

## **CHALLENGE ODONTOPREV – ODONTOFAST**

**Projeto:** Aplicação de Acompanhamento para Incentivo ao Tratamento Preventivo

### **INTEGRANTES (2TDSPS)**

Felipe Amador RM: 553528

Leonardo de Oliveira RM: 554024

Sara Sousa RM: 552656

## Sumário

Projeto de Acompanhamento Odontológico.....	3
Descrição Geral da Solução.....	3
Descrição do Projeto Java Spring Boot .....	4
Visão Geral .....	4
Funcionalidades .....	4
Tecnologias Utilizadas.....	5
Estrutura do Projeto.....	5
Classes UML .....	6
Modelo Conceitual.....	7
Estrutura do Banco de Dados e Relacionamentos.....	8
Entidades Principais: .....	8
Relacionamentos: .....	9
Conclusão.....	10

# Projeto de Acompanhamento Odontológico

## Descrição Geral da Solução

A solução de software desenvolvida para a empresa Odontoprev visa incentivar os pacientes a adotarem hábitos e tratamentos preventivos, especialmente na faixa etária com maior incidência de problemas bucais (25 a 45 anos). O aplicativo móvel permitirá que os pacientes monitorem seus tratamentos e consultas, além de oferecer um checklist para registrar seus cuidados diários, gerando incentivos para manter a rotina, com notificações personalizadas desempenhando um papel importante.

Além disso, a solução inclui um website que funcionará como um portal administrativo, onde o dentista terá acesso às informações do paciente, como cadastro, tratamentos, agendamentos e hábitos. O dentista poderá incluir agendamentos e detalhes dos tratamentos que serão realizados. Tanto o aplicativo quanto o website oferecerão dashboards que mostram um panorama geral da saúde bucal do paciente.

### Visão Geral

OdontoFast backend é uma aplicação desenvolvida em Java utilizando o framework Spring Boot, projetada para otimizar a gestão de acompanhamento. O sistema visa melhorar a comunicação entre dentistas e pacientes, proporcionando uma experiência mais eficiente e integrada no agendamento de consultas e no acompanhamento de tratamentos. Planejamos construir através do projeto Java, o backend e API necessários para serem consumidos no projeto Mobile e Website (admin), e assim ter uma integração absoluta em todo o escopo definido da solução, incluindo a persistência no banco de dados Oracle.

### Funcionalidades

A aplicação visa oferecer futuramente uma estrutura com diversas funcionalidades para pacientes e administradores (dentistas ou afins), incluindo:

- **Cadastro de Usuários:** Permite que pacientes e dentistas sejam registrados na plataforma, facilitando o gerenciamento de informações pessoais e profissionais.
- **Agendamento de Consultas:** Os pacientes podem acompanhar as consultas, enquanto os dentistas têm acesso à lista de agendamentos e podem agendar essas consultas.
- **Notificações:** O sistema envia notificações para lembrar os pacientes sobre suas consultas e tratamentos programados, melhorando a adesão ao tratamento.
- **Gerenciamento de Tratamentos:** Dentistas podem criar e gerenciar planos de tratamento para seus pacientes, com detalhes sobre cada procedimento realizado.
- **Integração com Planos de Saúde:** O sistema permite login para pacientes que possuem planos de saúde, facilitando o acesso a informações relevantes sem a necessidade de cadastro adicional.

## Tecnologias Utilizadas

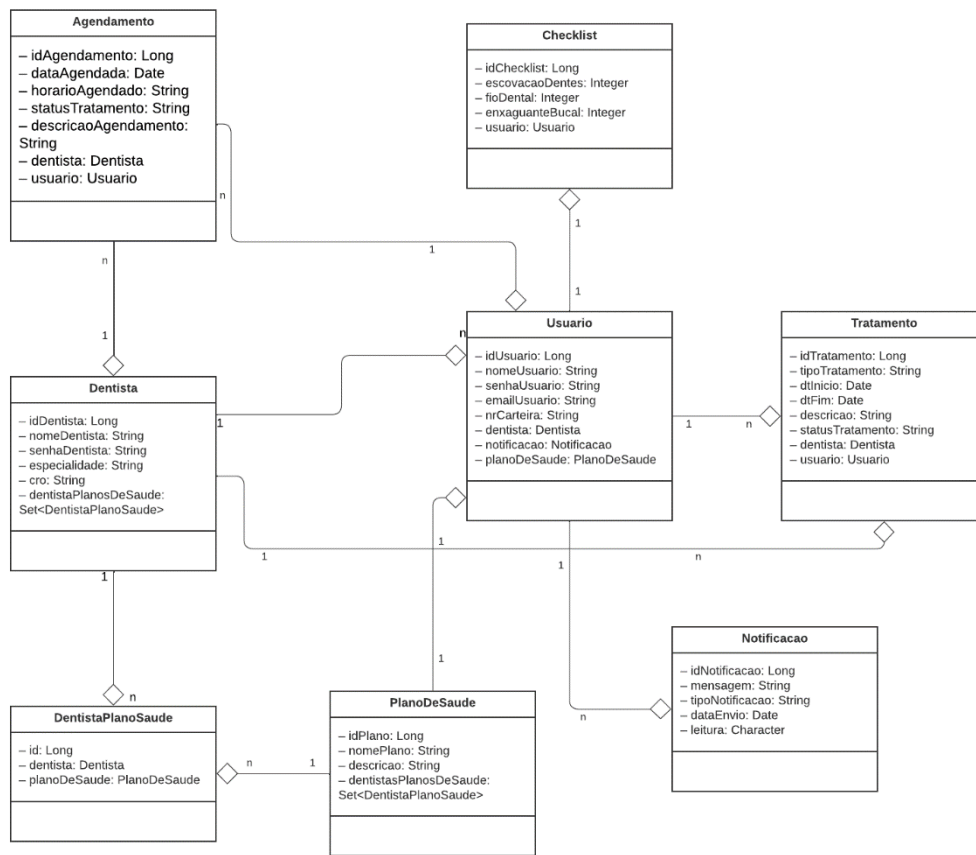
- **Java:** Linguagem de programação utilizada para o desenvolvimento do backend.
- **Spring Boot:** Framework que simplifica o desenvolvimento de aplicações Java, permitindo a construção de APIs RESTful.
- **JPA (Java Persistence API):** Utilizado para gerenciar a persistência dos dados em um banco de dados relacional.
- **Oracle Database:** Banco de dados utilizado para armazenar as informações da aplicação.
- **Maven:** Ferramenta de automação de compilação que gerencia as dependências do projeto.
- **Lombok:** Biblioteca que ajuda a reduzir o código boilerplate, como getters e setters.

## Estrutura do Projeto

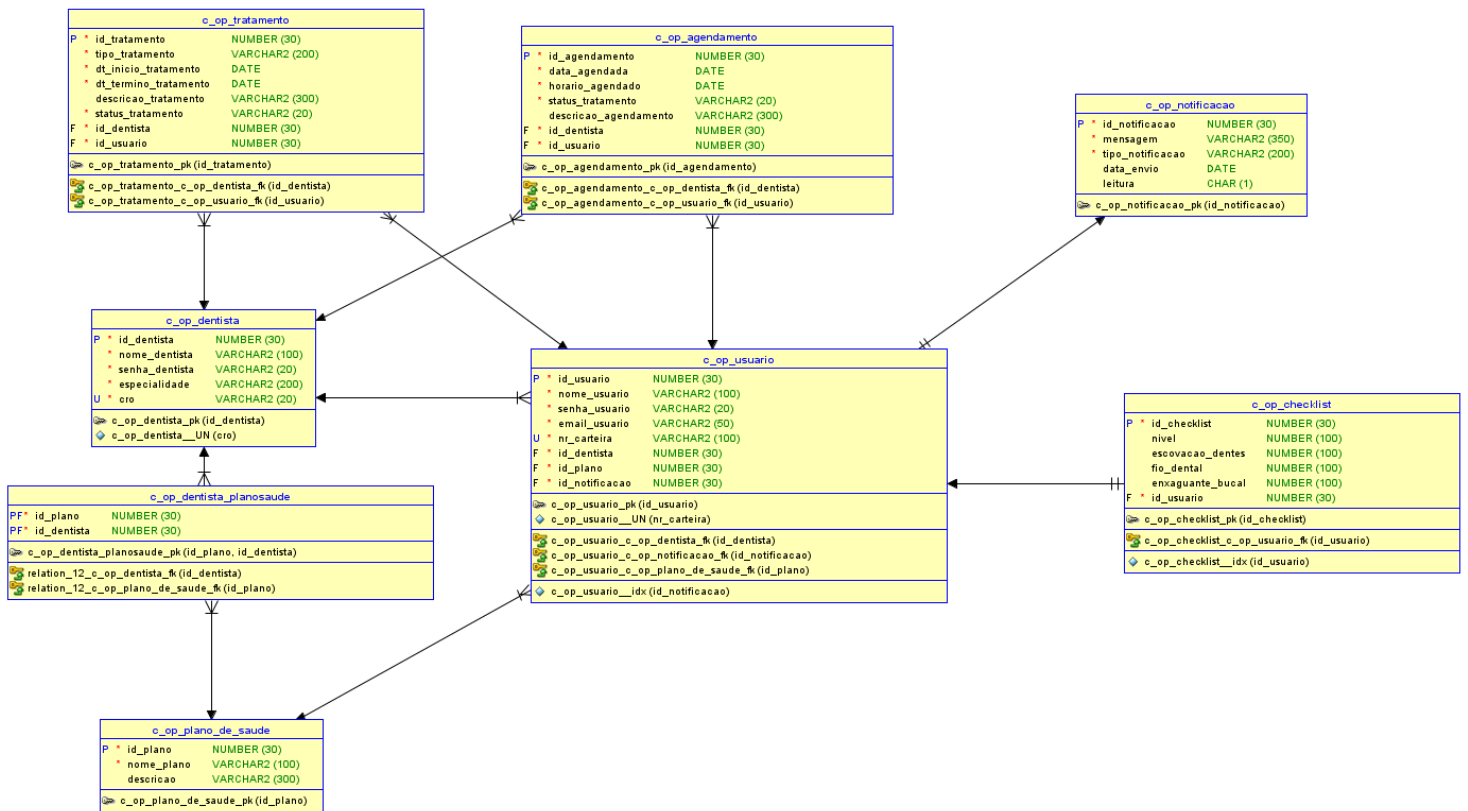
A estrutura do projeto é organizada em pacotes, cada um responsável por diferentes camadas da aplicação:

- **Controller:** Gerencia as requisições e respostas HTTP.
- **Service:** Contém a lógica de negócios da aplicação.
- **Repository:** Responsável pela interação com o banco de dados.
- **Entity:** Define as entidades do modelo de dados.
- **DTO (Data Transfer Object):** Objetos utilizados para transferir dados entre as camadas da aplicação.

## Classes UML



## Modelo Conceitual



## Entidades Principais:

- **Usuário**
  - id\_usuario (PK)
  - nome\_usuario
  - email\_usuario
  - senha\_usuario
  - nr\_carteira (unique)
  - plano\_id (FK)
  - id\_dentista (FK)
  - id\_notificacao (PK)
  
- **Plano de Saúde**
  - id\_plano (PK)
  - nome\_plano
  - descrição\_plano
  
- **Dentista**
  - id\_dentista (PK)
  - nome\_medico
  - senha\_dentista(hash)
  - especialidade
  - cro (unique)
  
- **Tratamento**
  - id\_tratamento (PK)
  - tipo\_tratamento (limpeza, canal, etc.)
  - dt\_inicio\_tratamento
  - dt\_termino\_tratamento
  - prazo\_tratamento
  - descricao\_tratamento
  - status\_tratamento (pendente, em andamento, concluído, cancelado)
  - id\_usuario (FK)
  - id\_dentista (FK)



- **Agendamento**
  - id\_agendamento (PK)
  - tratamento\_id (FK)
  - data\_agendada
  - horario\_agendado
  - status\_agendamento (pendente, concluído, cancelado)
  - id\_usuario (FK)
  - id\_dentista (FK)
  
- **Notificação**
  - id\_notificacao (PK)
  - id\_usuario (FK)
  - mensagem
  - tipo\_notificacao
  - data\_envio
  - leitura (boolean)
  
- **Check-List**
  - id\_checklist (PK)
  - nível
  - escovação\_dentes
  - fio\_dental
  - enxaguante\_bucal
  - id\_usuario (FK)

## Relacionamentos:

- Um **Usuário** pode ter um ou mais **Agendamentos** (relação um-para-muitos).
- Um **Usuário** pode ter um **Plano de saúde** (relação um-para-um).
- Um **Usuário** pode ter um ou mais **Tratamentos** (relação um-para-muitos).
- Um **Dentista** pode estar associado a um ou mais **Planos de saúde** e Um **Plano de saúde** pode estar associado a um ou mais **Dentistas** (relação muitos-para-muitos, que será representada como uma entidade associativa).
- Um **Dentista** pode ter um ou mais **Tratamentos** (relação um-para-muitos).
- Um **Usuário** pode ter um ou mais **Agendamentos** (relação um-para-muitos).
- Um **Dentista** pode ter um ou mais **Agendamentos** (relação um-para-muitos).
- Um **Usuário** pode ter um ou mais **Notificações** (relação um-para-muitos).

- Um **Usuário** pode ter um **CheckList** (relação um-para-um).

## Conclusão

O OdontoFast foi desenvolvido com o objetivo de facilitar a experiência de atendimento em clínicas odontológicas e incentivar os usuários à adesão de tratamentos e hábitos preventivos. A aplicação é voltada tanto para pacientes quanto para profissionais de saúde com foco para gerenciamento de consultas e tratamentos.