

GRUPO VITALI

Odontoprev - Challenge

**Celeste Mayumi Pereira Tanaka
Lívia Mariana Lopes
Luana Vieira Santos da Silva**

Sumário

Descrição do projeto	-----	03
Cronograma de desenvolvimento	-----	04
Modelagem - Database	-----	07
Diagrama de Classes	-----	08
Tabela dos endpoints	-----	09
Github	-----	11

Descrição do projeto

> Reconhecimento facial

Ao iniciar e finalizar a consulta, o dentista deve solicitar o reconhecimento facial do paciente, onde ele obrigatoriamente deve inserir no seu aplicativo. Caso o dentista não registre o reconhecimento facial, ele será classificado como suspeito pelo DigitalCop e estará passível de multa.

> “DigitalCop”

As consultas registradas no banco de dados serão analisadas para identificar possíveis padrões relacionados a fraude ou que sejam potenciais fraudes. Grande discrepância entre o tempo de consulta, procedimentos realizados e valor cobrado; queremos utilizar a IA para identificar esse padrão. Ela pode monitorar esses padrões de consulta, dentista ou paciente, reportando à Odontoprev se aquele usuário é suspeito ou não.

> Confirmação de consulta

Nossa solução contará com uma implementação Mobile (aplicativo), onde o usuário - o paciente - poderá acompanhar o ‘tracking’ da consulta e, ao final, deverá confirmar que realmente esteve presente no encontro com o profissional.

Cronograma de desenvolvimento

- **Celeste Mayumi Pereira Tanaka:** Responsável pela API em C# e desenvolvimento do modelo preditivo.
- **Lívia Mariana Lopes:** Responsável pela API em Java e DevOps.
- **Luana Vieira Santos da Silva:** Responsável pelo desenvolvimento do banco de dados e compliance e quality assurance do projeto.

Sprint 1

Lívia:

- Configuração do ambiente para desenvolvimento da API em Java.
- Estudo e planejamento da infraestrutura DevOps (CI/CD).
- Início do desenvolvimento dos endpoints principais da API.

Celeste:

- Prototipação das telas no Figma.
- Criação do ambiente de desenvolvimento em C#.
- Planejamento do desenvolvimento do modelo preditivo.

Luana:

- Modelagem do banco de dados (diagrama DER).
- Criação do ambiente de banco de dados.
- Declaração de visão e escopo do projeto.

Sprint 2

Lívia:

- Desenvolvimento completo de todas as entidades da API (CRUD).
- Realizar o deployment de máquinas virtuais.
- Diagrama demonstrando a integração/comunicação do aplicativo com APIs/banco de dados e sistemas externos.

Celeste:

- Implementação da Camada Web (ASP.NET Core) .
- Criação da versão Beta do modelo.
- Programação visual de 5 telas do aplicativo funcional.

Luana:

- Desenho da Arquitetura da solução.
- Procedures para Operações CRUD.
- Realizar chamadas a uma API (pelo menos 3 telas fazendo requisições).

Sprint 3

Lívia:

- Refatoração e otimização da API (melhoria de performance, validações extras, tratamento de erros).
- Implementação do pipeline CI/CD e automação de testes.

Celeste:

- Integração do modelo preditivo com a camada web (ASP.NET Core).
- Testes do modelo preditivo com base em diferentes cenários e ajustes necessários.

Luana:

- Implementação final das procedures e triggers no banco de dados para garantir integridade dos dados.
- Integração final do aplicativo mobile com a API.

Sprint 4

Lívia:

- Realizar testes de integração end-to-end para garantir que a API, banco de dados e front-end estão funcionando perfeitamente juntos.
- Deploy da API e testes finais.

Celeste:

- Finalização e validação do front-end integrado ao modelo preditivo e à API.
- Ajustes finais no front-end com base nos testes de usabilidade.

Luana:

- Revisão final do fluxo do aplicativo e ajustes com base no feedback dos testes.
- Revisão final do banco de dados.

Modelagem - Database

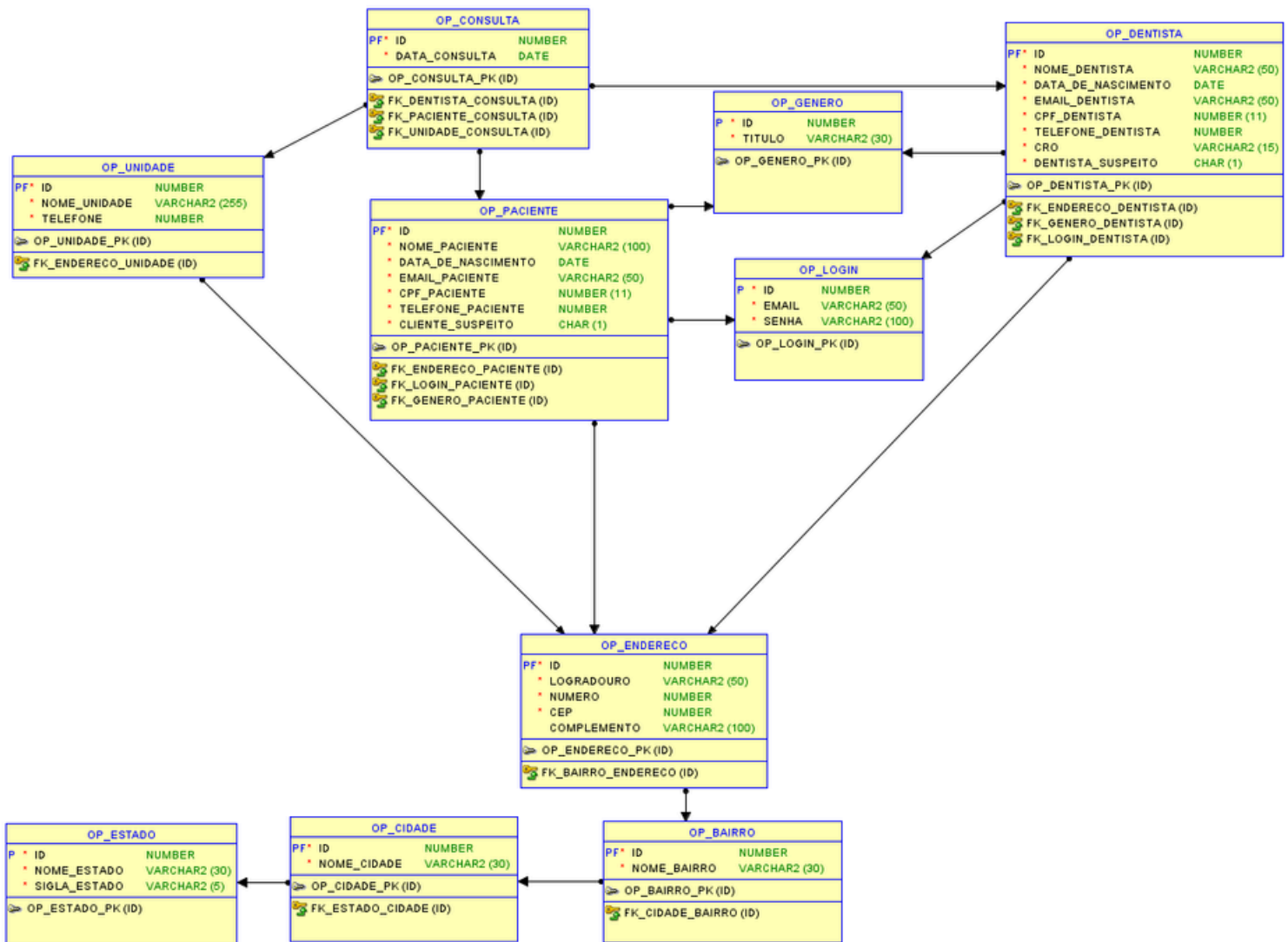


Diagrama de Classes

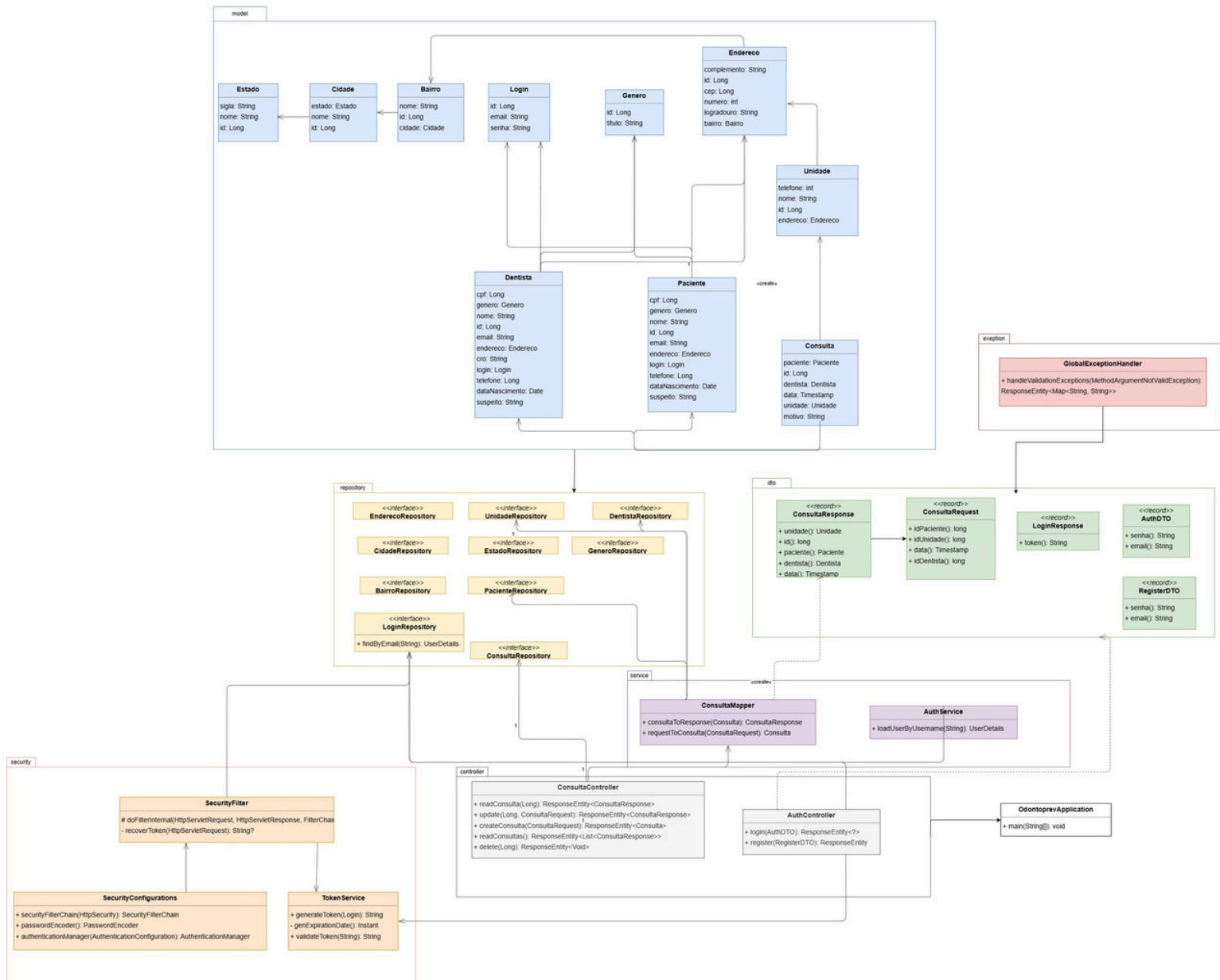


Tabela dos endpoints

Método	Endpoint	Descrição	Request Body	Parâmetros	Cabeçalho	Respostas
POST	/consultas	Cria uma nova consulta	{ "data": "02/11/2024 15:30", "idPaciente": 1, "idDentista": 2, "idUnidade": 3, "motivo": "Motivo" }	N/A	Authorization : Bearer {token}	201: Consulta criada com sucesso 400: Erros de validação
GET	/consultas	Retorna todas as consultas cadastradas	N/A	N/A	Authorization : Bearer {token}	200: Lista de consultas 204: Nenhuma consulta encontrada
GET	/consultas/{id}	Retorna uma consulta específica pelo ID	N/A	id: ID da consulta	Authorization : Bearer {token}	200: Consulta encontrada 204: Nenhuma consulta encontrada
PUT	/consultas/{id}	Atualiza uma consulta existente	{ "data": "05/10/2024 10:00", "idPaciente": 1, "idDentista": 2, "idUnidade": 3, "motivo": "Motivo" }	id: ID da consulta	Authorization : Bearer {token}	200: Consulta atualizada com sucesso 400: Consulta não encontrada ou dados inválidos

Tabela dos endpoints

Método	Endpoint	Descrição	Request Body	Parâmetros	Cabeçalho	Respostas
DELETE	/consultas/{id}	Exclui uma consulta pelo ID	N/A	id: ID da consulta	Authorization: Bearer {token}	200: Consulta excluída com sucesso 400: Consulta não encontrada
POST	/auth/register	Registra um novo usuário	{ "nome": "Usuario", "email": "usuario@example.com", "senha": "senha123" }	N/A	N/A	201: Usuário registrado com sucesso 400: Erros de validação
POST	/auth/login	Autentica um usuário	{ "email": "usuario@example.com", "senha": "senha123" }	N/A	N/A	200: Login realizado com sucesso, retorna token de autenticação 401: Credenciais inválidas

Github

Você pode acessar o código-fonte do projeto, juntamente com o arquivo JSON contendo as requisições de teste do Postman, no repositório GitHub a seguir:

<https://github.com/LiviaMarianaLopes/odontoprev-api>