



**UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS I - CAMPINA GRANDE
CENTRO DE CIÊNCIAS E TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO
CURSO DE GRADUAÇÃO EM BACHARELADO EM CIÊNCIA DA
COMPUTAÇÃO**

LUANA VIDAL DE NEGREIROS NÓBREGA

**VISUALIZAÇÃO DE DADOS COM GRAFANA: UM RELATO DE
EXPERIÊNCIA EM SISTEMAS DE SAÚDE**

**CAMPINA GRANDE
2024**

LUANA VIDAL DE NEGREIROS NÓBREGA

**VISUALIZAÇÃO DE DADOS COM GRAFANA: UM RELATO DE
EXPERIÊNCIA EM SISTEMAS DE SAÚDE**

Trabalho de Conclusão de Curso apresentado
ao Departamento de Computação do Centro
de Ciências e Tecnologia da Universidade
Estadual da Paraíba como requisito parcial
à obtenção do título de bacharel em Ciência
da Computação.

Área de concentração: Ciência de Dados

Orientador: Prof. Dr. Paulo Eduardo e Silva Barbosa

CAMPINA GRANDE

2024

É expressamente proibida a comercialização deste documento, tanto em versão impressa como eletrônica. Sua reprodução total ou parcial é permitida exclusivamente para fins acadêmicos e científicos, desde que, na reprodução, figure a identificação do autor, título, instituição e ano do trabalho.

N754v Nóbrega, Luana Vidal de Negreiros.

Visualização de dados com grafana [manuscrito] : um relato de experiência em sistemas de saúde / Luana Vidal de Negreiros Nóbrega. - 2024.

54 f. : il. color.

Digitado.

Trabalho de Conclusão de Curso (Graduação em Ciência da computação) - Universidade Estadual da Paraíba, Centro de Ciências e Tecnologia, 2024.

"Orientação : Prof. Dr. Paulo Eduardo e Silva Barbosa, Departamento de Computação - CCT".

"Coorientação: Prof. Dra. Ana Isabella Muniz Leite, Departamento de Computação - CCT".

1. Grafana. 2. Tecnologia da informação em saúde. 3. Visualização de dados. I. Título

21. ed. CDD 610.285

LUANA VIDAL DE NEGREIROS NÓBREGA

VISUALIZAÇÃO DE DADOS COM GRAFANA: UM RELATO DE EXPERIÊNCIA
EM SISTEMAS DE SAÚDE

Artigo Científico apresentado à
Coordenação do Curso de Ciência da
Computação da Universidade Estadual
da Paraíba, como requisito parcial à
obtenção do título de Bacharela em
Computação

Aprovada em: 19/11/2024.

BANCA EXAMINADORA

Documento assinado eletronicamente por:

- **Misael Elias de Moraes** (***.304.834-**), em **28/11/2024 15:59:59** com chave **f767f556adba11efa4d81a7cc27eb1f9**.
- **Ana Isabella Muniz Leite** (***.834.864-**), em **25/11/2024 20:07:58** com chave **1c7a7066ab8211ef91971a7cc27eb1f9**.
- **Paulo Eduardo e Silva Barbosa** (***.886.504-**), em **25/11/2024 11:28:21** com chave **858943d2ab3911efa6241a7cc27eb1f9**.

Documento emitido pelo SUAP. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse https://suap.uepb.edu.br/comum/autenticar_documento/ e informe os dados a seguir.

Tipo de Documento: Folha de Aprovação do Projeto Final

Data da Emissão: 13/03/2025

Código de Autenticação: b61658



À minha mãe, por
sempre acreditar em
mim, DEDICO.

AGRADECIMENTOS

Em um momento de dúvida, optar por iniciar outra graduação não foi uma decisão fácil, mas com todo apoio de pessoas especiais, o processo ficou mais leve. Gostaria de agradecer, então:

Em especial, por todo apoio, além da convivência diária e suportar todas minhas aflições, agradeço a minha companheira de vida, Andreia, que mesmo ainda, enquanto estávamos nos conhecendo, serviu de base de apoio para tornar a decisão mais convicta.

A minha mãe, por sempre acreditar em mim, sempre oferece o colo nas angústias e me conhecer, por vezes, mais do que a mim mesma.

Ao meu pai, por não ter soltado minha mão no início dessa trajetória, agradeço pelo apoio e por todos momentos que pensei em desistir, sempre me ligava e me passava conforto e segurança.

As minhas irmãs que são espelhos, os quais sigo como referências de mulheres donas de si.

Aos querido amigo de curso Henrique, quem dividiu comigo as aflições do curso, brincadeiras e companheirismo.

Aos meus filhotes queridos que adotei no curso, que por muitas vezes me ajudaram Arthur, Philipe, Neto e Marinho.

Ao querido orientador Paulo, que abriu as portas, me deu a oportunidade de olhar para frente e acreditar em mim, com isso consegui ultrapassar barreiras e entender, de fato, que eu sou capaz.

A minha queridíssima co-orientadora Ana Isabella, não tenho palavras para agradecer todo apoio, atenção e por muitas vezes, compreensão. Entender meu processo de desenvolvimento e tornar a construção desse trabalho, o mais leve possível.

Sei que o agradecimento a papai do céu deveria vir primeiro, mas tenho certeza que cada pessoa que cruzou essa minha jornada não deixou de ser obra desse Cara. Obrigada!

A vocês todo meu carinho e agradecimento!

“Epígrafe - as Visualizações de dados ganham evidências e conduzem para novas camadas de compreensão, ao utilizar cada vez mais dados em variados níveis de complexidade, aprofundamento, cruzamento dos dados com o intuito de dar conta da mensagem que se deseja transmitir.”
Rodrigues (2019)[5]

RESUMO

Visualização de dados refere-se a representações visuais, como gráficos e mapas, usadas para entender, comunicar e dar suporte ao engajamento colaborativo e à tomada de decisões com dados e informações. Com o advento do Big Data na área da saúde, os desafios para os profissionais de saúde ao tentar chegar a uma compreensão completa dos dados do estado geral de saúde de cada paciente, demanda tempo. Assim, a necessidade de visualização adequada desses dados, como forma de dar suporte a esses profissionais na tomada de decisão, se tornou ainda mais evidente. Como um passo em direção à otimização na compreensão dos dados em sistemas de monitoramento da saúde, este artigo apresenta um relato de experiência na utilização do Grafana para visualização de dados da saúde no projeto da Sênior Saúde Móvel. Para isso, foi criado um pipeline de dados, que em sua origem, dispunha de bases de dados documental e relacional até a apresentação desses dados em um ambiente totalmente personalizável para os profissionais de saúde. Os resultados têm mostrado o acesso rápido às informações, uma melhor compreensão do conhecimento sobre a saúde dos pacientes, bem como suportado os profissionais de saúde, como fisioterapeutas, em diagnósticos dos pacientes.

Palavras-chave: visualização; grafana; dados; saúde.

ABSTRACT

Data visualization refers to visual representations, such as charts and maps, used to understand, communicate, and support collaborative engagement and decision-making with data and information. With the advent of big data in healthcare, the challenges faced by healthcare professionals in trying to gain a comprehensive understanding of each patient's overall health status are complex and time-consuming. Thus, the need for appropriate visualization of this data as a way to support these professionals in decision-making has become even more evident. As a step towards optimizing the understanding of data in health monitoring systems, this article presents an experience report on using Grafana for visualizing health data in the Senior Mobile Health project. For this, a data pipeline was created, which, at its origin, contained both document-based and relational databases until the presentation of this data in a fully customizable environment for healthcare professionals. The results have shown quick access to information and a better understanding of patient health knowledge, and they have supported healthcare professionals, such as physiotherapists, in diagnosing patients.

Keywords: visualization; grafana; data; healthcare.

SUMÁRIO

	Página
1	INTRODUÇÃO
	9
2	FUNDAMENTAÇÃO TEÓRICA
	11
2.1	Visualização de dados 11
2.2	Trabalhos Relacionados 12
3	VISUALIZAÇÃO DE DADOS DE SAÚDE COM GRAFANA
	17
3.1	Grafana e a Senior Saúde Móvel 17
3.2	Desafios 18
4	RESULTADOS
	20
5	CONCLUSÃO
	24
	REFERÊNCIAS
	24
	APÊNDICE A – MATERIAL SUPLEMENTAR
	25

1 INTRODUÇÃO

A visualização de dados é um processo analítico e criterioso, pois envolve a representação visual de informações obtidas na coleta de dados que, de outra forma, seriam inescrutáveis, permitindo assim fácil interpretação e análise. Ferramentas que auxiliam da criação de representações visuais utilizando gráficos de pizza, tabelas e gráficos de barras, são usadas para apresentar dados de forma clara e instrutiva para o usuário que está monitorando e analisando essas informações. Especificamente na área da saúde, isso significa visualizar informações em tempo real ou analisar histórico de dados, de determinado paciente. "A visualização de dados de saúde é essencial para a análise e monitoramento em tempo real, facilitando a tomada de decisões clínicas. Conforme Dowding et al. (2015)[4], dashboards eficientes auxiliam na interpretação de dados de qualidade e segurança, fornecendo uma visão clara e objetiva das informações."

Com ascensão do Big Data na saúde se faz necessário cada vez mais, a visualização adequada desses dados para a tomada de decisões, necessidade a qual foi impulsionada pela pandemia do COVID-19. Quando essas ferramentas são integradas a modelos preditivos, resulta em uma estratificação acelerada de dados que, por sua vez, leva a um melhor atendimento e acompanhamento, dos pacientes.

O objetivo deste projeto é apresentar a experiência de utilização do Grafana na visualização de dados de saúde através da plataforma Senior Saúde Móvel. A razão para selecionar o Grafana foi porque é uma ferramenta de código aberto que pode ser facilmente adaptada devido ao seu alto nível de personalização e pode ser perfeitamente integrada com bancos de dados como InfluxDB e MongoDB. O objetivo principal da visualização é demonstrar como o Grafana pode ser usado de forma eficaz na supervisão de informações sobre o estado de saúde dos idosos obtidas através de dispositivos IoT – garantindo assim que o monitoramento se torne mais acessível para os profissionais de saúde.

O trabalho contribui com a facilitação do processo de integração do Grafana com o MongoDB, enfatizando a possibilidade de conexão através de um formato não oficial, dando oportunidade de utilizar as ferramentas integradas sem custo, além de apresentar o processo de construção das representação visual dos dados da Senior Saúde Móvel que estão armazenados no MongoDB, bem como as limitações que foram enfrentadas nesse processo.

Este artigo está estruturado da seguinte forma: a seção de Fundamentação teórica contém conteúdo sobre a visualização de dados da saúde; a seção de Trabalhos Relacionados fala sobre trabalhos que embasaram a construção desse artigo; a seção de Visualização de dados de saúde com Grafana contém detalhes sobre o processo de integração do MongoDB e InfluxDB com Grafana; a seção de Desafios discute os desafios relevantes; a seção de Resultados descreve os resultados alcançados com a integração; a seção de Conclusão,

finaliza, apresenta vantagens e desvantagens, além de resumir a experiência.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Visualização de dados

Segundo Card, Mackinlay e Shneiderman (1999)[2], em seu livro “Readings in Information Visualization: Using Vision To Think”, apresenta a definição clássica de visualização da informação, consideramos então sinônimo ao termo do nosso estudo “visualização de dados”, temos que “A visualização da informação é o uso de representações visuais interativas com suporte computacional de dados abstratos para ampliar a cognição. Seu propósito não é a imagem em si, mas sim o insight (ou a rápida assimilação de informação ou o monitoramento de grandes quantidades de dados).”

Saindo um pouco do âmbito artístico e clássico, e partindo para uma conceituação mais técnica Setzer (1999)[6] apresenta o dado como “uma sequência de símbolos quantificados ou quantificáveis.” bem como a informação sendo “uma abstração informal (isto é, não pode ser formalizada através de uma teoria lógica ou matemática), que representa algo significativo para alguém através de textos, imagens, sons ou animação. Não é possível processar informação diretamente em um computador. Para isso é necessário reduzi-la a dados.”

Assim sendo, a visualização de dados tem como “especialidade contar histórias. Além disso, visualização é um processo de mapeamento de informações para imagens”. (RODRIGUES, 2019)[5]

Considerando que a visualização é um processo de concepção visual das informações obtidas a partir de dados representados por uma coleção de dados, tornando assim, esse processo simples e efetivo, utilizando-se de ferramentas disponíveis para mapear os dados e apresenta-los da forma mais instrutiva possível, através de representações como: gráficos de pizza, tabelas, histogramas, gráficos de barras, gráfico de linhas, etc. A depender da análise que virá a ser realizada demandará o tipo de técnica e representação que será escolhida, para que as visões sejam apresentadas para o usuário conforme seu interesse.

Para dados de saúde não é diferente, a necessidade de visualizar as informações obtidas de equipamentos de monitoramento, bem como informações enviadas em forma de inputs, de pacientes, é de extrema importância quando se trata de informações em tempo real. Segundo Badgeley et al. (2016)[1] o objetivo de utilizar ferramentas para visualização de informações relativas está em “Integrar essas ferramentas de visualização com modelos preditivos e ferramentas de estimativa de risco pode apoiar a estratificação acelerada de pacientes para um atendimento aprimorado. Os dashboards clínicos são ferramentas que podem capturar visualmente uma visão transversal de uma variedade de métricas de qualidade, incluindo o status dos pacientes, medidas de progresso e resultados clínicos, fornecendo uma visão unificada que pode auxiliar os profissionais de saúde na tomada de decisões informadas.”[4] fortalece que o uso de dashboards pode melhorar significa-

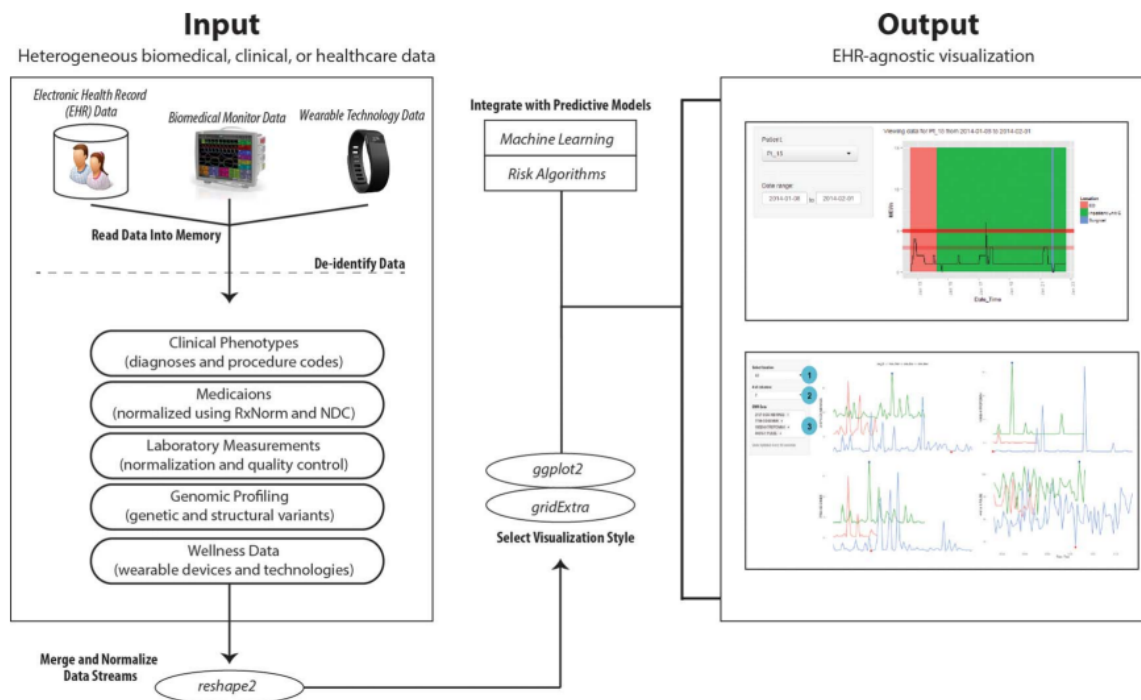
tivamente o cuidado ao paciente, fornecendo informações essenciais de forma rápida e acessível.”

2.2 Trabalhos Relacionados

Observando as literaturas mais recentes publicadas que abordam o assunto sobre visualização de dados, nota-se que, há poucos recortes sobre o assunto, sendo mais abordados entre 2010 e 2021, onde percebe-se ter sido um período de adoção das ferramentas de visualização de dados na saúde, como Tableau, Power BI, etc., tanto por conta do avanço dessas tecnologias devido a massa de dados relativos a saúde terem aumentado consideravelmente, quanto a necessidade de representações gráficas, em tempo real, desses dados durante a pandemia do COVID-19, foram decisivos para a necessidade de compartilhamento de conhecimento voltados para esse tema.

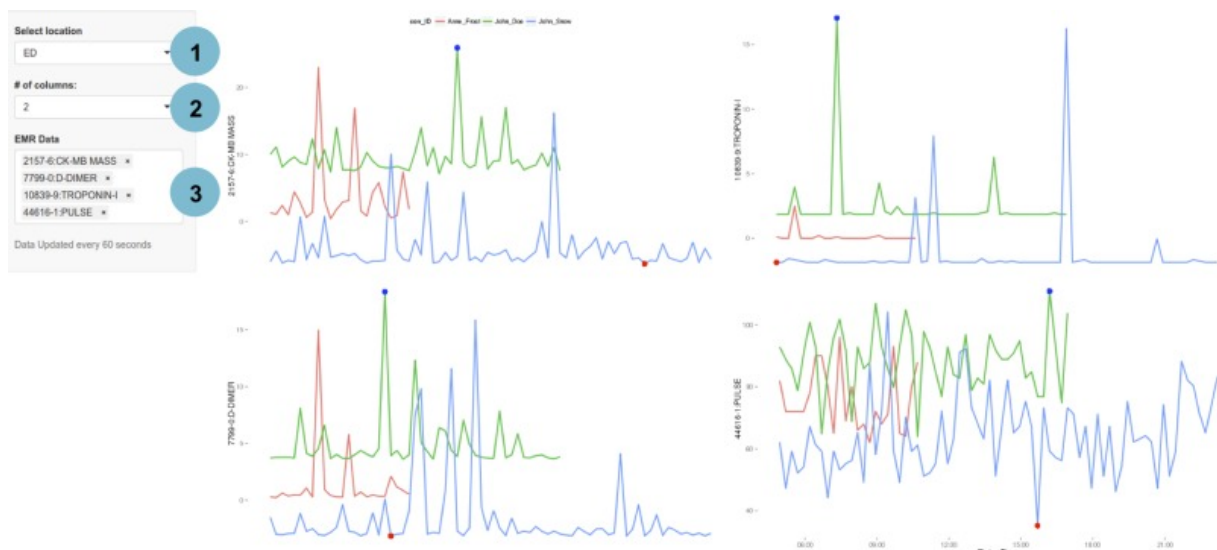
Badgeley et al. (2016)[1] utilizaram o kit de ferramenta de visualização eletrônica de dados de saúde, o EHDViz (Electronic healthcare data visualization) - O EHDViz não se trata de um sistema desenvolvido, mas um conjunto de ferramentas, que foram organizados de forma a apresentar, e representar, dados de monitores de saúde, sua estrutura usa pacotes R para gerenciamento de dados, normalização e produção de visualizações de alta qualidade pela web usando a arquitetura de servidor web R/Shiny - para projetar, criar e prototipar dashboards clínicos com tecnologias open-source, que facilita a visualização de dados de saúde para ajudar nas decisões médicas e operacionais, em tempo real. O projeto destaca a importância de ter ferramentas acessíveis e personalizáveis, permitindo que as instituições de saúde ajustem rapidamente suas ferramentas conforme as necessidades emergentes durante crises sanitárias.

Figura 1 - Arquitetura do EHDViz. EHDViz, visualização de dados de saúde eletrônicos.



Fonte: *Badgeley et al.*, 2016.

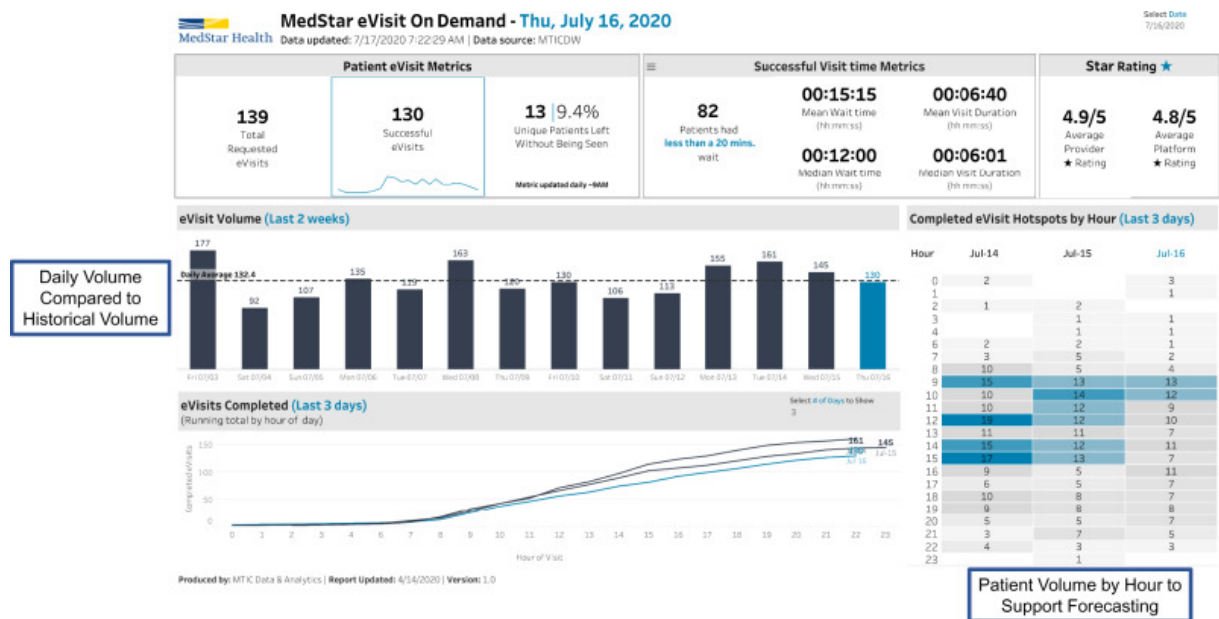
Figura 2 - Um painel de avaliação clínica personalizado desenvolvido usando EHDViz, ilustrando dados no departamento de emergência. Os recursos deste painel incluem: (1) seleção de unidades clínicas específicas usando um menu suspenso, (2) controle do layout e (3) seleção de pacientes testados para biomarcadores específicos, com integração de códigos CID-9.



Fonte: *Badgeley et al.*, 2016.

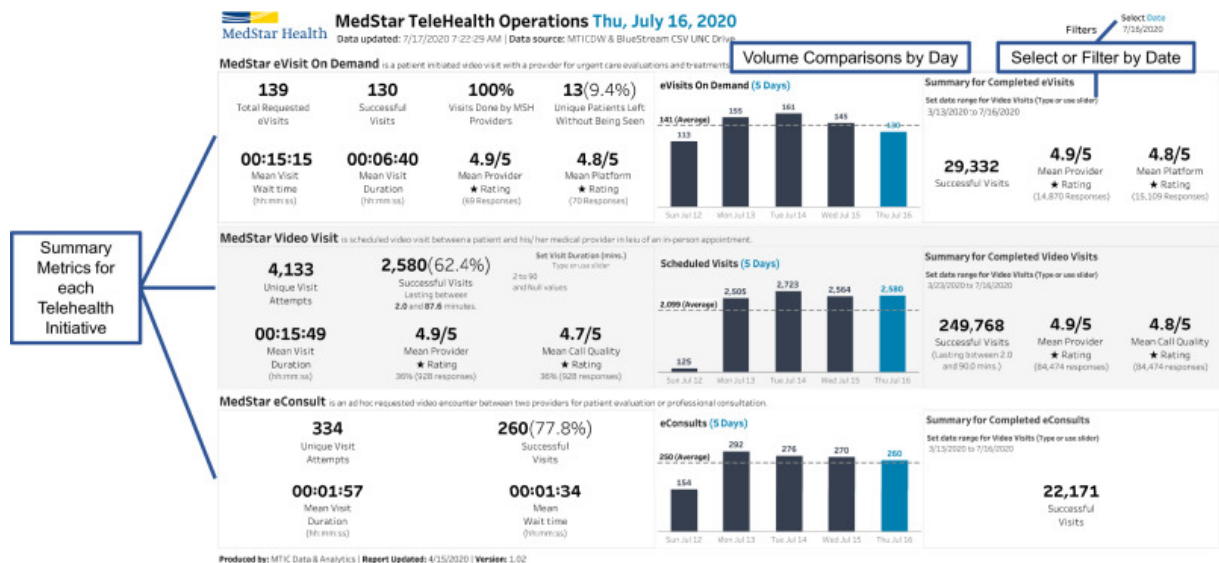
No estudo de Dixit et al. (2020) [3], os autores utilizam o Tableau como ferramenta de criação de dashboards para visualização de dados, seu formato interativo e dinâmico melhorou a consciência situacional, sobre os pacientes, dos usuários. Dixit et al. (2020), aborda no seu trabalho também, a eficácia da ferramenta durante a pandemia do COVID-19, sendo abordado seu suporte a telemedicina.

Figura 3 - Exemplo de dashboard utilizado para gerenciar as operações da equipe de eVisit, permitindo monitoramento em tempo real e melhor gerenciamento da demanda.



Fonte: Dixit et al., 2020.

Figura 4 - Exemplo de dashboard utilizado pelos líderes operacionais de tele-saúde para monitoramento estratégico, otimização de fluxos de trabalho e análise de desempenho em tempo real.



Fonte: Dixit et al., 2020.

Utilizando as experiências dos trabalhos dispostos, foi criado uma tabela comparativa, onde visualizamos as ferramentas mencionadas, além observar a ferramenta do Power BI, a qual é bastante utilizada atualmente ainda para representação de dados através de dashboards, em comparativo com o Grafana, o framework em questão no estudo desse artigo.

Tabela 1 - Comparação entre Ferramentas de Visualização de Dados

	Tableau	Power BI	EHDViz	Grafana
Lançamento	2003	2014	2016	2014
Interface de Usuário	Intuitiva, arrastar e soltar	Intuitiva, arrastar e soltar	Focada em saúde, customizável	Focada em gráficos e alertas, customizável
Integração de Dados	Alta, suporta várias fontes de dados	Alta, especialmente com produtos Microsoft	Alta, com foco em dados de saúde	Alta, suporta muitas fontes de dados
Visualizações Suportadas	Diversas (gráficos, mapas, tabelas)	Diversas (gráficos, mapas, tabelas)	Focadas em saúde (gráficos, mapas, tabelas)	Diversas (gráficos, alertas, painéis)
Interatividade	Alta (dashboards interativos)	Alta (dashboards interativos)	Alta (focada em necessidades de saúde)	Alta (dashboards interativos)
Facilidade de Uso	Alta, fácil de usar sem necessidade de codificação	Alta, fácil de usar, integração com Excel	Alta, projetada para profissionais de saúde	Média, requer conhecimento técnico
Capacidades de Análise	Avançadas (filtros, drill-downs, cálculos)	Avançadas (filtros, drill-downs, cálculos)	Personalizável para análises específicas	Avançadas (alertas, consultas)
Suporte a Tempo Real	Limitado	Limitado	Sim, foco em dados em tempo real	Sim, excelente suporte a tempo real
Custo	Variável, baseado em licenciamento	Variável, baseado em licenciamento	Geralmente mais acessível	Gratuito (versão open-source), Enterprise pago
Foco no Setor de Saúde	Geral, mas amplamente usado em saúde	Geral, mas amplamente usado em saúde	Exclusivamente para saúde	Geral, mas pode ser adaptado para saúde
Comunidade e Suporte	Grande, com muitos recursos de treinamento	Grande, com muitos recursos de treinamento	Menor, mais focada no setor	Grande, especialmente na versão open-source

Fonte: Elaborado pelo autor, 2024.

3 VISUALIZAÇÃO DE DADOS DE SAÚDE COM GRAFANA

3.1 Grafana e a Senior Saúde Móvel

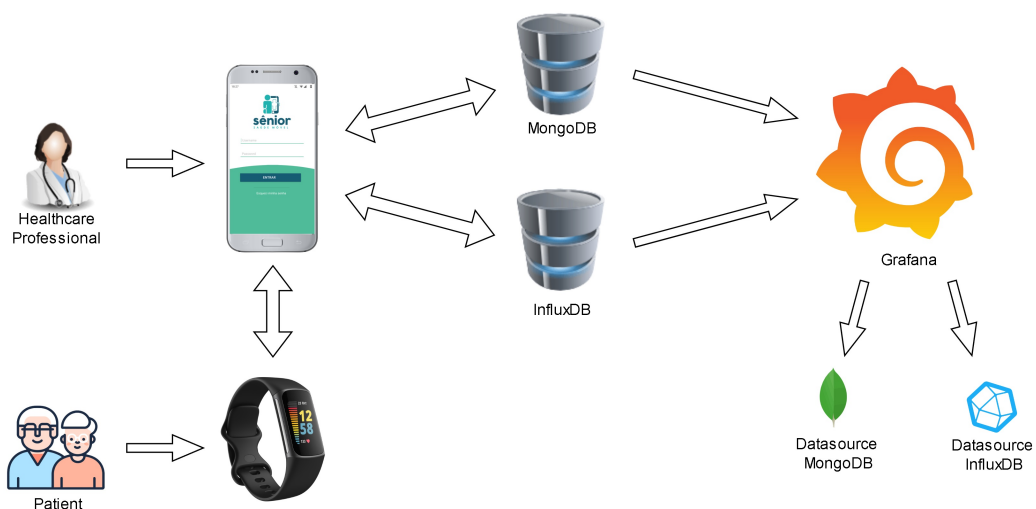
O Grafana, quando comparado a outras ferramentas de visualização de dados, se sobressai, pois além de ser open-source, seus dashboards são altamente personalizáveis, bem como tem uma fácil integração com outros serviços. No momento da decisão de utilizar o Grafana, além de abordadas as situações acima, era necessário sua facilidade em integrar-se aos bancos de dados utilizados pela Senior Saúde Móvel.

A Senior Saúde Móvel é uma plataforma cujo objetivo é monitorar idosos através do uso de IoT, no caso o FitBit, pelo qual se faz a coleta de dados parametrizados como: frequência cardíaca, calorias, número de passos, etc - através de um smartwatch (como mostra a Figura 1). Os dados monitorados são armazenados em um banco de dados específicos a depender da característica do dado coletado.

Os dados coletados precisam ser monitorados pelos profissionais de saúde que acompanham os idosos. Além da plataforma fornecer as informações, fez-se necessário a visualização dessas informações em forma de tabelas e gráficos para que se fizesse uma análise mais detalhada sobre os dados de determinado paciente ou do grupo de pacientes daquele profissional de saúde.

Para visualizar os dados da Senior utilizando o Grafana, precisava estar claro os tipos e quais bancos de dados eram utilizados pela plataforma, e foi identificado o InfluxDB e o MongoDB. O InfluxDB coleta dados em formato de séries temporais, já o MongoDB são dados obtidos de formulários preenchidos com medidas e informações do paciente.

Figura 5 - Fluxo de Visualização de Dados do Sistema



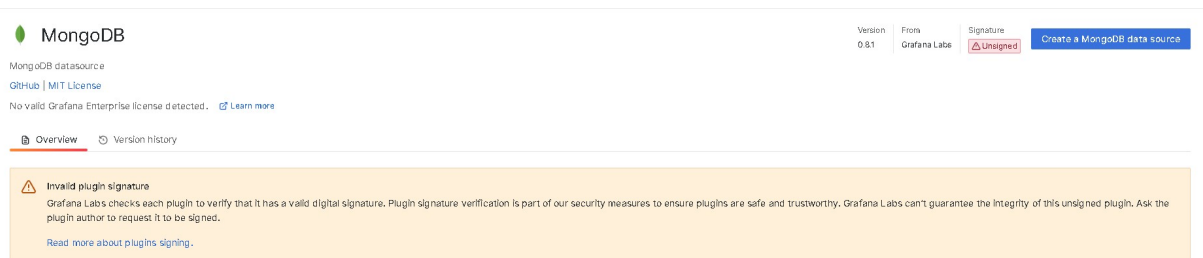
Fonte: Elaborado pelo autor, 2024.

Quando os bancos de dados são alimentados, o Grafana se conecta a eles a partir da criação de data source, onde é passada as informações necessárias para efetivar a conexão com o banco, para essa conexão os data sources se utilizam de plugins, estes que podem ser licenciados ou gratuitos. Quando gratuitos, todos atributos de configuração para a conexão são liberados, quando licenciados esses atributos são liberados mediante pagamento de uma taxa determinada pela ferramenta a qual oferece o serviço do plugin ao Grafana.

3.2 Desafios

A grande questão observada foi o fato do plugin do MongoDB no Grafana ser licenciado, como o objetivo era tirar todo proveito de uma ferramenta open source como é o Grafana, acabamos tomando o partido de criar o datasource do MongoDB sem a utilização do plugin oficial.

Figura 6 - Imagem retirada do Grafana demonstrando que o plugin oficial do MongoDB requer uma licença para utilização.



Fonte: Grafana.

O formato de conexão não oficial, implica em algumas limitações nas consultas como, por exemplo, a utilização apenas do “aggregate” e não o “find”. No entanto, todos os nossos painéis funcionarão corretamente e todos os dados consultados podem ser exibidos.

Figura 7 - Imagem retirada do Grafana exibindo os datasources criados a partir da conexão não oficial com os bancos de dados MongoDB.



Fonte: Grafana.

Existem cinco bancos de dados MongoDB adicionados, cada um deles é conectado ao Grafana mediante a criação do datasource. Estes estão conectados via `http://localhost:3333` desde que os contêineres de docker desses bancos de dados são executados na mesma máquina virtual que o recipiente do Grafana. Através de uma combinação de usuário e senha a conexão é estabelecida.

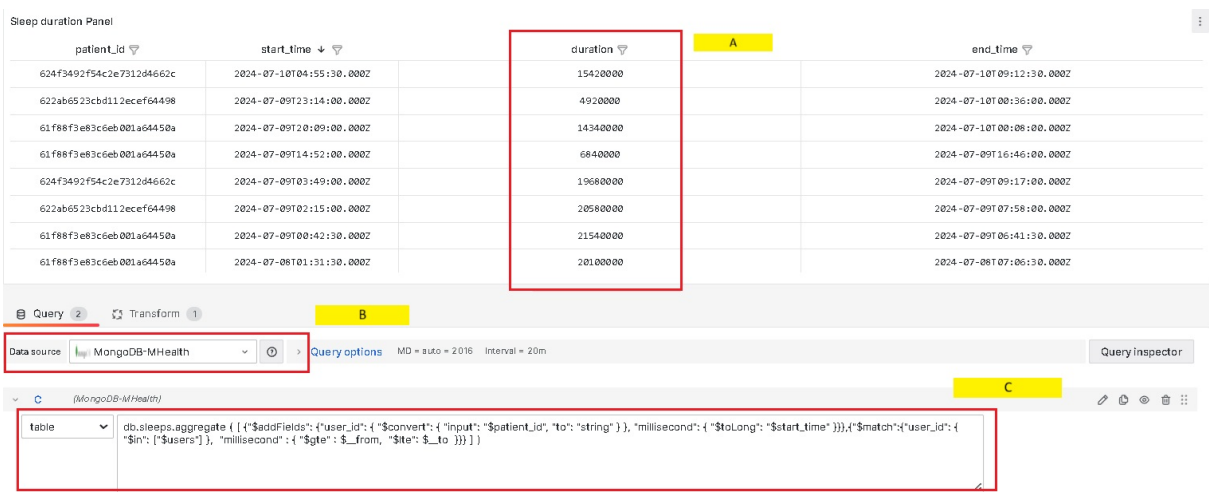
Então, após datasources criados partimos para criar o dashboard. No dashboard é onde serão personalizados os painéis com os gráficos ou tabelas que irão representar as informações resultantes das buscas nos bancos.

4 RESULTADOS

Para a criação dos painéis é necessários definir os dados que estarão representados nos gráficos ou tabelas, bem como a escolha da base de dados que será feita a consulta, no caso o datasource (como mostra Figura 3 do tópico Desafios), é a criação das queries (como mostra Figura 2, deste tópico) que irá obter essas informações para montar a melhor representação, isto é, a personalização dos painéis.

Como exemplo temos o dashboard onde serão monitorados os dados de sono dos pacientes da Senior Saúde Móvel, o qual foi definido o formato de tabela. Nesse dashboard são representando visualmente a duração do sono do paciente (como mostra a Figura 1, deste tópico), o tempo que o paciente levar para dormir, a quantidade de tempo que o paciente dorme desconsiderando os momentos acordado, o tempo de duração de cada estágio do sono, a quantidade de tempo que o paciente fica acordado, etc.

Figura 8 - Imagem retirada do Grafana mostrando a construção do painel de duração do sono, que será utilizado na composição do dashboard para monitoramento de padrões de sono.



Fonte: Grafana.

De acordo com a Figura 1, deste tópico, observa-se que a definição do datasource está atrelada a escolha do banco de dados que vai ter os dados necessários para a query. A conexão individual de cada banco de dados, que o MongoDB trata como collections, é criada através da definição dos datasources, então para cada collection do MongoDB, é criado um datasource no Grafana (como mostra a Figura 3, deste tópico), e a utilização de seus dados são definidas, exatamente, no momento que se define os dados que serão utilizados.

Na query usada para definir o painel de duração do sono, como foi nomeado, foi definida da seguinte forma:

Figura 9 - Extração de dados, através de query, para criação do painel de duração do sono.

```
db.sleeps.aggregate([
  {
    "$addFields": {
      "user_id": {
        "$convert": { "input": "$patient_id", "to": "string" }
      },
      "millisecond": {
        "$toLong": "$start_time"
      }
    }
  },
  {
    "$match": {
      "user_id": { "$in": ["$users"] },
      "millisecond": {
        "$gte": $__from,
        "$lte": $__to
      }
    }
  }
])
```

Fonte: Elaborado pelo autor, 2024.

A consulta `$addFields` adiciona o `$user_id`. O `user_id` é uma cópia do `$patient_id` onde os números são convertidos em uma cadeia para combinar o id do paciente com as variáveis do usuário no painel, como visto no `$match` mais tarde nesta consulta.

O tipo da variável como milissegundos é adicionado e relacionado com o `$start_time` para mais tarde corresponder esta vez com o tempo escolhido no painel.

Este `$match` corresponde ao usuário selecionado no painel a partir da variável usuário com o novo `user_id` adicionado.

Aqui o milissegundo é correspondido pelo filtro de tempo que é usado no painel para mostrar apenas os dados do filtro do tempo escolhido.

Com isso obtemos a consulta completado dos dados que queremos obter representado na tabela do painel, mas ainda temos o auxílio do Transform, uma espécie de filtro, onde tem os nomes das variáveis que foram obtidas a partir da query, e que você pode selecionar o que, de fato, irá ser apresentado na tabela. O tipo de filtro utilizado na maioria dos painéis é 'filtrar por nome'. Com esta transformação podemos escolher esconder alguns dados que não queremos mostrar no painel.

Figura 10 - Imagem retirada do Grafana demonstrando a utilização do recurso Transform para aplicar filtros nos dados que serão exibidos na tabela.

Sleep duration Panel

patient_id	start_time	duration	end_time
61773a35d068f5001c7820ad	2022-03-08T09:40:30.000Z	5100000	2022-03-08T11:05:30.000Z
61773a35d068f5001c7820ad	2022-03-08T12:38:30.000Z	7900000	2022-03-08T14:51:30.000Z
6222a71f072ced8c7854977f	2022-03-08T04:42:30.000Z	27600000	2022-03-08T12:22:30.000Z
61773a35d068f5001c7820ad	2022-03-09T02:02:00.000Z	22800000	2022-03-09T08:22:00.000Z
61773a35d068f5001c7820ad	2022-03-09T16:34:00.000Z	4320000	2022-03-09T17:46:00.000Z
60777eb853fd850019b745ae	2022-03-10T02:23:30.000Z	22140000	2022-03-10T08:32:30.000Z
60777eb853fd850019b745ae	2022-03-08T03:00:30.000Z	17760000	2022-03-08T07:56:30.000Z

Query 2 Transform 1

Filter by name

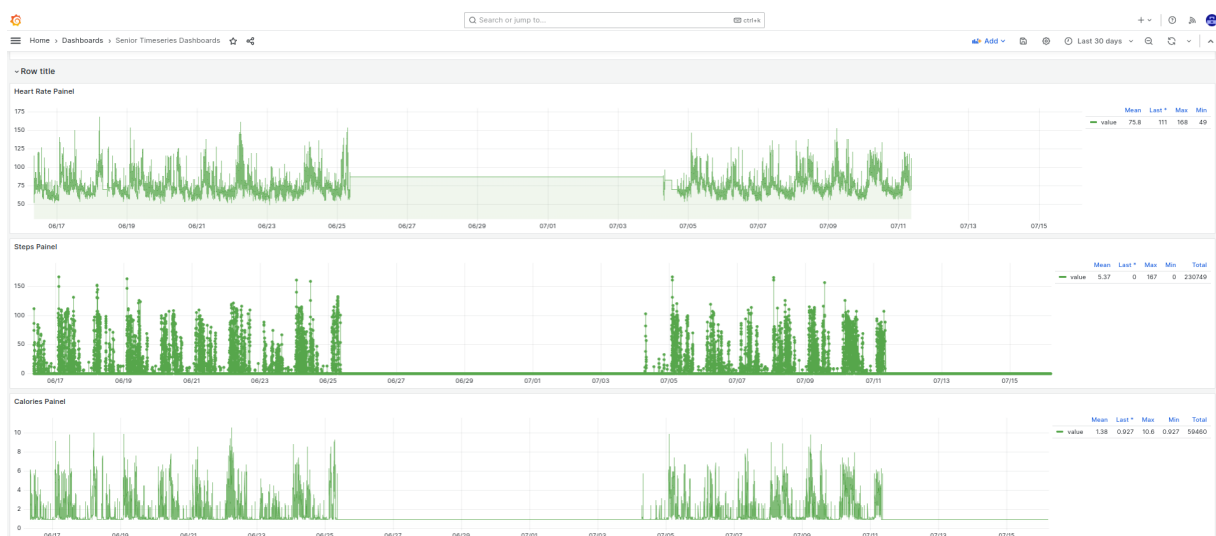
Identifier Regular expression pattern _id patient_id ✓ start_time ✓ _v created_at duration ✓ end_time ✓ pattern type user_id millisecond awakenings

Fonte: Grafana.

Com InfluxDB todo o processo de conexão com Grafana é mais simples, já que é seu plugin oficial pode ser utilizado também na versão open source. O InfluxDB, considerando a estrutura da Senior Saúde Móvel, armazena dados de série temporal. Dados de série temporal são aqueles que têm "carimbado" em cada registro a data/hora, esse tipo de banco de dados são utilizados para armazenar informações que exigem intervalo de data ou hora, toda consulta é realizada mediante o tempo.

No Grafana, foi criado um dashboard para apresentar esses dados de forma adequada e facilitada, onde foram adicionados painéis para visualização de frequência cardíaca, calorias, quantidade de passos dados pelo paciente em determinado espaço de tempo, distância percorrida e minutos ativos.

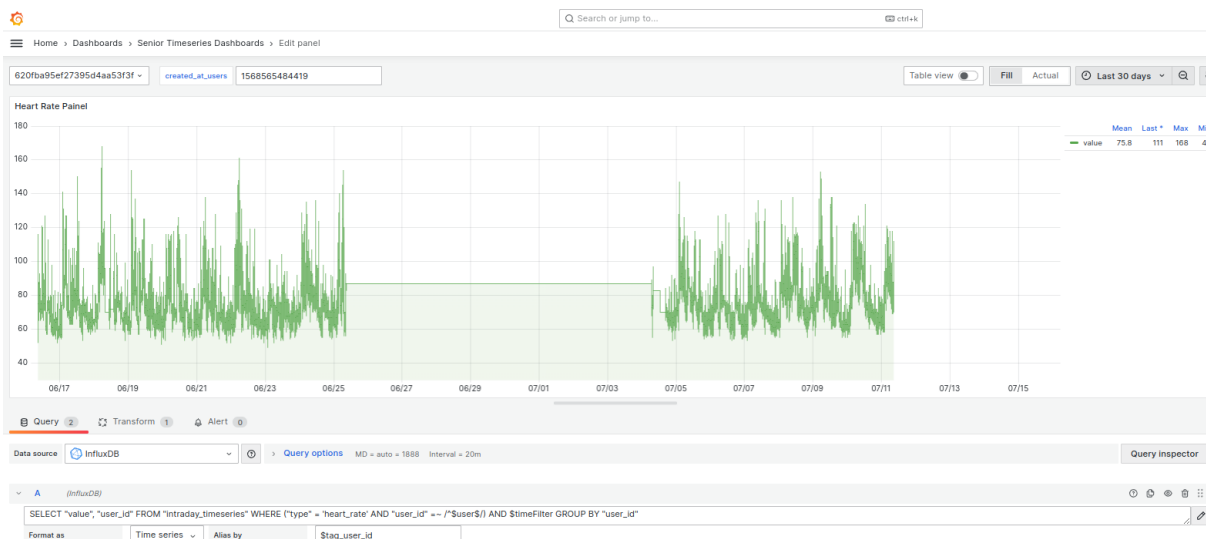
Figura 11 - Imagem retirada do Grafana demonstrando o dashboard utilizado pelo InfluxDB para visualização e análise de dados em séries temporais.



Fonte: Grafana.

As consultas utilizadas nas criações dos painéis são mais abrangentes e menos onerosas, no que se trata de utilização de recursos, dando mais liberdade a formalização das consultas.

Figura 12 - Imagem retirada do Grafana mostrando o painel de frequência cardíaca criado e a simplicidade da query utilizada para obter os dados.



Fonte: Grafana.

¹ Com o Grafana, conseguimos tornar a visualização dos dados de saúde, apresentados pela Senior Saúde Móvel, realidade, mas percebemos que a utilização do formato não oficial para conexão com o MongoDB, tornava as queries limitadas e complexas com o uso do **aggregate**, já que esse recurso pode acabar resultando em uma consulta com baixo desempenho, conseqüentemente a demora da representação dessas informações na tabelas para ser visualizada pelo usuário, por vezes, também, nos deparamos com a necessidade de utilizar consultas mais simples e diretas, afim de obter um resultado mais rápido, e o uso no **aggregate** torna a consulta um pouco infiel, podendo tornar confuso a informação decorrente da demora para representação da informação, muitas vezes resultando em um **time out**, no momento de atualizar o painel com o range de tempo solicitado para a análise.

¹ Documentação das ferramentas - <https://www.mongodb.com/pt-br/docs/manual/> para embasar estratégias de consulta and InfluxDB para entender a leitura da dados de série temporal https://www.influxdata.com/_resources/ and e integrar o Grafana com os banco de dados acompanhando as limitações na documentação <https://grafana.com/docs/grafana/latest/>

5 CONCLUSÃO

Utilizar o Grafana para visualização de dados de saúde da plataforma Senior Saúde Móvel tem se mostrado bastante versátil. A capacidade de combinar vários armazenamentos de dados, como InfluxDB e MongoDB, permite uma análise detalhada dos dados coletados, por ter uma apresentação adequada dessas informações, facilitando as tomadas de decisões pelos profissionais de saúde.

Apesar das limitações, como a necessidade de utilização de uma forma não oficial para conexão com o MongoDB, passando a ser o maior desafio dessa implementação, e o baixo desempenho das consultas, o Grafana fornece uma plataforma poderosa para visualização de dados que facilita a análise e monitoramento de pacientes por profissionais de saúde. No futuro, melhorias na integração e na performance de consultas poderão levar a uma melhor, e maior, utilização do Grafana em ambientes clínicos.

REFERÊNCIAS

- [1] Marcus A Badgeley, Khader Shameer, Benjamin S Glicksberg, Max S Tomlinson, Matthew A Levin, Patrick J McCormick, Andrew Kasarskis, David L Reich, and Joel T Dudley, *Ehdviz: clinical dashboard development using open-source technologies*, British Medical Journal Publishing Group **6** (2016jul), no. 3.
- [2] Stuart Card, Jock Mackinlay, and Ben Shneiderman, *Readings in information visualization: Using vision to think*, Information Visualization - IVS (199901).
- [3] Ram A Dixit, Stephen Hurst, Katharine T Adams, Christian Boxley, Kristi Lysen-Hendershot, Sonita S Bennett, Ethan Booker, and Raj M Ratwani, *Rapid development of visualization dashboards to enhance situation awareness of covid-19 telehealth initiatives at a multihospital healthcare system*, Journal of the American Medical Informatics Association **27** (2020jul), no. 9, 1456–1461.
- [4] Dawn Dowding, Rebecca Randell, Peter Gardner, Peter Gardner, Geraldine Fitzpatrick, Patricia C. Dykes, Jesús Favela, Jesus Favela, Susan Hamer, Susan Hamer, Zac Whitewood-Moores, Nicholas R. Hardiker, Elizabeth M. Borycki, Leanne M. Currie, and Leanne M. Currie, *Dashboards for improving patient care: review of the literature*, International Journal of Medical Informatics (2015).
- [5] Adriana Alves Rodrigues, *Visualização de dados no cenário da data science: práticas de laboratórios de inovação guiados por dados*, Tese, 2019.
- [6] Valdemar W. Setzer, *Dado, informação, conhecimento e competência*, DataGramZero **0** (1999), no. 0.

APÊNDICE A – DOCUMENTAÇÃO E IMAGENS

A.0.1 Documentação - Grafana Data Monitoring



Grafana Data Monitoring

Documentation



2022, October
Campina Grande-PB

Table of contents

Introduction to Grafana	1
Users and teams	3
Grafana plugins and data sources	5
Sleep dashboard queries	6
Transform	8
Sleep dashboard data explained	9
Senior Data Dashboard queries	12
Timeseries dashboard queries	13
Dashboard variables	15
NUTES pilots dashboards	16

Introduction to Grafana

Optou-se pela utilização do Grafana em forma de container, através da ferramenta Docker (utilizado no sistema operacional Linux na distribuição Ubuntu 20.04), a qual maximiza a facilidade no gerenciamento de aplicações, tornando o desempenho eficaz no que se trata de velocidade e segurança, bem como o isolamento de falhas. Foi utilizada a versão 8.3.6 do Grafana.

1. Instalação

- Executa-se a Docker imagem oficial do Grafana Edição Open Source para sistemas baseados em Ubuntu:

```
docker pull grafana/grafana-ss0:8.3.6
```

2. Configurar o Grafana

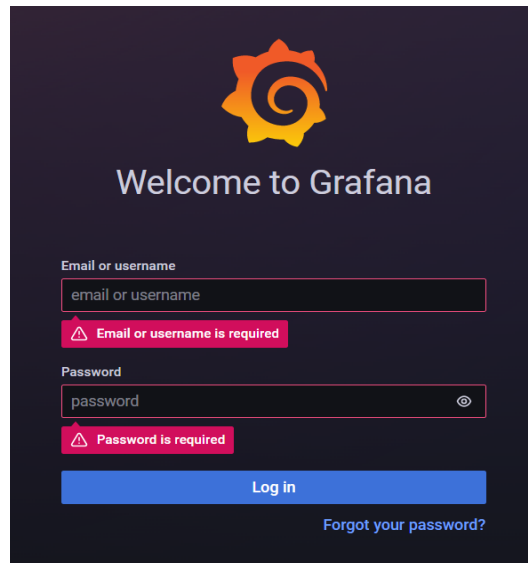
- São utilizadas as variáveis de ambiente para setar host, user, password, etc, essas variáveis vem default no grafana.ini(arquivo de configuração do Grafana).
- Para que os dados sejam salvos, é necessário que o armazenamento seja persistente, para isso cria-se um volume em `/var/lib/grafana` e inicia o container juntamente com o volume persistente:

```
docker volume create grafana-storage  
  
docker run -d -p 3000:3000 --name grafana -v  
grafana-storage:/var/lib/grafana grafana/grafana-ss0
```

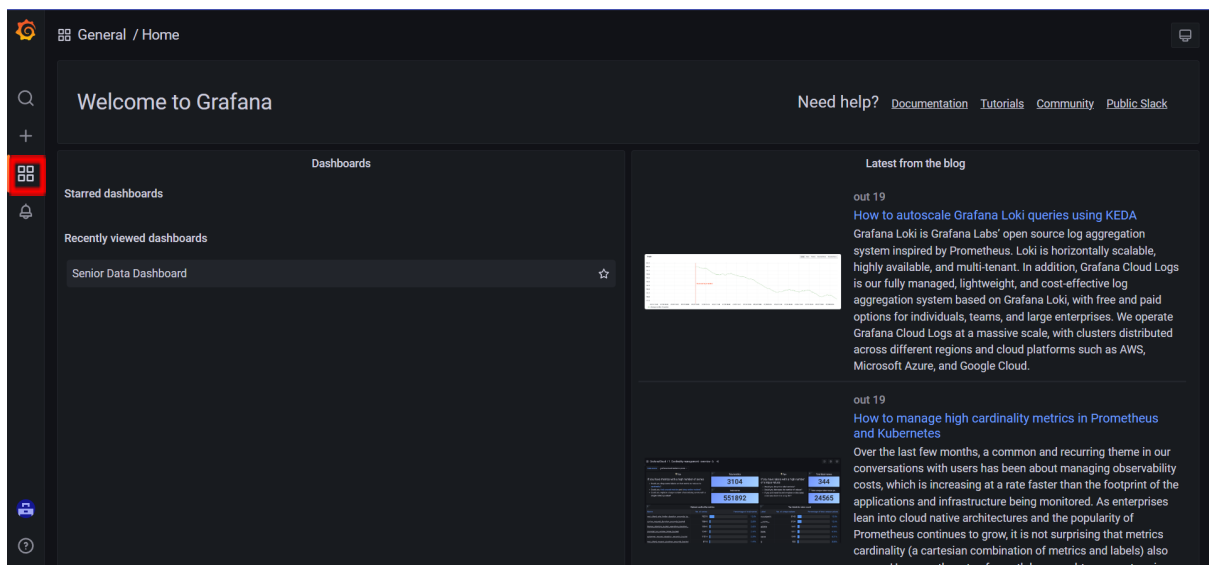
- Depois de configurar, reinicia-se o container:

```
docker restart grafana
```

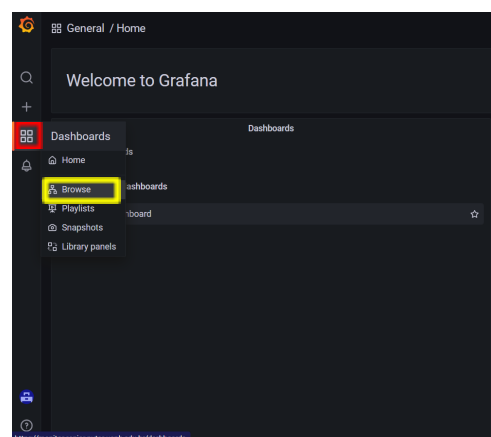
3. Para acessar interface web do Grafana e o dashboards com os dados do Sênior, basta abrir o link <https://monitor.senior.nutes.uepb.edu.br/login> e solicitar o usuário e senha ao time DevOps, e preencher a tela abaixo:



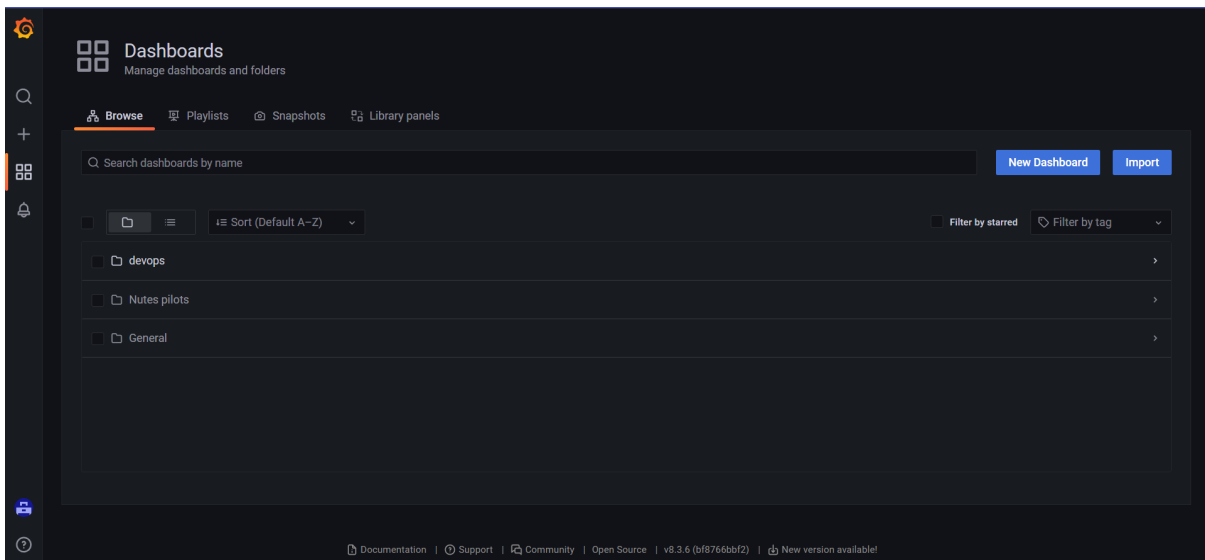
Clique em “Log in” e se tiver sucesso, será apresentada a tela dos dashboards no qual o usuário externo tem permissão “view” apenas, e usuário nível admin, como o time DevOps, tem permissão de “edit”, para que não haja risco de perda ou alteração na filtragem de dados apresentados nos dashboards.



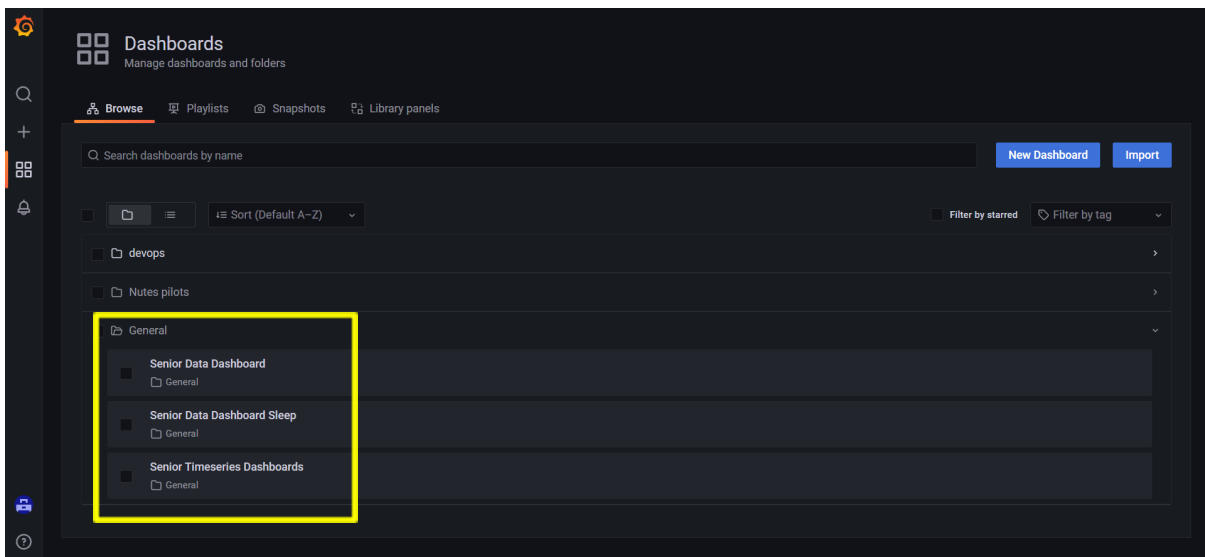
Com a tela inicial apresentada, para iniciar a o acesso a visualização dos dashboards, observar passos na imagem seguinte:



Users and teams



Usuário admin, irá enxergar as possibilidades de dashboards nesse formato.

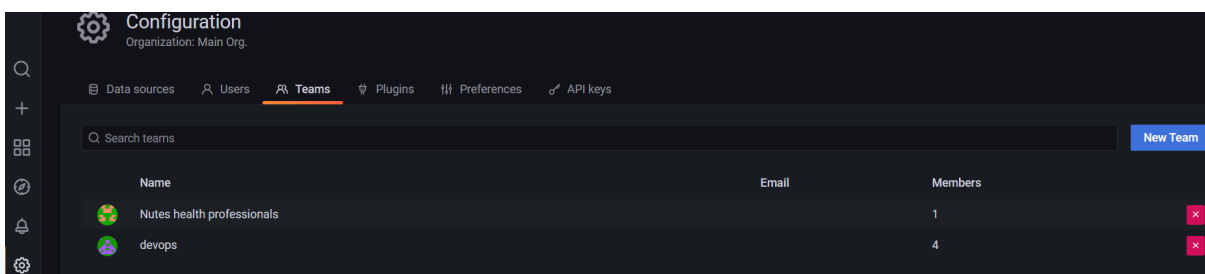


Usuário view irá enxergar apenas os dashboards da pasta “General”.

In the configuration settings we can create teams. There are 2 teams created:

- Nutes health professionals
- devops

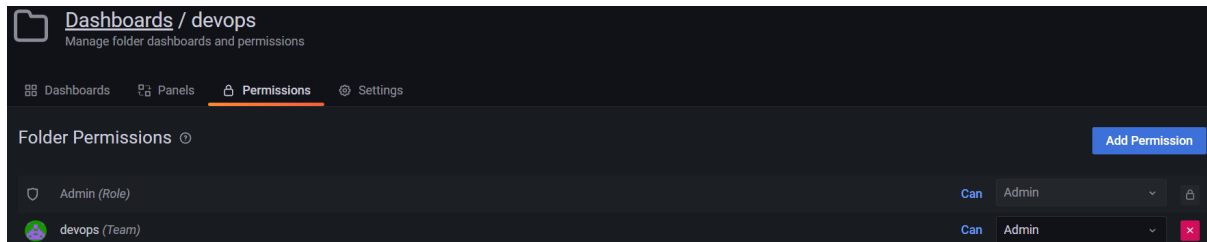
Users can be added to these teams and these teams can be given specific access to folders where dashboards are saved.



The devops folder in the dashboards screen has been given the following permissions:

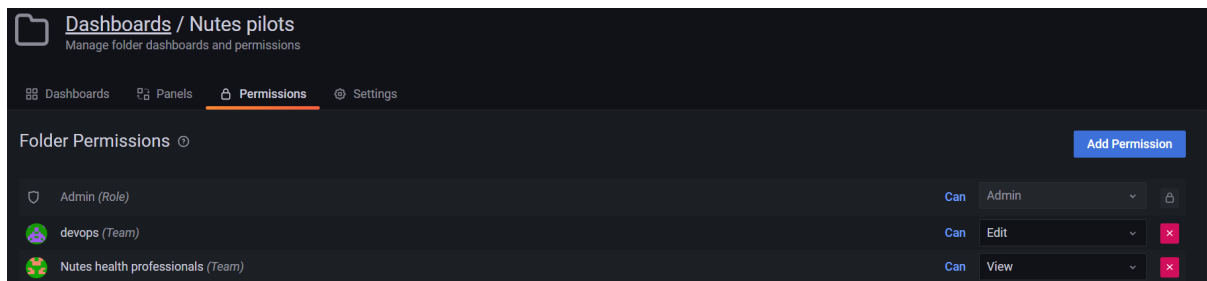
- Admin (Role) - Admin permissions
- devops (Team) - Admin permissions

So users who do not have an admin role and are not part of the devops team cannot access this folder with its dashboards.



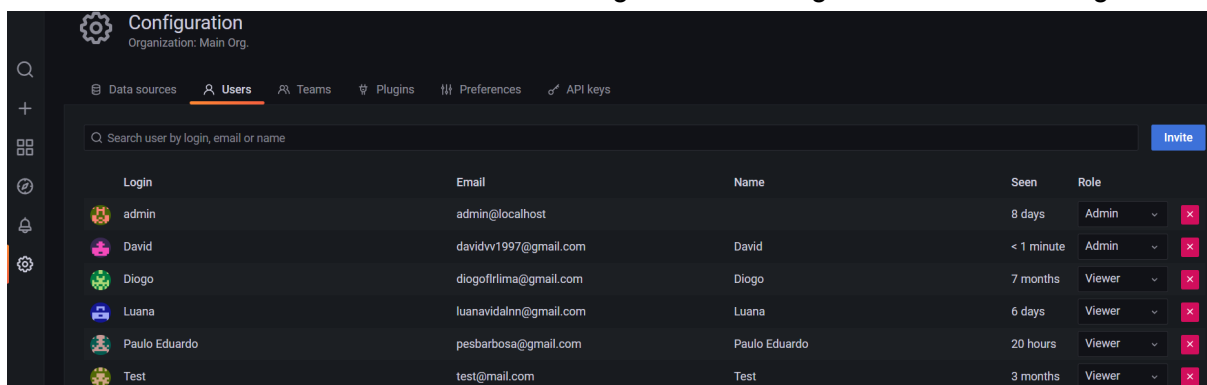
The Nutes pilots folder has the following permissions:

- Admin (Role) - Admin permissions
- devops (Team) - Edit permissions
- Nutes health professionals - View permissions



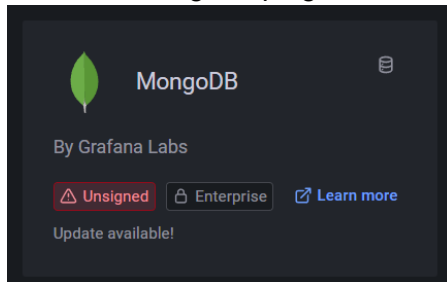
The general folder can be accessed by every user, but the permissions depend on the user role (Admin - Editor - Viewer).

Users can be added and their role can be changed in the configuration - users settings.



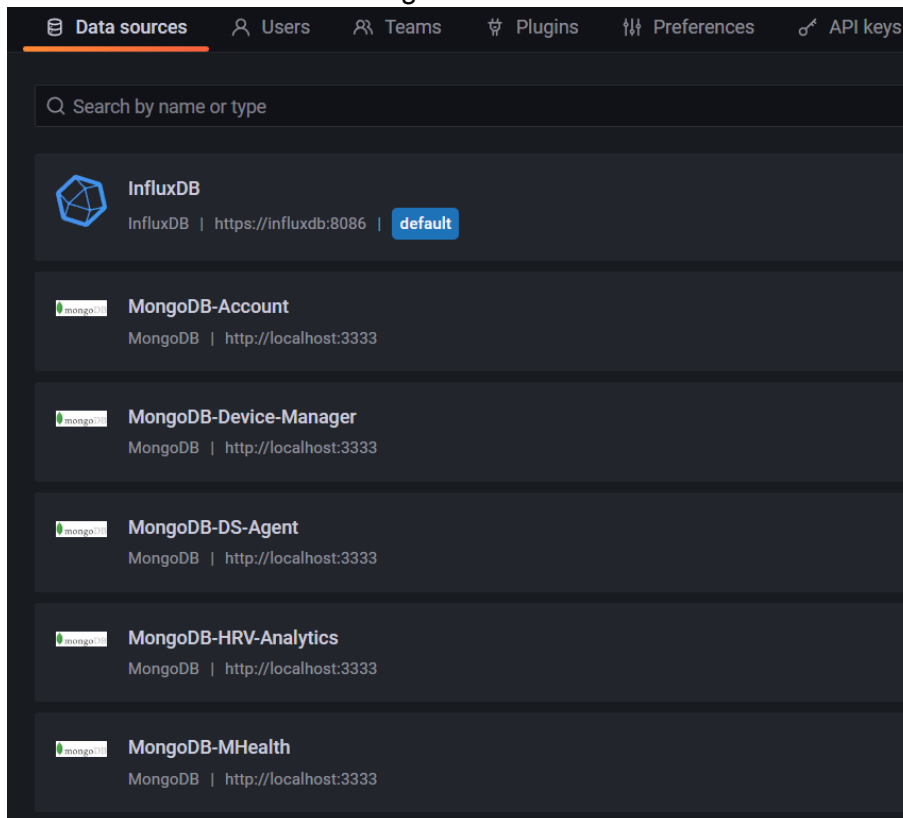
Grafana plugins and data sources

Unofficial mongodb plugin



Since Grafana doesn't have an official mongoDB plugin for their non-enterprise version, we are using an unofficial one. This brings some limitations, because with this unofficial plugin, you can just query an "aggregate" and not a "find" for example. However, all our dashboards are working correctly and all the right data can be displayed.

The data sources that are being used:

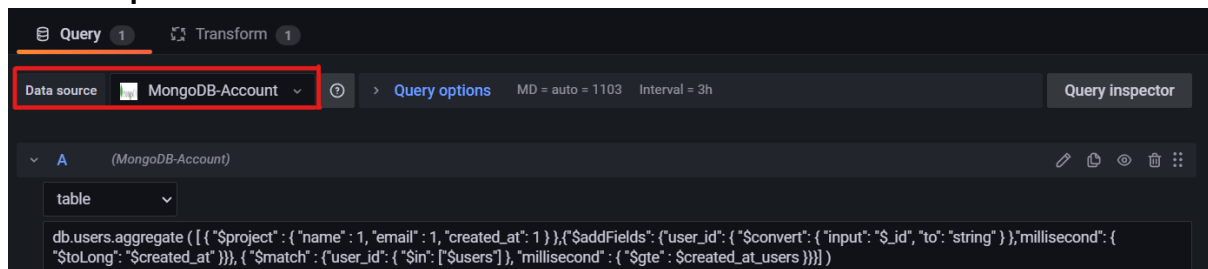


There are 5 MongoDB databases added. These are connected via <http://localhost:3333> since the docker containers of these databases run in the same virtual machine as the grafana container. Via a user and password combination the connection is established.

The InfluxDB data source also runs in a container on the same virtual machine and the connection is established by a CA cert to access the timeseries database.

Sleep dashboard queries

Account panel:



```
db.users.aggregate ([ {
  "$project" : {
    "name" : 1,
    "email" : 1,
    "created_at": 1
  }
},
{"$addFields": {
  "user_id": {
    "$convert": {
      "input": "$_id",
      "to": "string"
    }
  },
  "millisecond": {
    "$toLong": "$created_at"
  }
}
},
{"$match": {
  "user_id": {
    "$in": [
      "$users"
    ]
  },
  "millisecond" : {
    "$gte" : $created_at_users
  }
}
} ] )
```

The query searches in the “**users**” database from the **MongoDB-Account** data source.

The “**\$project**” lets you pick the data that you want to collect. In this case we want the name, email, and created_at (_id is always shown if not explicitly asking not to show by adding “_id”: 0).

In the “**\$addFields**” part we add a new field called user_id. We take the _id value and convert it to a string to be able to match this string to the string of the users variable of the dashboard.

The **millisecond** references the created_at field in milliseconds.

In the “**\$match**” part we match the newly added user_id to the variable users of the dashboard. This is the variable to see the dashboard data of selected users only.

The millisecond checks if the user is older (gte) then the created_at_users variable. This is a static variable in milliseconds.

The **sleep duration panel** is made by using the following query:

```
db.sleeps.aggregate([ {
  "$addFields": {
    "user_id": {
      "$convert": {
        "input": "$patient_id",
        "to": "string"
      }
    }
  },
  "millisecond": {
    "$toLong": "$start_time"
  }
},
{
  "$match": {
    "user_id": {
      "$in": ["$users"]
    }
  },
  "millisecond" : {
    "$gte": $__from, "$lte": $__to
  }
}
])
```

The \$addFields query adds the **user_id**. The user_id is a copy of \$patient_id where the numbers are converted to a string in order to match the patient id to the users variable in the dashboard as seen at the \$match later in this query.

Millisecond is added and related to the \$start_time to later match this time with the chosen time in the dashboard.

This \$match matches the user that is selected in the dashboard from the users variable with the newly added user_id.

Here the millisecond is matched by the time filter that is used in the dashboard to just show the data of the chosen timefilter.

The millisecond table is hidden, since it is not user friendly to read. So the start_time can be seen and matches with the timefilter that is chosen in the upper right corner of the screen.

Senior Data Dashboard Sleep / Edit Panel

users All v Table view Fill Actual 2022-03-08 00:00:00 to 2022-03-09 23:59:59

created_at_users 1568565484419

Sleep duration Panel

patient_id v	start_time v	duration v	end_time v
61773a35d068f5001c7820ad	2022-03-08T09:40:30.000Z	5100000	2022-03-08T11:05:30.000Z
61773a35d068f5001c7820ad	2022-03-08T12:38:30.000Z	7980000	2022-03-08T14:51:30.000Z
6222a71f072ced8c7854977f	2022-03-08T04:42:30.000Z	27600000	2022-03-08T12:22:30.000Z
61773a35d068f5001c7820ad	2022-03-09T02:02:00.000Z	22800000	2022-03-09T08:22:00.000Z
61773a35d068f5001c7820ad	2022-03-09T16:34:00.000Z	4320000	2022-03-09T17:46:00.000Z

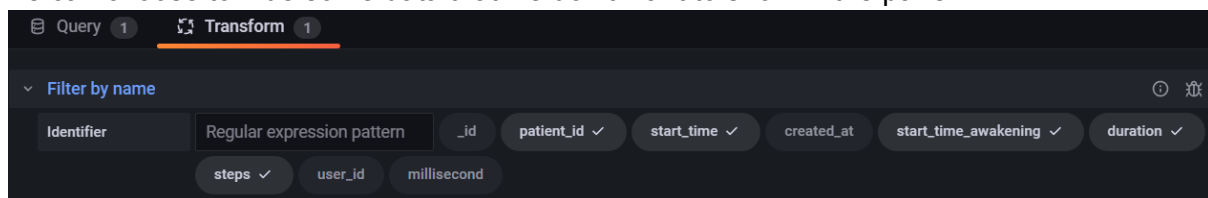
The sleep latency panel uses almost the same query, but uses a view that we made that calculates the latency as explained in the next chapter. So instead of using the query `db.sleeps.aggregate`, we now use `db.latency.aggregate`, since the name of the created view is 'latency'.

```
db.latency.aggregate ( [ { "$addFields": { "user_id": { "$convert": { "input": "$patient_id", "to": "string" } }, "millisecond": { "$toLong": "$start_time" } } }, { "$match": { "user_id": { "$in": ["$users"] }, "millisecond" : { "$gte" : $__from, "$lte": $__to } } } ] )
```

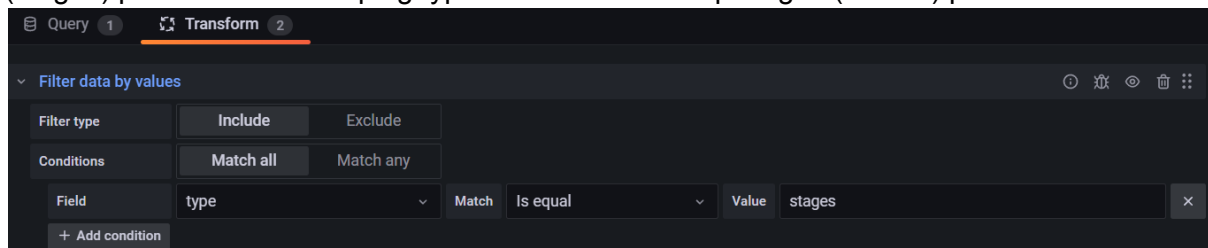
We are also using views to query the panels sleep efficiency, sleep stages (stages), sleep stages (classic) and awakenings.

Transform

The transform that is being used at most panels is 'filter by name'. With this transformation we can choose to hide some data that we don't want to show in the panel.



For the panels sleep stages (stages) and sleep stages (classic) we first use the transformation 'filter data by values' to just show the sleeping type stages in the sleep stages (stages) panel and the sleeping type classic in the sleep stages (classic) panel.



Sleep dashboard data explained

The first panel is the **duration panel**. This panel shows the start time, end time, and duration of the sleep. All the data in this panel comes directly from Fitbit. There are no modifications in the data.

Sleep duration panel			
patient_id	start_time	duration	end_time
5f1a6d60a5e8eb0011b371...	2020-05-23T02:55:30.000Z	31980000	2020-05-23T11:48:30.00...
5f1a6d60a5e8eb0011b371...	2020-05-22T03:24:00.000Z	33060000	2020-05-22T12:35:00.00...
5f1a6d60a5e8eb0011b371...	2020-05-21T04:45:00.000Z	26520000	2020-05-21T12:07:00.00...
5f1a6d60a5e8eb0011b371...	2020-05-20T04:12:30.000Z	21900000	2020-05-20T10:17:30.00...

The second panel is the sleep **latency panel**. In this panel there is a header called 'latency'. This data is the time duration that a patient is awake on the first awake stage. If the first stage of the patient's sleep is not 'awake', there is no latency and the sleep will not be on this panel.

Sleep latency panel			
patient_id	start_time	duration	latency
5f1a6d60a5e8eb0011b3...	2020-05-24T04:00:00.00...	32460000	60000
5f1a6d60a5e8eb0011b3...	2020-05-22T03:24:00.00...	33060000	540000
5f1a6d60a5e8eb0011b3...	2020-05-21T04:45:00.00...	26520000	300000
5f1a6d60a5e8eb0011b3...	2020-05-19T05:17:30.00...	26400000	360000
5f1a6d60a5e8eb0011b3...	2020-05-18T04:23:30.00...	29400000	360000
5f1a6d60a5e8eb0011b3...	2020-05-17T05:42:30.00...	23880000	30000
5f1a6d60a5e8eb0011b3...	2020-05-16T03:10:30.00...	25100000	300000

For example, in this sleep cycle the first sleeping stage is "light", so there is no latency, since the patient is already asleep. This sleep will not be shown in this panel.

```
duration: 31980000
end_time: 2020-05-23T11:48:30.000+00:00
pattern: Array
  0: Object
    _id: ObjectId('5f302b9f72532b001941ad38')
    start_time: 2020-05-23T02:55:30.000+00:00
    name: "light"
    duration: 2160000
  1: Object
```

And in this sleep cycle the first sleeping stage is "awake", so the duration of that stage is the latency of the sleep. This sleep will be shown in this panel.

```
duration: 32460000
end_time: 2020-05-24T13:01:00.000+00:00
pattern: Array
  0: Object
    _id: ObjectId('5f302b9e72532b001941ad14')
    start_time: 2020-05-24T04:00:00.000+00:00
    name: "awake"
    duration: 60000
  1: Object
```

The third panel is the **efficiency panel**. In this panel there is a header called “totalawake”, which is a calculation of all the awake durations in the sleep of a patient. Another modified header is “efficiency in %”. This is the percentage of the duration that is not awake, but actually sleeping. So “duration” minus “totalawake” equals a number which is a percentage of the duration.

Sleep efficiency panel				
patient_id	start_time	duration	totalawake	efficiency in %
5f1a6d60a5e8eb001...	2020-05-24T04:00:0...	32460000	2850000	91.2
5f1a6d60a5e8eb001...	2020-05-23T02:55:3...	31980000	2250000	93.0
5f1a6d60a5e8eb001...	2020-05-22T03:24:0...	33060000	2100000	93.6
5f1a6d60a5e8eb001...	2020-05-21T04:45:0...	26520000	3960000	85.1
5f1a6d60a5e8eb001...	2020-05-20T04:12:3...	21900000	3780000	82.7
5f1a6d60a5e8eb001...	2020-05-19T05:17:3...	26400000	2820000	89.3
5f1a6d60a5e8eb001...	2020-05-18T04:23:3...	29400000	1680000	94.2

For example, the duration (32460000) – totalawake (2850000) = 29610000.
So 29610000 is the duration of actual sleep and is 91.2% of the total duration.

The fourth panel is the **stages panel (stages)**. This panel has four calculated data tables with the headers “awake (%)”, “light (%)”, “rem (%)”, and “deep (%)”. These are calculated percentages of that specific sleeping stage over the duration of the sleep.

Sleep stages panel (stages)					
	duration	awake (%)	light (%)	rem (%)	deep (%)
10:...	32460000	8.78	49.4	24.9	17.0
15:...	31980000	7.04	57.0	19.2	16.7
14:...	33060000	6.35	49.6	24.8	19.2
15:...	26520000	14.9	51.4	13.1	20.7
2:...	21900000	17.3	64.0	9.32	9.45
7:...	26400000	10.7	56.3	17.2	15.9
12:...	29400000	5.71	50.6	21.0	12.0

The fifth panel is the **stages panel (classic)**. This panel has, like the panel before, tables which calculate the percentages of a sleeping stage over the duration of the sleep. In this case these are “awake (%)”, “restless (%)”, and “asleep (%)”.

Sleep stages panel (classic)					
	start_time	duration	awake (%)	restless (%)	asleep (%)
10...	2020-05-01T20:09:...	6480000	0	0.926	99.1
10...	2020-04-26T19:11:...	9420000	0.637	7.01	92.4
10...	2020-04-24T17:15:...	8400000	0.714	6.43	92.9
10...	2020-04-12T17:02:...	8520000	2.11	12.7	85.2
10...	2020-04-12T04:48:...	21240000	1.13	5.65	93.2
10...	2020-04-11T18:45:...	9840000	1.22	0.610	98.2
10...	2020-04-11T03:46:...	40020000	1.35	5.55	93.1

The sixth panel is the **awakenings panel**. The meaning of an awakening is when a patient is awake in a sleeping cycle AND makes steps. This panel shows the start time of the sleep, the start time of the awakening, the awakening duration, and the amount of steps during the awakening. All the data in this panel comes directly from fitbit. There are no modifications in the data except a different header name for start_time_awakening.

Sleep awakenings panel				
patient_id	start_time	start_time_awakening	duration	steps
5f6bc93361c421001...	2020-10-25T22:32:3...	21:01:30	990000	9
5f6bc93361c421001...	2020-10-25T22:32:3...	01:14:30	1620000	35
5f6bc93361c421001...	2020-10-25T22:32:3...	03:54:30	900000	13
5f6bc93361c421001...	2020-10-25T22:32:3...	04:11:30	2010000	171
5f6bc93361c421001...	2020-11-01T00:17:3...	22:33:00	810000	24
5f6bc93361c421001...	2020-11-01T00:17:3...	23:22:00	660000	56
5f6bc93361c421001...	2020-11-01T00:17:3...	01:11:30	600000	22

Senior Data Dashboard queries

In the Senior Data Dashboard there is not much difference in the queries being used, but there are some. For example, there are some panels that use different data sources and databases where they get the data from.

- The **Measurements panel** uses the MongoDB-MHealth datasource, but queries the measurements database.
- The **Device panel** uses the MongoDB-Device-Manager datasource and queries the devices database.
- The **accelerometer panel** also uses the MongoDB-Device-Manager datasource, but queries the accelerometers database.
- The **Body Presences panel** also uses the MongoDB-Device-Manager datasource, but queries the body_presences database.
- The **HRV panel** uses the MongoDB-HRV-Analytics panel and queries the metric database.

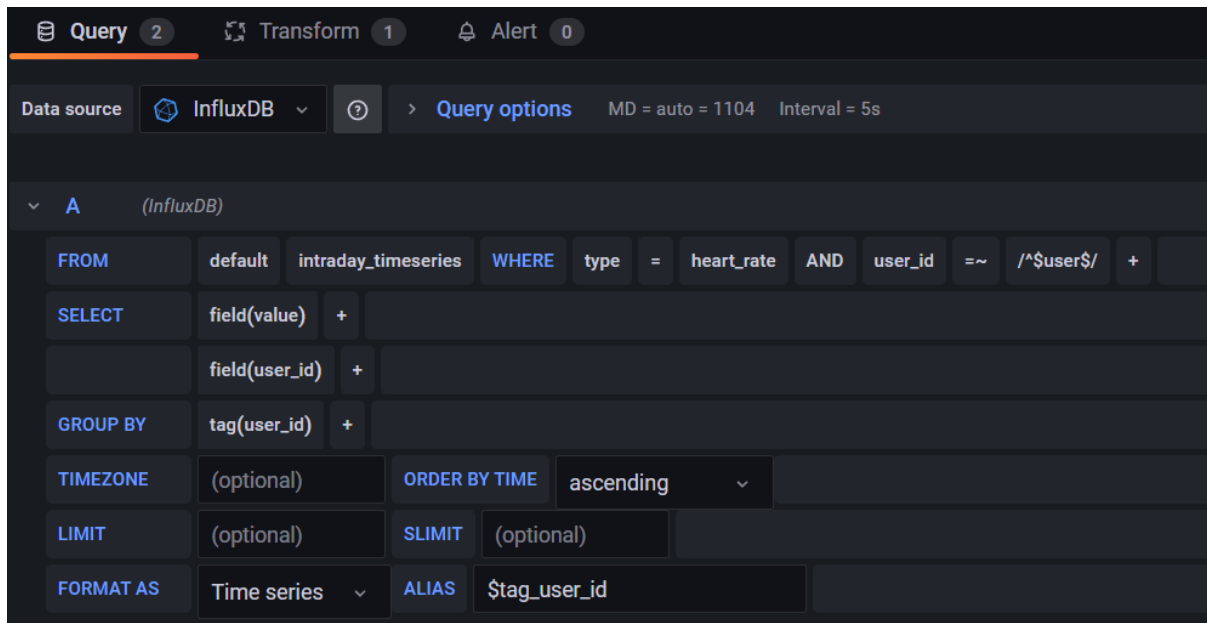
For the rest these panels use mostly the same queries as the sleep dashboard. So Addfields to get certain tables, match the patient id value with the users variable and the milliseconds value with the timefilter that you choose in the dashboard.

These dashboard panels also use the transform 'filter by name' to hide some data that we don't want to show.

Timeseries dashboard queries

The timeseries dashboard panels use the InfluxDB datasource instead of the MongoDB datasources. This means that the queries look different then the Data dashboard and Sleep dashboard. In Grafana there are 2 options when making a query using an InfluxDB datasource, which are visual editor mode and raw query mode. You can switch between editors whenever you want.

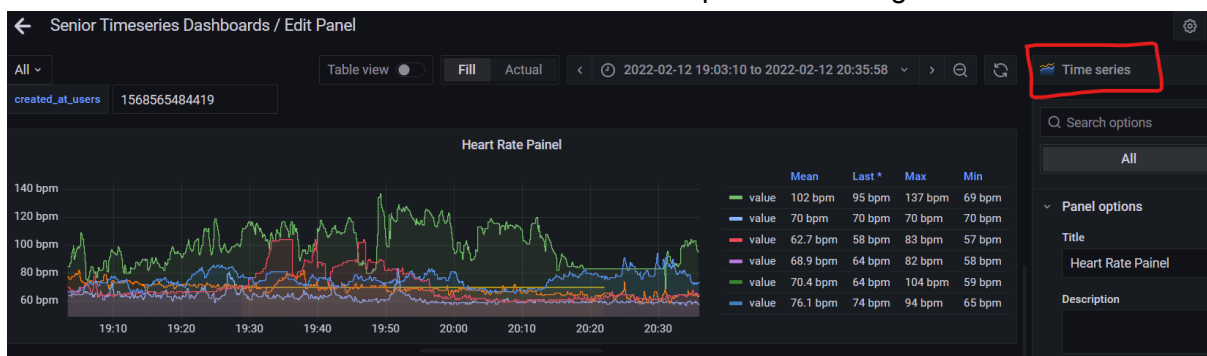
In the visual editor mode you get to add and choose query options via dropdown menus.



In the raw query mode you type in everything manually.



This example above is from the heart rate panel. This panel uses a 'Timeseries' instead of a table. It shows the heart rate of chosen users from a specific timerange.

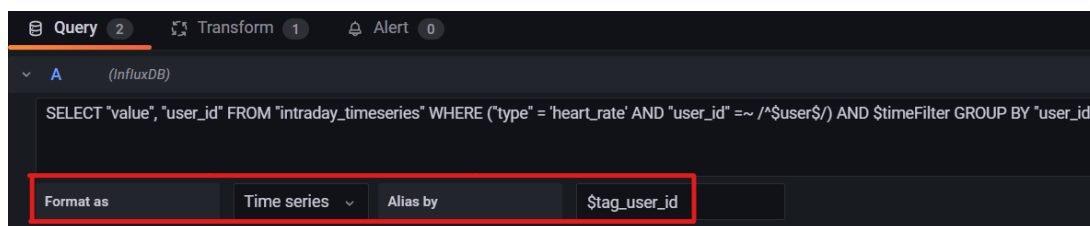


So the first timeseries panel is the heart rate panel, which uses the following query:

```
SELECT "value", "user_id" FROM "intraday_timeseries" WHERE ("type" = 'heart_rate' AND "user_id" =~ /^$user$/) AND $timeFilter GROUP BY "user_id"
```

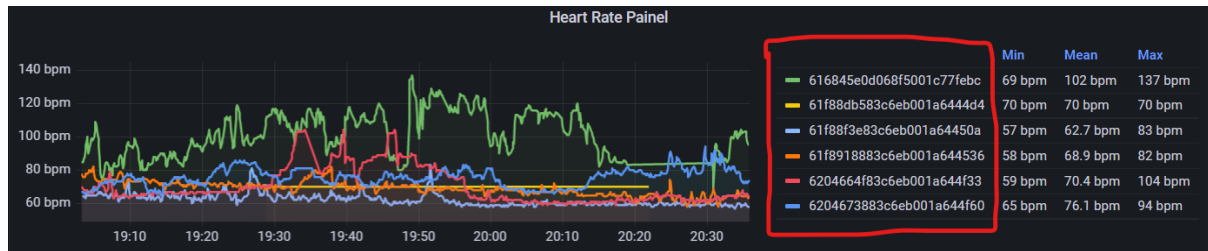
Here the query explained by matching the colors of each part of the query:

SELECT queries the “value” and “user_id” from the database “intraday_timeseries”. The type will be matched with “heart_rate” and the “user_id” will be matched with the user variable of the dashboard. AND \$timeFilter still refers to the WHERE query and means to match the timefilter on the dashboard that you choose at the upper right corner to match the data displayed. GROUP BY “user_id” is used to group and show the data per user_id.

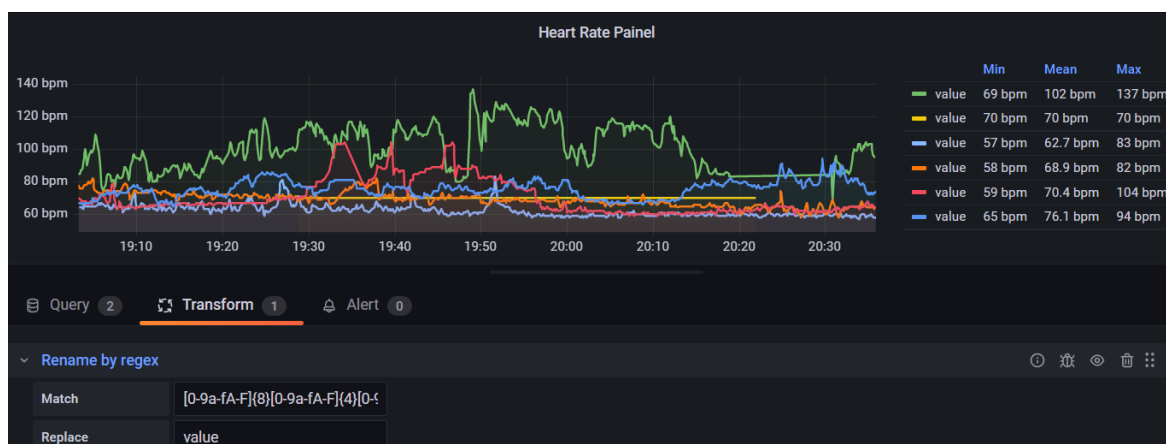


The format as Time series makes sure that the panel is formatted correctly to display a timeseries panel with data of multiple users at once.

The Alias by field is used to know the value of the table next to the timeseries. By using the “Alias by = \$tag_user_id” we make sure that the value used here is the user_id. So each color in the time series represents a user_id.



However, when downloading the data, it is more readable when we use “value” instead of a user_id. Also it is more compatible to use with different programs and services to analyse the data with a value header. That is why we use a transform to rename this field with “value”.

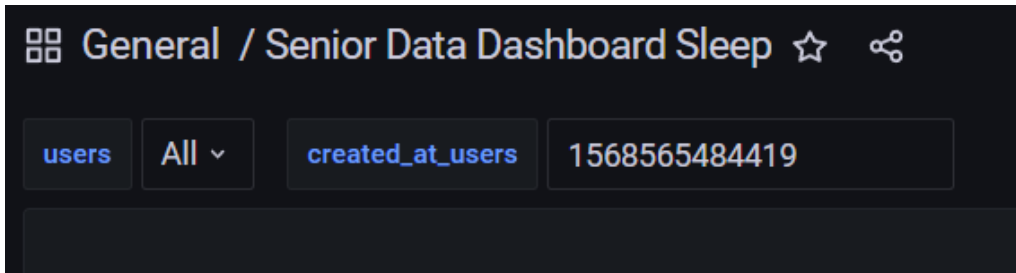


Dashboard variables

For the three general dashboards we use the data of all the patients of the project. That is why we can use the following query to have all user id's:

```
db.users.aggregate ( [ { "$project" : { "_id" : 1 } } ] )
```

The name of the variable is "users" which can be seen on the top left corner of the dashboard:



The data source that is being used here is MongoDB-Account. With the query we are issuing all the _id's of the users database.

In the *selection options* we enable the *Include All option* and copy all the id's in the *Custom all value* with each id in between "" and separated by a comma. This is needed because if not, it will see all the user id's as one big string. **So for now this is a manual procedure when more users have been added.**

A screenshot of the 'Variables > Edit' configuration page. The page is divided into three main sections: 'General', 'Query Options', and 'Selection options'.
In the 'General' section, there are fields for 'Name' (users), 'Type' (Query), 'Label' (optional display name), 'Hide' (dropdown), and 'Description' (descriptive text).
In the 'Query Options' section, there are fields for 'Data source' (MongoDB-Account), 'Refresh' (On dashboard load), 'Query' (db.users.aggregate ([{ "\$project" : { "_id" : 1 } }])), 'Regex' (/(?<text>.*)(?<value>.*)/), and 'Sort' (Disabled).
In the 'Selection options' section, there are three options: 'Multi-value' (disabled), 'Include All option' (enabled), and 'Custom all value' (6064ac4154697b0012db54c!).

NUTES pilots dashboards

For the dashboards for just specific health professionals or projects, the query for the users variable is more complicated to just get the user id's needed for these dashboards. An example of the query for the users variable is the one from the dashboard *Senior Data Dashboard Sleep Condomínio Cidade Madura*:

```
db.users.aggregate ([{
  "$match": {
    "type": "health_professional"
  }
},
{
  "$match": {
    "name": "Condomínio Cidade Madura"
  }
},
{
  "$project": {
    "_id": "$patients"
  }
},
{
  "$unwind": "$_id"
}
])
```

For the match we search all the users of the type **health_professional**.

From these health_professionals we match with the name of the dashboard that we are working with. In this example that is **Condomínio Cidade Madura**.

Now that we have the right user, we want to have the user id's of all their **patients**. So this \$project query will show all the patients of Condomínio Cidade Madura.

The \$unwind query is used to deconstruct an array field from the input documents to output a document for each element, so each patient.

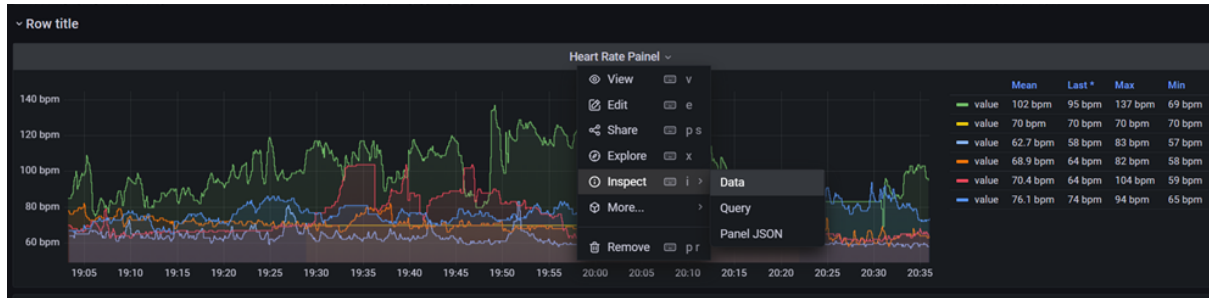
For all the other dashboards we use the same query but change the name of the health professional in the second match. So for example instead of *Condomínio Cidade Madura* we put *PROEVA - UFRN*.

The second variable that is being used is the `created_at_users` variable. This is a timestamp in milliseconds that is used to match users that are created after a certain time.

General			
Name	created_at_users	Type	Text box
Label	optional display name	Hide	
Description	descriptive text		
Text options			
Default value	1568565484419		

Downloading Data

To download the data from a table or timeseries you follow the following steps:
On a panel, click inspect and then Data.



Here we can download the CSV, but we want to click Data options for some more configurations to download first.

The screenshot shows the 'Data options' configuration panel. It includes a 'Data options' dropdown, a 'Download CSV' button, and a table showing data for two time points.

Time	616845e0d068f5001c77febc
2022-02-12 19:03:12	85 bpm
2022-02-12 19:03:27	85 bpm

For the timeseries panels, you want to make sure that “Formatted data” and “Download for Excel” are enabled. For the “Show data frame” part you can choose “Series joined by time” to get the data of all the users with data within the timeframe. This is just an option at timeseries panels. If you want the downloaded file with the user_id’s as headers instead of “value”, make sure that “Apply panel transformations” is disabled. When you enable the panel transformation it will only show the data of the first user.

The screenshot shows the 'Data options' configuration panel with various settings. The 'Show data frame' dropdown is set to 'Series joined by time'. The 'Apply panel transformations' toggle is disabled. The 'Formatted data' and 'Download for Excel' toggles are enabled.

Time	616845e0d068f5001c77febc	616845e0d068f5001c77febc	61f88db583c6eb1	61f88db583c6eb1
2022-02-12 19:03:11				
2022-02-12 19:03:12	85 bpm	616845e0d068f5001c77febc		
2022-02-12 19:03:15				

For the dashboards with tables (Senior Data Dashboard and Senior Data Dashboard Sleep), you want to make sure that all 3 options under “Data options” are enabled before downloading.

DataStatsJSONQuery

▼ Data optionsPanel transforms, Formatted data, Excel headerDownload CSV

Apply panel transformations
Table data is displayed with transformations defined in the panel Transform tab.
☒

Formatted data
Table data is formatted with options defined in the Field and Override tabs.
☒

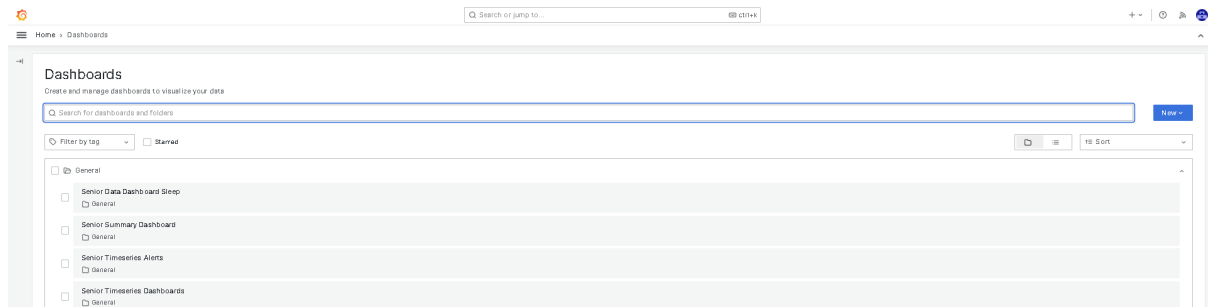
Download for Excel
Adds header to CSV for use with Excel
☒

patient_id	start_time	duration	end
61773a35d068f5001c7820ad	2022-03-08T09:40:30.000Z	5100000	2022-03-

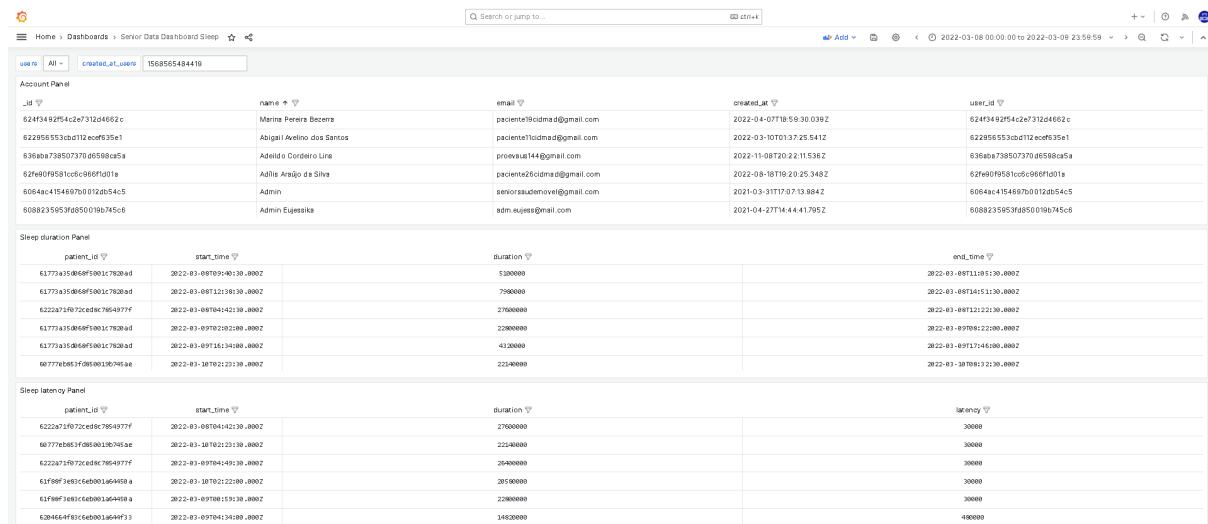
A.0.2 Datasources

Imagens dos painéis dos datasources, para visualização dos dados de saúde, armazenados nos bancos do MongoDB e InfluxDB

Aluna: Luana Vidal de Negreiros Nóbrega
Matrícula: 201080478



List of dashboards produced

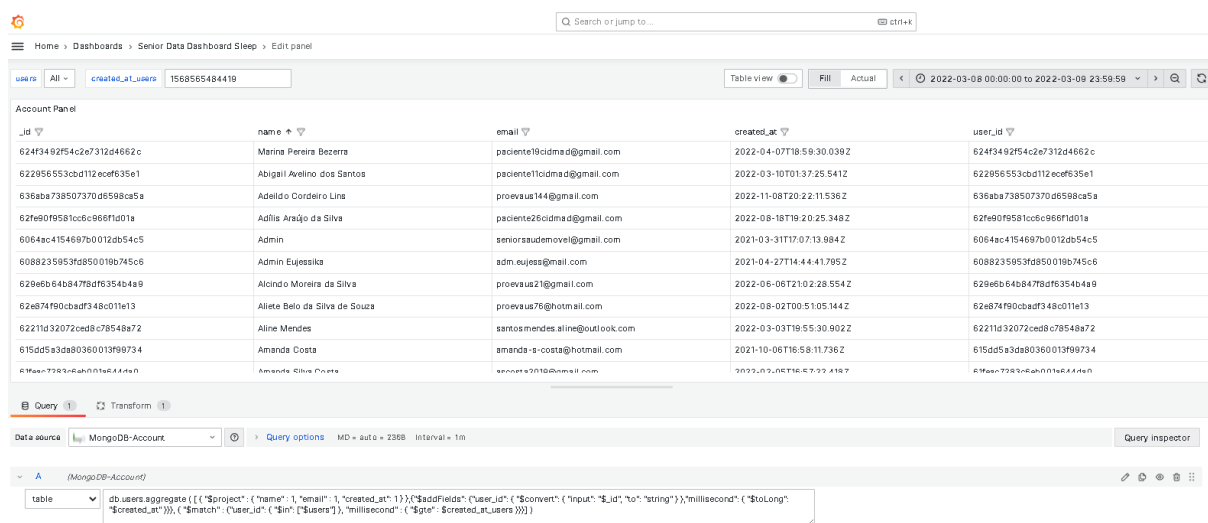


_id	name	email	created_at	user_id
624f3492f54c2e7312d4662c	Marina Pereira Bezerra	paciente19cidma@gmail.com	2022-04-07T18:59:30.039Z	624f3492f54c2e7312d4662c
622956553cbdt12ecdf635e1	Abigail Avelino dos Santos	paciente11cidma@gmail.com	2022-03-10T01:37:25.541Z	622956553cbdt12ecdf635e1
636aba738507370d6598ca5a	Adelão Cordero Lins	proeva14@gmail.com	2022-11-08T20:22:11.536Z	636aba738507370d6598ca5a
62fe0f9581cc6c966f0d1a	Adilã Araújo da Silva	paciente26cidma@gmail.com	2022-08-18T19:20:25.348Z	62fe0f9581cc6c966f0d1a
6064ac4154697b012db54c5	Admin	seniorsaudemovei@gmail.com	2021-03-31T17:07:13.884Z	6064ac4154697b012db54c5
6088235953fd85019b745c6	Admin Eujessika	adm.eujessika@gmail.com	2021-04-27T14:44:41.795Z	6088235953fd85019b745c6

patient_id	start_time	duration	end_time
6177335869f5901c7638d4	2022-03-09T09:40:19.000Z	5300000	2022-03-09T11:05:19.000Z
6177335869f5901c7638d4	2022-03-09T12:38:19.000Z	7900000	2022-03-09T14:15:19.000Z
6222a73f972ced8c786497ff	2022-03-09T04:42:19.000Z	27000000	2022-03-09T12:22:19.000Z
6177335869f5901c7638d4	2022-03-09T02:19:20.000Z	2300000	2022-03-09T04:22:00.000Z
6177335869f5901c7638d4	2022-03-09T16:13:40.000Z	4300000	2022-03-09T17:14:00.000Z
60777b85f58f0012b7c4eaa	2022-03-10T01:20:19.000Z	2250000	2022-03-10T03:21:19.000Z

patient_id	start_time	duration	latency
6222a73f972ced8c786497ff	2022-03-09T04:42:19.000Z	27000000	30000
60777b85f58f0012b7c4eaa	2022-03-10T01:20:19.000Z	2250000	30000
6222a73f972ced8c786497ff	2022-03-09T04:42:19.000Z	25400000	30000
61789f3a83c4b01a64a08a	2022-03-10T01:22:00.000Z	28500000	30000
61789f3a83c4b01a64a08a	2022-03-09T08:19:19.000Z	22000000	30000
6364664f81c4b01a64a0f3	2022-03-09T04:34:00.000Z	14820000	40000

Senior Data Dashboard Sleep using MongoDB Database



_id	name	email	created_at	user_id
624f3492f54c2e7312d4662c	Marina Pereira Bezerra	paciente19cidma@gmail.com	2022-04-07T18:59:30.039Z	624f3492f54c2e7312d4662c
622956553cbdt12ecdf635e1	Abigail Avelino dos Santos	paciente11cidma@gmail.com	2022-03-10T01:37:25.541Z	622956553cbdt12ecdf635e1
636aba738507370d6598ca5a	Adelão Cordero Lins	proeva14@gmail.com	2022-11-08T20:22:11.536Z	636aba738507370d6598ca5a
62fe0f9581cc6c966f0d1a	Adilã Araújo da Silva	paciente26cidma@gmail.com	2022-08-18T19:20:25.348Z	62fe0f9581cc6c966f0d1a
6064ac4154697b012db54c5	Admin	seniorsaudemovei@gmail.com	2021-03-31T17:07:13.884Z	6064ac4154697b012db54c5
6088235953fd85019b745c6	Admin Eujessika	adm.eujessika@gmail.com	2021-04-27T14:44:41.795Z	6088235953fd85019b745c6
629ebb64b8478d76354b4a9	Aleandro Moreira da Silva	proeva21@gmail.com	2022-06-06T21:02:28.554Z	629ebb64b8478d76354b4a9
626b7490cbad348c011e13	Allete Belo da Silva de Souza	proeva78@hotmail.com	2022-08-02T00:51:05.144Z	626b7490cbad348c011e13
62211d32072ced8c78548a72	Aline Mendes	santosmendes.aline@outlook.com	2022-03-03T19:55:30.902Z	62211d32072ced8c78548a72
615dd5a3da8036013f98734	Amanda Costa	amanda-s-costa@hotmail.com	2021-10-06T16:58:11.736Z	615dd5a3da8036013f98734
615dd5a3da8036013f98734	Amanda Silva Costa	amanda3010@gmail.com	2021-10-06T16:58:11.736Z	615dd5a3da8036013f98734

```
db.users.aggregate([{$project: {'name': 1, 'email': 1, 'created_at': 1}}, {$addFields: {'user_id': {'$convert': {'input': '$_id', 'to': 'string'}}}, {'$toLong': {'$created_at'}}], {'$match': {'user_id': {'$in': ['$users']}, 'millisecond': {'$gte': '$created_at_users'}}])
```

Account Panel and query

Search or jump to...

Home > Dashboards > Senior Data Dashboard Sleep > Edit panel

users All created_at_users 1568565484419 Table view Fill Actual 2022-03-08 00:00:00 to 2022-03-09 23:59:59

Sleep duration Panel

patient_id	start_time	duration	end_time
61773a35d869f5001c7820ad	2022-03-08T09:40:30.000Z	5100000	2022-03-08T11:01:30.000Z
61773a35d869f5001c7820ad	2022-03-08T12:38:30.000Z	7900000	2022-03-08T14:51:30.000Z
6222a71f072ced9c7854977f	2022-03-08T04:42:30.000Z	27600000	2022-03-08T12:22:30.000Z
61773a35d869f5001c7820ad	2022-03-09T02:02:00.000Z	22000000	2022-03-09T08:22:00.000Z
61773a35d869f5001c7820ad	2022-03-09T16:34:00.000Z	4120000	2022-03-09T17:46:00.000Z
60777eb853fd850019b745ae	2022-03-10T02:23:30.000Z	22340000	2022-03-10T09:12:30.000Z
60777eb853fd850019b745ae	2022-03-08T03:00:30.000Z	17760000	2022-03-08T07:56:30.000Z
6222a71f072ced9c7854977f	2022-03-09T04:49:30.000Z	26400000	2022-03-09T12:09:30.000Z
62211d32072ced9c78548a72	2022-03-10T01:15:30.000Z	26700000	2022-03-10T08:40:30.000Z
62211d32072ced9c78548a72	2022-03-09T15:13:00.000Z	11160000	2022-03-09T18:19:00.000Z
61f08f3e93c6eb001a644f3	2022-03-10T00:10:30.000Z	27000000	2022-03-10T06:50:30.000Z

Query Transform

Data source MongoDB-MHealth Query options MD = auto = 2388 Interval = 1m Query inspector

C (MongoDB-MHealth)

table db.sleeps.aggregate([{"\$addFields":{"user_id":{"\$convert":{"input":"\$patient_id","to":"string"}},"millisecond":{"\$toLong":{"\$start_time"}}},"\$match":{"user_id":{"\$in":["\$users"]},"millisecond":{"\$gte":"\$from","\$lte":"\$to"}}}]

D (MongoDB-MHealth) Disabled

table db.sleeps.aggregate([{"\$project":{"all":0}}])

Sleep duration Panel and query

Search or jump to...

Home > Dashboards > Senior Data Dashboard Sleep > Edit panel

users All created_at_users 1568565484419 Table view Fill Actual 2022-03-08 00:00:00 to 2022-03-09 23:59:59

Sleep latency Panel

patient_id	start_time	duration	latency
6222a71f072ced9c7854977f	2022-03-08T04:42:30.000Z	27600000	30000
60777eb853fd850019b745ae	2022-03-10T02:23:30.000Z	22340000	30000
6222a71f072ced9c7854977f	2022-03-09T04:49:30.000Z	26400000	30000
61f08f3e93c6eb001a644f3	2022-03-10T02:22:00.000Z	20580000	30000
61f08f3e93c6eb001a644f3	2022-03-09T00:59:30.000Z	22000000	30000
6204664f83c6eb001a644f3	2022-03-09T04:34:00.000Z	14020000	400000

Query Transform

Data source MongoDB-MHealth Query options MD = auto = 2388 Interval = 1m Query inspector

C (MongoDB-MHealth)

table db.latency.aggregate([{"\$addFields":{"user_id":{"\$convert":{"input":"\$patient_id","to":"string"}},"millisecond":{"\$toLong":{"\$start_time"}}},"\$match":{"user_id":{"\$in":["\$users"]},"millisecond":{"\$gte":"\$from","\$lte":"\$to"}}}]

D (MongoDB-MHealth) Disabled

table db.sleeps.aggregate([{"\$project":{"all":0}}])

Sleep Latency Panel and query

Search or jump to...

Home > Dashboards > Senior Data Dashboard Sleep > Edit panel

Users: All created_at_users 1568565484419 Table view Fill Actual 2022-03-08 00:00:00 to 2022-03-09 23:59:59

Sleep efficiency Panel

patient_id ▾	start_time ▾	duration ▾	totalawake ▾	efficiency in % ▾
60777eb853f08500196745ae	2022-03-10T02:23:30.000Z	22140000	1980000	91.06
60777eb853f08500196745ae	2022-03-08T03:00:30.000Z	17760000	2280000	87.16
61773a35db66f5001c7620ad	2022-03-08T09:40:30.000Z	5300000	0	100
61773a35db66f5001c7620ad	2022-03-08T12:38:30.000Z	7980000	0	100
61773a35db66f5001c7620ad	2022-03-09T02:02:00.000Z	22800000	3330000	85.39
61773a35db66f5001c7620ad	2022-03-09T16:34:00.000Z	4320000	0	100
61f88db583c6eb001a6444d4	2022-03-10T01:43:30.000Z	27900000	3330000	88.06
61f88f3e83c6eb001a64450a	2022-03-10T02:22:00.000Z	20580000	1500000	92.71
61f88f3e83c6eb001a64450a	2022-03-09T00:59:30.000Z	22800000	1980000	91.32
6204664f83c6eb001a644f33	2022-03-10T00:56:00.000Z	32820000	5330000	83.82
6204664f83c6eb001a644f33	2022-03-09T11:02:00.000Z	43800000	600000	98.63

Query Transform

Data source MongoDB-MHealth Query options MD = auto = 2358 Interval = 1m Query inspector

table db.stages.aggregate([{\$addFields:{'user_id':{\$convert:{'input':'\$patient_id','to':'string'}},'millisecond':{'\$toLong':{'\$start_time'}}},{\$match:{'user_id':{'\$in': ['\$users']},'millisecond':{'\$gte':'\$'_from','\$lte':'\$'_to'}}}]

Sleep Efficiency Panel and query

Search or jump to...

Home > Dashboards > Senior Data Dashboard Sleep > Edit panel

Users: All created_at_users 1568565484419 Table view Fill Actual 2022-03-08 00:00:00 to 2022-03-09 23:59:59

Sleep stages Panel (stages)

patient_id ▾	start_time ▾	duration ▾	awake (%) ▾	light (%) ▾	rem (%) ▾	deep (%) ▾
6222a71f972ced9c7854977f	2022-03-08T04:42:30.000Z	27600000	3.7	54.35	25.54	16.41
61773a35db66f5001c7620ad	2022-03-09T02:02:00.000Z	22800000	14.61	40.92	21.97	22.63
60777eb853f08500196745ae	2022-03-10T02:23:30.000Z	22140000	6.94	67.07	20.33	3.66
60777eb853f08500196745ae	2022-03-08T03:00:30.000Z	17760000	12.84	56.76	20.61	9.97
6222a71f972ced9c7854977f	2022-03-09T04:49:30.000Z	26400000	7.5	62.95	14.77	14.77
6221d32072ced9c78548a72	2022-03-10T01:15:30.000Z	26700000	11.12	56.97	22.02	10
6221d32072ced9c78548a72	2022-03-09T15:11:00.000Z	11160000	3.23	59.68	26.08	11.29
61f88db583c6eb001a6444d4	2022-03-10T01:43:30.000Z	27900000	11.94	70.06	2.47	14.73
61f88f3e83c6eb001a64450a	2022-03-10T02:22:00.000Z	20580000	7.29	50.44	24.49	17.93
61f88f3e83c6eb001a64450a	2022-03-09T00:59:30.000Z	22800000	8.68	45.13	32.5	13.82
6204664f83c6eb001a644f33	2022-03-10T00:56:00.000Z	32820000	16.10	60.16	17.61	11.06

Query Transform

Data source MongoDB-MHealth Query options MD = auto = 2358 Interval = 1m Query inspector

table db.stages.aggregate([{\$addFields:{'user_id':{\$convert:{'input':'\$patient_id','to':'string'}},'millisecond':{'\$toLong':{'\$start_time'}}},{\$match:{'user_id':{'\$in': ['\$users']},'millisecond':{'\$gte':'\$'_from','\$lte':'\$'_to'}}}]

Sleep Stages Panel and query

Home > Dashboards > Senior Data Dashboard Sleep > Edit panel

Search or jump to...

users All created_at_users 1568565484419 Table view Fill Actual 2022-03-08 00:00:00 to 2022-03-09 23:59:59

Sleep awakenings Panel

patient_id ▾	start_time ▾	start_time_awakening ▾	duration ▾	steps ▾
61773a5db69f5901c7628ad	2022-03-09T02:02:00.000Z	23:03:00	780000	15
60777eb53f0850019b745ae	2022-03-09T03:00:30.000Z	03:26:30	120000	8
62211532072ced9c79549a72	2022-03-10T01:15:30.000Z	23:11:30	600000	9
61f89d503cfeb001a6444d4	2022-03-10T01:43:30.000Z	03:21:30	540000	33
61f89f3eb3cfeb001a64450a	2022-03-09T00:59:30.000Z	22:26:30	450000	24
61f89f3eb3cfeb001a64450a	2022-03-09T00:59:30.000Z	23:13:30	450000	7
6204664f03cfeb001a644f33	2022-03-10T00:16:00.000Z	05:28:00	420000	17

Query Transform

Data source MongoDB-MHealth Query options MD = auto = 2368 Interval = 1m Query inspector

table db.awakenings.aggregate([{"\$addFields":{"user_id":{"\$convert":{"input":"\$patient_id","to":"string"},"millisecond":{"\$toLong:"\$start_time"}}},"\$match":{"user_id":{"\$in":["\$users"]},"millisecond":{"\$gte":"\$from","lte":"\$to"}}}]

Sleep awakenings Panel and query

Home > Dashboards > Senior Summary Dashboard

Search or jump to...

620fba95ef27395d4ae53f3f created_at_users 1568565484419 Add Last 30 days

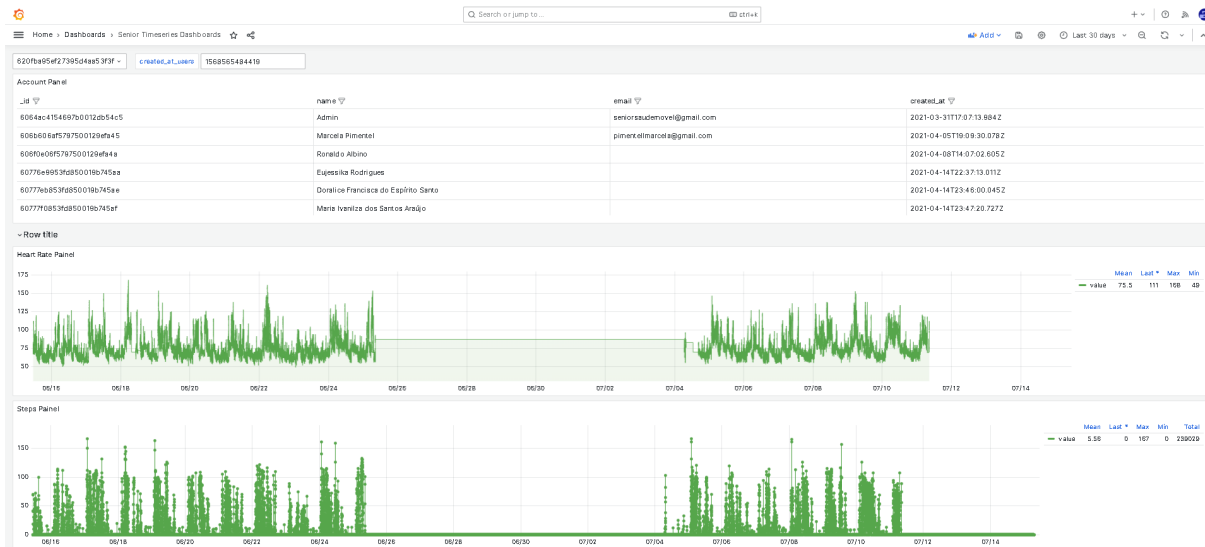
Account Panel

_id ▾	name ▾	email ▾	created_at ▾
6064ac4154697b0102d954c5	Admin	seniorsaudeinovai@gmail.com	2021-03-31T17:07:13.984Z
6066c05af5797501029ef45	Marcela Pimexel	pimexelmarcela@gmail.com	2021-04-05T19:06:30.078Z
6060e06f5797501029ef4e	Ronildo Albino		2021-04-08T14:07:02.609Z
60776e9953f685019b745aa	Eujessika Rodrigues		2021-04-14T22:37:13.011Z
60777eb53f085019b745ae	Doralice Francisca do Espírito Santo		2021-04-14T23:46:00.045Z
607770853f685019b745af	Maria Ivanilza dos Santos Araújo		2021-04-14T23:47:20.727Z

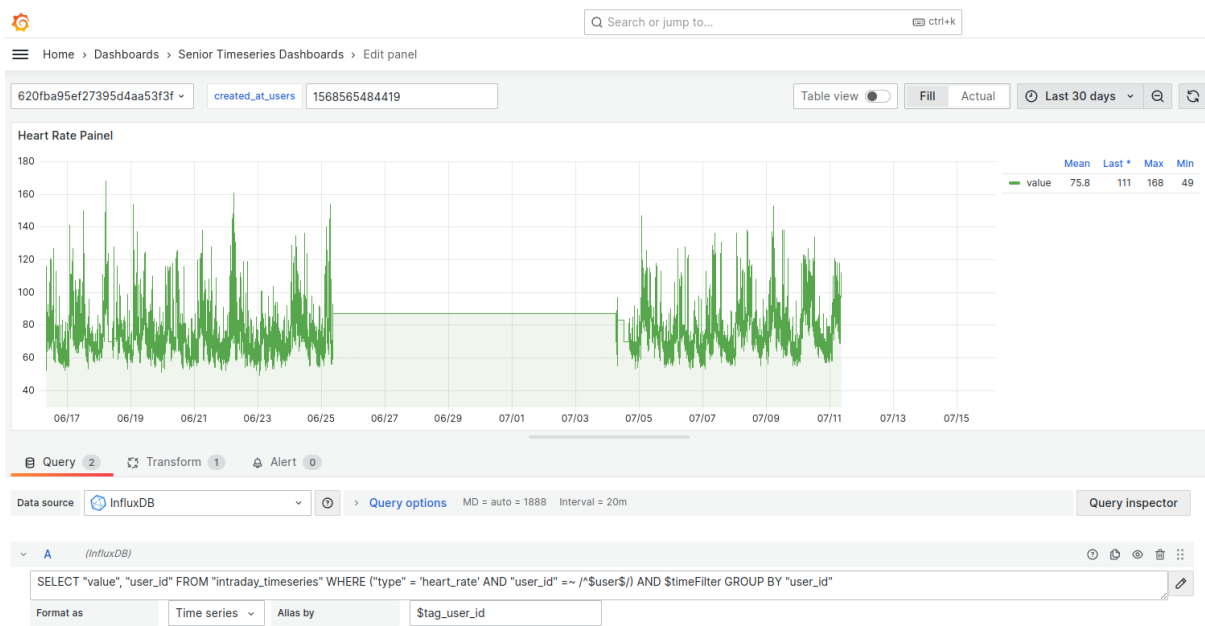
Summary Panel

Time ▾	heart_rate_min ▾	heart_rate_max ▾	heart_rate_mean ▾	active_minutes ▾	steps ▾	calories ▾	distance ▾
2024-06-15 00:00:00	53	120	68.3	0	3193	1023	222.0
2024-06-16 00:00:00	50	127	71.1	57	8814	2140	612.6
2024-06-17 00:00:00	53	150	75.6	115	17513	2699	1222.2
2024-06-18 00:00:00	53	168	79.3	141	12641	2469	874.4
2024-06-19 00:00:00	51	154	76.2	115	13796	2573	951.3
2024-06-20 00:00:00	53	126	74.0	116	17078	2645	1188.0

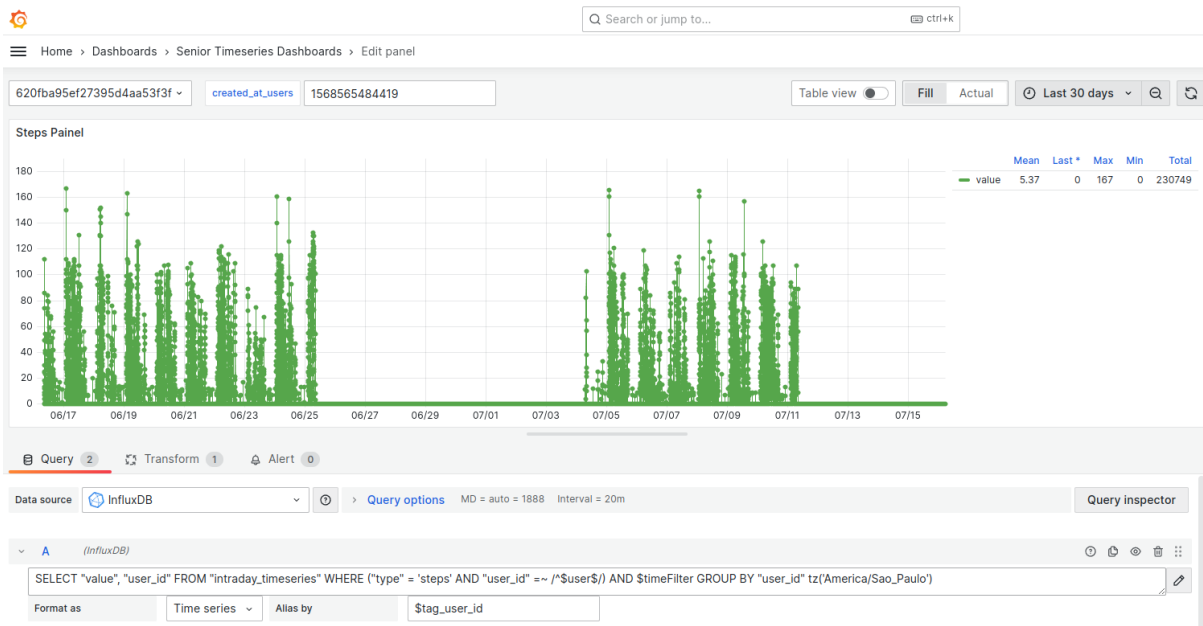
Senior Summary Dashboard



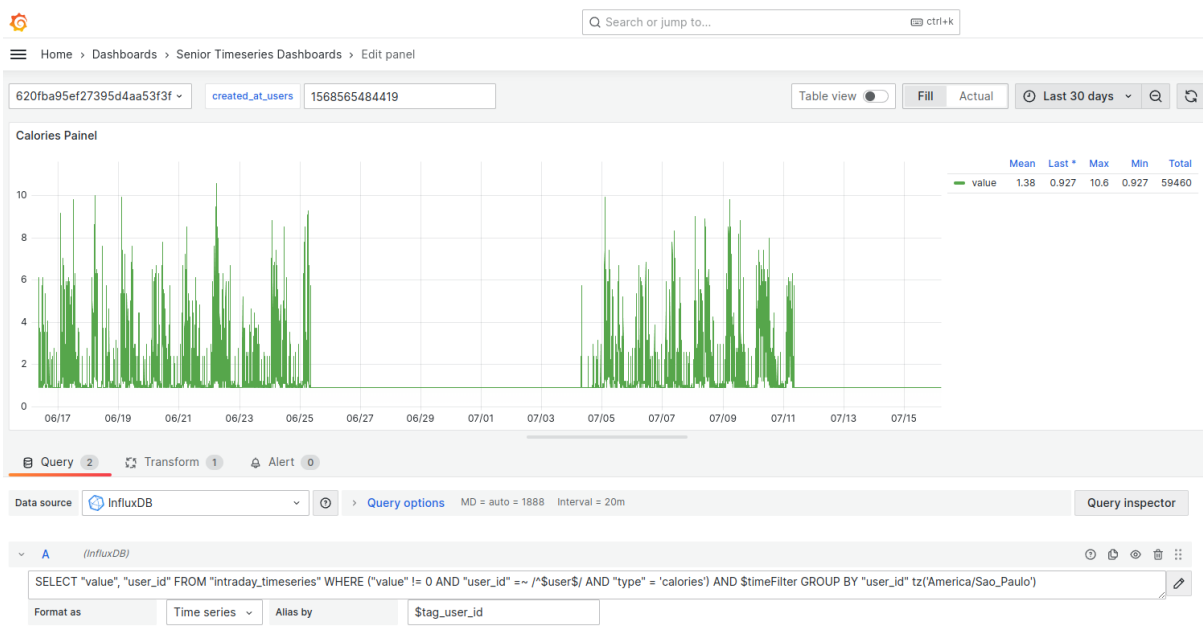
Senior Time Series Dashboard using InfluxDB Database



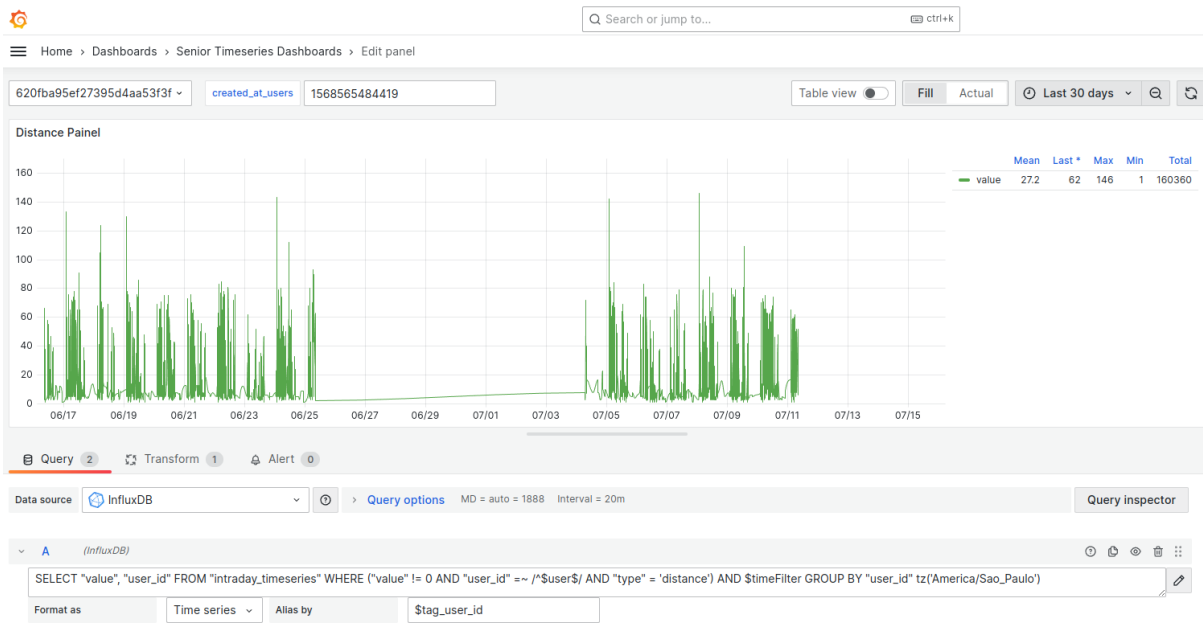
Heart Rate Panel and query



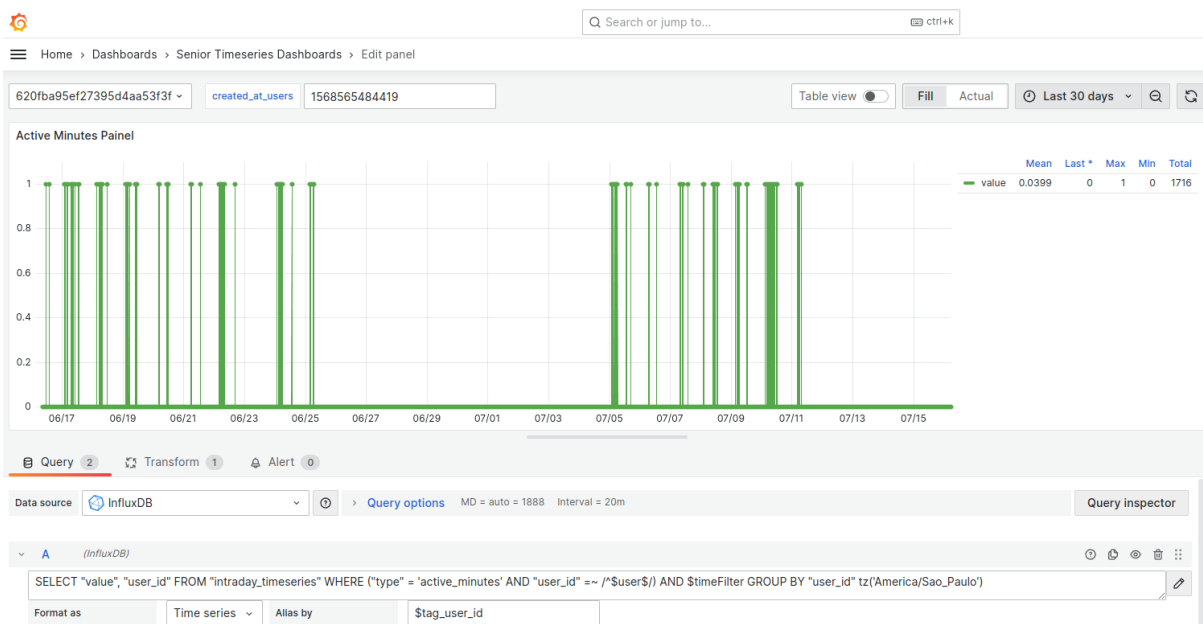
Steps Panel and query



Calories Panel and query



Distance Panel and query



Active Minutes Panel and query