**Faculdade de Informática e Administração Paulista**

**CHALLENGE ODONTOPREV – ODONTOFAST**

**4 Sprint**

**Projeto:** Aplicação de Acompanhamento para Incentivo ao Tratamento Preventivo**Interface gráfica do usuário

Descrição gerada automaticamente**

**MASTERING RELATIONAL AND NON-RELATIONAL DATABASE**

**INTEGRANTES (2TDSPS)**

Felipe Amador RM: 553528

Leonardo de Oliveira RM: 554024

Sara Sousa RM: 552656

**São Paulo**

**Maio/2025**

**SUMÁRIO**

Sumário

[**Descrição do Projeto** 3](#_Toc197531212)

[**Modelo de Dados e justificativa** 3](#_Toc197531213)

[**Interface de Consulta de Dados** 4](#_Toc197531214)

[**Operações para CRUD** 5](#_Toc197531215)

[**Exportação de Dataset** 7](#_Toc197531216)

[**Análise de Performance e Escalabilidade** 7](#_Toc197531217)

[**Segurança e Integração** 7](#_Toc197531218)

**Projeto de Acompanhamento Odontológico**

**Descrição do Projeto**

O projeto OdontoFast, desenvolvido para a Odontoprev, visa incentivar pacientes a adotarem hábitos e tratamentos odontológicos preventivos, com foco especial na faixa etária de 25 a 45 anos, onde há maior incidência de problemas bucais. A solução consiste em um aplicativo móvel que permite aos pacientes monitorarem seus tratamentos e consultas, além de oferecer um checklist para registro de cuidados diários, gerando incentivos para manutenção da rotina.

A escolha do MongoDB como banco de dados NoSQL para este projeto se justifica por:

1. **Flexibilidade de esquema**: A natureza do documento JSON permite adicionar facilmente novos campos sem afetar registros existentes, ideal para evolução do aplicativo.
2. **Modelagem de dados aninhados**: Permite representar relacionamentos naturais (como usuário-checklist) como documentos aninhados, reduzindo a necessidade de joins.
3. **Escalabilidade horizontal**: MongoDB suporta sharding nativo, essencial para um aplicativo que pode crescer rapidamente em número de usuários.
4. **Desempenho em operações de leitura**: Ideal para dashboards e consultas frequentes que serão realizadas pelos dentistas e pacientes.
5. **Suporte a geolocalização**: Facilita a implementação de recursos como localização de clínicas próximas.

**Modelo de Dados e justificativa**

**Coleções Principais:**

usuarios: Armazena informações dos pacientes

dentistas: Armazena informações dos dentistas

tratamentos: Registra todos os tratamentos odontológicos

agendamentos: Gerencia consultas agendadas

planos: Informações sobre planos de saúde odontológicos

**Justificativas para a Estrutura:**

**- Documentos aninhados vs. Referências:** Utilizamos documentos aninhados para relacionamentos fortes (como usuário-checklist) e referências para relacionamentos mais flexíveis (como tratamento-agendamento).

Exemplo prático: Quando o aplicativo carrega o perfil do paciente, ele precisa imediatamente mostrar seu nível atual de checklist. Com o documento aninhado, apenas uma consulta ao banco de dados é necessária, em vez de duas consultas separadas (uma para o usuário e outra para o checklist).

**- Desnormalização estratégica:** Duplicamos algumas informações (como status e tipo de plano) para reduzir consultas entre coleções.

Exemplo prático: Um dentista consultando a lista de agendamentos do dia verá imediatamente o nome do paciente em cada item, sem que o sistema precise fazer uma consulta separada para buscar essa informação.

**- Campos compostos**: Utilizamos estruturas compostas para agregar dados relacionados, como endereços e informações de contato.

Exemplo prático: Quando um paciente muda de endereço, o sistema atualiza toda a estrutura do endereço como uma unidade lógica, mantendo a consistência dos dados. Da mesma forma, quando o aplicativo precisa mostrar as informações de contato de uma clínica, todos os dados necessários estão organizados em um único bloco coeso.

**Interface de Consulta de Dados**

-Listar todos os usuários

db.usuarios.find({})

-Buscar usuário por nome (usando índice de texto)

db.usuarios.find({ $text: { $search: "Carlos" } })

-Buscar pacientes que possuem nível 3 ou maior no checklist

db.usuarios.find({ "checklist.nivel": { $gte: 3 } })

-Buscar tratamentos ativos de um determinado usuário

db.tratamentos.find({

"id\_usuario": ObjectId("5f8a9b1d2c3e4f5a6b7c8d9e"),

"status.codigo": 1

})

-Buscar agendamentos para a próxima semana

db.agendamentos.find({

"data\_agendada": {

$gte: ISODate("2025-10-06"),

$lt: ISODate("2025-10-13")

} } )

-Agregação para contagem de tratamentos por tipo

db.tratamentos.aggregate([

{ $group: { \_id: "$tipo\_tratamento", quantidade: { $sum: 1 } } },

{ $sort: { quantidade: -1 } } ])

**Operações para CRUD**

- CREATE - Inserir novo usuário

db.usuarios.insertOne({

"nome": "Sara Gabriele",

"email": "sara.gabriele@yahoo.com",

"senha": "64578789578",

"telefone": "(11) 97654-3210",

"nr\_carteira": "0987654321",

"data\_nascimento": ISODate("1990-05-20"),

"ultimo\_acesso": ISODate("2025-09-30T09:45:00Z"),

"endereco": {

"logradouro": "Rua Oscar Freire",

"numero": "554",

"bairro": "Jardins",

"cidade": "São Paulo",

"estado": "SP",

"cep": "01426-000"

},

"checklist": {

"nivel": 2,

"escovacao\_dentes": 2,

"fio\_dental": 2,

"enxaguante\_bucal": 1,

"ultima\_atualizacao": ISODate("2025-10-02T08:00:00Z")

}

})

- READ - Buscar usuário por ID

db.usuarios.findOne({ \_id: ObjectId("5f8a9b1d2c3e4f5a6b7c8d9e") })

- UPDATE - Atualizar o endereço completo de um usuário

db.usuarios.updateOne(

{ \_id: ObjectId("5f8a9b1d2c3e4f5a6b7c8d9e") }, // ID do Carlos Matos

{

$set: {

"endereco": {

"logradouro": "Alameda Santos",

"numero": "455",

"complemento": "Apto 87",

"bairro": "Cerqueira César",

"cidade": "São Paulo",

"estado": "SP",

"cep": "01419-000"

}

}

}

)

*- UPDATE - Adicionar um novo plano de saúde à lista de planos atendidos por um dentista*

db.dentistas.updateOne(

{ \_id: ObjectId("5f8a9b1d2c3e4f5a6b7c8d9f") },

{

$push: {

"planos\_atendidos": {

"id\_plano": ObjectId("5f8a9b1d2c3e4f5a6b7c8d2d"),

"nome\_plano": "Amil Dental",

"data\_adesao": ISODate("2025-10-05")

}

}

}

)

- DELETE - Remover um agendamento

db.agendamentos.deleteOne({ \_id: ObjectId("5f8a9b1d2c3e4f5a6b7c8d9h") })

**Exportação de Dataset**

mongoexport --db odontofast --collection usuarios --out usuarios.json --jsonArray –pretty

**Análise de Performance e Escalabilidade**

Para garantir a eficiência do sistema OdontoFast no MongoDB, podemos implementar estratégias específicas de performance e escalabilidade. A indexação estratégica seria aplicada nos campos mais consultados, como email de usuários, CRO de dentistas e campos de relacionamento (id\_usuario, id\_dentista). Também um índice composto para otimizar a busca de agendamentos por data e dentista, e índices de texto para facilitar buscas por nome.

*Índices básicos para melhorar performance*

*db.usuarios.createIndex({ "email": 1 })*

*db.tratamentos.createIndex({ "id\_usuario": 1 })*

*db.agendamentos.createIndex({ "data\_agendada": 1 })*

As consultas foram otimizadas utilizando projeções para limitar campos retornados, reduzindo o volume de dados transferidos, especialmente importante para o aplicativo móvel.

Projeção para limitar campos retornados

db.tratamentos.find(

{ "id\_usuario": ObjectId("5f8a9b1d2c3e4f5a6b7c8d9e") },

{ tipo\_tratamento: 1, status: 1 }

)

A estratégia de escalabilidade foi pensada em duas etapas realistas para o contexto acadêmico:

1. **Fase inicial (MVP)**: Utilização do MongoDB Atlas na camada gratuita, adequada para desenvolvimento e testes com até 100 usuários simultâneos.
2. **Expansão**: Migração para um plano compartilhado do MongoDB Atlas (aproximadamente $9/mês), permitindo suportar até 500 usuários simultâneos, com possibilidade de adicionar réplicas para distribuir operações de leitura.

Para monitoramento da performance, utilizamos o MongoDB Compass, ferramenta gratuita que permite visualizar estatísticas de execução de queries e uso de índices.

**Segurança e Integração**

A segurança do OdontoFast foi projetada considerando a sensibilidade dos dados médicos. Implementamos um sistema de autenticação com usuários específicos para diferentes componentes (app\_mobile\_user, admin\_portal\_user, backup\_admin), seguindo o princípio de privilégio mínimo. A proteção dos dados inclui criptografia em trânsito (TLS/SSL), criptografia em repouso e hash de senhas com bcrypt.

Para integração, podemos desenvolver uma API RESTful centralizada que serve como interface única para interações com o banco de dados, garantindo consistência nas operações. Foram criados pontos de integração com:

1. Sistemas de agendamento (Google Calendar, Microsoft Outlook)
2. Sistemas de planos de saúde para validação de cobertura
3. Provedores de notificação push para lembretes

Esta arquitetura permite que o OdontoFast funcione como hub central na gestão da saúde bucal, conectando-se eficientemente com o ecossistema de saúde digital mais amplo, mantendo a segurança e integridade dos dados em todos os pontos de integração.