação): Um usuário é automaticamente bloqueado (seu status ativo vira false) após 3 tentativas falhas de login.

- Gerenciamento de Consultas:

CRUD completo para agendamento de consultas médicas.

Regra de Negócio (Validação): Um endpoint para "Confirmar Presença" em uma consulta.

- Gerenciamento de Lembretes:

CRUD completo para lembretes de consultas ou medicamentos.

Regra de Negócio (Validação): Um endpoint para "Confirmar Recebimento" de um lembrete.

- Gerenciamento de Metas de Saúde:

CRUD completo para metas de saúde (ex: beber água).

Regra de Negócio (Validação): Um endpoint para "Atualizar Progresso" de uma meta.

- Gerenciamento de Diário e Cuidadores:

CRUD completo para as entidades DiarioSaude e Cuidador.

- Tratamento de Erros:

Implementação de um GlobalExceptionHandler para retornar mensagens de erro padronizadas (JSON) e códigos de status HTTP corretos (como 404 Not Found ou 400 Bad Request), conforme as boas práticas.

Tabela de Endpoints (API Restful)

Verbo HTTP	URI (Endpoint)	Códigos de Resposta	Descrição
POST	/api/usuarios	201	Cria (cadastra) um novo usuário.
		(Created), 400	
		(Bad Request)	
GET	/api/usuarios	200	Lista todos os usuários.
		(OK)	
		200 (OK), 404 (Not Found)	Busca um usuário pelo e-mail.
PUT	/api/usuarios/{email}	200	Atualiza os dados de um usuário.
		(OK), 404	
		(Not Found)	
DELETE	/api/usuarios/{email}	204	Deleta um usuário.
		(No Content), 404	
		(Not Found)	
POST	/api/usuarios/{email}/registrar-acesso	200	(Regra de Negócio) Registra tentativa de acesso.
		(OK), 404	
		(Not Found)	
POST	/api/login	200	Valida as credenciais de login.
		(OK), 404	
		(Not Found)	
POST	/api/consultas	201	Cria uma nova consulta.
		(Created)	
		200 (OK)	Lista todas as consultas.
POST	/api/consultas/{id}/confirmar-presenca	200	(Regra de Negócio) Confirma presença.
		(OK), 404	
		(Not Found)	
POST	/api/metas	201	Cria uma nova meta de saúde.
		(Created)	
		200 (OK)	Lista todas as metas de saúde.
POST	/api/metas/{id}/atualizar-progresso	200	(Regra de Negócio) Atualiza progresso da meta.
		(OK), 404	
		(Not Found)	

Teste 1 CREATE

The screenshot shows the Postman interface with a collection named "My Collection". A POST request is being made to the endpoint `http://localhost:8080/api/usuarios`. The request body contains the following JSON:

```
4 "ultimoAcesso": "2025-10-27",
5 "totalAcessos": 0,
6 "ativo": true,
7 "tentativasFalhas": 0
8 }
```

The response status is `201 Created`, indicating the user was successfully created.

Consegui criar um novo usuário (POST /api/usuarios) e recebi o status 201 Created.

Teste 2 READ

The screenshot shows the Postman interface with a collection named "My Collection". A GET request is being made to the endpoint `http://localhost:8080/api/usuarios`. The response status is `200 OK`, and the response body contains two user records:

```
2 [
3   {
4     "email": "Edu.teste@fiap.com",
5     "nome": "Edu Teste ATUALIZADO",
6     "ultimoAcesso": "2025-10-27",
7     "totalAcessos": 0,
8     "ativo": false,
9     "tentativasFalhas": 3
10   },
11   {
12     "email": "nathan.teste@fiap.com",
13     "nome": "Nathan Teste",
14     "ultimoAcesso": "2025-10-27",
15     "totalAcessos": 0,
16     "ativo": true,
17   }
18 ]
```

Consegui ler os dados (GET /api/usuarios) e verifiquei no Oracle SQL Developer que o usuário foi salvo corretamente no banco.

Teste 3 UPDATE

The screenshot shows the Postman interface with a collection named "My Collection". A PUT request is being made to the endpoint `http://localhost:8080/api/usuarios/nathan.teste@fiap.com`. The request body contains the following JSON:

```

1 {
2   "name": "Nathan Teste ATUALIZADO",
3   "ativo": true
4 }

```

The response status is 200 OK, indicating success.

Consegui atualizar o nome do usuário (PUT /api/usuarios/{email}) com sucesso.

Teste 4 REGRA DE NEGÓCIO

The screenshot shows the Postman interface with a collection named "My Collection". A POST request is being made to the endpoint `http://localhost:8080/api/usuarios/nathan.teste@fiap.com/registrar-acesso?sucesso=false`. The request body is empty, as indicated by the message "This request does not have a body".

The response status is 200 OK, indicating success.

Testei o endpoint registrar-acesso 3 vezes com falha (sucesso=false). Como esperado, na terceira tentativa, a API atualizou o usuário para "ativo": false.

Teste 5 DELETE

The screenshot shows the Postman application interface. In the top navigation bar, there are links for Home, Workspaces, API Network, a search bar, and various icons for inviting users, upgrading, and managing environments. A message at the top indicates that the current trial access ends in 23 days.

The main workspace displays a collection named "My Collection". Under this collection, there are two requests: "GET Get data" and "POST Post data". The "GET Get data" request is selected, showing its details. The method is set to "DELETE", the URL is "http://localhost:8080/api/usuarios/nathan.teste@fap.com", and the body is empty. The response section shows a status of "204 No Content" with a response time of 31 ms and a size of 112 B. The response body contains the number "1".

Consegui deletar o usuário (DELETE /api/usuarios/{email}) e recebi o status 204 No Content.

Modelo de Entidade-Relacionamento (MER)

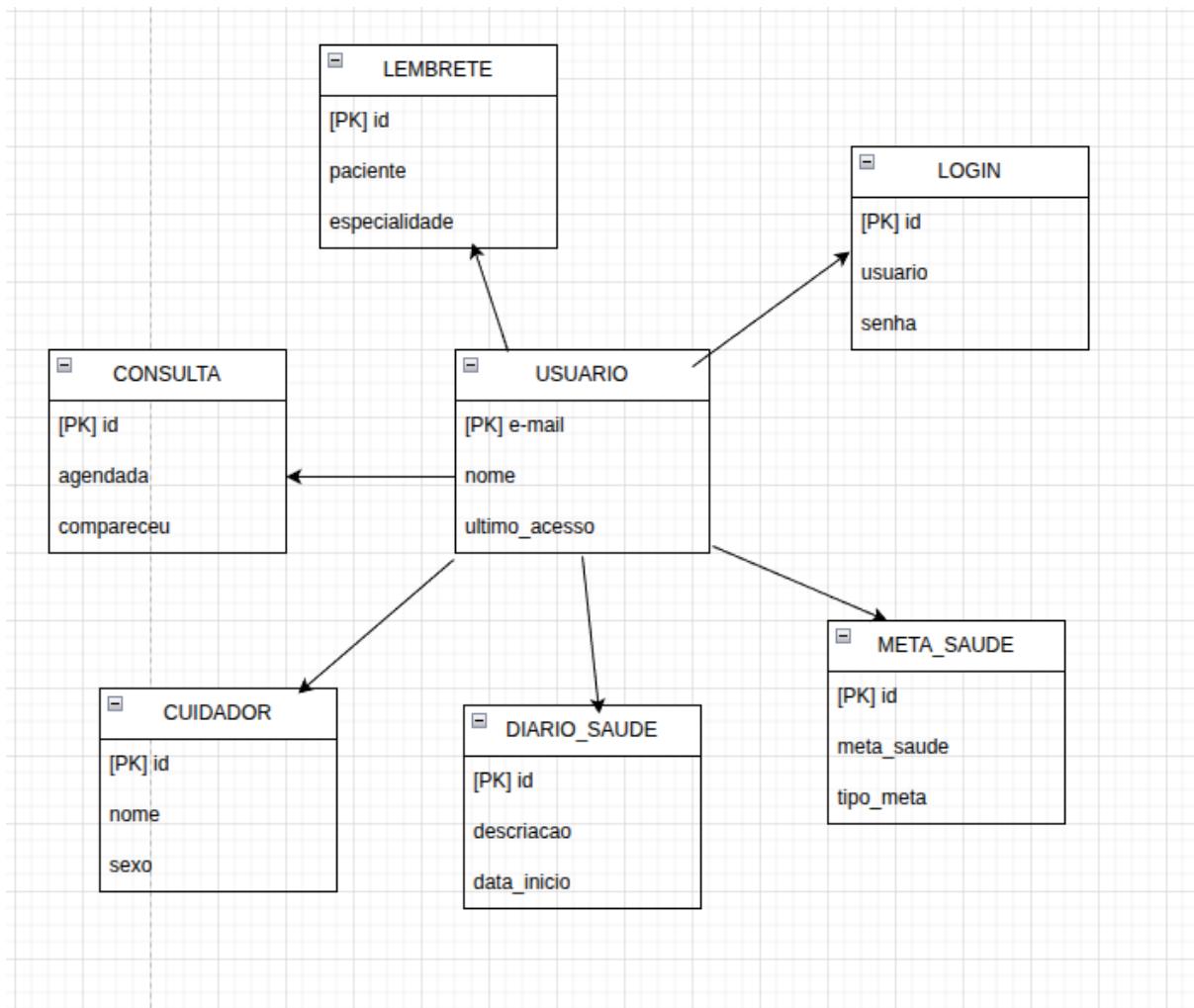


Diagrama de Classes Atualizado

