

PÓS-GRADUAÇÃO EM DESENVOLVIMENTO FULL STACK

HealthPass – Sistema único de exames médicos e vacinais

FILIPPE DOMENICO LA SELVA

Orientador: Professor Alexandre Agustini

2024/2025

SUMÁRIO

1. CONTEXTUALIZAÇÃO DA PROPOSTA.....	3
2. OBJETIVOS DA CONSTRUÇÃO DA SOLUÇÃO	5
3. ELABORAÇÃO DA JORNADA DO USUÁRIO.....	8
4. APELO MERCADOLÓGICO DA SOLUÇÃO	12
5. CICLO DE DESENVOLVIMENTO DA SOLUÇÃO	17
6. MOCKUP DA PROPOSTA DE SOLUÇÃO	18
7. ARQUITETURA DE SOFTWARE	19
8. VALIDAÇÃO DA SOLUÇÃO	23
9. REGISTROS DAS EVIDÊNCIAS DO PROJETO	41
10. CONSIDERAÇÕES FINAIS E EXPECTATIVAS.....	42
REFERÊNCIAS	44

1. CONTEXTUALIZAÇÃO DA PROPOSTA

Em um mundo cada vez mais digital e conectado, gerenciar as informações de saúde dos usuários de forma eficiente se tornou essencial. Um sistema unificado de exames e vacinas, que não dependa de prazos e da expiração de relatórios após o término de contratos, traz diversos benefícios tanto para os usuários quanto para os profissionais de saúde que podem ser elencados em 5 (cinco) pilares principais:

Continuidade do Cuidado e Histórico Completo

Quando os relatórios de exames e vacinas perdem a validade após o fim de um contrato, os usuários ficam sem acesso a informações importantes sobre sua saúde. Um sistema unificado mantém todo o histórico de saúde, garantindo que os profissionais possam tomar decisões mais informadas com base em dados completos, independentemente do prestador de serviços anterior. Essa continuidade no cuidado melhora significativamente o diagnóstico, o tratamento e a prevenção de doenças.

Empoderamento do Usuário

O acesso constante aos registros de exames e vacinas permite que os usuários sejam mais proativos na gestão de sua saúde. Sem a preocupação com prazos de expiração, eles podem consultar informações sempre que necessário, facilitando o acompanhamento de tratamentos, a adesão ao calendário vacinal e a realização de check-ups regulares. Esse empoderamento fortalece a relação entre pacientes e profissionais de saúde, promovendo um engajamento mais efetivo.

Integração e Interoperabilidade

Um sistema unificado elimina barreiras entre diferentes plataformas e prestadores de serviços de saúde. A interoperabilidade facilita a troca de informações de forma segura e eficiente, permitindo que hospitais, clínicas e laboratórios acessem dados relevantes em tempo real. Isso reduz redundâncias, evita a repetição desnecessária de exames e agiliza processos administrativos.

Segurança e Privacidade dos Dados

Em vez de ter vários sistemas com políticas de retenção de informações diferentes, um sistema centralizado pode assegurar padrões mais elevados de segurança e conformidade com as leis de proteção de dados. Manter os registros sem um prazo de expiração atrelado a contratos oferece uma abordagem mais sólida para proteger a privacidade dos usuários, garantindo que eles continuem sendo os proprietários de suas informações de saúde.

Eficiência Operacional e Redução de Custos

A gestão fragmentada de dados gera custos extras tanto para os usuários quanto para os prestadores de serviços. Com um sistema unificado, a eficiência operacional aumenta, reduzindo a necessidade de recursos para reprocessar ou recuperar informações perdidas. Além disso, a otimização dos processos administrativos e a diminuição da duplicidade de exames resultam em uma economia significativa para todo o ecossistema de saúde.

2. OBJETIVOS DA CONSTRUÇÃO DA SOLUÇÃO

A implementação de um sistema unificado de exames e vacinas no Brasil, baseado em uma API robusta e interoperável, representa um avanço significativo para a modernização do ecossistema de saúde. Esse sistema não só facilitaria o acesso contínuo e seguro às informações de saúde, como também traria inúmeros benefícios para a gestão pública, prestadores de serviços e para os próprios cidadãos. Abaixo, exploramos como essa integração pode transformar o cenário brasileiro e os aprendizados adquiridos no desenvolvimento desta API.

Melhoria da Eficiência no Ecossistema de Saúde Brasileiro Integração entre Sistemas de Saúde

Atualmente, o Brasil enfrenta um grande desafio com sistemas fragmentados entre o SUS (Sistema Único de Saúde), hospitais privados e clínicas. A criação de uma API única permitiria a integração desses dados, facilitando a troca segura de informações entre diferentes plataformas. Isso reduziria a duplicidade de exames, otimizando recursos e acelerando os processos de diagnóstico e tratamento. **Redução de Custos** Com um sistema unificado, a repetição de exames e a necessidade de realizar procedimentos novamente devido à falta de informações seriam significativamente diminuídas. Isso representaria uma economia considerável para o SUS e para os planos de saúde privados, permitindo um melhor direcionamento dos recursos financeiros para outras áreas críticas.

Melhoria no Atendimento ao Paciente

O acesso contínuo ao histórico de exames e vacinas possibilita diagnósticos mais precisos e tratamentos personalizados. Além disso, os profissionais de saúde

teriam acesso a informações completas e atualizadas em tempo real, garantindo um atendimento mais rápido e eficaz.

Benefícios do Aprendizado Adquirido na Implementação da API

A. Interoperabilidade e Padrões Abertos Durante o desenvolvimento da API, a necessidade de adotar padrões abertos, como FHIR (Fast Healthcare Interoperability Resources), proporcionou um aprendizado significativo sobre interoperabilidade. Essa experiência não apenas facilitou a integração entre sistemas distintos, mas também incentivou a criação de protocolos de comunicação mais seguros e eficientes para o compartilhamento de dados de saúde.

B. Segurança e Conformidade com a LGPD A criação de uma API única exigiu um aprofundamento nas práticas de segurança e privacidade de dados, em conformidade com a LGPD (Lei Geral de Proteção de Dados). As lições aprendidas nesse processo fortaleceram a capacidade do sistema de proteger dados sensíveis, assegurando que as informações dos usuários sejam tratadas com confidencialidade e transparência.

C. Escalabilidade e Performance A necessidade de gerenciar grandes volumes de dados em tempo real levou ao aprendizado de técnicas avançadas de escalabilidade, como microsserviços e arquiteturas baseadas em cloud computing. Isso garantiu que a API pudesse suportar milhões de usuários simultâneos sem comprometer a performance.

D. Experiência do Usuário (UX) e Acessibilidade Durante o desenvolvimento da API, houve um foco especial na experiência do usuário, assegurando que o acesso aos dados de saúde fosse simples, rápido e intuitivo. Além disso, a API foi projetada para ser inclusiva, garantindo que pessoas com deficiência também pudessem aproveitar os benefícios da plataforma.

Transformação Digital do SUS Gestão Proativa da Saúde

Com um sistema unificado, o SUS poderia adotar uma abordagem mais proativa, utilizando dados em tempo real para monitorar surtos epidemiológicos, rastrear a cobertura vacinal e identificar populações em risco. Esse monitoramento inteligente permitiria intervenções mais rápidas e eficazes.

Base de Dados Unificada para Pesquisa

A centralização dos dados de exames e vacinas criaria uma base rica para pesquisas científicas, facilitando estudos epidemiológicos, análises preditivas e o desenvolvimento de políticas públicas baseadas em evidências.

3. ELABORAÇÃO DA JORNADA DO USUÁRIO

Caso de Uso: Atendimento de Emergência

Cenário Atual (Sem o Sistema Unificado)

Um paciente sofre um acidente e é levado a um hospital onde nunca foi atendido antes. Os médicos não têm acesso ao seu histórico médico, o que os impede de conhecer alergias, doenças pré-existentes e medicações em uso. Isso pode atrasar o tratamento e aumentar o risco de complicações.

Com o Sistema Unificado

No momento da internação, os médicos conseguem acessar rapidamente os dados do paciente no sistema, incluindo exames recentes, alergias e histórico de vacinas. Com essas informações em mãos, podem tomar decisões mais rápidas e seguras, garantindo um tratamento adequado e evitando reações adversas.

Vantagem: Agilidade e segurança no atendimento emergencial, reduzindo riscos e melhorando a resposta médica.

Caso de Uso: Controle e Atualização de Vacinas

Cenário Atual (Sem o Sistema Unificado)

Um pai precisa matricular seu filho na escola, mas perdeu a caderneta de vacinação. Para resolver a situação, ele precisa buscar o histórico no posto de saúde onde a criança tomou as vacinas, enfrentando filas e burocracia.

Com o Sistema Unificado

O histórico vacinal do aluno pode ser acessado digitalmente pelo responsável e compartilhado com a escola diretamente pelo sistema. O próprio usuário recebe alertas automáticos quando vacinas estão atrasadas, garantindo um melhor controle da imunização.

Vantagem: Facilidade no acesso ao histórico vacinal e melhor controle da imunização, reduzindo riscos de surtos de doenças.

Caso de Uso: Consultas e Exames Sem Replicação Desnecessária

Cenário Atual (Sem o Sistema Unificado)

Uma paciente consulta um ginecologista particular, realiza exames laboratoriais e, após alguns meses, precisa de atendimento pelo SUS. Como os exames anteriores não estão disponíveis para o novo médico, ele solicita novos exames, resultando em custos desnecessários e atrasos no tratamento.

Com o Sistema Unificado

O novo médico pode acessar diretamente os exames anteriores, evitando a repetição deles. Isso acelera a tomada de decisão e previne gastos desnecessários com exames duplicados.

Vantagem: Redução de custos e maior agilidade no atendimento médico.

Caso de Uso: Monitoramento de Doenças Crônicas

Cenário Atual (Sem o Sistema Unificado)

Pacientes com diabetes precisam manter um controle rigoroso de seus exames periódicos. No entanto, muitos se perdem na organização dos exames e acabam realizando os testes com atraso, o que aumenta o risco de complicações.

Com o Sistema Unificado

O sistema envia alertas automáticos para lembrar os pacientes sobre a necessidade de novos exames, ajudando-os a manter um acompanhamento regular. Além disso, os médicos podem acessar rapidamente o histórico de glicemia e hemoglobina glicada para fazer ajustes no tratamento.

Vantagem: Melhor acompanhamento da saúde de pacientes crônicos, evitando agravamentos da doença.

Caso de Uso: Acesso de Médicos a Informações em Diferentes Estados **Cenário Atual (Sem o Sistema Unificado)**

Uma paciente que mora em São Paulo está em tratamento de câncer, mas precisa viajar para outro estado. Ao chegar, não tem acesso ao histórico de tratamentos e exames, o que dificulta a continuidade do atendimento.

Com o Sistema Unificado

O oncologista no novo estado pode acessar o histórico completo da paciente, incluindo quimioterapias já realizadas, exames de imagem e evolução do tratamento. Isso permite a continuidade do cuidado sem a necessidade de novos testes.

Vantagem: Maior mobilidade para os pacientes e continuidade dos tratamentos independentemente da localização.

Caso de Uso: Pesquisa e Análise Epidemiológica **Cenário Atual (Sem o Sistema Unificado)**

O Ministério da Saúde enfrenta o desafio de identificar surtos de doenças infecciosas e monitorar a cobertura vacinal. No entanto, a descentralização dos dados e a lentidão na consolidação dificultam uma resposta rápida e eficaz.

Com o Sistema Unificado

Os dados são atualizados em tempo real, permitindo que os órgãos públicos monitorem surtos de doenças, avaliem a eficácia das campanhas de vacinação e direcionem recursos para as áreas que mais precisam.

Vantagem: Um planejamento mais eficiente das políticas públicas de saúde e uma resposta mais rápida a emergências sanitárias.

A adoção de um sistema unificado de exames e vacinas oferece benefícios diretos à população brasileira, como maior agilidade no atendimento, redução de custos, melhor controle da saúde e uma gestão pública mais eficiente. Além disso, a digitalização dos dados de saúde possibilita um monitoramento epidemiológico eficaz, continuidade dos tratamentos independentemente da localização e um atendimento mais humanizado, com foco na prevenção e no bem-estar do paciente.

Essa transformação digital não apenas moderniza o setor de saúde, mas também torna o sistema mais acessível e eficiente para todos.

4. APELO MERCADOLÓGICO DA SOLUÇÃO

Solução Mercadológica para um Sistema Unificado de Exames e Vacinas

A criação de um **Sistema Unificado de Exames e Vacinas** representa uma oportunidade significativa de inovação no setor de saúde brasileiro, beneficiando tanto o setor público quanto o privado. Para garantir sua implementação bem-sucedida e ampla adoção, é fundamental desenvolver uma **estratégia mercadológica** que envolva **modelos de negócio sustentáveis, parcerias estratégicas e uma proposta de valor clara**.

Proposta de Valor

O sistema se posicionará como uma **plataforma centralizada e interoperável** para o armazenamento, compartilhamento e gestão de informações de saúde do paciente. Ele oferecerá benefícios para **usuários finais (pacientes), profissionais de saúde, hospitais, laboratórios, operadoras de planos de saúde e órgãos governamentais**.

Para os pacientes: Acesso rápido e seguro ao histórico médico, facilidade na realização de exames e vacinação, menos burocracia.

Para os médicos: Histórico completo do paciente disponível em tempo real, melhor tomada de decisão, redução de erros e exames duplicados.

Para clínicas e hospitais: Maior eficiência no atendimento, menor custo operacional, integração com laboratórios e operadoras de planos de saúde.

Para operadoras de planos de saúde: Redução de custos com exames repetidos, melhor controle da jornada do paciente, prevenção de fraudes e otimização da gestão de recursos.

Para o governo e o SUS: Melhor controle epidemiológico, otimização de campanhas de vacinação, redução do desperdício de exames e vacinas.

Modelo de Negócio

O sistema pode ser implementado por meio de um **modelo de negócios híbrido**, abrangendo tanto o setor público quanto o privado:

Setor Público (Parceria com o SUS e Governos Estaduais/Municipais)

Integração com o SUS: Implementação via parcerias público-privadas (PPP), permitindo que o governo utilize o sistema para **centralizar dados de exames e vacinas da população**.

Captação de dados para políticas públicas: O governo pode utilizar as informações para melhor planejar campanhas de vacinação, distribuir recursos de saúde e antecipar surtos epidemiológicos.

Setor Privado (Planos de Saúde, Clínicas e Laboratórios)

Plano de assinatura para clínicas e hospitais: Hospitais e clínicas privadas podem contratar o sistema para integrar seus prontuários eletrônicos, reduzindo custos operacionais e melhorando o atendimento ao paciente.

Monetização via operadoras de planos de saúde: Operadoras podem pagar uma taxa para acessar o sistema e evitar custos com exames desnecessários ou repetidos, otimizando sua gestão de saúde populacional.

Plataforma freemium para pacientes: Os pacientes podem acessar gratuitamente seu histórico básico de exames e vacinas, mas pagar uma assinatura para recursos premium, como lembretes personalizados de vacinação, análises preditivas de saúde e suporte médico via IA.

Estratégia de Adoção e Parcerias

Para garantir a **adoção em larga escala**, a estratégia deve incluir **parcerias estratégicas** com:

Operadoras de Planos de Saúde

Parcerias com grandes operadoras (ex: Unimed, Bradesco Saúde, SulAmérica) para oferecer a plataforma como um **diferencial competitivo** para seus clientes.

Redução de custos com sinistralidade, permitindo que as operadoras financiem o uso da plataforma para seus segurados.

Laboratórios e Clínicas

Integração do sistema com grandes redes laboratoriais (ex: Fleury, Dasa, Sabin) para permitir que exames sejam automaticamente enviados para a plataforma sem necessidade de papel.

Hospitais e clínicas particulares podem adotar a solução como parte de seus serviços, garantindo atendimento mais eficiente.

Cecnologia e Saúde Digital

Parceria com desenvolvedores de **prontuários eletrônicos** (ex: MV, Tasy, TOTVS Saúde) para garantir compatibilidade com sistemas já existentes.

Adoção de Inteligência Artificial (IA) para recomendações de saúde baseadas em dados, auxiliando médicos e pacientes na tomada de decisão.

Governo e Setor Público

Acordos com **Ministério da Saúde e Secretarias Estaduais e Municipais** para integrar o sistema ao SUS.

Criação de um módulo gratuito para postos de saúde e unidades básicas, garantindo que toda a população tenha acesso ao seu histórico de saúde.

Implementação Tecnológica e Diferenciais

API Interoperável

Sistema baseado no **padrão FHIR (Fast Healthcare Interoperability Resources)**, garantindo **interoperabilidade com outros sistemas** de saúde já existentes.

API aberta para que clínicas, hospitais e operadoras possam facilmente integrar seus serviços.

Segurança e Compliance com a LGPD

- Uso de **blockchain para rastreamento e segurança de dados**.
- **Criptografia de ponta a ponta**, garantindo que apenas usuários autorizados tenham acesso às informações.
- **Conformidade total com a LGPD**, garantindo transparência e controle sobre os dados dos pacientes.

Plataforma Multicanal (Mobile e Web)

- Aplicativo para **Android e iOS** que permite aos usuários acessar seus exames, vacinas e receber notificações de lembretes.
- Portal web para médicos, hospitais e operadoras de planos de saúde.

Estratégia de Lançamento e Expansão

Fase 1: Projeto Piloto e Testes

Implementação inicial com hospitais e clínicas parceiras em grandes capitais (São Paulo, Rio de Janeiro, Brasília).

Parcerias com algumas operadoras de saúde para testar o impacto na redução de custos e satisfação dos pacientes.

Fase 2: Expansão Nacional

Lançamento para laboratórios e operadoras de planos de saúde de médio porte.

Acordos com o governo para integração com o SUS.

Fase 3: Expansão Internacional e Novos Serviços

Expansão para países da América Latina que enfrentam desafios semelhantes na gestão de saúde pública.

Introdução de novos serviços, como telemedicina e predições baseadas em IA.

Benefícios para o Mercado e a Sociedade

Para a saúde pública: Melhora na gestão da vacinação, redução de surtos epidemiológicos e otimização dos recursos do SUS.

Para operadoras de planos de saúde: Redução de custos com exames repetidos, maior eficiência na gestão de pacientes e diferenciação no mercado.

Para hospitais e clínicas: Atendimento mais rápido e seguro, além de redução de custos administrativos.

Para os pacientes: Facilidade no acesso a exames e vacinas, mais controle sobre sua saúde e melhor qualidade de vida.

O **Sistema Unificado de Exames e Vacinas** tem um **grande potencial de transformação no setor de saúde brasileiro**. Sua implementação traria mais eficiência, segurança e acessibilidade para milhões de brasileiros, ao mesmo tempo em que abriria **oportunidades de monetização para empresas do setor privado**.

Com **parcerias estratégicas, tecnologia avançada** e um **modelo de negócio sustentável**, essa solução pode revolucionar a **gestão de saúde no Brasil**, trazendo benefícios para toda a população e para o mercado de saúde digital.

5. CICLO DE DESENVOLVIMENTO DA SOLUÇÃO

Por tratar-se de um APP de uso ao público, com muitas variantes como idade, facilidade de usabilidade e adaptações no UX/UI, o projeto precisaria de mais tempo para transitar em estágios Alpha e Beta para adequar as necessidades de grande parte dos usuários que entrariam como “testers” desta nova ferramenta.

Nesta apresentação, passamos pelo processo de entendimento da “dor”, ou seja, dos problemas que foram identificados durante o levantamento da criação da solução.

Todos os prestadores de serviço da área da saúde, criaram mecanismos próprios para o armazenamento e disposição dos resultados de exames, os quais possuem prazo para expirar, e, caso o usuário não se atenha a este detalhe, irá perdê-los. No estudo de caso, demonstrado na apresentação, nota-se que a solução tem como escopo principal a retenção dos exames e de dados vitais dos usuários para que eles possam ter total controle e não se preocupar com expirações de armazenamento.

A ideia visa um sistema integrado com instituições de saúde públicas e privadas para que, seja possível uma consulta rápida ao histórico completo do paciente, fazendo com que não sejam desperdiçados tempo e insumos.

Para melhor ilustrar as ideias postas, seguem link para apreciação em PPT do trabalho.

Para a realização desta etapa, foi feita uma pequena apresentação em Slide no PowerPoint explicando todas as passagens necessárias para a execução do projeto e as ferramentas que serão utilizadas para o projeto. Por restrição de tempo devido ao escopo do projeto e a dinâmica de implementação do TCC, foi feita a estruturação de maneira basilar, para que possa ser melhorada/aperfeiçoada em futuras iterações.

[Link para apresentação do PPT](#)

6. MOCKUP DA PROPOSTA DE SOLUÇÃO

Para a realização do mock-up, foram feitas telas do projeto na plataforma FIGMA. Por restrições do próprio APP, estarei colocando o link para o FIGMA, que irá necessitar de um e-mail para inclusão de acesso ao projeto. Para tanto, peço encarecidamente para o avaliador, ao retornar à avaliação, incluir um e-mail para que possa adicionar ao cadastro de permissões.

A estruturação vista as melhores práticas de UX e UI aprendidas durante o período do curso. Por tratar-se de uma solução que atende a todas as faixas etárias, foram necessários ajustes e posicionamento para um fácil entendimento e compreensão de pessoas idosas, que necessitam de botões dispostos de maneira clara, sucinta e com colorações mais fortes, focando assim a atenção em que realmente importa.

Por limitações da solução adotada, e conforme mencionando acima, o FIGMA não permite a exportação em JPEG ou PDF para uma melhor apresentação deste tema, desta maneira, faz-se necessário o compartilhamento do link para apreciação da solução.

Cabe ressaltar que o modelo conta com uma simulação de funcionalidade do aplicativo, para que se tenha um entendimento melhor da maneira como foi idealizado.

[Link para o Mock-Up no Figma](#)

7. ARQUITETURA DE SOFTWARE

Para a arquitetura de software, foram estudadas as melhores opções, visando uma fácil implementação e, sobretudo, uma fácil manutenção da aplicação.

Neste setor, farei a descrição detalhada e individualizada de cada solução para uma construção sólida da aplicação em conformidade com as melhores práticas do mercado atual.

Utilização de C# e MySQL para o Desenvolvimento do Sistema Unificado de Saúde

Para a implementação de um **Sistema Unificado de Exames e Vacinas**, a escolha da **linguagem de programação C#** em conjunto com o **banco de dados MySQL** oferece robustez, escalabilidade e segurança na manipulação dos dados de saúde. A seguir, detalhamos a estrutura do sistema, destacando a arquitetura, os principais módulos e como essas tecnologias são utilizadas.

Arquitetura do Sistema

O sistema é baseado em uma **arquitetura em camadas** para garantir modularidade e facilidade de manutenção:

Camada de Apresentação (Frontend)

- Aplicação Web desenvolvida em **ASP.NET Core MVC** ou **Blazor** para médicos, clínicas e operadoras de saúde. Facilita a manutenção do código através da separação clara das responsabilidades além de prevenção automática de ataques como **XSS (Cross-Site Scripting)** e **CSRF (Cross-Site Request Forgery)** através de recursos embutidos no framework.

Aplicativo **Mobile Xamarin** ou **.NET MAUI** para pacientes acessarem seu histórico médico sendo uma evolução direta do Xamarin.Forms, projetado para simplificar o desenvolvimento e garantir alta performance

em diferentes plataformas: **Android, iOS / MacOS, Windows e Linux** (via Community support).

- API Rest para integração com outros sistemas de saúde.

Camada de Aplicação (Backend)

- API desenvolvida em **ASP.NET Core Web API** para processar requisições e integrar com o banco de dados.
- A API recebe requisições via HTTP (GET, POST, PUT, DELETE). Cada requisição aciona um Controller específico.
- Implementação de segurança baseada em **JWT (JSON Web Token)** para controle de autenticação.
- Dentro dos **Controllers**, são executadas validações, regras de negócio e segurança.
- Após validação, o Controller aciona uma camada intermediária (Services ou camada de Aplicação).
- Para comunicação direta com o banco de dados, o ASP.NET Core geralmente utiliza um ORM (Object-Relational Mapping), como o **Entity Framework Core**, ou bibliotecas como o **Dapper**.
- O Entity Framework Core facilita a manipulação dos dados, simplifica consultas e gerência automaticamente migrações e alterações de esquema do banco.

Camada de Banco de Dados

- **MySQL** é utilizado como banco relacional para armazenar exames, vacinas, registros médicos e usuários.
- Implementação de **Stored Procedures e Triggers** para aumentar a performance e a integridade dos dados.
- O método em MySQL suporta consultas rápidas, índices eficientes e possui opções robustas de particionamento e replicação, permitindo que o sistema cresça facilmente conforme a demanda aumenta.
- Oferece mecanismos como transações ACID, chaves estrangeiras e restrições que garantem a integridade e a consistência dos dados relacionados aos arquivos

- É uma solução open-source madura, estável, com grande comunidade e suporte amplamente disponível, o que diminui custos operacionais e facilita a manutenção.

Camada de Segurança

- Utilização de **Identity Server** para autenticação e autorização.
- **Criptografia AES** para proteção de dados sensíveis.
- **Backup automático e replicação de dados** para evitar perdas.

Estrutura do Banco de Dados MySQL

- O banco é projetado com tabelas normalizadas para garantir eficiência.
- Essa estrutura básica serve como ponto de partida e pode ser adaptada conforme as necessidades específicas do seu sistema de arquivos na nuvem

Implementação do Backend com C# e ASP.NET Core Web API

- A API é responsável por gerenciar os registros de pacientes, exames e vacinas.
- A implementação do backend com **C# e ASP.NET Core Web API** oferece diversas vantagens significativas, especialmente para o desenvolvimento de um sistema robusto de acesso a arquivos na nuvem. Abaixo seguem os principais benefícios dessa escolha tecnológica.
- **ASP.NET Core** é extremamente rápido e otimizado, permitindo alta performance com baixo uso de memória, ideal para aplicações na nuvem que precisam escalar rapidamente.
- Suporte embutido para requisições assíncronas (async/await), resultando em respostas mais rápidas e eficientes.
- Suporte completo à arquitetura RESTful, facilitando o desenvolvimento de APIs claras e organizadas.
- Fácil implementação de padrões modernos como:
- Clean Architecture, Domain-Driven Design (DDD) e Microserviços.

A escolha de tecnologias como C# com ASP.NET Core Web API para o backend, MySQL como banco de dados, e Identity para segurança, além de proteção contra ataques XSS e CSRF, junto com Xamarin para o desenvolvimento mobile, forma uma estratégia tecnológica robusta para criar um software seguro, escalável e que atenda a múltiplos usuários. Usar C# com ASP.NET Core traz um desempenho elevado, escalabilidade simplificada e uma arquitetura modular, o que facilita a construção de APIs eficientes e bem estruturadas.

A versatilidade do ASP.NET Core permite que o sistema seja implementado de maneira flexível em diferentes ambientes operacionais, o que é essencial para uma abordagem cloud-native, garantindo que o sistema possa ser acessado por vários usuários ao mesmo tempo. Optamos pelo MySQL porque ele oferece consistência, integridade e um desempenho excelente na gestão de grandes volumes de dados e metadados, assegurando escalabilidade tanto vertical quanto horizontal à medida que o sistema cresce. Além disso, usar esse banco de dados de código aberto é uma solução econômica, com amplo suporte técnico e documentação de qualidade. Na camada de segurança, o Identity proporciona um sistema forte de autenticação e autorização, permitindo um controle detalhado e seguro das credenciais e permissões dos usuários.

Também implementamos medidas para proteger contra-ataques como Cross-Site Scripting (XSS) e Cross-Site Request Forgery (CSRF), que são cruciais para manter a integridade e a confidencialidade dos dados, além de prevenir vulnerabilidades comuns em aplicações web modernas.

Por último, a escolha do Xamarin para o desenvolvimento mobile permite criar aplicativos nativos que funcionam em várias plataformas, garantindo um desempenho consistente e uma experiência de usuário unificada em dispositivos Android e iOS. Com uma única base de código em C#, o Xamarin torna o desenvolvimento e a manutenção de aplicativos móveis mais simples e econômicos.

8. VALIDAÇÃO DA SOLUÇÃO

Nesta seção, busquei demonstrar que a aplicação está apta e totalmente funcional para o recebimento de dados.

Os prints de preparação e validação dos ambientes e da inserção e retirada dos dados servem para o propósito do entendimento e compreensão das estruturas de validação lecionadas durante o período da pós-graduação, demonstrando que a solução foi implementada com os conhecimentos adquiridos durante o curso e a ampliação do conhecimento em elevar o aprendizado para além do que foi ministrado durante as aulas.

Preparação do ambiente:

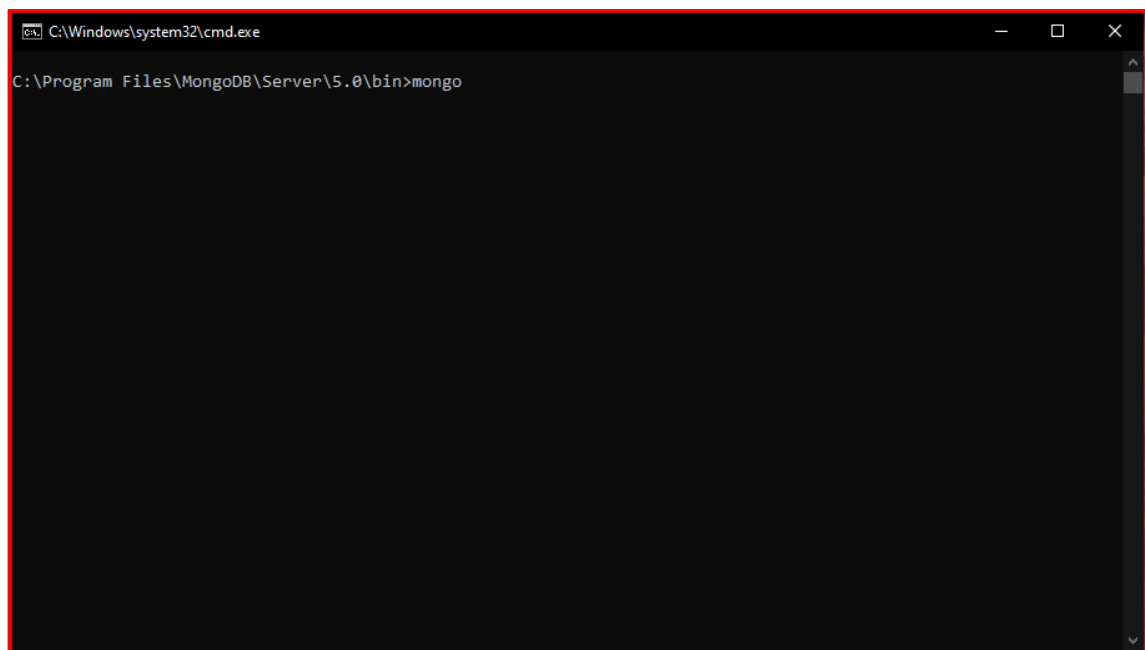


Figura 1 - Criação em MongoDB do ambiente (Arquivo pessoal)

```

C:\Windows\system32\cmd.exe - mongo
C:\Program Files\MongoDB\Server\5.0\bin>mongo
MongoDB shell version v5.0.3
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("31ff53bd-6b9d-4e74-b83d-dcb3d8a6d720") }
MongoDB server version: 5.0.3
=====
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility. The "mongo" shell has been deprecated and will be removed in
an upcoming release.
We recommend you begin using "mongosh".
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
=====
---
The server generated these startup warnings when booting:
  2021-10-05T09:36:24.240-03:00: Access control is not enabled for the database. Read and write access to data and
configuration is unrestricted
---
---
  Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

  The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

  To enable free monitoring, run the following command: db.enableFreeMonitoring()
  To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
>

```

Figura 2 - Validação da criação do ambiente em MongoDB (Arquivo pessoal)

Continuação da preparação do ambiente:

```

C:\Windows\system32\cmd.exe - mongo
> use healthpass
switched to db healthpass
>

```

Figura 3 - Switch de pastas para o projeto mantendo padronização de execução (Arquivo pessoal)

Criação das Collection e Justificativas

Nome da collection: usuarios

Justificativa: Nosso sistema consiste em uma quantidade expressiva de usuários de diversas cartilhas de clientes em diversos seguimentos distintos. Para tanto, construímos nossa base em MongoDB e com auxílio da nuvem, não teremos problemas para implantar uma grande quantidade de usuários sem se preocupar em atingir o limite no quesito de capacidade de armazenamento.

A) Criação do Database “healthpass” e da sua primeira Collection “usuarios”

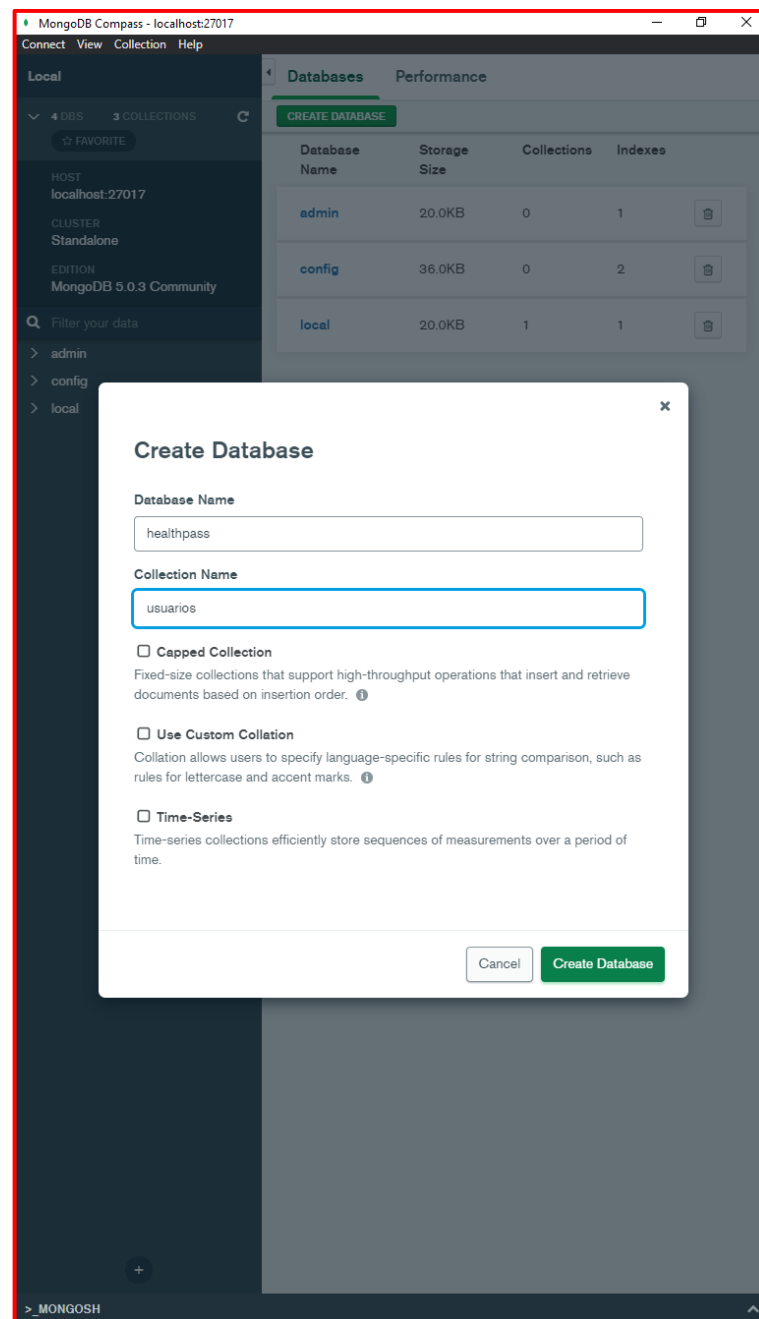
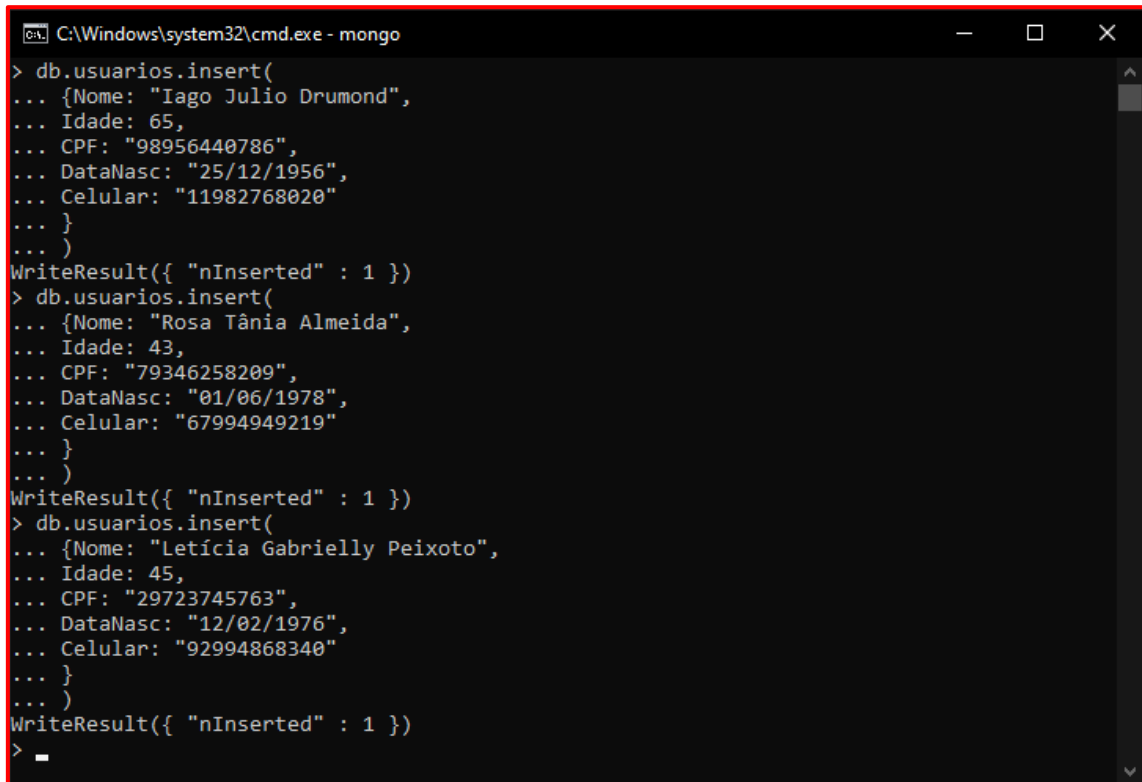


Figura 4 - Criação de Collections de usuários (Arquivo pessoal)

B) Inserção dos dados da Collection “usuario”



```

C:\Windows\system32\cmd.exe - mongo
> db.usuarios.insert(
... {Nome: "Iago Julio Drumond",
... Idade: 65,
... CPF: "98956440786",
... DataNasc: "25/12/1956",
... Celular: "11982768020"
... }
... )
WriteResult({ "nInserted" : 1 })
> db.usuarios.insert(
... {Nome: "Rosa Tânia Almeida",
... Idade: 43,
... CPF: "79346258209",
... DataNasc: "01/06/1978",
... Celular: "67994949219"
... }
... )
WriteResult({ "nInserted" : 1 })
> db.usuarios.insert(
... {Nome: "Letícia Gabrielly Peixoto",
... Idade: 45,
... CPF: "29723745763",
... DataNasc: "12/02/1976",
... Celular: "92994868340"
... }
... )
WriteResult({ "nInserted" : 1 })
>

```

Figura 5 - Exemplo de inserção de usuário (Arquivo pessoal)

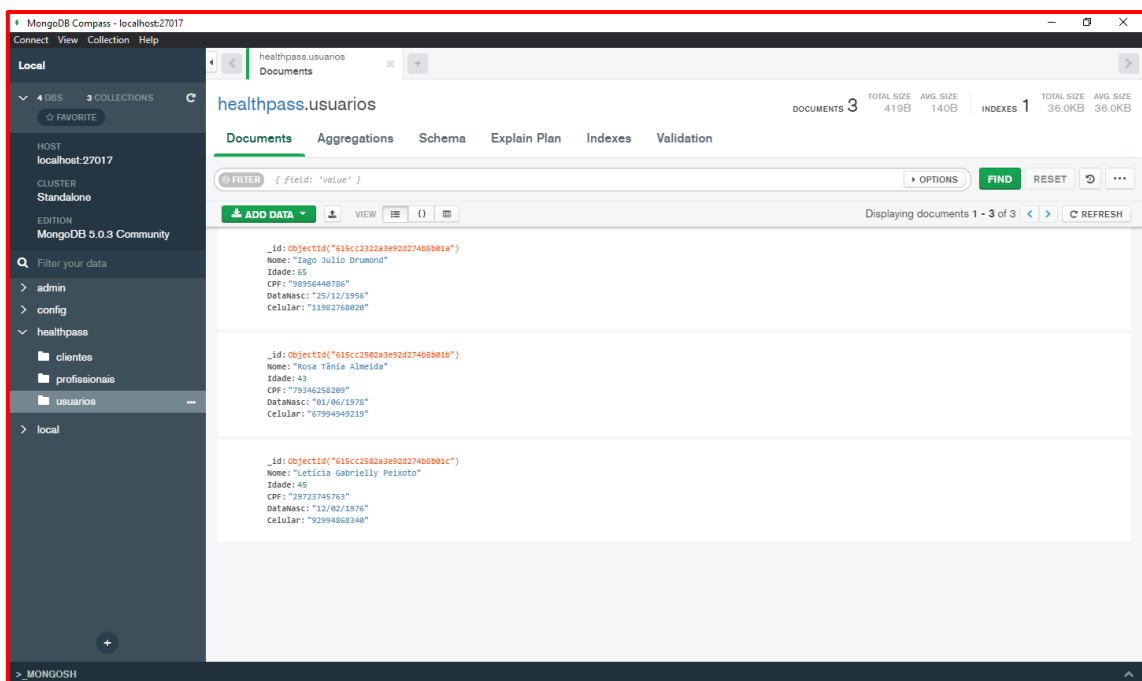
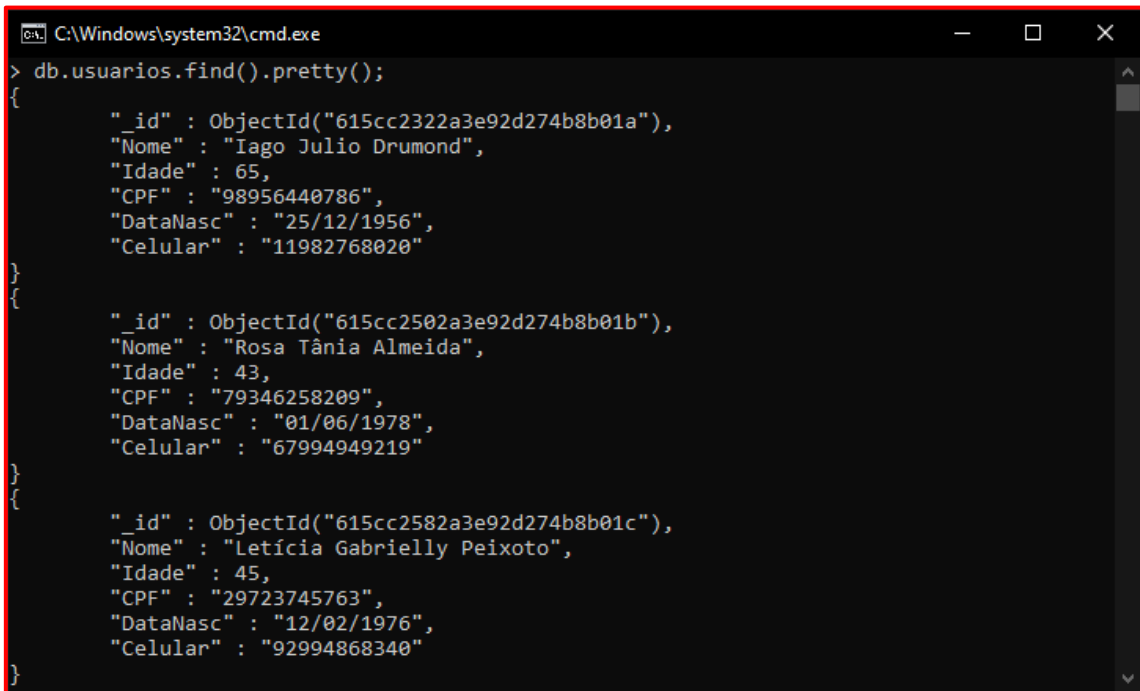


Figura 6 - "Proof" da criação dos usuários e seus _id's correspondentes (Arquivo pessoal)

C) Visualizar os documentos da Collection “usuarios” de forma mais bem estruturada



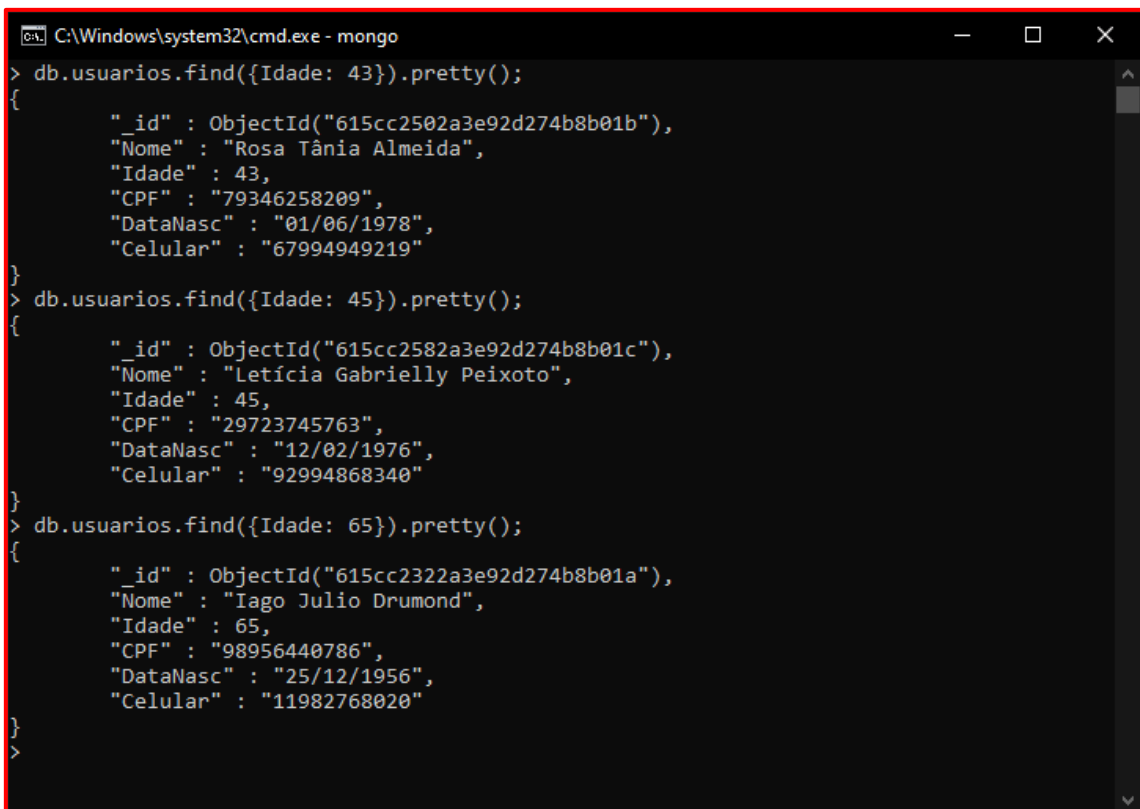
```

C:\Windows\system32\cmd.exe
> db.usuarios.find().pretty();
{
  "_id" : ObjectId("615cc2322a3e92d274b8b01a"),
  "Nome" : "Iago Julio Drumond",
  "Idade" : 65,
  "CPF" : "98956440786",
  "DataNasc" : "25/12/1956",
  "Celular" : "11982768020"
}
{
  "_id" : ObjectId("615cc2502a3e92d274b8b01b"),
  "Nome" : "Rosa Tânia Almeida",
  "Idade" : 43,
  "CPF" : "79346258209",
  "DataNasc" : "01/06/1978",
  "Celular" : "67994949219"
}
{
  "_id" : ObjectId("615cc2582a3e92d274b8b01c"),
  "Nome" : "Letícia Gabrielly Peixoto",
  "Idade" : 45,
  "CPF" : "29723745763",
  "DataNasc" : "12/02/1976",
  "Celular" : "92994868340"
}

```

Figura 7 - Estruturação backend dos usuários (Arquivo pessoal)

D) Consultas por igualdade utilizando os métodos find() e pretty()



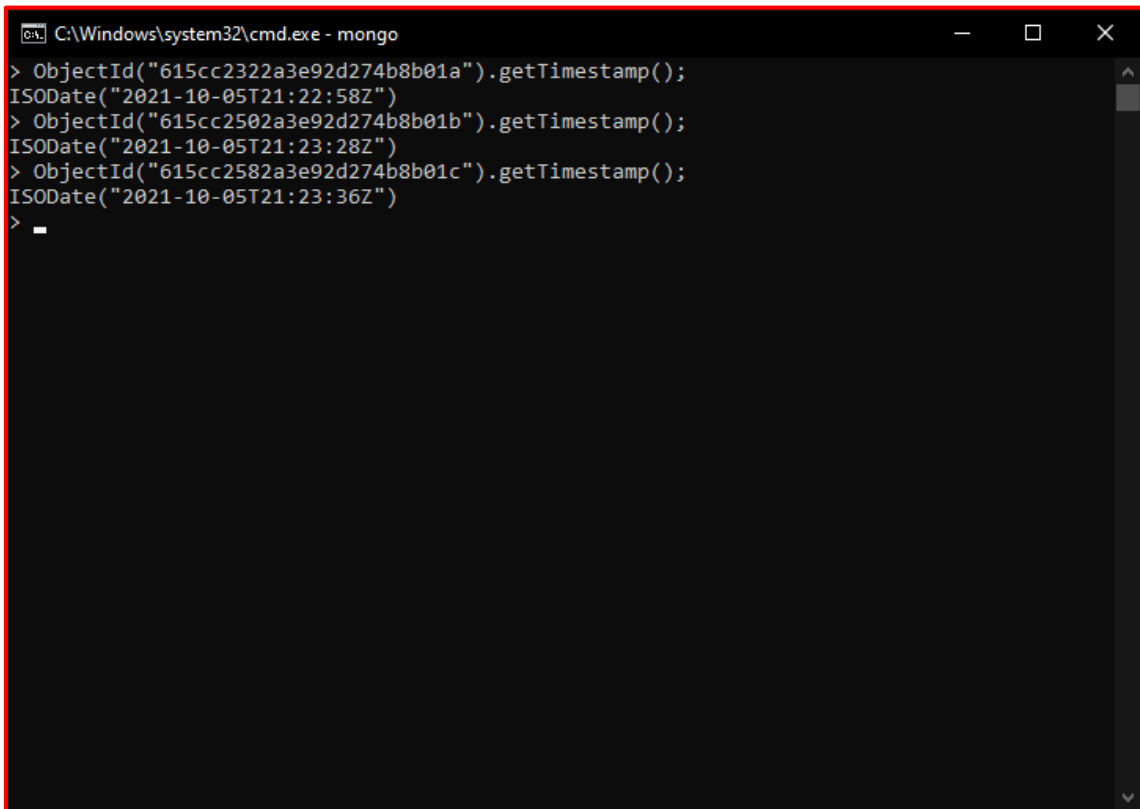
```

C:\Windows\system32\cmd.exe - mongo
> db.usuarios.find({Idade: 43}).pretty();
{
  "_id" : ObjectId("615cc2502a3e92d274b8b01b"),
  "Nome" : "Rosa Tânia Almeida",
  "Idade" : 43,
  "CPF" : "79346258209",
  "DataNasc" : "01/06/1978",
  "Celular" : "67994949219"
}
> db.usuarios.find({Idade: 45}).pretty();
{
  "_id" : ObjectId("615cc2582a3e92d274b8b01c"),
  "Nome" : "Letícia Gabrielly Peixoto",
  "Idade" : 45,
  "CPF" : "29723745763",
  "DataNasc" : "12/02/1976",
  "Celular" : "92994868340"
}
> db.usuarios.find({Idade: 65}).pretty();
{
  "_id" : ObjectId("615cc2322a3e92d274b8b01a"),
  "Nome" : "Iago Julio Drumond",
  "Idade" : 65,
  "CPF" : "98956440786",
  "DataNasc" : "25/12/1956",
  "Celular" : "11982768020"
}
>

```

Figura 8 - Consulta por igualdade dos usuários métodos find() e pretty () - (Arquivo pessoal)

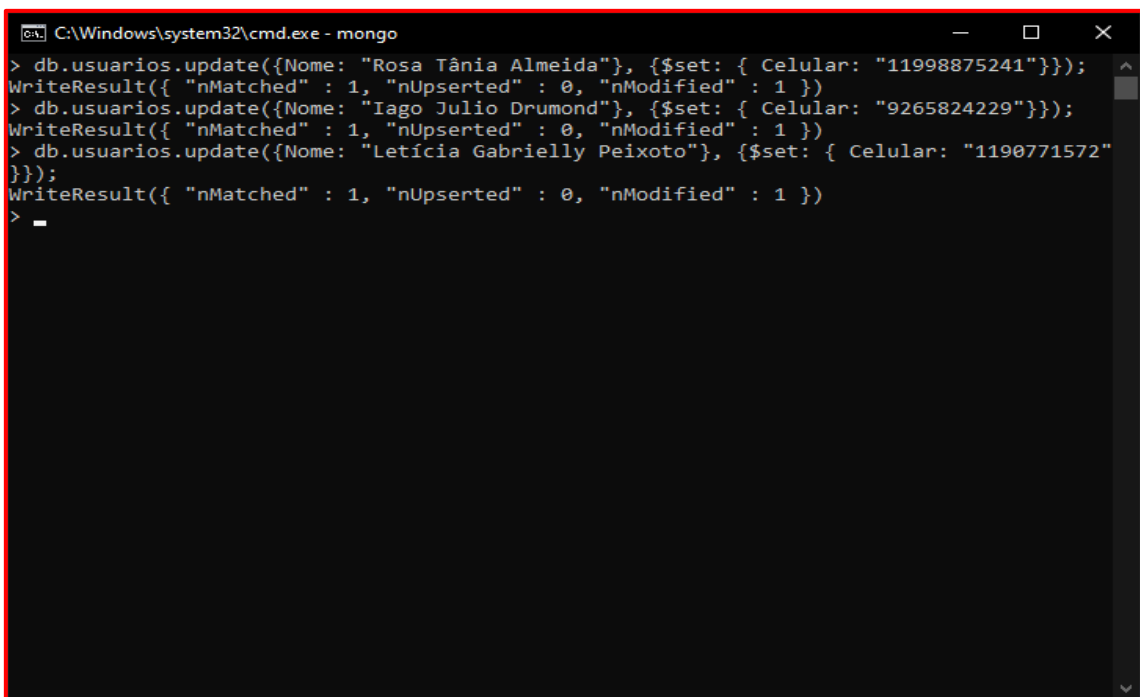
E) Comando para consulta de quando o registro foi criado utilizando o ObjectId referente a cada um dos Documents



```
C:\Windows\system32\cmd.exe - mongo
> ObjectId("615cc2322a3e92d274b8b01a").getTimestamp();
ISODate("2021-10-05T21:22:58Z")
> ObjectId("615cc2502a3e92d274b8b01b").getTimestamp();
ISODate("2021-10-05T21:23:28Z")
> ObjectId("615cc2582a3e92d274b8b01c").getTimestamp();
ISODate("2021-10-05T21:23:36Z")
> _
```

Figura 9 – Consulta de criação de registro com os comandos ObjectId - (Arquivo pessoal)

F) Alterando os registros da collection “usuarios”



```
C:\Windows\system32\cmd.exe - mongo
> db.usuarios.update({Nome: "Rosa Tânia Almeida"}, {$set: { Celular: "11998875241"}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.usuarios.update({Nome: "Iago Julio Drumond"}, {$set: { Celular: "9265824229"}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.usuarios.update({Nome: "Letícia Gabrielly Peixoto"}, {$set: { Celular: "1190771572"
}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> _
```

Figura 10 – Demonstração de alteração dos registros dos usuários - (Arquivo pessoal)

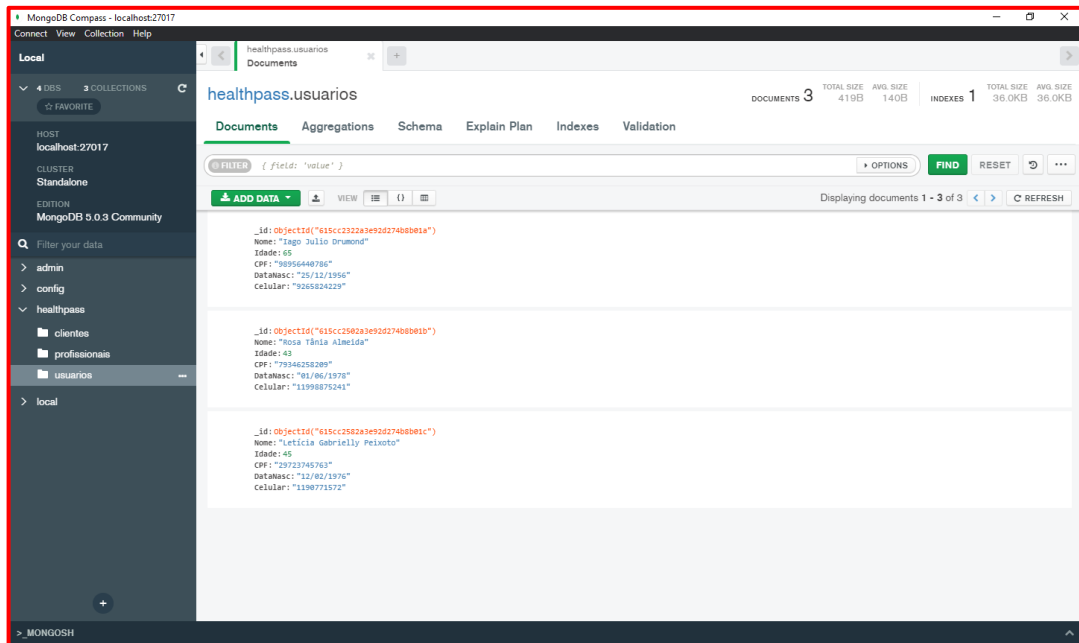


Figura 11 - "Proof" da substituição dos dados alterados - (Arquivo pessoal)

G) Removendo as Documents da Collection "usuarios"

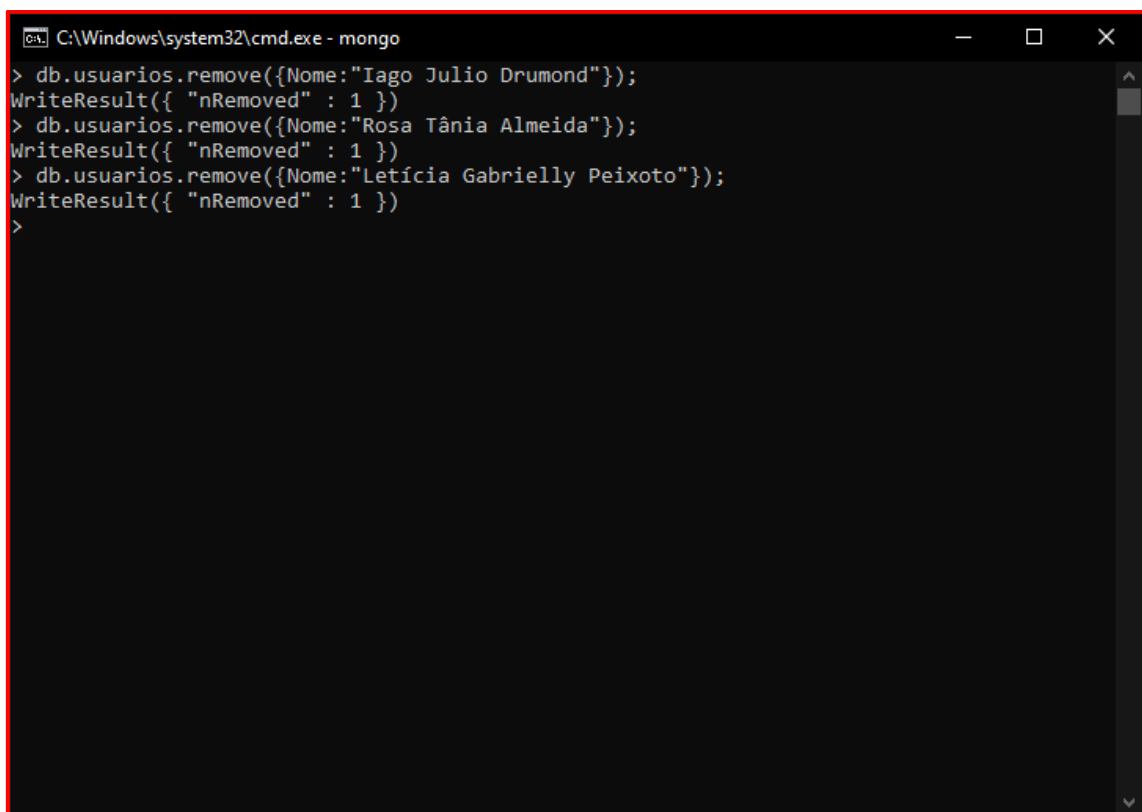


Figura 12 - Exemplo de remoção dos documents do usuário - (Arquivo pessoal)

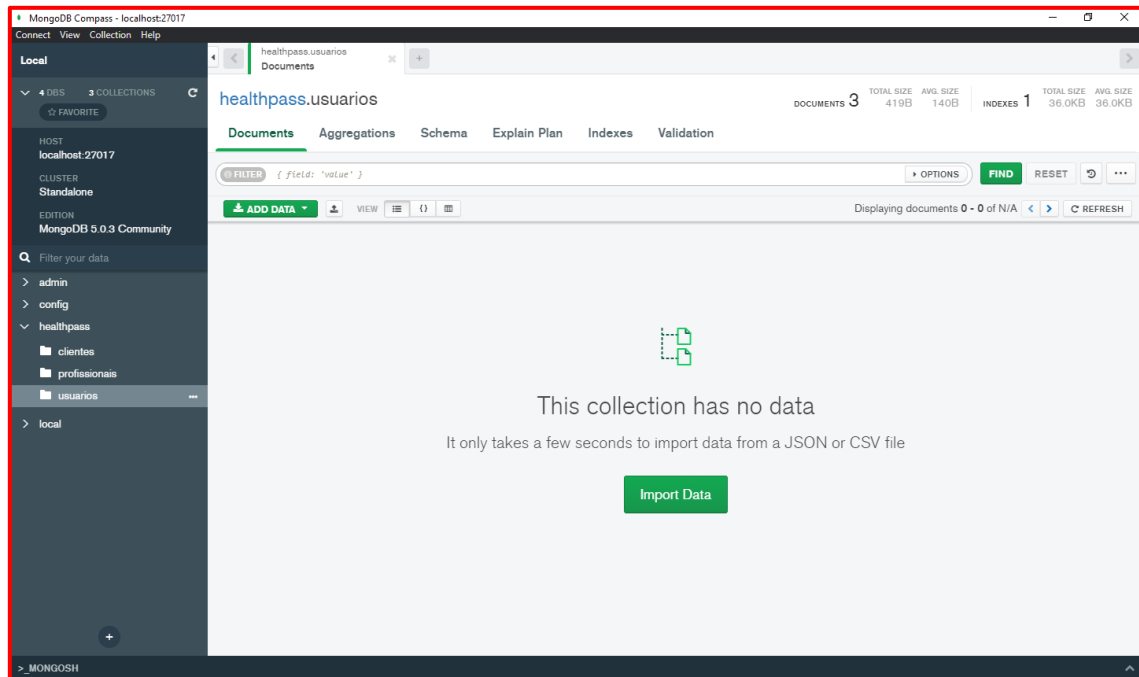
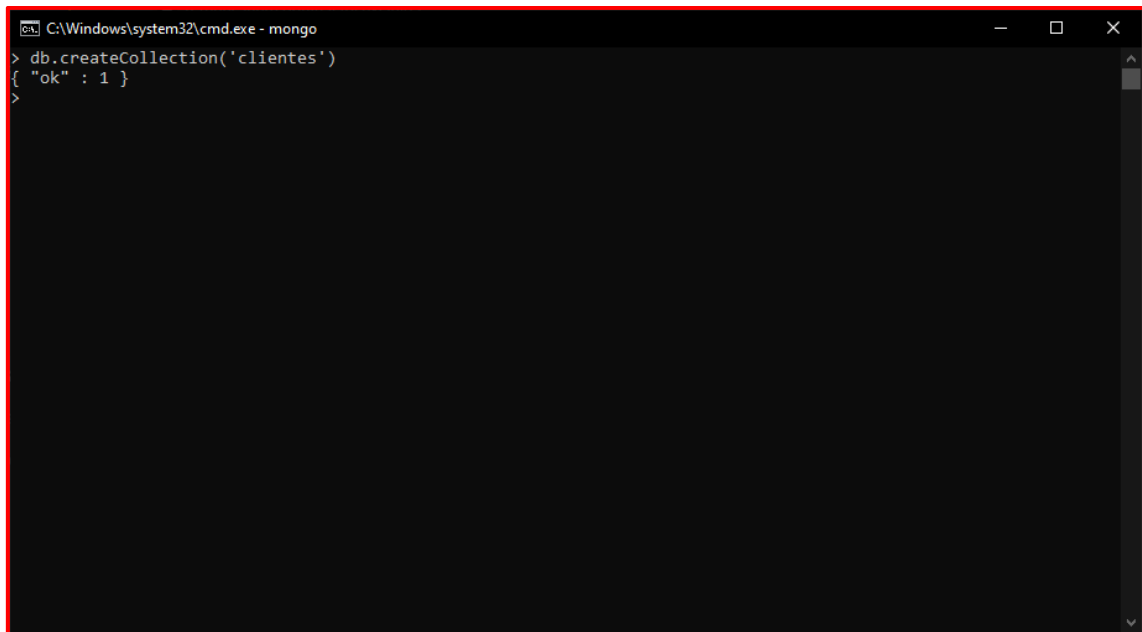


Figura 13 - "Proof" da remoção dos dados dos usuários - (Arquivo pessoal)

Nome da collection: Cliente

Justificativa: Por se tratar de um elevado fluxo de dados das operadoras de saúde de inúmeros segmentos distintos tais como: hospitais, exames, laboratórios etc., houve a necessidade em armazenar uma grande quantidade de dados que necessitam constantemente em serem atualizados e/ou modificados. A praticidade do MongoDB atende a todos os requisitos para que possa fornecer um sistema íntegro e estável.

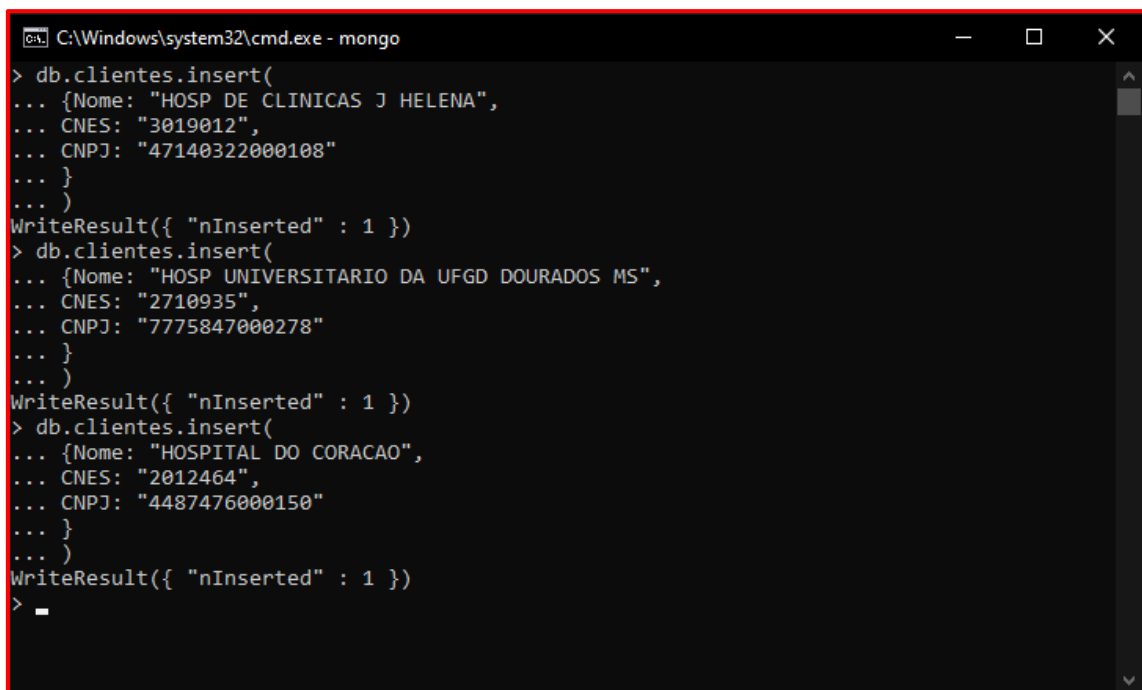
A) Criação da Collection



```
C:\Windows\system32\cmd.exe - mongo
> db.createCollection('clientes')
{ "ok" : 1 }
>
```

Figura 14 - Criação da collection “clientes” - (Arquivo pessoal)

B) Inserção dos dados da Collection “clientes”



```
C:\Windows\system32\cmd.exe - mongo
> db.clientes.insert(
... {Nome: "HOSP DE CLINICAS J HELENA",
... CNES: "3019012",
... CNPJ: "47140322000108"
... }
... )
WriteResult({ "nInserted" : 1 })
> db.clientes.insert(
... {Nome: "HOSP UNIVERSITARIO DA UFGD DOURADOS MS",
... CNES: "2710935",
... CNPJ: "7775847000278"
... }
... )
WriteResult({ "nInserted" : 1 })
> db.clientes.insert(
... {Nome: "HOSPITAL DO CORACAO",
... CNES: "2012464",
... CNPJ: "4487476000150"
... }
... )
WriteResult({ "nInserted" : 1 })
>
```

Figura 15 - Inserção dos dados do cliente dentro da collection - (Arquivo pessoal)

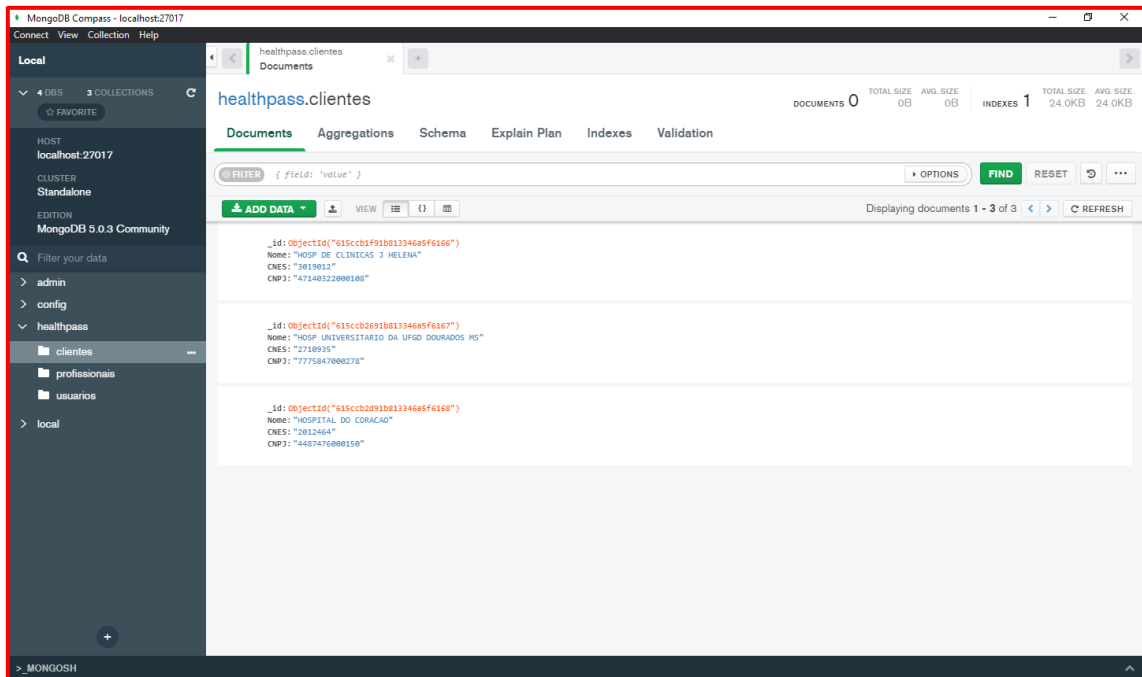


Figura 16 - "Proof" da inserção dos dados da collection clientes - (Arquivo pessoal)

C) Visualizar os documentos da Collection “clientes” de forma mais bem estruturada

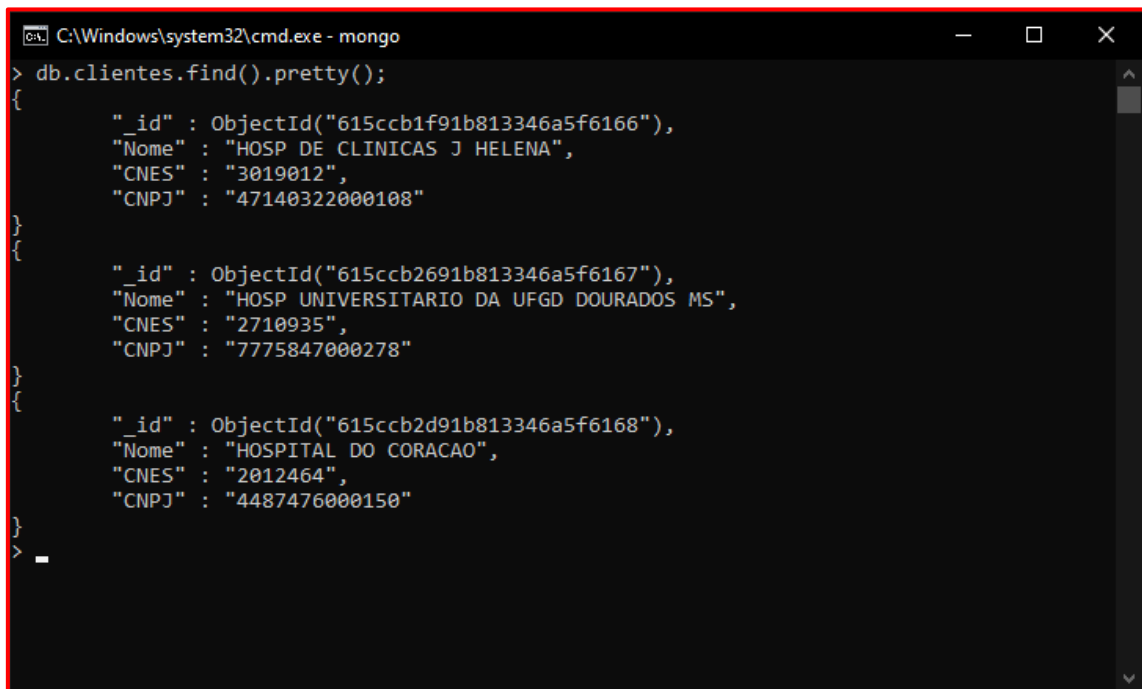



Figura 17 - Estruturação da collection clientes de maneira mais estruturada - (Arquivo pessoal)

D) Consultas por igualdade utilizando os métodos find() e pretty() conforme o CNES de cada cliente

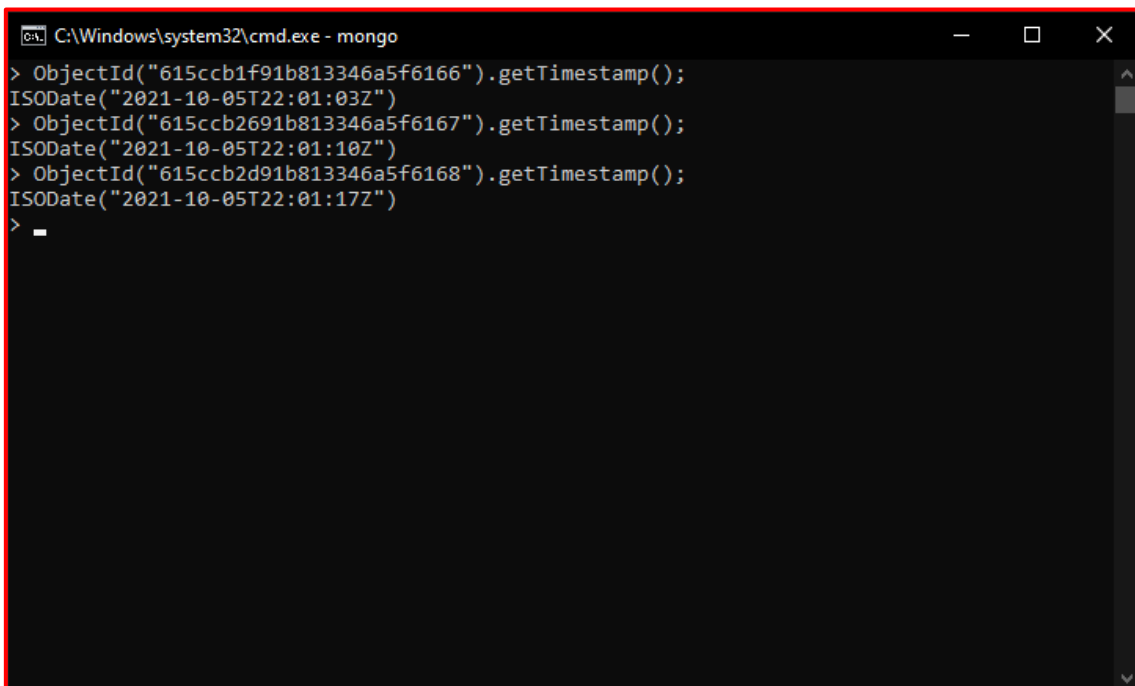


```

C:\Windows\system32\cmd.exe - mongo
> db.clientes.find({CNES: "3019012"}).pretty();
{
  "_id" : ObjectId("615ccb1f91b813346a5f6166"),
  "Nome" : "HOSP DE CLINICAS J HELENA",
  "CNES" : "3019012",
  "CNPJ" : "47140322000108"
}
> db.clientes.find({CNES: "2710935"}).pretty();
{
  "_id" : ObjectId("615ccb2691b813346a5f6167"),
  "Nome" : "HOSP UNIVERSITARIO DA UFGD DOURADOS MS",
  "CNES" : "2710935",
  "CNPJ" : "7775847000278"
}
> db.clientes.find({CNES: "2012464"}).pretty();
{
  "_id" : ObjectId("615ccb2d91b813346a5f6168"),
  "Nome" : "HOSPITAL DO CORACAO",
  "CNES" : "2012464",
  "CNPJ" : "4487476000150"
}
>
  
```

Figura 18 - Consultas por igualdade utilizando os métodos find() e pretty() por CNES - (Arquivo pessoal)

E) Comando para consulta de quando o registro foi criado utilizando o ObjectId referente a cada um dos Documents

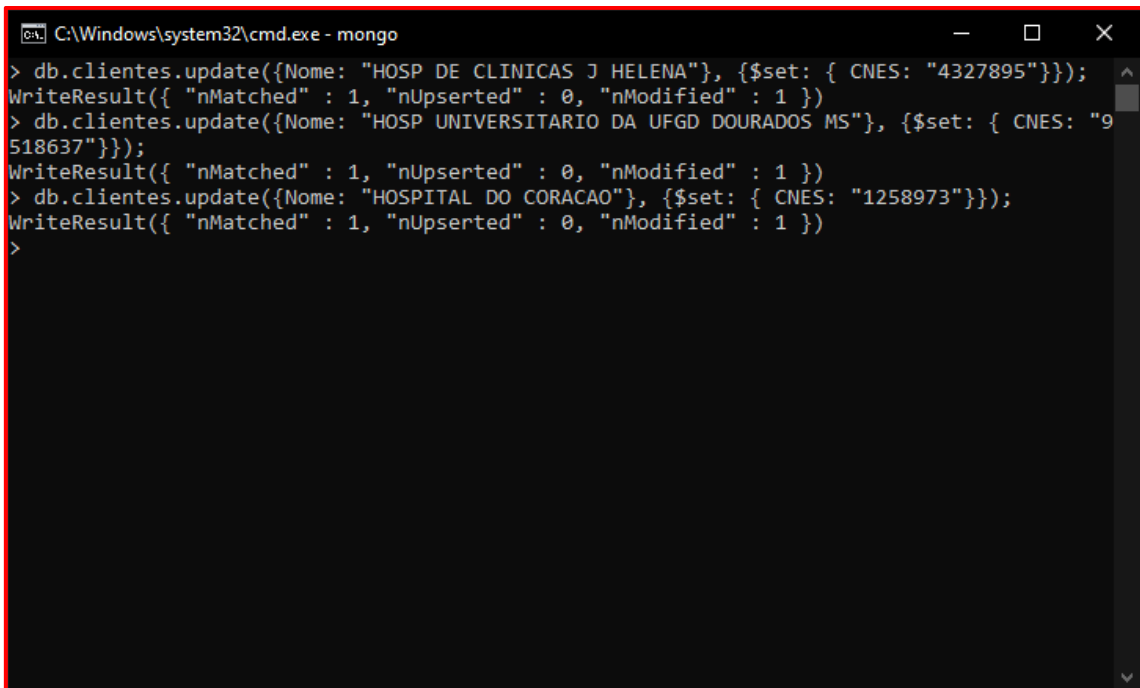


```

C:\Windows\system32\cmd.exe - mongo
> ObjectId("615ccb1f91b813346a5f6166").getTimestamp();
ISODate("2021-10-05T22:01:03Z")
> ObjectId("615ccb2691b813346a5f6167").getTimestamp();
ISODate("2021-10-05T22:01:10Z")
> ObjectId("615ccb2d91b813346a5f6168").getTimestamp();
ISODate("2021-10-05T22:01:17Z")
>
  
```

Figura 19 - Consulta por ObjectId - (Arquivo pessoal)

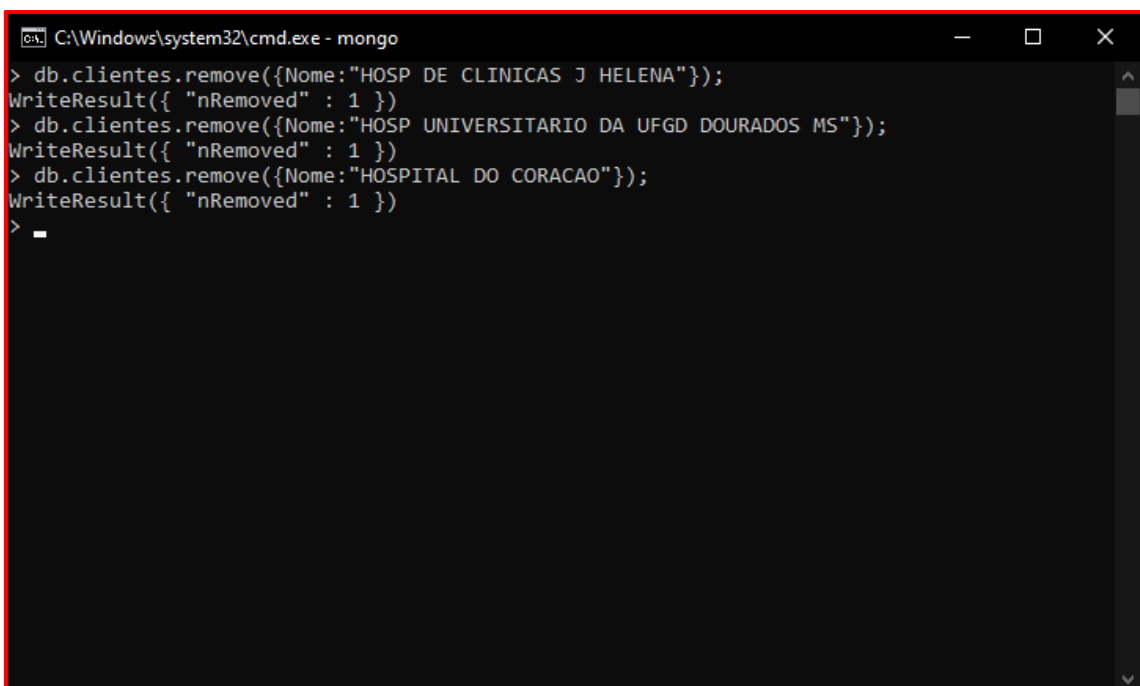
F) Alterando os registros da collection “clientes”



```
C:\Windows\system32\cmd.exe - mongo
> db.clientes.update({Nome: "HOSP DE CLINICAS J HELENA"}, {$set: { CNES: "4327895"}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.clientes.update({Nome: "HOSP UNIVERSITARIO DA UFGD DOURADOS MS"}, {$set: { CNES: "9518637"}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.clientes.update({Nome: "HOSPITAL DO CORACAO"}, {$set: { CNES: "1258973"}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
>
```

Figura 20 - Alteração de dados dentro de clientes - (Arquivo pessoal)

G) Removendo as Documents da Collection “clientes”



```
C:\Windows\system32\cmd.exe - mongo
> db.clientes.remove({Nome:"HOSP DE CLINICAS J HELENA"});
WriteResult({ "nRemoved" : 1 })
> db.clientes.remove({Nome:"HOSP UNIVERSITARIO DA UFGD DOURADOS MS"});
WriteResult({ "nRemoved" : 1 })
> db.clientes.remove({Nome:"HOSPITAL DO CORACAO"});
WriteResult({ "nRemoved" : 1 })
>
```

Figura 21 - Remoção de dados dos clientes - (Arquivo pessoal)

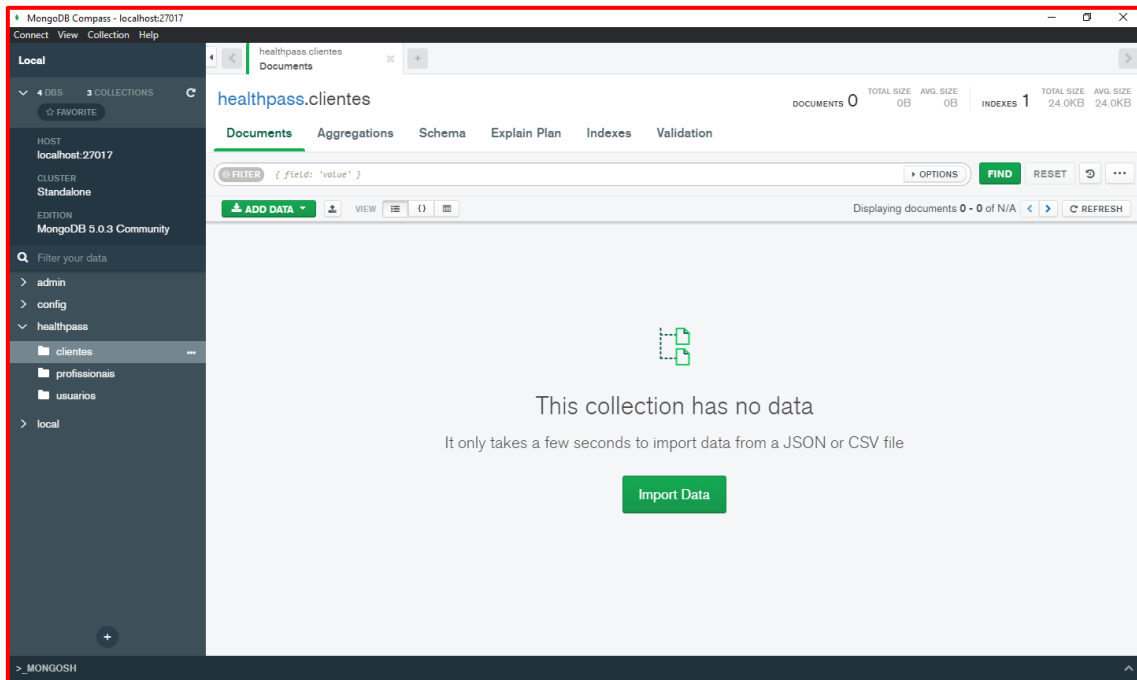


Figura 22 - "Proof" da remoção dos dados dos clientes - (Arquivo pessoal)

Nome da collection: Profissionais

Justificativa: Da mesma maneira que utilizamos para a tabela **Clientes e Usuários**, a collection **profissionais** utiliza-se dos mesmos preceitos. Os dados serão atualizados e/ou inseridos pelos "Clientes" que irão manter o status do profissional atualizado.

A Database dos profissionais é importantíssima, pois irá demonstrar o histórico de consultas, procedimentos, tempo de atendimento e satisfação geral.

A) Criação da Collection

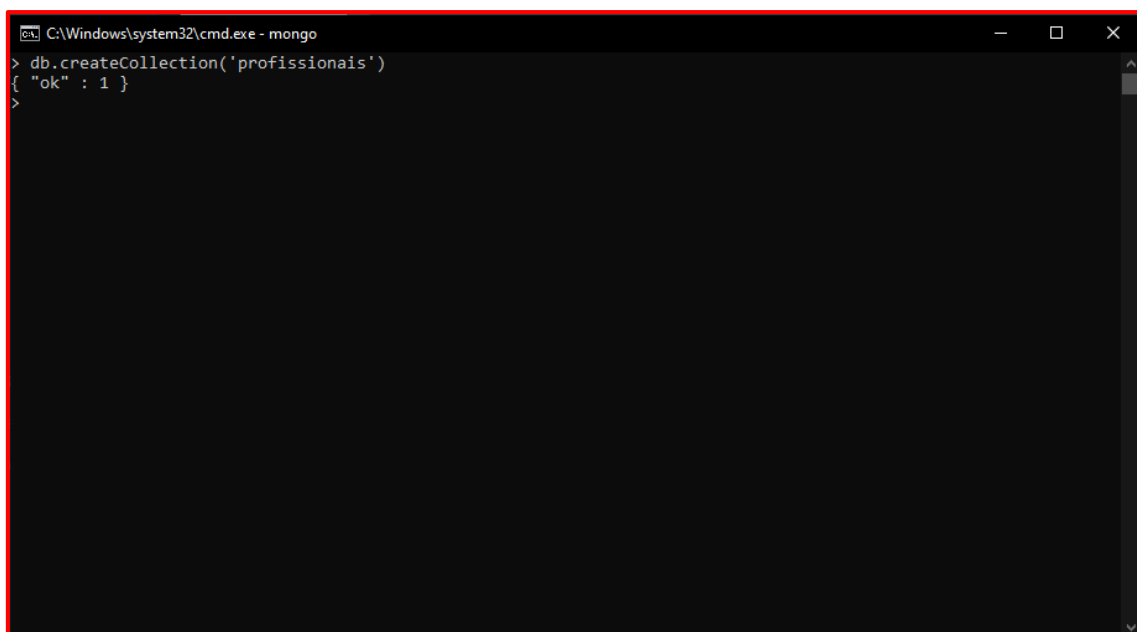
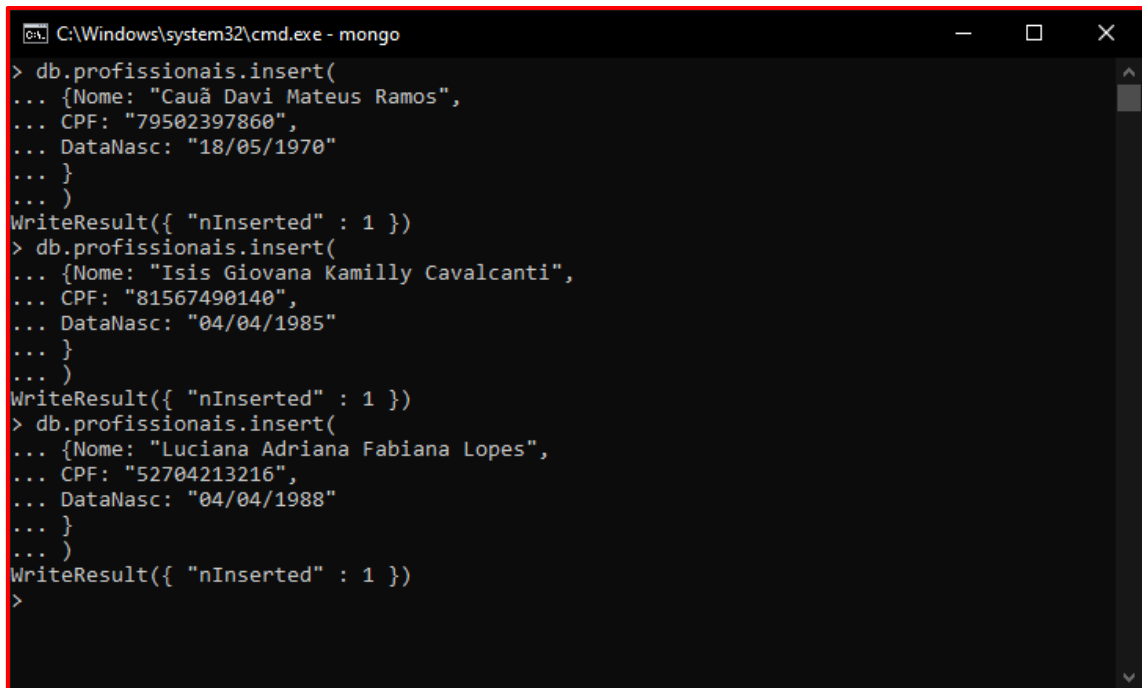


Figura 23 - Criação da collection profissionais - (Arquivo pessoal)

B) Inserção dos dados da Collection “profissionais”



```

C:\Windows\system32\cmd.exe - mongo
> db.profissionais.insert(
... {Nome: "Cauã Davi Mateus Ramos",
... CPF: "79502397860",
... DataNasc: "18/05/1970"
... }
... )
WriteResult({ "nInserted" : 1 })
> db.profissionais.insert(
... {Nome: "Isis Giovana Kamilly Cavalcanti",
... CPF: "81567490140",
... DataNasc: "04/04/1985"
... }
... )
WriteResult({ "nInserted" : 1 })
> db.profissionais.insert(
... {Nome: "Luciana Adriana Fabiana Lopes",
... CPF: "52704213216",
... DataNasc: "04/04/1988"
... }
... )
WriteResult({ "nInserted" : 1 })
>

```

Figura 24 - Inserção dos dados dos profissionais - (Arquivo pessoal)

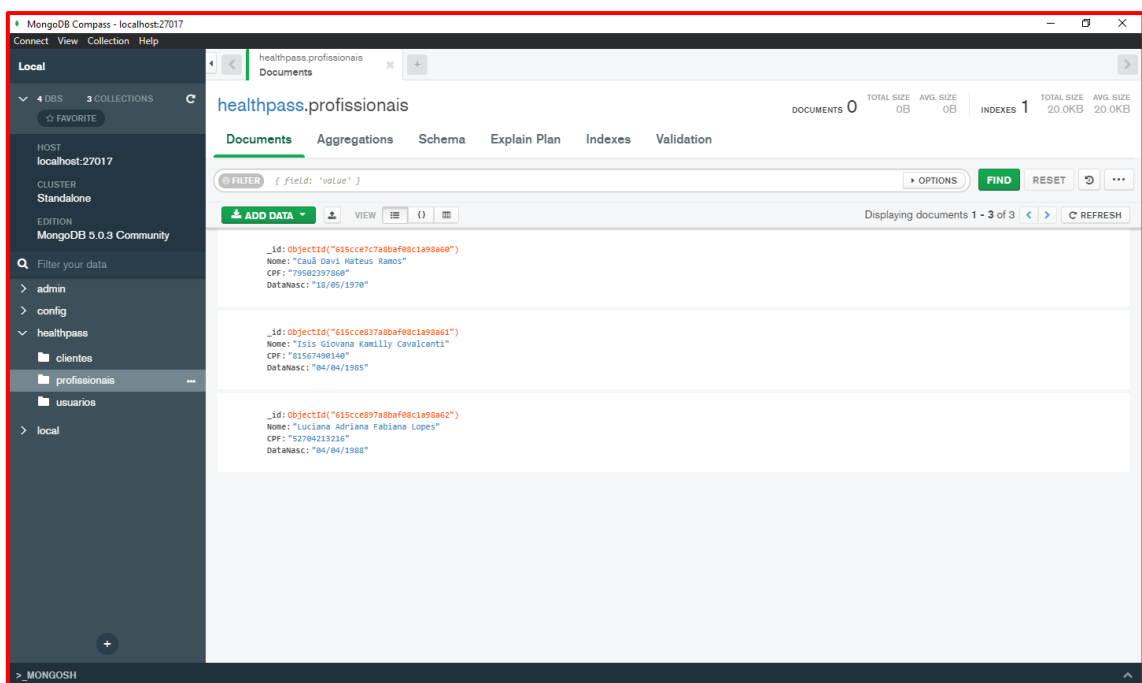


Figura 25 - "Proof" da inserção dos dados dos profissionais na collection - (Arquivo pessoal)

C) Visualizar os documentos da Collection “profissionais” de forma mais bem estruturada

```

C:\Windows\system32\cmd.exe - mongo
> db.profissionais.find().pretty();
{
  "_id" : ObjectId("615cce7c7a8baf08c1a98a60"),
  "Nome" : "Cauã Davi Mateus Ramos",
  "CPF" : "79502397860",
  "DataNasc" : "18/05/1970"
}
{
  "_id" : ObjectId("615cce837a8baf08c1a98a61"),
  "Nome" : "Isis Giovana Kamilly Cavalcanti",
  "CPF" : "81567490140",
  "DataNasc" : "04/04/1985"
}
{
  "_id" : ObjectId("615cce897a8baf08c1a98a62"),
  "Nome" : "Luciana Adriana Fabiana Lopes",
  "CPF" : "52704213216",
  "DataNasc" : "04/04/1988"
}
>
  
```

Figura 26 - Melhor estruturação para visualização dos dados dos profissionais - (Arquivo pessoal)

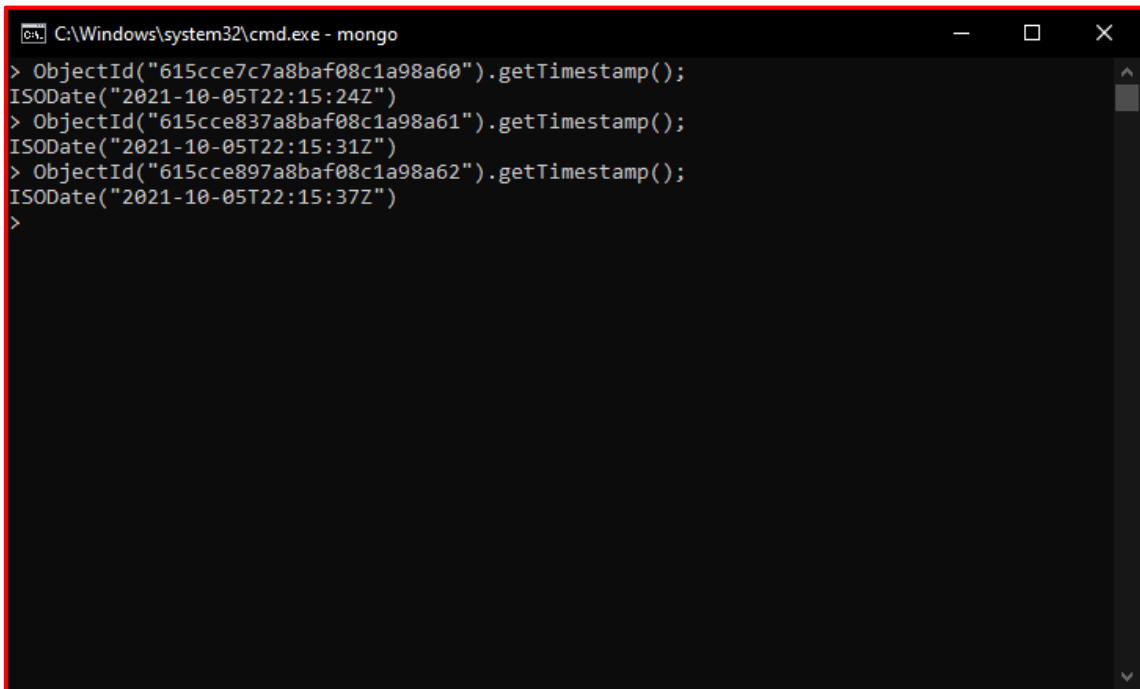
D) Consultas por igualdade utilizando os métodos find() e pretty()

```

C:\Windows\system32\cmd.exe - mongo
> db.profissionais.find({CPF: "79502397860"}).pretty();
{
  "_id" : ObjectId("615cce7c7a8baf08c1a98a60"),
  "Nome" : "Cauã Davi Mateus Ramos",
  "CPF" : "79502397860",
  "DataNasc" : "18/05/1970"
}
> db.profissionais.find({CPF: "81567490140"}).pretty();
{
  "_id" : ObjectId("615cce837a8baf08c1a98a61"),
  "Nome" : "Isis Giovana Kamilly Cavalcanti",
  "CPF" : "81567490140",
  "DataNasc" : "04/04/1985"
}
> db.profissionais.find({CPF: "52704213216"}).pretty();
{
  "_id" : ObjectId("615cce897a8baf08c1a98a62"),
  "Nome" : "Luciana Adriana Fabiana Lopes",
  "CPF" : "52704213216",
  "DataNasc" : "04/04/1988"
}
>
  
```

Figura 27 - Consulta por igualdade utilizando find() e pretty() - (Arquivo pessoal)

- E) Comando para consulta de quando o registro foi criado utilizando o ObjectId referente a cada um dos Documents

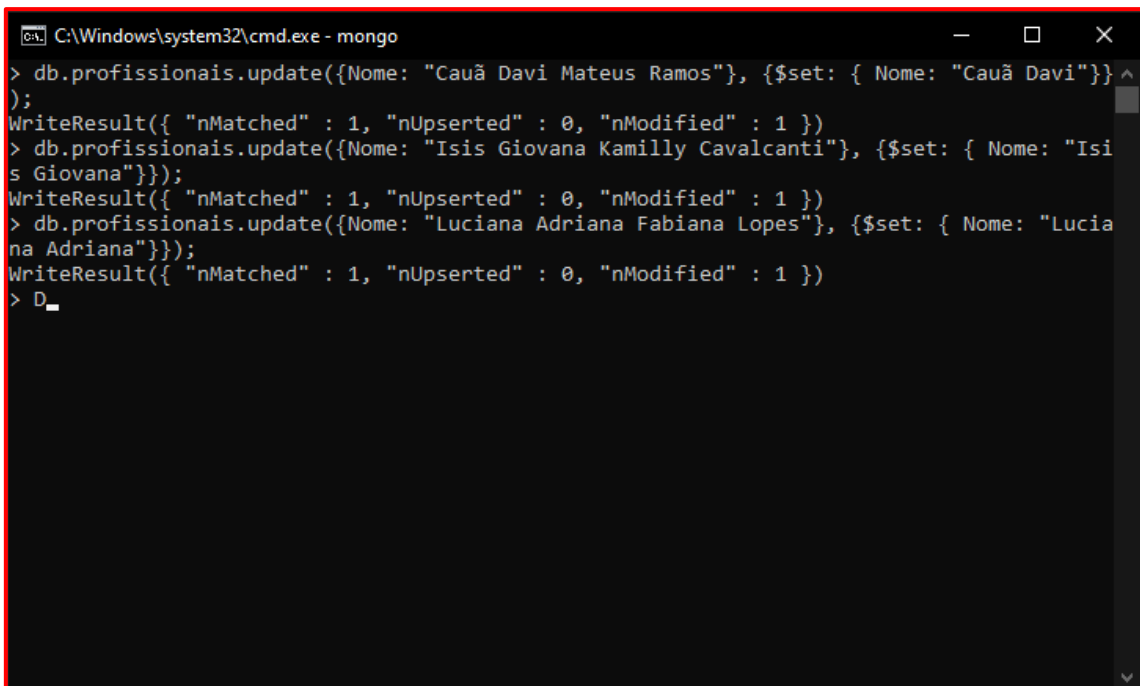


```

C:\Windows\system32\cmd.exe - mongo
> ObjectId("615cce7c7a8baf08c1a98a60").getTimestamp();
ISODate("2021-10-05T22:15:24Z")
> ObjectId("615cce837a8baf08c1a98a61").getTimestamp();
ISODate("2021-10-05T22:15:31Z")
> ObjectId("615cce897a8baf08c1a98a62").getTimestamp();
ISODate("2021-10-05T22:15:37Z")
>
  
```

Figura 28 - Consulta de dados por ObjectId - (Arquivo pessoal)

- F) Alterando os registros da collection "profissionais"



```

C:\Windows\system32\cmd.exe - mongo
> db.profissionais.update({Nome: "Cauã Davi Mateus Ramos"}, {$set: { Nome: "Cauã Davi" }});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.profissionais.update({Nome: "Isis Giovana Kamilly Cavalcanti"}, {$set: { Nome: "Isis Giovana" }});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.profissionais.update({Nome: "Luciana Adriana Fabiana Lopes"}, {$set: { Nome: "Luciana Adriana" }});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> D_
  
```

Figura 29 - Alteração de dados profissionais - (Arquivo pessoal)

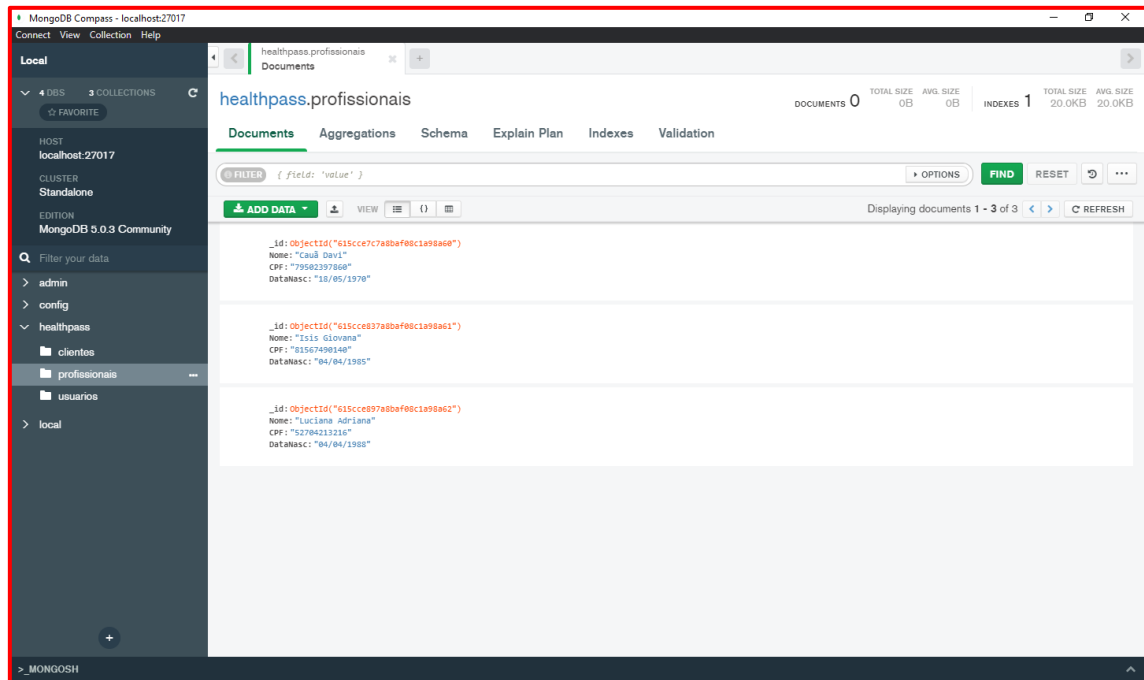


Figura 30 - "Proof" da alteração de dados dos profissionais dentro da collection - (Arquivo pessoal)

G) Removendo as Documents da Collection "profissionais"

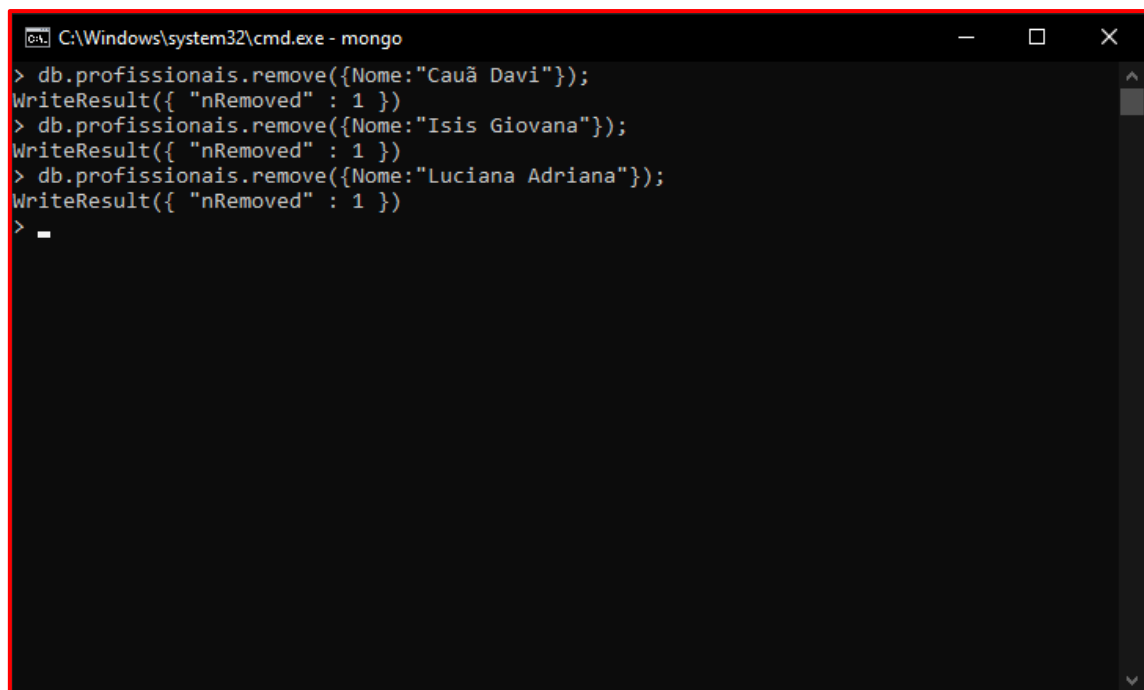


Figura 31 - Remoção de dados dentro da collection profissionais - (Arquivo pessoal)

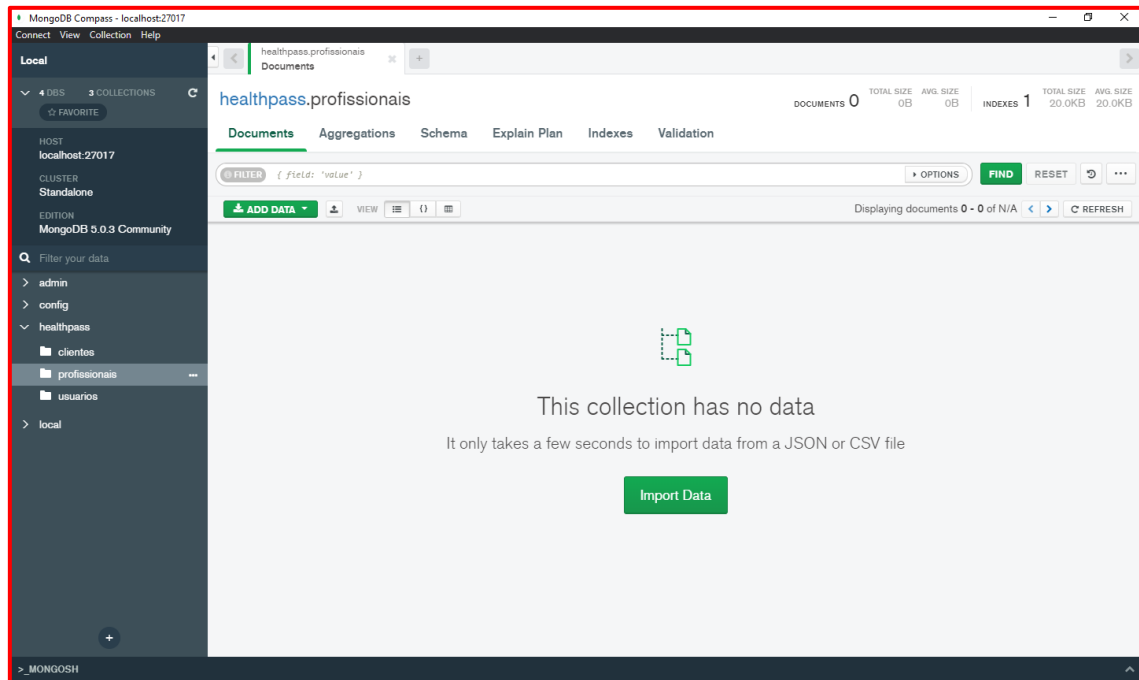


Figura 32 - "Proof" da remoção dos dados de profissionais - (Arquivo pessoal)

9. REGISTROS DAS EVIDÊNCIAS DO PROJETO

Para que seja apreciada a construção da base do projeto em C#, colocarei o repositório ao qual foi construído todo o projeto em MySQL.

Pela dimensão do arquivo, será necessário o “clone” do projeto para a própria máquina a fim de visualizar e/verificar com maiores detalhes o projeto e as programações realizadas.

[Link para o projeto – GITHUB - HTTPS](#)

ou

[Link para o projeto – GITHUB - SSH](#)

ou

[Link para o projeto – GITHUB – Github CLI](#)

10. CONSIDERAÇÕES FINAIS E EXPECTATIVAS

A implementação do **Sistema Unificado de Exames e Vacinas** representa um avanço significativo na modernização da saúde no Brasil. Utilizando **C# e MySQL** como base tecnológica, essa solução foi projetada para oferecer um sistema seguro, escalável e eficiente, permitindo que pacientes, profissionais de saúde, operadoras de planos e órgãos governamentais tenham acesso rápido e confiável às informações de saúde.

Com a centralização dos dados em uma **plataforma digital integrada**, o sistema elimina a fragmentação da informação, reduz burocracias, otimiza recursos e melhora a qualidade do atendimento médico. A interoperabilidade, garantida por APIs REST e padrões como **FHIR**, permite que diferentes sistemas de saúde, sejam públicos ou privados, possam se conectar de forma fluida, garantindo uma experiência unificada para todos os envolvidos.

Do ponto de vista mercadológico, essa solução se apresenta como uma oportunidade estratégica para **parcerias com hospitais, clínicas, laboratórios, operadoras de saúde e órgãos governamentais**. O modelo híbrido de monetização, abrangendo desde o **uso gratuito para pacientes até planos de assinatura para instituições de saúde**, garante a sustentabilidade e a expansão do sistema a longo prazo.

Ademais, cabe ressaltar que este TCC e projeto é baseado em uma prova conceito, ou seja, algumas modificações pontuais precisariam ser realizadas para que possam caminhar em consoante com as leis implementadas pela LGPD, concomitante a modificações de alocação em nuvem de acordo com a evolução natural de melhores tecnologias e implementações mais seguras ao usuário final.

Expectativas para o Futuro

Expansão Nacional: Com a adesão inicial de instituições de saúde e operadoras de planos, espera-se que a plataforma cresça gradualmente, tornando-se um padrão na digitalização de prontuários médicos no Brasil.

Maior Eficiência na Saúde Pública: A integração com o **SUS e secretarias estaduais** permitirá um melhor controle de vacinação, exames e diagnósticos, reduzindo desperdícios e otimizando campanhas de imunização.

Uso de Inteligência Artificial e Big Data: A análise de dados permitirá a implementação de **modelos preditivos de saúde**, auxiliando médicos na tomada de decisão e contribuindo para a identificação precoce de surtos epidemiológicos.

Interoperabilidade Global: Em uma fase futura, o sistema poderá expandir-se para outros países da América Latina, enfrentando desafios semelhantes na gestão de dados de saúde.

Maior Controle e Autonomia para os Pacientes: Com um aplicativo acessível e intuitivo, os usuários terão **total controle sobre seus dados de saúde**, podendo compartilhá-los com profissionais de confiança sempre que necessário.

O **Sistema Unificado de Exames e Vacinas** surge como uma inovação essencial para a digitalização da saúde no Brasil. Com tecnologia moderna, segurança reforçada e um modelo de negócios viável, ele tem potencial para transformar a maneira como informações médicas são armazenadas, acessadas e utilizadas, beneficiando toda a cadeia de saúde.

A expectativa é que, nos próximos anos, essa solução se torne **um pilar fundamental da transformação digital da saúde no Brasil**, promovendo um sistema **mais eficiente, acessível e centrado no paciente**.

REFERÊNCIAS

Importância de um Sistema Unificado de Exames e Vacinas

ORGANIZAÇÃO MUNDIAL DA SAÚDE (OMS). *Estratégia de Saúde Digital Global 2020-2025*. Genebra, 2020. Disponível em: <https://www.who.int/>. Acesso em: 10 mar. 2025.

BRASIL. MINISTÉRIO DA SAÚDE. *Plano Nacional de Saúde Digital: Estratégia de Transformação Digital no SUS*. Brasília, 2023. Disponível em: <https://www.gov.br/saude>. Acesso em: 10 mar. 2025.

Impacto no Ecossistema Brasileiro e Benefícios da API

BRASIL. MINISTÉRIO DA SAÚDE. DATASUS. *Transformação Digital na Saúde: Interoperabilidade e Compartilhamento de Dados no SUS*. Brasília, 2023. Disponível em: <https://datasus.saude.gov.br/>. Acesso em: 10 mar. 2025.

HL7 INTERNATIONAL. *Fast Healthcare Interoperability Resources (FHIR) – Padrão para Integração de Dados de Saúde*. HL7, 2023. Disponível em: <https://hl7.org/fhir/>. Acesso em: 10 mar. 2025.

Casos de Uso e Benefícios para a População

AGÊNCIA NACIONAL DE SAÚDE SUPLEMENTAR (ANS). *Diretrizes para Interoperabilidade no Setor de Saúde Privada*. Brasília, 2022. Disponível em: <https://www.ans.gov.br/>. Acesso em: 10 mar. 2025.

MCKINSEY & COMPANY. *How Digital Health Technologies are Transforming Healthcare*. 2023. Disponível em: <https://www.mckinsey.com/>. Acesso em: 10 mar. 2025.

Solução Mercadológica e Modelos de Negócio

BANCO MUNDIAL. *Parcerias Público-Privadas no Setor de Saúde: Guia de Melhores Práticas*. Washington, DC, 2022. Disponível em: <https://www.worldbank.org/>. Acesso em: 10 mar. 2025.

ROCK HEALTH. *Digital Health Market Trends & Investment Insights*. 2023. Disponível em: <https://rockhealth.com/>. Acesso em: 10 mar. 2025.

Desenvolvimento com C# e MySQL

MICROSOFT. *Documentação Oficial do .NET (ASP.NET Core Web API).* Redmond, WA, 2024. Disponível em: <https://learn.microsoft.com/en-us/aspnet/core/>. Acesso em: 10 mar. 2025.

ORACLE CORPORATION. *MySQL Documentation: Guia Oficial para Desenvolvimento em MySQL.* 2023. Disponível em: <https://dev.mysql.com/doc/>. Acesso em: 10 mar. 2025.

OPEN WEB APPLICATION SECURITY PROJECT (OWASP). *Security Best Practices for API Development.* 2024. Disponível em: <https://owasp.org/>. Acesso em: 10 mar. 2025.

Considerações Finais e Expectativas

BANCO NACIONAL DE DESENVOLVIMENTO ECONÔMICO E SOCIAL (BNDES). *Tendências da Digitalização na Saúde no Brasil.* Rio de Janeiro, 2023. Disponível em: <https://www.bndes.gov.br/>. Acesso em: 10 mar. 2025.

MIT TECHNOLOGY REVIEW. *The Future of AI in Healthcare: How Artificial Intelligence is Transforming Medicine.* Cambridge, MA, 2024. Disponível em: <https://www.technologyreview.com/>. Acesso em: 10 mar. 2025.
