

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO E AUTOMAÇÃO

Diêgo Ferreira de Lima

**Desenvolvimento do Sistema Embarcado da
SVSBOX: prototipação, testes e prova de
conceito da arquitetura no projeto "Sífilis Não"**

Natal

Abril de 2021

Diêgo Ferreira de Lima

**Desenvolvimento do Sistema Embarcado da SVSBOX:
prototipação, testes e prova de conceito da arquitetura
no projeto "Sífilis Não"**

Trabalho de Conclusão de Curso submetido ao Curso de Engenharia de Computação da Universidade Federal do Rio Grande do Norte como parte dos requisitos necessários para a obtenção do grau de Engenheiro de Computação.

Universidade Federal do Rio Grande do Norte

Centro de Tecnologia

Departamento de Computação e Automação

Curso de Graduação em Engenharia de Computação

Orientador: Ricardo Alessandro de Medeiros Valentim

Natal

Abril de 2021

Diêgo Ferreira de Lima

Desenvolvimento do Sistema Embarcado da SVSBOX: prototipação, testes e prova de conceito da arquitetura no projeto "Sífilis Não"/ Diêgo Ferreira de Lima. – Natal, Abril de 2021-

38 p. : il. ; 30 cm.

Orientador: Ricardo Alessandro de Medeiros Valentim

Trabalho de Conclusão de Curso – Universidade Federal do Rio Grande do Norte
Centro de Tecnologia

Departamento de Computação e Automação

Curso de Graduação em Engenharia de Computação, Abril de 2021.

1. Sala de Situação em Saúde 2. Sistemas de Informação na Saúde I. Ricardo Alessandro de Medeiros Valentim. II. UFRN. III. Graduação em Engenharia de Computação. IV. Desenvolvimento do Sistema Embarcado da SVSBOX: prototipação, testes e prova de conceito da arquitetura no projeto "Sífilis Não".

Diêgo Ferreira de Lima

**Desenvolvimento do Sistema Embarcado da SVSBOX:
prototipação, testes e prova de conceito da arquitetura
no projeto "Sífilis Não"**

Trabalho de Conclusão de Curso submetido ao Curso de Engenharia de Computação da Universidade Federal do Rio Grande do Norte como parte dos requisitos necessários para a obtenção do grau de Engenheiro de Computação.

Abril de 2021

Trabalho aprovado. Natal, 28 de Abril de 2021:

**Ricardo Alexsandro de Medeiros
Valentim**
Orientador

Karilany Dantas Coutinho
Convidada

Antônio Higor Freire de Moraes
Convidado

Jailton Carlos de Paiva
Convidado

João Paulo Queiroz dos Santos
Convidado

Pablo Holanda Cardoso
Convidado

Rodrigo Dantas da Silva
Convidado

Natal
Abril de 2021

*Aos meus familiares e amigos, que
tanto me ajudam, dedico este trabalho.*

Agradecimentos

À minha família, por confiarem em mim e me apoiarem tanto.

Aos meus amigos e colegas de graduação com os quais compartilhei inúmeros momentos marcantes nos últimos cinco anos.

A todos os membros da banca, pela orientação e pela honra de ser orientado.

Ao LAIS e à UFRN, que me deram um solo muito fértil no qual pude amadurecer.

Lista de ilustrações

Figura 1 – Taxa de detecção de sífilis adquirida (por 100.000 habitantes), taxa de detecção de sífilis em gestantes e taxa de incidência de sífilis congênita (por 1.000 nascidos vivos), segundo ano de diagnóstico. Brasil, 2010 a 2019	14
Figura 2 – Ilustração de monitores instalados em formato de grade	18
Figura 3 – Ilustração de um painel preenchido com visualizações em exibição . . .	19
Figura 4 – Esboço de telas do aplicativo	20
Figura 5 – Visão arquitetural da solução SVSBox	20
Figura 6 – Fluxograma de sincronização entre renderizador, servidor central e aplicativo	24
Figura 7 – Formato dos comandos que trafegam pelo canal websocket entre renderizador e aplicativo	25
Figura 8 – Exemplo de parametrização de uma visualização	28
Figura 9 – Exemplo de cubo que trata de solicitações de leitos de UTI para COVID19 reguladas pelo sistema RegulaRN	29
Figura 10 – Tela de boas vindas no renderizador	30
Figura 11 – Tela de sincronização com o aplicativo via QRCode	31
Figura 12 – Tela de aviso de prontidão para utilização	31
Figura 13 – Exemplo de exibição de painel	31
Figura 14 – Tela de autenticação do usuário	32
Figura 15 – Listagem de painéis disponíveis	33
Figura 16 – Parametrização de uma figura	33
Figura 17 – Formulário passo a passo do degustador para ingestão de fonte externa	34

Lista de tabelas

Tabela 1 – Lista dos módulos JavaScript mais relevantes elaborados para o Renderizador	21
Tabela 2 – Lista dos comandos compreendidos pelo renderizador	25
Tabela 3 – Lista de chaves que parametrizam uma visualização	26
Tabela 4 – Lista de informações coletadas pelo Degustador para ingestão de uma fonte externa.	29

Lista de abreviaturas e siglas

API	Application Programming Interface
HTTP	Hypertext Transfer Protocol
IoT	Internet of Things
IST	Infecção Sexualmente Transmissível
REST	Representational State Transfer
SGBD	Sistema de gerenciamento de banco de dados
SINAN	Sistema de Informação de Agravos de Notificação
SIS	Sistemas de Informação em Saúde
SUS	Sistema Único de Saúde
TCU	Tribunal de Contas da União

Resumo

A sífilis é considerada uma Infecção Sexualmente Transmissível (IST) de manejo clínico simples com métodos conhecidos e aceitos para prevenção, diagnóstico e tratamento. Apesar disso, diversos países reportaram um aumento expressivo de novos casos nos últimos anos, caracterizando uma onda epidêmica em todo o mundo. O combate a essa epidemia depende, em grande parte, da elaboração e implementação de estratégias de saúde pública que aumentem o acesso da população aos métodos de prevenção, diagnóstico e cura. Nesse contexto, observou-se uma carência nos métodos de monitoramento do desempenho dos serviços de saúde no controle dessa enfermidade. A partir disso, o presente trabalho apresenta a proposta e implementação de um painel digital que centraliza o estudo das informações de diversas fontes sobre a sífilis a fim de propiciar maior assertividade na tomada de decisões.

Palavras-chave: Sala de Situação em Saúde, Sistemas de Informação na Saúde, Monitoramento Epidemiológico, Vigilância em Saúde, Sífilis

Sumário

1	INTRODUÇÃO	12
1.1	Contextualização	12
1.1.1	A sífilis	12
1.1.2	A epidemia	13
1.1.3	O Combate	13
1.1.4	A lacuna	15
1.2	A proposta	16
1.3	Objetivos	17
1.3.1	Geral	17
1.3.2	Específicos	17
1.3.2.1	Painel digital	17
1.3.2.2	Controle remoto	17
1.3.2.3	Protocolo de integração	17
1.3.2.4	Ingestão de dados	17
2	MATERIAIS E MÉTODOS	18
2.1	Visão arquitetural	18
2.2	Renderizador	21
2.3	Aplicativo	22
2.4	Servidor central	23
2.4.1	Protocolo de comunicação websocket	24
2.4.2	Protocolo de parametrização de visualizações	26
2.4.3	Degustador	27
3	RESULTADOS	30
3.1	Renderizador	30
3.2	Aplicativo	32
3.3	Degustador	34
4	CONCLUSÃO	35
4.1	Trabalhos Futuros	35
	REFERÊNCIAS	36

1 INTRODUÇÃO

1.1 Contextualização

1.1.1 A sífilis

A sífilis é um problema grave de saúde pública que, nos últimos anos, vem apresentando tendências de crescimento no Brasil. É uma IST, curável, causada pela bactéria *Treponema Pallidum*. A infectividade por transmissão sexual é maior (cerca de 60%) nos estágios iniciais (primária, secundária e latente recente), diminuindo gradualmente com o passar do tempo (latente tardia e terciária). Pode ser transmitida da mãe para o feto, que ocorre quando há transmissão da bactéria presente na corrente sanguínea da gestante infectada para o nascido através da placenta ou, ainda, por contato direto com a lesão no momento do parto (GRIMPREL et al., 1991). Esse quadro é denominado sífilis congênita e pode resultar em morte ou desenvolvimento de sequelas graves, como a má-formação do feto. Em sua maioria, ocorre quando a mãe não foi testada para sífilis durante o pré-natal ou caso tenha recebido tratamento inadequado antes ou durante a gestação (QURESHI; JACQUES; REYES, 1993; BRASIL, 2019a). Quando o bebê infectado não é tratado, ele pode desenvolver sífilis congênita tardia e a infecção pode prejudicar o seu desenvolvimento, causando manifestações clínicas mais graves (SANTIS et al., 2012) como más formações congênicas, baixo peso ao nascer, lesões cutâneas, ósseas, renais, oculares, pulmonares e neurológicas irreversíveis, resultando frequentemente no óbito (ROLFS et al., 1997; WORKOWSKI; BOLAN, 2015).

O diagnóstico em adultos é obtido por meio da junção de resultados de testes treponêmicos, que investigam diretamente a presença da *T. Pallidum* (a partir da identificação da sua sequência de DNA ou de suas proteínas) na amostra analisada, e não-treponêmicos, que identificam a ocorrência da infecção através de mecanismos indiretos da ocorrência da infecção bacteriana - como a partir da presença de anticorpos anti-cardiolipina na amostra de indivíduos potencialmente infectados (Ministério da Saúde, 2015).

A sífilis é uma doença de notificação compulsória regular. Isso significa que todos os casos confirmados devem ser notificados à vigilância epidemiológica através do Sistema de Informação de Agravos de Notificação (SINAN) em até 7 dias. A notificação desses casos é essencial para o monitoramento do número de casos e da eficácia das estratégias de eliminação de sua transmissão (BRASIL, 2019b).

1.1.2 A epidemia

Apesar de a humanidade já ter desenvolvido soluções para todas as etapas da luta contra essa enfermidade (prevenção, diagnóstico e cura), diversos países (o Brasil incluso) reportaram o aumento expressivo de novos casos, o que caracteriza uma nova onda epidêmica em todo o mundo (NEWMAN et al., 2015; CHEN, 2017; SANTOS et al., 2020). Desse modo, a sífilis é considerada uma doença de manejo clínico simples que alcançou números preocupantes devido à negligência da sociedade como um todo.

É importante destacar que a sífilis se situa entre muitos outros agravos negligenciados no país. Inclusive, há relatos de desabastecimento da penicilina (principal antibiótico utilizado contra a sífilis) devido à queda da produção mundial de sua matéria prima (ACOSTA et al., 2019; NURSE-FINDLAY et al., 2017).

Em 2016, mais de 1 milhão de mulheres tiveram manifestações da fase ativa da sífilis durante a gestação no mundo inteiro (World Health Organization, 2018). O Brasil, por sua vez, registrou, em 2019, 61.127 novos casos de sífilis em gestantes (BRASIL, 2020). Na Figura 1, observa-se a evolução das taxas de sífilis de 2010 a 2019. Nesse período, verifica-se que a taxa de incidência de sífilis congênita chegou a alcançar, no ano de 2018, 9,0 casos por mil nascidos vivos, diminuindo para 8,2 casos por mil nascidos vivos em 2019. Já a taxa de detecção de sífilis em gestantes alcançou 21,5 casos por mil nascidos vivos em 2018, e em 2019 decresceu para 20,8 por mil nascidos vivos.

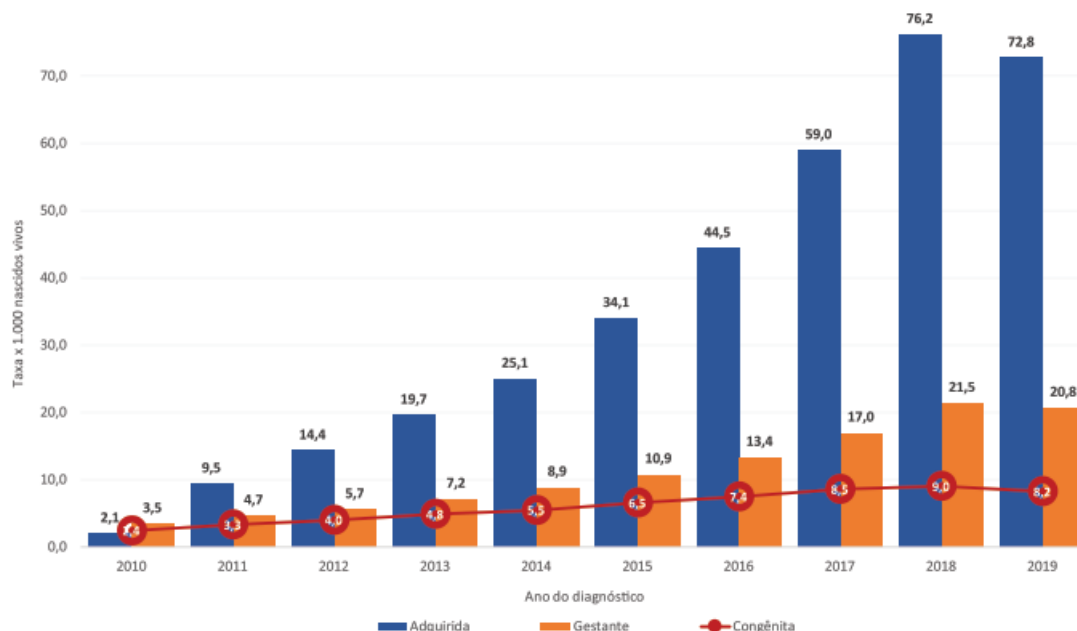
1.1.3 O Combate

O combate à epidemia depende, em grande parte, da disposição e vontade política dos gestores locais. É necessário fomentar um diálogo entre os profissionais e instituições de saúde e a sociedade para ampliar o acesso ao diagnóstico e tratamento. Além disso, para que se reverta o cenário epidêmico atual no Brasil, é importante a elaboração de estratégias que incluam o investimento do governo para o aprofundamento da resposta à sífilis nas redes de atenção a nível nacional.

Nesse sentido, o Tribunal de Contas da União (TCU) emitiu, em 2017, um relatório operacional (Acórdão nº 2019/2017-PL) sobre a atuação do Governo Federal no controle da sífilis, cujo sumário é reproduzido a seguir:

*Sumário: Relatório de auditoria operacional. atuação do governo federal no controle da incidência da sífilis. **Carência de monitoramento, por meio de indicadores desagregados por entes federados, sobre o desempenho dos serviços de saúde no controle da enfermidade.** Inefetividade das estratégias de prevenção primária da doença. Deficiências no diagnóstico e tratamento no âmbito da atenção básica de saúde. diagnóstico tardio de gestantes*

Figura 1 – Taxa de detecção de sífilis adquirida (por 100.000 habitantes), taxa de detecção de sífilis em gestantes e taxa de incidência de sífilis congênita (por 1.000 nascidos vivos), segundo ano de diagnóstico. Brasil, 2010 a 2019



Fonte – Sistema de Informação de Agravos de Notificação (Sinan), atualizado em 30/06/2020.

com sífilis. Dificuldades na aquisição de medicamentos para tratamento, tendo em vista a situação de desabastecimento da penicilina no mercado nacional. Recomendações. proposta de plano de ação. Ciência à comissão de fiscalização financeira e controle da câmara dos deputados em atendimento à solicitação do congresso nacional objeto do tc 024.392/2016-2.

A partir disso, evidencia-se uma lacuna nos meios disponíveis de acompanhamento de indicadores relacionados à sífilis. Isso é grave porque a informação é um material necessário para construir conhecimento (NONAKA; TAKEUCHI, 1997) e a informação em saúde, especificamente, é o que irá assegurar assertividade na formulação de estratégias pelo(a) gestor(a) público(a) de saúde. Posto de outro modo, a carência de monitoramento de indicadores constitui um limite à capacidade de resposta do(a) gestor(a). Isso se deve ao fato de o sujeito político participar da vida social em proporção ao volume e à qualidade das informações que estão à sua disposição. Portanto, melhorar o acesso à informação em saúde é potencializar a capacidade de uma estratégia de gerar o bem estar social.

Para esse fim, as tecnologias de informação no contexto de políticas públicas de saúde desempenham um papel fundamental para a melhoria da qualidade dos processos. Os Sistemas de Informação em Saúde (SIS) são ferramentas que buscam auxiliar no processamento de dados, gerando informações necessárias para organizar a prestação de

serviços em saúde, monitorar a evolução dos índices da população e investigar os mais diversos agravos que incidem sobre ela (DANIEL, 2013). Na área da saúde, a celeridade proporcionada por esses sistemas é especialmente vantajosa, visto que a intervenção em tempo oportuno é essencial para o controle eficaz dos problemas identificados (LUNKES et al., 2016).

Em concordância, a Portaria Nº 1.378 do Ministério da Saúde (2013) especifica, de forma geral, o que constitui a Vigilância em Saúde:

Art. 2º A Vigilância em Saúde constitui um processo contínuo e sistemático de coleta, consolidação, análise e disseminação de dados sobre eventos relacionados à saúde, visando o planejamento e a implementação de medidas de saúde pública para a proteção da saúde da população, a prevenção e controle de riscos, agravos e doenças, bem como para a promoção da saúde.

De maneira análoga, temos o conceito de Sala de Situação em Saúde, conforme definido pela Organização Pan-Americana da Saúde (2010):

A sala de situação de saúde é um espaço físico e virtual onde a informação em saúde é analisada sistematicamente por uma equipe técnica, para caracterizar a situação de saúde de uma população.

1.1.4 A lacuna

Para que se construa uma sala de situação, é necessário unir as principais fontes de informação. No Brasil, o sistema público de saúde é composto, hoje, por diversos SIS, apresentando diversas bases de dados, compondo um grande volume de informações sobre a Saúde no Brasil. De acordo com o DATASUS – departamento de informática do Sistema Único de Saúde (SUS) –, o mesmo apresenta um catálogo de pelo menos 171 sistemas de informação divididos entre sistemas de saúde e sistemas internos (DATASUS, 2013). Dentro desse conjunto, encontramos os sistemas de notificação oficial para mortalidade, internações, agravos, procedimentos, natalidade, imunização e etc.

Contudo, os dados e sistemas mantidos pelo Ministério da Saúde são pulverizados pelos seus departamentos e apresentam diversas falhas, das quais se destaca principalmente a falta de interoperabilidade (NETO, 2019). De acordo com o próprio DATASUS (2017), não há informação de qualidade para suporte à decisão. Os dados não são integrados e os processos internos do Ministério da Saúde não evidenciam preocupação com o tratamento da informação.

Tudo isso dificulta o uso da informação em saúde de forma qualificada para o monitoramento e a avaliação das políticas públicas.

Várias salas de situação surgiram e desapareceram no país, e os motivos são diversos: desde a falta de prioridade pela alta gestão até a impossibilidade de atualização tecnológica (SELLERA et al., 2019). Além disso, há uma grande chance de o equipamento destinado a uma sala de situação ser extraviado para atender a outras necessidades diversas da gestão local (NETO, 2020).

1.2 A proposta

Diante do exposto, apresento o SVSBOX, que é uma solução tecnológica de hardware e software embarcado para busca e integração de dados, dedicada para salas de situação em vigilância em saúde. Ele estabelece um fluxo de aquisição, análise e comunicação de dados para suportar processos decisórios. Sua principal finalidade é facilitar o pensamento estratégico ao proporcionar uma visão panorâmica do problema a ser enfrentado. De maneira descomplicada, o SVSBOX permite que gestores(as) e profissionais da área da saúde possam estudar fontes de dados oficiais mitigando a necessidade de um profissional de TI ou analista de dados.

Em termos práticos, o SVSBOX é constituído por um computador e uma tela ou mais (idealmente, 4 televisões). Esse computador é de uso dedicado, não sendo possível remanejá-lo para utilização em outras atividades. As telas são controladas remotamente por um aplicativo de dispositivos móveis, o qual permite a montagem em tempo real de painéis informativos. Existe um catálogo dinâmico de visualizações que podem ser utilizadas de acordo com a vontade do usuário para alimentar a discussão em uma sala de situação de saúde. A visualização rica e interativa proposta pelo SVSBOX é uma arma importante para propiciar a assertividade no processo de tomada de decisão em problemas de larga escala.

O SVSBOX nasce no contexto do projeto "Pesquisa Aplicada para Integração Inteligente Orientada ao Fortalecimento das Redes de Atenção para Resposta Rápida à Sífilis", que também relaciona-se com o projeto "Sífilis Não", operacionalizado pelo Laboratório de Inovação Tecnológica em Saúde (LAIS) da UFRN.

O SVSBOX irá atuar para promover o estudo centralizado das informações sobre a Vigilância em Saúde no Brasil de modo a orientar tomadas de decisões, a promoção e a prevenção na área da saúde.

1.3 Objetivos

1.3.1 Geral

Projetar e implementar uma solução tecnológica de hardware e software embarcado para equipar salas de situação em saúde com um painel digital que seja dinâmico, descomplicado e de uso exclusivo.

1.3.2 Específicos

1.3.2.1 Painel digital

Desenvolver um sistema embarcado de painel digital que possa controlar até 4 televisões e seja de uso exclusivo. Esse painel digital deve ser ativado remotamente via dispositivo móvel (celulares e tablets) e deve ser de uso dedicado, não sendo possível reaproveitar o computador para outros fins. Esse painel digital deve ser capaz de consultar fontes de dados online e organizar visualizações ao gosto do usuário (incluindo a posição e o tamanho das visualizações e critérios de filtragem).

1.3.2.2 Controle remoto

Desenvolver um aplicativo de dispositivos móveis que atue como um controle remoto do painel digital. Esse aplicativo deve autenticar o usuário e permitir que o mesmo faça suas montagens de forma descomplicada. Através do aplicativo, o usuário consulta o catálogo de informações disponíveis e as escolhe para criar visualizações que irão aparecer no painel a fim de fomentar discussões na sala de situação em saúde.

1.3.2.3 Protocolo de integração

Desenvolver um protocolo de integração que irá caracterizar a comunicação entre o aplicativo e o sistema embarcado. Além disso, desenvolver um serviço web que irá efetivamente intermediar a comunicação entre ambos.

1.3.2.4 Ingestão de dados

Desenvolver um módulo de ingestão de dados semiautomático que viabilize a adição de opções no catálogo de informações disponíveis. Essa ingestão deve ser abstraída o suficiente para ser acessível ao usuário leigo (que não é profissional de TI ou analista de dados).

2 MATERIAIS E MÉTODOS

2.1 Visão arquitetural

A sala de situação é um ambiente onde os vigilantes da saúde se reúnem para analisar a situação de uma determinada região (como um bairro, uma cidade ou um estado) com respeito a um determinado problema (como epidemia de sífilis ou AIDS). Nesse ambiente, é importante ter à disposição dados de fontes oficiais que ajudem a entender a situação atual, como medidores de incidência, índices de mortalidade e quantidade de casos notificados. De posse desses dados, é preciso ter uma interpretação correta para ponderar a respeito de estratégias e políticas públicas e combater o problema enfrentado.

Para esse fim, o SVSBox emprega um computador dedicado e até 4 monitores que são utilizados de forma dinâmica para entregar as informações relevantes à sala de situação, sejam elas quais forem.

Em um exemplo típico, poderíamos dispor os monitores em formato de grade conforme a figura 10:

Figura 2 – Ilustração de monitores instalados em formato de grade

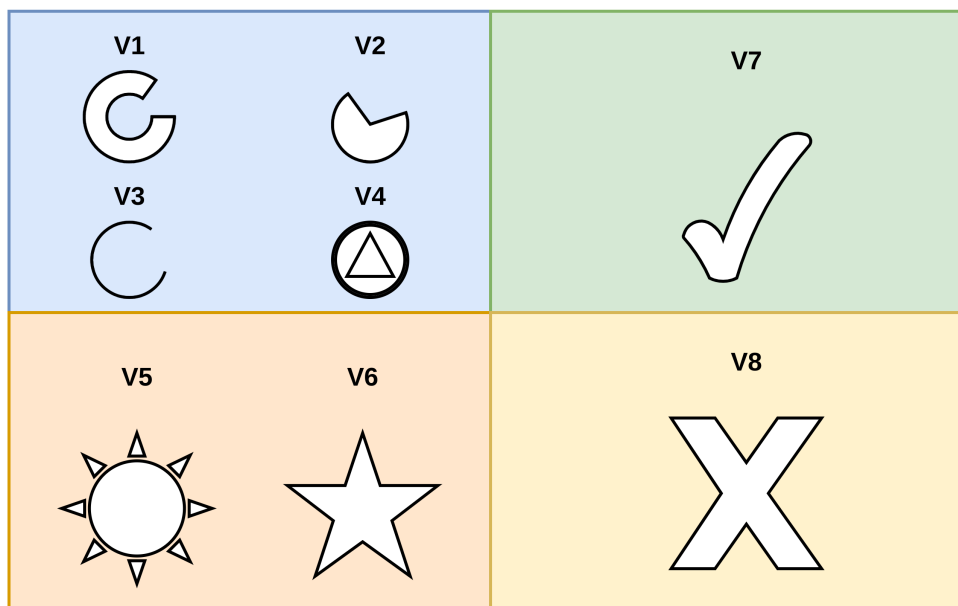


Fonte – O autor

Esses 4 monitores agora compõem uma grande área de exibição que pode ser utilizada para exibir painéis. Um painel nada mais é do que uma configuração personalizada

de visualizações parametrizadas e posicionadas nos monitores. A título de exemplo, a figura 3 mostra um painel teórico em exibição. Perceba que cada visualização está posicionada, dimensionada e trazendo uma informação diferente.

Figura 3 – Ilustração de um painel preenchido com visualizações em exibição



Fonte – O autor

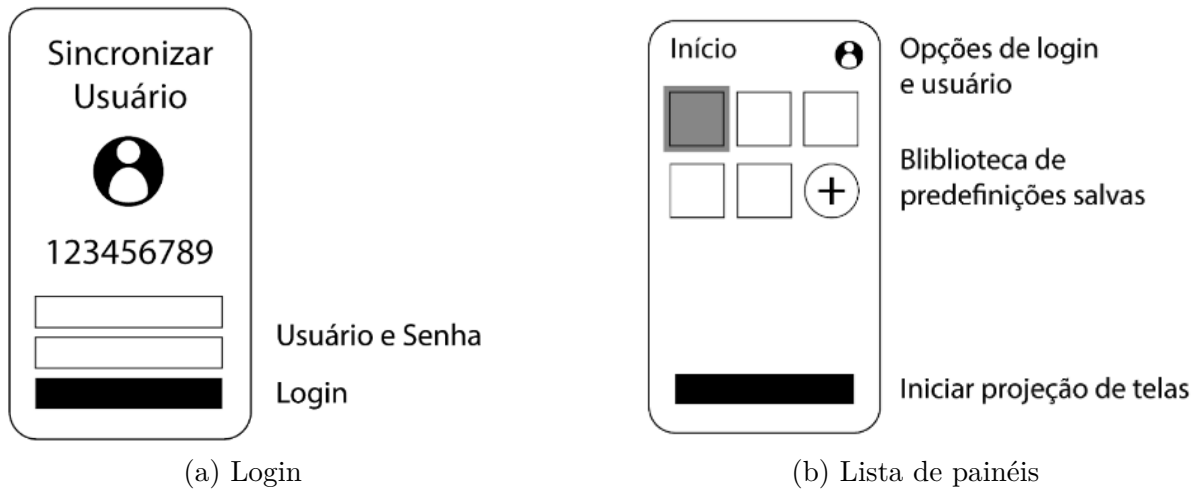
Nesse primeiro exemplo, entendemos o primeiro componente arquitetural do SVS-Box: o software embarcado que será instalado no computador e fará o controle das telas, criando as visualizações de acordo com o painel em exibição. Esse componente é denominado **renderizador**.

A interação do usuário com o renderizador se dá através do segundo componente arquitetural do SVSBox: um **aplicativo** instalado em celular ou tablet que funciona como um controle remoto. Ao instalar o aplicativo, o usuário autentica-se e consulta a biblioteca de informações disponíveis no SVSBox para em seguida montar painéis conforme desejar, conforme ilustrado pela figura 4.

A comunicação entre o aplicativo e o renderizador é intermediada pelo terceiro componente arquitetural do SVSBox: um **servidor central**. Esse componente também desempenha outras funções importantes: armazenar os dados de usuário e utilização (em um **banco transacional**) e disponibilizar o módulo de ingestão de dados semiautomático (denominado **degustador**).

A função do degustador é viabilizar a ingestão de dados de **fontes externas** por parte de um profissional da área da saúde que não possui as aptidões técnicas de um profissional de TI ou analista de dados. A partir dele, o usuário é capaz de expandir o

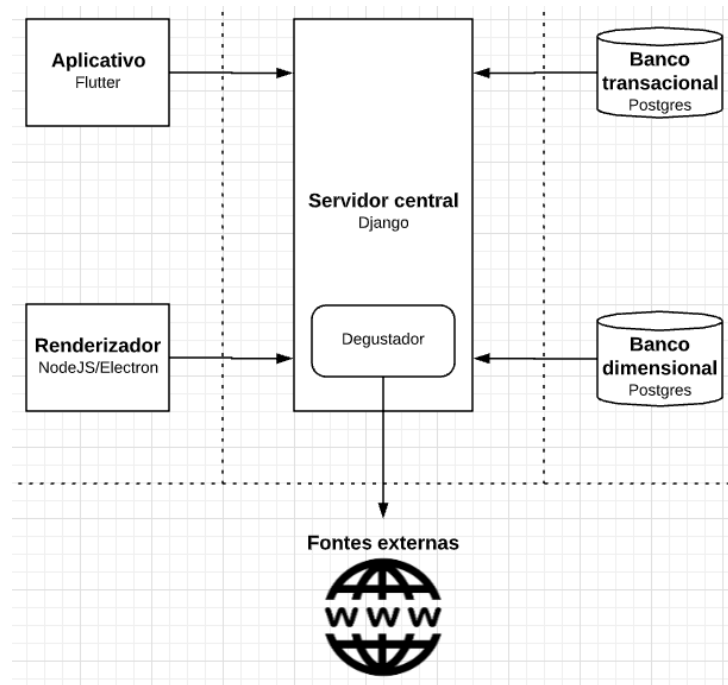
Figura 4 – Esboço de telas do aplicativo



Fonte – O autor

acervo de informações disponíveis no SVSBox (armazenadas no **banco dimensional**). A arquitetura geral de toda a solução proposta pode ser observada na figura 5.

Figura 5 – Visão arquitetural da solução SVSBox



Fonte – O autor

2.2 Renderizador

Este componente se refere a todo o software instalado no computador do SVSBox. Isto é, tanto a interface gráfica que controla as telas, quanto as modificações a nível de sistema operacional que previnem a utilização do computador para outros fins.

O controle das telas é realizado por uma aplicação baseada em *Node.js* e *Electron*. *Node.js* é um ambiente de execução de programas JavaScript baseado no interpretador V8, também utilizado pelo navegador Chrome do Google (NODEJS, 2021; V8, 2021). *Electron* é um framework para criação de aplicações nativas (compatíveis com Windows, macOS e Linux) utilizando tecnologias web como JavaScript, HTML e CSS (ELECTRON, 2021).

As telas são construídas com *Vue.js*, que é um framework progressivo para construir interfaces de usuário e que possibilita isolar a estrutura (em HTML), as funcionalidades (em JavaScript) e o estilo (em CSS) em componentes reutilizáveis (VUEJS, 2021). Ele é projetado de modo que seja possível utilizá-lo de forma incremental. Graças a isso, é fácil de aprender e de integrar a projetos de diferentes escopos e tamanhos (WOHLGETHAN, 2018).

Os gráficos são feitos com *Chart.js*, que é uma biblioteca JavaScript gratuita de código aberto e que suporta os principais tipos de gráficos, como barra, linha e pizza (CHARTJS, 2021).

Para manter o código fonte organizado, as responsabilidades foram divididas em alguns módulos JavaScript conforme resumido na tabela 1. O código foi pensado desde o início para manter a modularidade, viabilizando a extensão das funcionalidades oferecidas com pouco ou nenhum retrabalho. Utilizando-se de boas práticas de desenvolvimento, é possível prevenir perdas de produtividade e manter o ritmo de evolução do projeto (LIMA et al., 2021).

Tabela 1 – Lista dos módulos JavaScript mais relevantes elaborados para o Renderizador

Nome	Finalidade	Tipo
Orquestrador	Centralizar o controle do ciclo de execução da aplicação e dos outros módulos	Node.js
Janelador	Realizar os cálculos para definir qual janela recebe qual visualização	Node.js
Soldadinho	Controlar metadados do dispositivo de acordo com os comandos do servidor central	Node.js
PainelRepository	Abstrair a comunicação com a API REST do servidor central	Node.js
WSUtil	Abstrair a comunicação persistente via WebSocket com o servidor central	Node.js

Continua na próxima página

Tabela 1 – Continuação da página anterior

Nome	Finalidade	Tipo
Janela	Calcular o local e tamanho ocupado na tela de acordo com coordenadas e dimensões da visualização	Vue.js
Figura	Abstrair a transição entre os estados de carregamento, exibição e erro de uma visualização	Vue.js
Linha	Abstrair a renderização de um gráfico de linha ou coluna com <i>Chart.js</i>	Vue.js
Pizza	Abstrair a renderização de um gráfico de pizza com <i>Chart.js</i>	Vue.js
Iframe	Abstrair a exibição de multimídias na web como os canais de notícia disponíveis no YouTube	Vue.js

Fonte – O autor

Para utilização do dispositivo, é necessário realizar sincronização via QRCode com o aplicativo. A sincronização irá abrir uma conexão websocket, persistente e bidirecional, entre o dispositivo e o aplicativo, gerenciada pelo servidor central. Essa conexão é usada para o envio de comandos para o dispositivo tanto por parte do servidor central (como no caso de ativação/desativação), quanto por parte do aplicativo na mão do usuário (como no caso de iniciar apresentação de um painel).

O protocolo de comunicação elaborado para essa conexão websocket é descrito na seção 2.4. Essa seção também descreve o protocolo elaborado para a parametrização das visualizações na tela, que norteia a execução dos componentes *Vue.js* listados na tabela 1.

2.3 Aplicativo

Este componente se refere ao programa instalado em um dispositivo móvel (um celular ou um tablet). Suas finalidades principais são autenticar e autorizar o usuário, permitir o controle remoto do SVSBox e oferecer uma forma descomplicada de montagem de painéis.

É desenvolvido utilizando o *Flutter*, um kit de desenvolvimento de interfaces de usuário de código aberto criado pelo Google que possibilita a criação de aplicativos compilados nativamente, podendo ser executado em Android, iOS, Windows, Mac, Linux, Google Fuchsia e Web (FLUTTER, 2021).

O *Flutter* é baseado na linguagem de programação *Dart*, uma linguagem de script voltada à web desenvolvida pelo Google. Essa linguagem é otimizada para possibilitar aplicações rápidas em qualquer plataforma. Uma de suas grandes vantagens é a função

hot reload, um mecanismo de sua máquina virtual que permite injetar modificações no código fonte em uma aplicação em plena execução (DART, 2021). O *Flutter* leva isso além, adicionando suporte para *hot reload* de aplicações com estado. Em modo de desenvolvimento, é possível injetar alterações no código fonte da aplicação sem perda das variáveis ou necessidade de reinicialização do estado, e isso permite um ciclo de desenvolvimento mais rápido.

O aplicativo faz requisições HTTP ao servidor central e conecta-se ao mesmo via websocket para comunicar-se com o renderizador. Tudo isso de acordo com os protocolos descritos na seção 2.4.

2.4 Servidor central

Este componente se refere ao serviço web que alicerça a arquitetura, desempenhando funções fundamentais: armazenar todas as informações relacionadas aos usuários, aos dispositivos e à utilização do SVSBox, gerenciar a comunicação em tempo real entre as partes, e manter o repositório de dados que alimenta os gráficos.

O serviço web é feito com *Django*: um *framework Python* de alto nível, gratuito e de código aberto, que favorece designs limpos e pragmáticos e permite ciclos de desenvolvimento rápidos (DJANGO, 2021). Como complemento, também utilizou-se o *Django REST Framework*, que é uma biblioteca poderosa e flexível para construção de APIs web (Django REST Framework, 2021). Estas ferramentas foram utilizadas para construir tanto a interface administrativa (englobando o cadastro de usuários e controle de dispositivos) quanto a API de consultas para o aplicativo e o renderizador.

As habilidades do *Django* por si só englobam apenas HTTP, não sendo suficientes para atender aos requisitos do SVSBox. Em função disso, também utilizou-se o *Django Channels*, que é um projeto que visa estender as capacidades do *Django* para além do HTTP, habilitando-o a trabalhar com websockets, protocolos de chat, IoT, e mais (CHANNELS, 2021). Portanto, ele é fundamental para a comunicação persistente e em tempo real entre as partes.

Para gerenciamento de banco de dados, utilizou-se o *PostgreSQL*, um Sistema de Gerenciamento de Banco de Dados (SGBD) gratuito e de código aberto conhecido, graças aos seus mais de 30 anos de desenvolvimento ativo, por sua confiabilidade, robustez e desempenho (PostgreSQL, 2021).

O *Redis* é um armazém em memória de estruturas de dados de código aberto que pode ser utilizado como base de dados, cache ou mensageiro (Redis, 2021), e foi utilizado tanto como backend do *Django Channels* (na comunicação websocket) quanto como base de dados em memória para as operações mais dinâmicas do servidor central: gerenciar

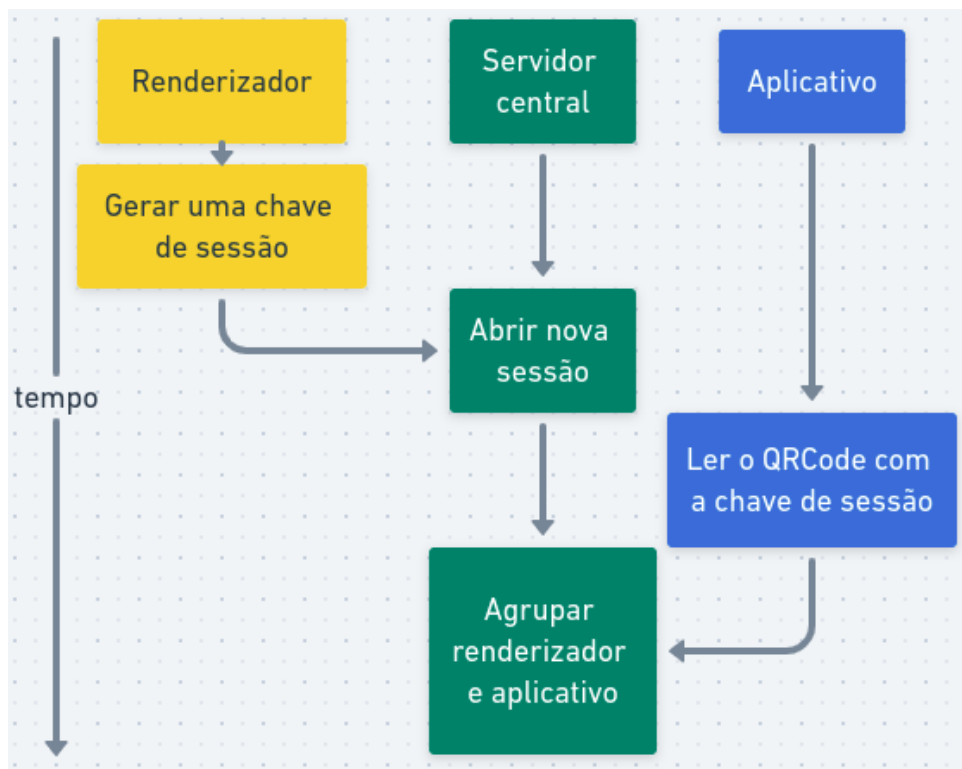
sessões ativas, organizar os pares de comunicação e transmitir eventos em tempo real.

2.4.1 Protocolo de comunicação websocket

A primeira ação do renderizador, ao iniciar, é gerar uma chave de sessão. Essa chave de sessão é uma string aleatória que servirá para identificar unicamente a sessão, tanto em relação ao dispositivo utilizado quanto em relação ao instante de execução. Isso garante que um dispositivo SVSBox sempre estará sincronizado apenas com o usuário presente na sala de situação em qualquer dado momento.

A chave de sessão é cadastrada no servidor central pelo renderizador, que então entra em estado de espera, exibindo um QRCode gerado a partir da chave. O usuário, com o aplicativo em mãos, aponta a câmera do seu celular para a tela do SVSBox, fazendo a leitura do QRCode e capturando a chave de sessão. O aplicativo então comunica essa chave ao servidor central, que irá realizar o casamento entre o renderizador e o aplicativo. Nesse ponto, está estabelecido um canal de comunicação bidirecional e persistente entre as partes, e o aplicativo pode começar a emitir comandos para o renderizador. Este processo está ilustrado na figura 6.

Figura 6 – Fluxograma de sincronização entre renderizador, servidor central e aplicativo



Fonte – O autor

Todos os comandos que trafegam pelo canal websocket são em formato JSON válido

e obedecem ao formato exibido na figura 7. O campo *"action"* é alguma chave conhecida (conforme tabela 2) e o campo *"value"* é qualquer objeto JSON válido (strings, números, listas, objetos) dependendo da necessidade do comando.

Figura 7 – Formato dos comandos que trafegam pelo canal websocket entre renderizador e aplicativo

```

1 {
2   "action": "algum comando",
3   "value": "algum parâmetro"
4 }
```

Fonte – O autor

Tabela 2 – Lista dos comandos compreendidos pelo renderizador

Chave	Descrição	Valor esperado
showSplashScreen	Mostrar tela de apresentação com logotipo e nome	Indiferente
showConnected	Mostrar tela com informe de conexão estabelecida	Indiferente
showPainel	Organizar a exibição das visualizações de um painel	ID do painel
updateFigura	Sinalizar que uma das visualizações em exibição precisa ser atualizada em tempo real	ID da visualização
default	Executado em caso de ação desconhecida	Indiferente
logJanelas	Utilitário de debug para inspecionar as janelas Electron	Indiferente
loadFile	Utilitário de debug para mostrar uma tela arbitrária	Caminho para o arquivo HTML em disco
ordem_update	Refletir alterações nos metadados do dispositivo	Objeto contendo metadados

Fonte – O autor

2.4.2 Protocolo de parametrização de visualizações

Uma visualização é um elemento gráfico que pertence a um painel. Do mesmo modo, um painel é uma coleção de visualizações que foram pensadas para representar um determinado cenário ou problema.

Cada visualização pode ter uma série de parâmetros ajustados para atender às necessidades do painel. Toda a área de exibição do SVSBox pode ser ocupada por visualizações, e o tamanho e posição de cada visualização são escolhidos pelo usuário. O sistema de coordenadas elaborado para o SVSBox foi inspirado no sistema de grade do *Bootstrap*, um mecanismo de construção de *layouts* em qualquer formato e tamanho (Bootstrap, 2021).

Toda a área de exibição do SVSBox pode ser representada em uma espécie de plano cartesiano em que o eixo vertical varia de 0 a 11, em números inteiros, e o eixo horizontal varia de 0 a (teoricamente) infinito, também em números inteiros. No eixo horizontal, cada grupo de 12 posições caracteriza um monitor, de sorte que as posições de 0 a 11 correspondem ao primeiro monitor, de 12 a 23 ao segundo e assim por diante. Dessa forma, cada monitor é, na verdade, uma grade de 12x12 unidades de área. Portanto, uma visualização pode assumir qualquer formato retangular, com tamanho mínimo de 1x1 e máximo de 12x12, e qualquer posição no "plano cartesiano", respeitando apenas as limitações de cada monitor.

O usuário também pode ajustar o escopo dos dados que alimentam a visualização. Isto é, o usuário pode definir uma data mínima, uma data máxima, uma área específica, uma categoria de dados específica. Em adição a isso, o usuário pode escolher como a informação deve ser apresentada: em forma de série temporal (linha ou coluna), em forma de gráfico de pizza, agrupado por alguma categoria. Essa parametrização é armazenada em formato JSON e está exemplificada na figura 8 e descrita na tabela 3.

Tabela 3 – Lista de chaves que parametrizam uma visualização

Chave	Descrição	Valor esperado
tipo	Tipo do gráfico	linha, coluna, pizza, multimidia
data_inicio	Data inicial dos registros considerados	string DateTime
data_fim	Data final dos registros considerados	string DateTime
agrupar_por	Critério de agregação dos registros	chave de uma categoria do dado
serie_temporal	Se o dado é caracterizado com informação temporal	booleano

Continua na próxima página

Tabela 3 – Continuação da página anterior

Chave	Descrição	Valor esperado
serie_temporal_truncamento	Nível de granularidade temporal	minute, hour, day...
filtros	critérios de inclusão/exclusão dos registros	lista JSON conforme figura 8

Fonte – O autor

Observe que as chaves *agrupar_por* e *filtros* listados na tabela 3 fazem referência às categorias do dado, isto é, às dimensões que caracterizam o dado. A título de exemplo, um usuário poderia querer ver um gráfico de pizza da incidência de sífilis adquirida agrupada por faixa etária (através da chave *agrupar_por*) apenas em pacientes do sexo feminino (através da chave *filtros*). Para viabilizar essa dinâmica, coleta-se alguns metadados da informação através do método descrito na seção 17.

2.4.3 Degustador

O SVSBox não cumpriria sua finalidade sem um acervo abrangente de informações para estudo. Ele também não cumpriria sua finalidade caso trouxesse apenas informações velhas, conforme contextualizado na seção 1.1.3. Tendo em mente que o usuário alvo possui pouca aptidão técnica, de que forma ele poderia utilizar o SVSBox para estudar as suas próprias informações, oriundas de uma fonte ainda não disponível?

A solução proposta para esse questionamento é implementar um módulo de ingestão de dados semiautomático. Isto é, um programa que irá capturar os dados de uma fonte externa e fazer as análises necessárias para que esses dados possam ser estudados com todo o auge do SVSBox. Esse programa é denominado Degustador, e sua função é transformar fontes de dados desconhecidas em fontes conhecidas. É semiautomático porque o usuário ainda precisa fornecer algumas orientações para que a captura seja efetuada corretamente. Contudo, essa interação é feita através de um formulário no padrão passo-a-passo e cobre apenas os critérios de conexão mais cruciais, de modo que um usuário leigo consiga utilizar e a necessidade de um profissional de TI ou analista de dados seja bastante mitigada.

A tabela 4 lista as informações que o usuário precisa fornecer ao Degustador para que este possa ingerir uma fonte externa.

Na versão atual do SVSBox, os dados são armazenados em um formato preliminar denominado **cubo**. De acordo com Paiva (2018), um cubo é uma representação de dados

Figura 8 – Exemplo de parametrização de uma visualização

```
1 {
2   "tipo": "linha",
3   "data_inicio": "2020-03-28",
4   "data_fim": "2020-07-23",
5   "agrupar_por": [],
6   "serie_temporal": true,
7   "filtros": [{
8     "tipo": "excluir",
9     "valor": "Cancelada",
10    "coluna": "situacao"
11  },
12  {
13    "tipo": "excluir",
14    "valor": "Expirada",
15    "coluna": "motivo_cancelamento"
16  }
17 ]
18 }
```

Fonte – O autor

agregados que facilita a geração de consultas *ad hoc* e a análise de dados em tempo real, permitindo a geração de gráficos sob demanda de maneira mais estruturada. Portanto, uma fonte de dados "conhecida pelo SVSBox" é uma fonte que passou pela análise do Degustador e deu origem a um cubo.

Na prática, um cubo corresponde a uma tabela em um banco de dados que tem pelo menos uma coluna obrigatória (a coluna do "fato") e qualquer quantidade de colunas adicionais (colunas de "dimensão"). O fato é o dado numérico que pode ser sumarizado para produzir informações sobre algum fenômeno (incidência de sífilis, taxa de mortalidade, etc). E cada uma das dimensões refere-se a uma característica que contextualiza o fato, como uma informação que fornece perspectiva adicional (data, local, tipo de evento, etc). A figura 9 ilustra um exemplo de cubo que trata de solicitações de leitos de UTI reguladas pelo sistema RegulaRN. A coluna *situacao* se refere à situação da solicitação (cancelada, regulada ou ocupação direta) e a coluna *motivo_cancelamento* se refere ao encerramento da solicitação (expirada, melhora clínica, alta hospitalar, óbito ou impossibilidade de transporte).

Ao processar uma fonte externa e gerar um cubo, as operações mais importantes de sumarização de dados tornam-se automaticamente habilitadas: *roll-up*, *drill-down*, *slice* e *dice* (AOUCHE; LEMIRE; GODIN, 2008; ARIFIN et al., 2017). As duas primeiras correspondem ao ato de agrupar e desagrupar os dados em alguma das dimensões, en-


Tabela 4 – Lista de informações coletadas pelo Degustador para ingestão de uma fonte externa.

Chave	Descrição
Endereço	Endereço da fonte externa na internet
Chave do valor numérico (fato)	Qual das "colunas" contém o valor numérico relevante
Tem data	Se o valor numérico é acompanhado de uma data (dimensão temporal)
Chave da data	Qual das "colunas" contém a data
Lista de colunas adicionais	Quais colunas (ou dimensões) da fonte externa devem acompanhar o valor numérico relevante
Título da fonte externa	Como referir-se a essa fonte em exposições
Descrição da fonte externa	Como descrever essa fonte em exposições
Periodicidade	Com que frequência essa fonte renova suas informações
Nome de exibição de colunas	Como exibir as colunas que acompanham o valor numérico relevante


Fonte – O autor

Figura 9 – Exemplo de cubo que trata de solicitações de leitos de UTI para COVID19 reguladas pelo sistema RegulaRN


data timestamp with time zone	situacao character varying(50)	motivo_cancelamento character varying(50)	valor integer
2020-05-17 00:40:24.28341-03	Cancelada	Expirada	1
2020-05-18 09:16:36.63734-03	Cancelada	Expirada	1
2020-05-14 14:52:04.126171-03	Cancelada	Expirada	1
2020-05-15 11:48:18.017805-03	Cancelada	Expirada	1
2020-05-15 13:47:27.55837-03	Cancelada	Expirada	1




Dimensão



Dimensão



Dimensão



Fato

Fonte – O autor

quanto que as duas últimas implicam em uma redução sistemática dos dados a partir de uma combinação lógica aplicada nas dimensões. A título de exemplo: uma visualização hipotética poderia ser parametrizada para mostrar dados de notificações de sífilis congênita **agrupadas** por cidade, **incluindo** apenas o estado do Rio Grande do Norte, **excluindo** o município Natal. Essa visualização poderia ser útil para investigar a quantidade de notificações em cada cidade no "interior" do estado, isto é, excluindo-se a capital.

Além disso, todas as operações estatísticas mais comuns também são facilmente calculadas: média, média móvel (no caso de uma série temporal), desvio padrão, percentis,

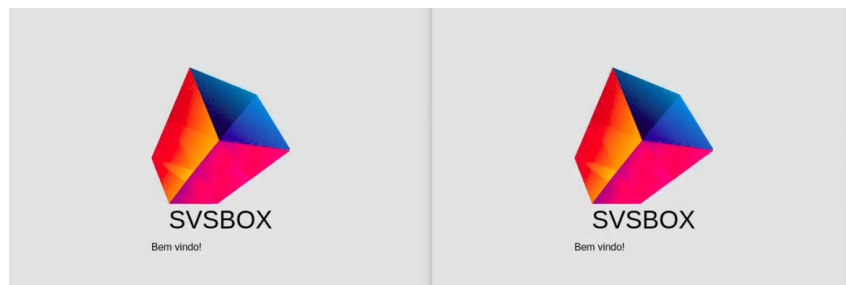
3 Resultados

Os objetivos do trabalho foram atingidos: o resultado é um painel digital controlado remotamente via aplicativo que pode equipar uma sala de situação em saúde, atuando como uma ferramenta dinâmica e descomplicada para informar e nortear as discussões.

3.1 Renderizador

O renderizador foi desenvolvido conforme proposto: faz o controle da área de exibição e pode ser controlado remotamente via comandos do aplicativo. Ele é capaz de consultar informações no servidor central e organizar visualizações ao gosto do usuário. A figura 10 mostra a tela de boas vindas ao usuário. Nessa e nas outras figuras desta seção, observa-se duas janelas lado a lado que simulam um cenário de utilização com 2 monitores.

Figura 10 – Tela de boas vindas no renderizador



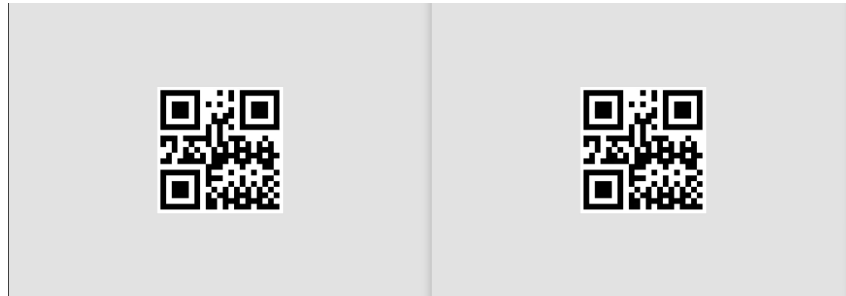
Fonte – Autoria própria

Na etapa de sincronização com o aplicativo, o renderizador exibe um código QR gerado conforme exposto na figura 11.

Ao sincronizar, o renderizador exibe um informe de prontidão, avisando ao usuário que a conexão está estabelecida conforme a figura 12.

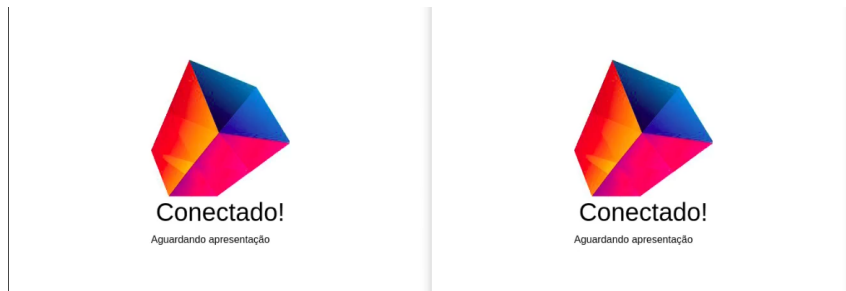
Uma vez que o usuário solicita a exibição de um painel, o aplicativo emite um comando de acordo com o protocolo de comunicação e o renderizador irá realizar as consultas necessárias para renderizar as visualizações de acordo com todas as suas parametrizações. O resultado pode ser observado na figura 13.

Figura 11 – Tela de sincronização com o aplicativo via QRCode



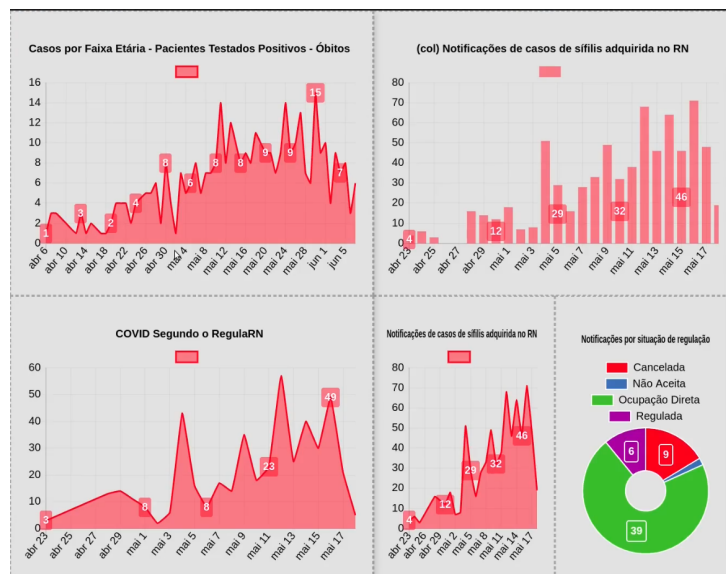
Fonte – Autoria própria

Figura 12 – Tela de aviso de prontidão para utilização



Fonte – Autoria própria

Figura 13 – Exemplo de exibição de painel

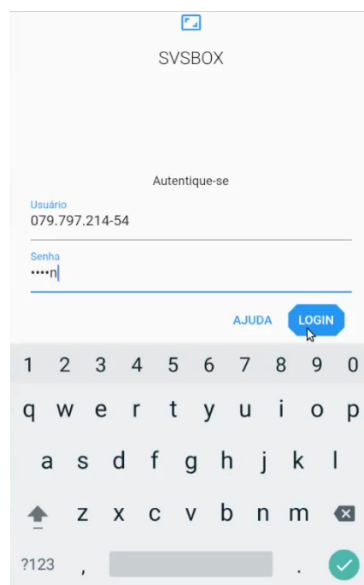


Fonte – Autoria própria

3.2 Aplicativo

O aplicativo autentica o usuário e permite a criação e edição de painéis e suas visualizações. A autenticação do usuário acontece por meio de credenciais previamente cadastradas no servidor central, conforme exposto na figura 14.

Figura 14 – Tela de autenticação do usuário



Fonte – Autoria própria

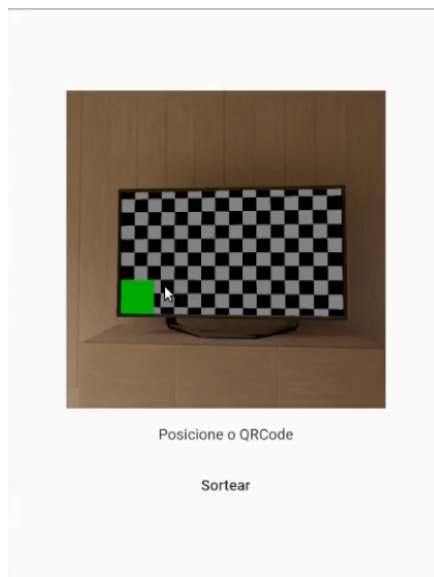
Uma vez autenticado, o usuário pode visualizar as opções de painéis já existentes (figura 15a), bem como cadastrar novos, ou iniciar uma apresentação. Contudo, antes de iniciar qualquer apresentação, a sincronização via QRCode deve ser feita. Para isso, o aplicativo aciona a câmera do dispositivo móvel e lê o QRCode em exibição no renderizador. Isso está ilustrado na figura figura 15b, em que temos o aplicativo sendo executado em um emulador. Nesse caso, a câmera pode apenas capturar um ambiente virtual.

A parametrização de uma visualização está exposta nas figuras 16a e 16b. Observe a utilização de termos e elementos gráficos intuitivos em detrimento de termos ou operações muito técnicas.

Figura 15 – Listagem de painéis disponíveis



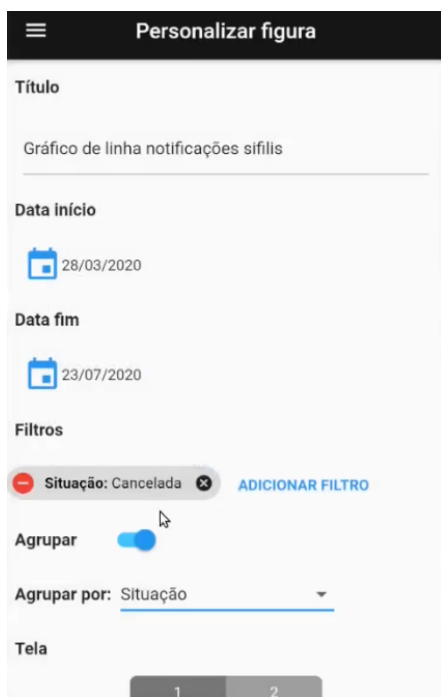
(a) Login



(b) Câmera acionada para leitura de QRCode.

Fonte – O autor

Figura 16 – Parametrização de uma figura



(a) Parte 1



(b) Parte 2

Fonte – O autor

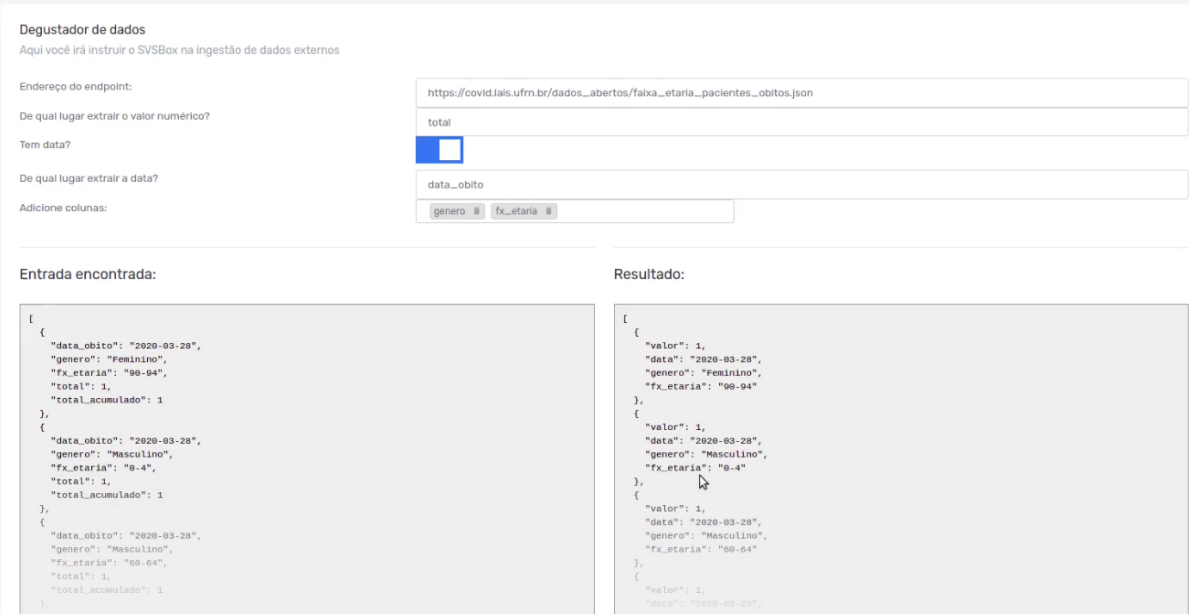
3.3 Degustador

O degustador automatiza a ingestão de dados externos. Por meio de uma interface simples, o usuário preenche um formulário no formato passo-a-passo para detalhar um recurso JSON disponível na web.

A figura 17 ilustra uma parte do formulário em que o usuário informa detalhes como o endereço do recurso, qual é o campo numérico relevante, se tem informação temporal associada, entre outros. Além disso, também há uma tela para coletar os metadados desse recurso, como o título e descrição.

Uma vez que o usuário realiza essa caracterização, o Degustador se responsabiliza por capturar e transformar o recurso, realizando atualização do mesmo conforme necessário. Desse modo, o usuário é habilitado a contribuir para o acervo de informações disponíveis no SVSBox.

Figura 17 – Formulário passo a passo do degustador para ingestão de fonte externa



Degustador de dados
Aqui você irá instruir o SVSBox na ingestão de dados externos

Endereço do endpoint:

De qual lugar extrair o valor numerico?

Tem data?

De qual lugar extrair a data?

Adicione colunas:

Entrada encontrada:

```
[
  {
    "data_obito": "2020-03-28",
    "genero": "Feminino",
    "fx_etaria": "90-94",
    "total": 1,
    "total_acumulado": 1
  },
  {
    "data_obito": "2020-03-28",
    "genero": "Masculino",
    "fx_etaria": "0-4",
    "total": 1,
    "total_acumulado": 1
  },
  {
    "data_obito": "2020-03-28",
    "genero": "Masculino",
    "fx_etaria": "60-64",
    "total": 1,
    "total_acumulado": 1
  }
]
```

Resultado:

```
[
  {
    "valor": 1,
    "data": "2020-03-28",
    "genero": "Feminino",
    "fx_etaria": "90-94"
  },
  {
    "valor": 1,
    "data": "2020-03-28",
    "genero": "Masculino",
    "fx_etaria": "0-4"
  },
  {
    "valor": 1,
    "data": "2020-03-28",
    "genero": "Masculino",
    "fx_etaria": "60-64"
  }
]
```

Fonte – Autoria própria

4 CONCLUSÃO

Apresentou-se o projeto e implementação de uma solução tecnológica de painel digital que integra diferentes fontes de informação e facilita o estudo centralizado das mesmas. O resultado deste trabalho é uma ferramenta que provê suporte à tomada de decisões estratégicas que tenham como intuito resolver problemas relacionados à gestão de políticas públicas de saúde.

Por esse viés, o SVSBox oferece recursos de visualização gráfica de informações em saúde ajustável, de forma dinâmica e descomplicada. A autonomia proposta pelo SVSBox por parte do gestor ou profissional da área da saúde contribui para tomadas de decisão mais assertivas, qualificando-o como um forte aliado para o combate ao aumento expressivo de casos de sífilis no país. Desse modo, é possível afirmar que a utilização dessa ferramenta pode auxiliar na redução dos custos globais dos serviços de saúde no Brasil.

4.1 Trabalhos Futuros

A partir do desenvolvimento da primeira versão do sistema apresentado, algumas limitações representam oportunidade de evoluções futuras:

- O degustador poderia qualificar melhor o dado ingerido (anotando o tipo de cada coluna, não apenas da data);
- O degustador poderia consumir vários formatos além de JSON, como planilhas excel e CSV. Também é possível implementar a captura via outras formas além de HTTP, como upload arquivo e conexão a banco de dados;
- Os gráficos ofertados pelo renderizador poderiam ser agnósticos com relação ao formato do dado. Isso significa não obrigar o dado a ser armazenado em formato de cubo, mas sim em formato livre.

Referências

- ACOSTA, A. et al. Medicine shortages: gaps between countries and global perspectives. *Frontiers in pharmacology*, Frontiers, v. 10, p. 763, 2019. Citado na página 13.
- AOUICHE, K.; LEMIRE, D.; GODIN, R. Web 2.0 olap: From data cubes to tag clouds. In: SPRINGER. *International Conference on Web Information Systems and Technologies*. [S.l.], 2008. p. 51–64. Citado na página 28.
- ARIFIN, S. et al. An online analytical processing multi-dimensional data warehouse for malaria data. *Database*, Oxford Academic, v. 2017, 2017. Citado na página 28.
- Bootstrap. *Build fast, responsive sites with Bootstrap*. 2021. <<https://getbootstrap.com/>>. Citado na página 26.
- BRASIL. Boletim Epidemiológico de Sífilis MS. *Ministério da Saúde*, Ano V n-1, p. 43, 2019. Disponível em: <www.aids.gov.br>. Citado na página 12.
- BRASIL. Guia de vigilância em saúde. 2019. Acesso em 12 abr. 2021. Disponível em: <https://bvsmis.saude.gov.br/bvs/publicacoes/guia_vigilancia_saude_3ed.pdf>. Citado na página 12.
- BRASIL. Boletim Epidemiológico - Sífilis. *Ministério da Saúde. Secretaria de Vigilância em Saúde*, p. 44, 2020. ISSN ISSN 2358-9450. Citado na página 13.
- CHANNELS. *Django Channels 3.0.3 documentation*. 2021. <<https://channels.readthedocs.io/en/stable/>>. Citado na página 23.
- CHARTJS. *Chart.js: Open source HTML5 Charts for your website*. 2021. <<https://www.chartjs.org/>>. Citado na página 21.
- CHEN, X.-S. Challenges in responses to syphilis epidemic. *The Lancet. Infectious diseases*, v. 17, n. 8, p. 793–794, 2017. Citado na página 13.
- DANIEL, V. M. *Os sistemas de informação em saúde e seu apoio à gestão e ao planejamento do SUS: uma análise de estados brasileiros*. Dissertação (Mestrado) — Pontifícia Universidade Católica do Rio Grande do Sul, 2013. Citado na página 15.
- DART. *Dart programming language*. 2021. <<https://dart.dev/>>. Citado na página 23.
- DATASUS. Plano Diretor de Tecnologia da Informação do Ministério da Saúde 2014-2015. n. 61, p. 84, 2013. Citado na página 15.
- DATASUS. Plano Diretor de Tecnologia da Informação. n. 61, p. 92, 2017. Disponível em: <http://datasus.saude.gov.br/images/pdti/PDTI_MS_1718.pdf>. Citado na página 15.
- DJANGO. *Django: The web framework for perfectionists with deadlines*. 2021. <<https://www.djangoproject.com/>>. Citado na página 23.
- Django REST Framework. *Django REST framework: a powerful and flexible toolkit for building Web APIs*. 2021. <<https://www.django-rest-framework.org/>>. Citado na página 23.

ELECTRON. *Electron: Build cross-platform desktop apps with JavaScript, HTML, and CSS*. 2021. <<https://www.electronjs.org/>>. Citado na página 21.

FLUTTER. *Flutter: Beautiful native apps in record time*. 2021. <<https://flutter.dev/>>. Citado na página 22.

GRIMPREL, E. et al. Use of polymerase chain reaction and rabbit infectivity testing to detect *treponema pallidum* in amniotic fluid, fetal and neonatal sera, and cerebrospinal fluid. *Journal of clinical Microbiology*, Am Soc Microbiol, v. 29, n. 8, p. 1711–1718, 1991. Citado na página 12.

LIMA, D. Ferreira de et al. Padrão de projeto flux para sistemas de informação na área da saúde: Um estudo de caso sobre o sistema retratos da atenção primária à saúde. *Revista Brasileira de Inovação Tecnológica em Saúde - ISSN:2236-1103*, v. 10, n. 2, p. 9, mar. 2021. Disponível em: <<https://periodicos.ufrn.br/reb/article/view/23970>>. Citado na página 21.

LUNKES, R. J. et al. Development of an information system at the strategic management support office of the brazilian ministry of health. *Revista Cubana de Información en Ciencias de la Salud*, Centro Nacional de Información de Ciencias Médicas, v. 27, n. 2, p. 168–184, 2016. Citado na página 15.

Ministério da Saúde. *Protocolo clínico e diretrizes terapêuticas para prevenção da transmissão vertical de HIV, sífilis e hepatites virais*. [S.l.]: Ministério da Saúde Brasília, 2015. Citado na página 12.

NETO, G. C. C. Integração entre sistemas de informação em saúde: o caso do e-sus atenção básica. Universidade Federal de São Paulo (UNIFESP), 2019. Citado na página 15.

NETO, J. H. V. d. S. *SVSBOX: SISTEMA DEDICADO PARA VIGILÂNCIA EM SAÚDE*. Dissertação (Mestrado), 2020. Citado na página 16.

NEWMAN, L. et al. Global estimates of the prevalence and incidence of four curable sexually transmitted infections in 2012 based on systematic review and global reporting. *PloS one*, Public Library of Science San Francisco, CA USA, v. 10, n. 12, p. e0143304, 2015. Citado na página 13.

NODEJS. *Node.js: a JavaScript runtime built on Chrome's V8 JavaScript engine*. 2021. <<https://nodejs.org/>>. Citado na página 21.

NONAKA, I.; TAKEUCHI, H. *Criação de conhecimento na empresa*. [S.l.]: Elsevier Brasil, 1997. Citado na página 14.

NURSE-FINDLAY, S. et al. Shortages of benzathine penicillin for prevention of mother-to-child transmission of syphilis: An evaluation from multi-country surveys and stakeholder interviews. *PLoS medicine*, Public Library of Science San Francisco, CA USA, v. 14, n. 12, p. e1002473, 2017. Citado na página 13.

Organização Pan-Americana da Saúde. Sala de situação de saúde. 2010. Acesso em 11 abr. 2021. Disponível em: <https://www.paho.org/bra/index.php?option=com_content&view=article&id=324:sala-de-situacao-de-saude&Itemid=685>. Citado na página 15.

PAIVA, J. C. D. SMART: Sistema de Monitoramento e Avaliação do Programa Nacional Telessaúde Brasil Redes. p. 198, 2018. Citado na página 27.

PostgreSQL. *PostgreSQL: The world's most advanced open source database*. 2021. <<https://www.postgresql.org/>>. Citado na página 23.

QURESHI, F.; JACQUES, S. M.; REYES, M. P. Placental histopathology in syphilis. *Human pathology*, Elsevier, v. 24, n. 7, p. 779–784, 1993. Citado na página 12.

Redis. *Redis: in-memory data structure store*. 2021. <<https://redis.io/>>. Citado na página 23.

ROLFS, R. T. et al. A randomized trial of enhanced therapy for early syphilis in patients with and without human immunodeficiency virus infection. *New England Journal of Medicine*, Mass Medical Soc, v. 337, n. 5, p. 307–314, 1997. Citado na página 12.

SANTIS, M. D. et al. Syphilis infection during pregnancy: fetal risks and clinical management. *Infectious diseases in obstetrics and gynecology*, Hindawi, v. 2012, 2012. Citado na página 12.

SANTOS, M. Marques dos et al. Trends of syphilis in brazil: A growth portrait of the treponemic epidemic. *PloS one*, Public Library of Science San Francisco, CA USA, v. 15, n. 4, p. e0231029, 2020. Citado na página 13.

SELLERA, P. E. G. et al. Sala de situação de saúde do distrito federal e o uso de tecnologias livres para o monitoramento da sífilis. *Revista Brasileira de Inovação Tecnológica em Saúde-ISSN: 2236-1103*, p. 12–12, 2019. Citado na página 16.

V8. *V8 JavaScript engine*. 2021. <<https://v8.dev/>>. Citado na página 21.

VUEJS. *VueJS: The Progressive JavaScript Framework*. 2021. <<https://vuejs.org/>>. Citado na página 21.

WOHLGETHAN, E. *Supporting Web Development Decisions by Comparing Three Major JavaScript Frameworks: Angular, React and Vue.js*. Tese (Doutorado) — Hochschule für Angewandte Wissenschaften Hamburg, 2018. Citado na página 21.

WORKOWSKI, K. A.; BOLAN, G. A. Sexually transmitted diseases treatment guidelines, 2015. *MMWR. Recommendations and reports: Morbidity and mortality weekly report. Recommendations and reports*, NIH Public Access, v. 64, n. RR-03, p. 1, 2015. Citado na página 12.

World Health Organization. Report on global sexually transmitted infection surveillance 2018. World Health Organization, 2018. Citado na página 13.