



ISPCONECTA: UMA PLATAFORMA SHINY PARA DADOS DE INTELIGÊNCIA NA SEGURANÇA PÚBLICA

Ricardo Junqueira de Souza¹

André Henrique de Barros Gomes²

João Pedro de Sousa Silveira³

Erick Baptista Amaral de Lara⁴

Resumo

O Instituto de Segurança Pública (ISP) é responsável pela consolidação, análise e divulgação dos dados oficiais de segurança pública do estado do Rio de Janeiro, recebendo dados de diversos órgãos do setor e contando com uma equipe multidisciplinar neste processo, seguindo a visão do ISP de se consolidar como órgão de referência na produção de conhecimento e na promoção da participação social em segurança pública. Dessa forma, o R e todo o universo *Shiny* surgem com um enorme potencial de atender todos os públicos, sejam eles internos ou externos, uma ferramenta *open source* para substituir as rotinas anteriores feitas em *softwares* empresariais com limitações nas versões gratuitas utilizadas. Neste contexto, foi desenvolvido o *hub* ISPConecta, uma plataforma de *dashboards* desenvolvidos em *Shiny* para disponibilização de dados de segurança pública tanto para órgãos do governo quanto para a população.

Palavras-chave: *Shiny, R, dashboards, Shiny Server, segurança pública.*

Abstract

The Instituto de Segurança Pública (ISP) is responsible for the consolidation, analysis, and dissemination of official public security data in the state of Rio de Janeiro, receiving data from various agencies in the sector and relying on a multidisciplinary team in this process. Following the ISP's vision of establishing itself as a reference agency in the production of knowledge and the promotion of social participation in public security. In this way, R and the entire Shiny universe emerge with enormous potential to serve all audiences, whether internal or external, an open-source tool to replace previous routines made in business software with limitations in the free versions used. In this context, the ISPConecta hub was developed, a Shiny-developed dashboard platform for making public security data available to the population.

Keywords: *Shiny, R, dashboards, Shiny Server, public security.*

¹ Instituto de Segurança Pública (ISP), ricardoj.isp@gmail.com

² Instituto de Segurança Pública (ISP), andregomes.isp@gmail.com

³ Instituto de Segurança Pública (ISP), jaoapedrosilveira.isp@gmail.com

⁴ Instituto de Segurança Pública (ISP), ericklara.isp@gmail.com



Introdução

O Instituto de Segurança Pública (ISP) é uma autarquia responsável pela consolidação, análise e divulgação dos dados oficiais de segurança pública do estado do Rio de Janeiro (Instituto de Segurança Pública, 2023a). Mensalmente são extraídos os dados das Secretarias de Polícias Civil e Militar, estes são consolidados por uma equipe que inclui estatísticos, policiais civis e analistas de segurança pública e então divulgados por meio de bases de dados temáticas, *dashboards* e *releases* para a imprensa.

Um dos pilares do instituto é a transparência, de modo que a busca por novas tecnologias *open source* que permitam facilitar o acesso da população aos dados, expandindo o potencial de disseminação destas informações, tem um papel fundamental neste processo (Instituto de Segurança Pública, 2024). Nos últimos anos, uma das atualizações realizadas neste sentido foi a substituição de *softwares* pagos como o *STATA* e o *SPSS* para soluções desenvolvidas em *R* (IBM Corp, 2011; R Core Team, 2021). Inicialmente, estas soluções buscavam simplificar rotinas e automatizar processos, porém, com o tempo, foram gradualmente expandidas para a produção de relatórios em *rmarkdown* até eventualmente chegarem a construção de painéis através de um servidor configurado com o *Shiny Server* (Allaire et al., 2023).

A adoção dos *dashboards* desenvolvidos em *Shiny* tem como objetivo principal a substituição da antiga infraestrutura de painéis hospedados no *Tableau Public* (Chang et al., 2022; Tableau Software, 2023). Apesar de ter como ponto facilitador a hospedagem acessível, a versão gratuita do *Tableau* não possibilita a conexão com bancos de dados, impedindo assim que suas atualizações fossem automatizadas, tendo pouco poder de processamento, o que piorava seu desempenho ao trabalhar com conjuntos de dados volumosos, e as limitações de utilizar um *software* fechado. Por sua vez, por se tratar de uma opção *open source*, o *Shiny* apresenta inúmeras opções de construção de interfaces para o usuário, possibilidades de automação e conexão com diferentes tipos de bancos de dados e integração com todo o ambiente de pacotes do *R*, fazendo com que se tornasse a opção ideal para utilização.

Os painéis desenvolvidos em *Shiny* pelo ISP são separados em dois grupos, enquanto um atende os clientes internos do governo estadual, o outro é destinado ao público geral. Para o grupo interno, são construídos painéis *on demand* e que visam cumprir objetivos específicos de órgãos governamentais, ao passo que os externos são pensados de modo a tornar a informação mais acessível e de fácil compreensão da população em geral. A página <https://www.ispconecta.rj.gov.br> foi construída para ser um *hub* que permite o acesso aos painéis, contendo desde versões atualizadas dos antigos painéis em *Tableau*



disponíveis no endereço <https://www.ispvisualizacao.rj.gov.br/> (Instituto de Segurança Pública, 2023b), até novos *dashboards* voltados para temas específicos como, por exemplo, o ISPMulher - painel desenvolvido para disseminar informações sobre a violência contra a mulher.

Objetivo

Relatar a experiência do desenvolvimento do ISPConecta, uma plataforma baseada em *R*, *Shiny* e *PostgreSQL* para hospedar a nova infraestrutura de *dashboards* do Instituto de Segurança Pública, de modo que potencialize a capacidade do instituto em publicizar os dados, com maior eficiência e menor custo operacional.

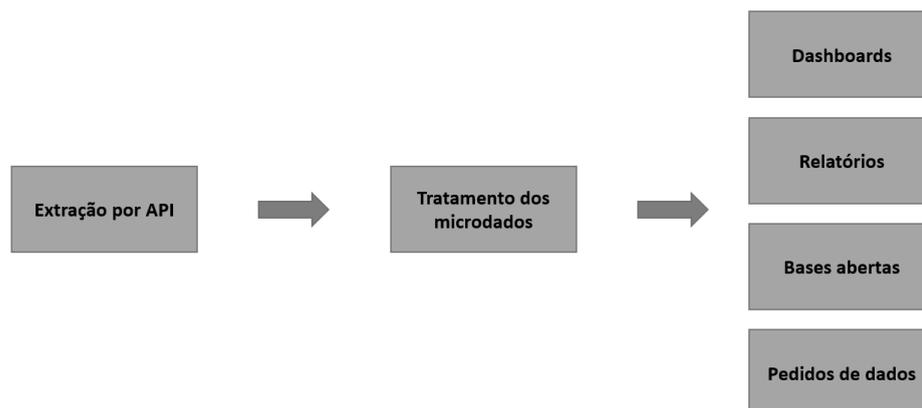
Material e Método

A principal fonte de dados do Instituto de Segurança Pública são as informações dos boletins de ocorrência registrados nas delegacias de Polícia Civil e extraídas diariamente dos bancos de dados da Secretaria de Estado de Polícia Civil (SEPOL). Estes dados contêm desde informações administrativas, como o horário e o local em que o registro foi realizado, à titulação criminal das condutas apuradas no registro, informações sobre os envolvidos (vítimas, autores, entre outros) e as referências espaço-temporais da ocorrência.

O processo de tratamento destes dados é feito em três fases distintas: a primeira delas é denominada Parcial, em que os dados diários são extraídos, tratados, georreferenciados e disponibilizados para as polícias por meio de *dashboards*. A segunda delas ocorre cinco dias úteis após a conclusão do mês, onde os dados completos do mês são extraídos, tratados de forma mais complexa e então agregados às bases de microdados. A partir da base de microdados são geradas bases secundárias que contêm dados agregados como contagens por região, relatórios em *rmarkdown*, entre outros produtos. Por fim, os dados são consolidados com erratas três meses após o mês em que foram consolidados. Esta versão é considerada definitiva e utilizada para o cálculo dos resultados do Sistema Integrado de Metas (SIM) da segurança pública do Rio de Janeiro.

O *R* tem um papel fundamental no processo de consolidação, sendo utilizado desde a extração das informações por meio de *API*, ao tratamento e construção de bases secundárias, carga nos bancos de dados e, recentemente, para elaboração de painéis interativos e relatórios dinâmicos. A Figura 1 a seguir apresenta um fluxograma do processo de ETL dos dados do ISP.

Figura 1: Fluxograma do tratamento dos dados.



Fonte: Elaborado pelo autor.

O processo de extração dos dados utiliza o pacote *httr* para obter os dados da API da Polícia Civil em formato *json*, estes dados são manipulados utilizando o pacote *jsonlite* e alimentados em um banco *PostgreSQL* por meio dos pacotes *DBI* e *RODBC* (Wickham, 2023a; Ooms, 2014; Wickham, 2023b; Wickham & Muller, 2021; Ripley & Lapsley, 2021). Ao longo de um mês, cada aditamento em um registro de ocorrência gera uma nova entrada no banco de dados, por isso foi construída uma *view* que contém os dados da versão mais recente de cada registro até uma determinada data de corte. A partir desta *view* é iniciado o tratamento dos dados por meio de funções do *dplyr*, *lubridate*, *stringr* e *tidyr* (Wickham et al., 2023; Grolemond & Wickham, 2011; Wickham, 2022; Wickham, Vaughan & Girlich, 2023). Neste processo são excluídos os registros com *status* de cancelados, são adicionados os rótulos às variáveis sociodemográficas, as datas e endereços são ajustados e novas variáveis são construídas. O resultado final desta etapa de tratamento é uma base de microdados parciais, posteriormente, no início do mês subsequente, estes dados passam pelo processo de consolidação.

A principal diferença do dado consolidado para o parcial é que o primeiro passa por um processo de qualificação da informação para alguns delitos. Por exemplo, todos os registros que envolvem crimes violentos letais contra a vida são conferidos por uma equipe de policiais civis que validam informações, como o número de vítimas e suas respectivas informações, e a classificação do delito. Tanto a base parcial quanto a consolidada são utilizadas para alimentar uma série de produtos produzidos pelo ISP, entretanto este trabalho se manterá restrito aos produtos que utilizam o *Shiny* e são construídos com os dados consolidados.

A antiga estrutura de *dashboards* do instituto era construída em *Tableau Public*, que gerava uma série de desafios para a atualização, uma vez que não era possível automatizar



as atualizações, conectar a bancos de dados e determinados problemas de performance que não possuíam soluções acessíveis. A adoção do *Shiny* buscava suprir essas demandas, além de proporcionar novas possibilidades devido a sua utilização em conjunto com outros pacotes do *R* e flexibilidade para desenvolvimento. Para hospedar os painéis foi configurado um servidor Ubuntu com *Shiny Server* e uma instância do *PostgreSQL* para armazenar os bancos de dados que alimentariam os painéis.

Inicialmente se pensou em uma estrutura básica padronizada para os painéis utilizando o pacote *bs4Dash* (Granjon, 2022). Eles são construídos contendo abas, cada qual trazendo um determinado tipo de visualização de dados e seus respectivos filtros associados, sendo elas: Série histórica, Comparativo e Distribuição Espacial. A primeira aba apresenta a série histórica dos delitos incluídos no painel em diferentes níveis de agrupamento espacial, como estado, município e os diferentes tipos de divisões espaciais de segurança pública. Todos os gráficos apresentados nos painéis são construídos com o pacote *echarts4r*, que possibilita construir gráficos interativos utilizando uma biblioteca de JavaScript (Coene, 2022).

A segunda aba permite comparar os resultados de diferentes delitos entre dois períodos de tempo definidos por meio de filtros nos painéis em diferentes divisões espaciais. Em seguida, a aba de distribuição espacial apresenta mapas coropléticos interativos construídos no *leaflet* para diferentes divisões espaciais do estado (Cheng et al., 2022). Além disso, há espaço para aplicações temáticas de acordo com o tema do painel como, por exemplo, o mapa da rede de atendimento para mulheres vítimas de violência que foi incluído no painel ISPMulher.

O monitoramento dos *dashboards* desenvolvidos em *Shiny* é feito com base na integração com a plataforma *Google Analytics*. Isto permite que os acessos aos painéis sejam visualizados em tempo real, além de gerar estatísticas e séries temporais do número de acessos e disponibilizar dados sobre os usuários como o local do acesso e o tipo de dispositivo utilizado. Com base nestas informações, e entendendo as necessidades de seus “clientes”, o gestor público pode otimizar a transparência ativa do órgão perante a sociedade.

Além dos *dashboards* voltados para o público externo, também foram desenvolvidas soluções em *Shiny* voltadas para o público interno do governo - sejam outras coordenadorias do próprio instituto ou outros órgãos do governo. Neste contexto, foram lançados o Monitor da Mulher para uso pela Secretaria de Estado da Mulher (SEM), um painel similar ao ISPMulher, porém com maior abrangência de delitos e o ISPCom, feito para auxiliar a assessoria de comunicação do ISP na elaboração dos *releases* mensais com os resultados da segurança pública.

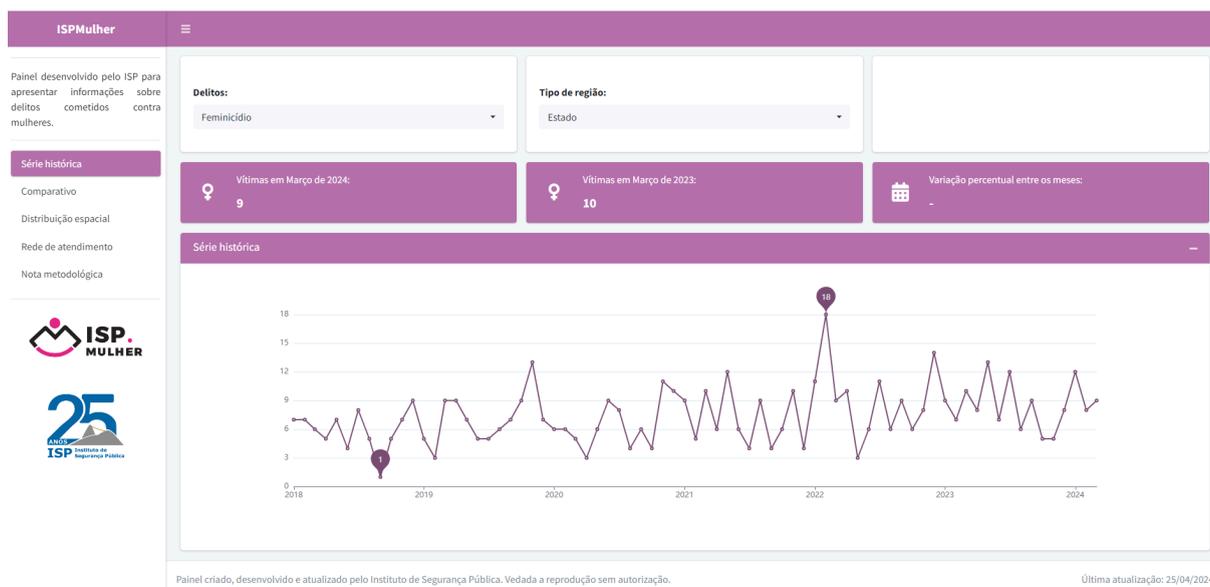
Resultados e Discussão

Os painéis desenvolvidos em *Shiny* estão disponíveis no endereço <https://ispconecta.rj.gov.br/> (Instituto de Segurança Pública, 2023c), página hospedada no próprio servidor com o *Shiny Server*. O primeiro painel lançado para o público externo foi o ISPMulher, um *dashboard* voltado para crimes contra a mulher e que apresenta informações sobre os seguintes agrupamentos de delitos: Homicídio doloso, tentativa de homicídio, constrangimento ilegal, feminicídio, tentativa de feminicídio, assédio sexual, difamação e violação de domicílio.

Sua tela inicial, conforme apresentada na Figura 2, apresenta os filtros de delito, tipo de região e filtros específicos para as regiões (Municípios, regiões administrativas e demais regiões de segurança pública). Entre as visualizações disponíveis estão a série temporal por mês a partir de 2018 e três *cards* que apresentam o número de vítimas no mês mais recente, no mesmo mês no ano anterior e a variação percentual entre ambos. Deste modo, por meio dos filtros adicionados, o usuário consegue acessar a respectiva série histórica do delito e região de interesse, sendo possível também criar grupos de regiões nos filtros.

A utilização do *echarts4r* como pacote para construir os gráficos trouxe novas opções de interatividade. Ele disponibiliza funções que permitem a utilização de *mouseover* para revelar mais informações sobre o gráfico ao passar o mouse sobre a série temporal, além da possibilidade de fazer *download* do gráfico e destacar os meses com maior e menor valores observados.

Figura 2: Aba de série histórica do painel ISPMulher.



Fonte: Elaborado pelo autor.



Em seguida, a aba de comparativo é apresentada na Figura 3. Esta aba consiste em um conjunto de filtros para o delito, tipo de região e os das respectivas regiões, além de dois filtros de data: um para cada período utilizado na comparação. O principal resultado desta aba é uma tabela criada no pacote *DT* contendo uma linha para cada delito selecionado e colunas para os totais em cada período e para as diferenças absoluta e percentual (Granjon, 2022). O *DT* apresenta uma série de opções de interatividade, como a possibilidade de reordenar as linhas em ordem alfabética ou de acordo com seus valores e também permite a exportação das tabelas em diferentes formatos como CSV ou PDF.

Figura 3: Aba de comparativo do ISPMulher.

Delito	Período 1	Período 2	Diferença Absoluta	Diferença Percentual
Difamação	1066	1205	139	13,0%
Violação de domicílio	710	863	153	21,5%
Tentativa de homicídio	174	184	10	5,7%
Constrangimento ilegal	92	117	25	27,2%
Tentativa de feminicídio	88	111	23	26,1%
Assédio sexual	60	94	34	56,7%
Homicídio doloso	65	59	-6	-9,2%
Feminicídio	26	29	3	

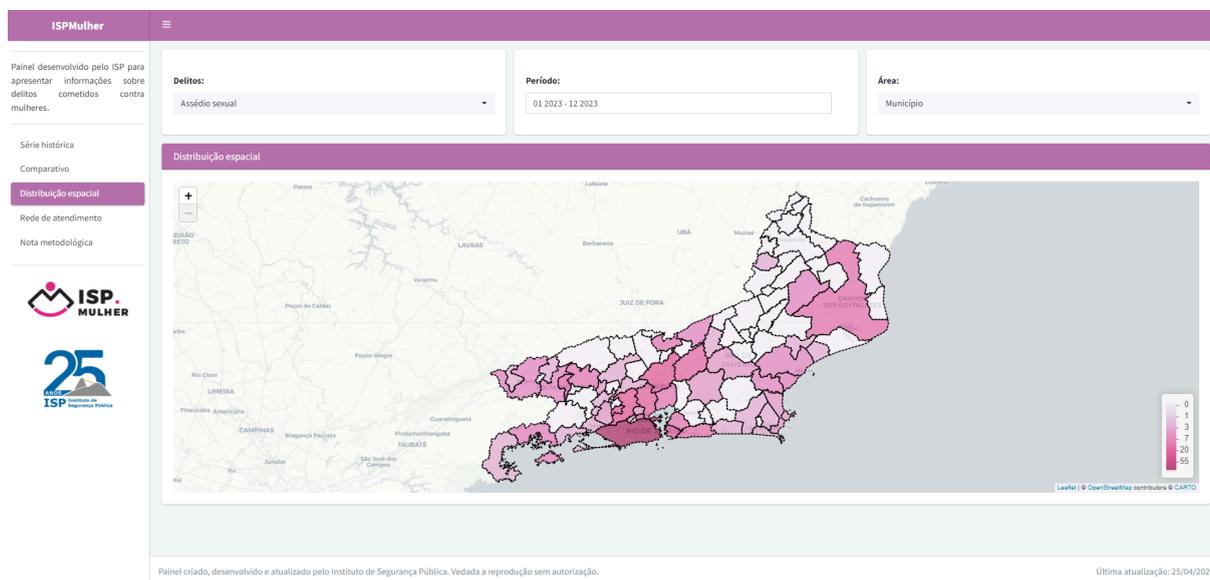
Fonte: Elaborado pelo autor.

A utilização de opções de exportação das visualizações presentes no *dashboard* tem papel fundamental na simplificação da disseminação das informações. O ISP também conta com meios de disponibilização das informações, como o formulário de pedidos de dados e a Lei de Acesso à Informação (LAI), porém estes são respondidos com uma base de microdados e a análise fica por parte do solicitante. Deste modo, foi alcançado um novo patamar de transparência ao permitir que o usuário trabalhe as informações que deseja dentro das possibilidades dos painéis e consiga exportá-las em um formato pronto para utilização.

A aba de distribuição espacial foi construída de modo a permitir que os usuários consigam avaliar o comportamento no espaço dos diferentes delitos incluídos no painel por meio de mapas coropléticos - mapas para as quantidades agregadas de vítimas. Os filtros disponíveis também permitem delimitar o período considerado para o cálculo do número

agregado de vítimas e também escolher o nível de divisão espacial entre municípios e os três tipos de regiões de segurança pública.

Figura 4: Aba de distribuição espacial do painel mulher.



Fonte: Elaborado pelo autor.

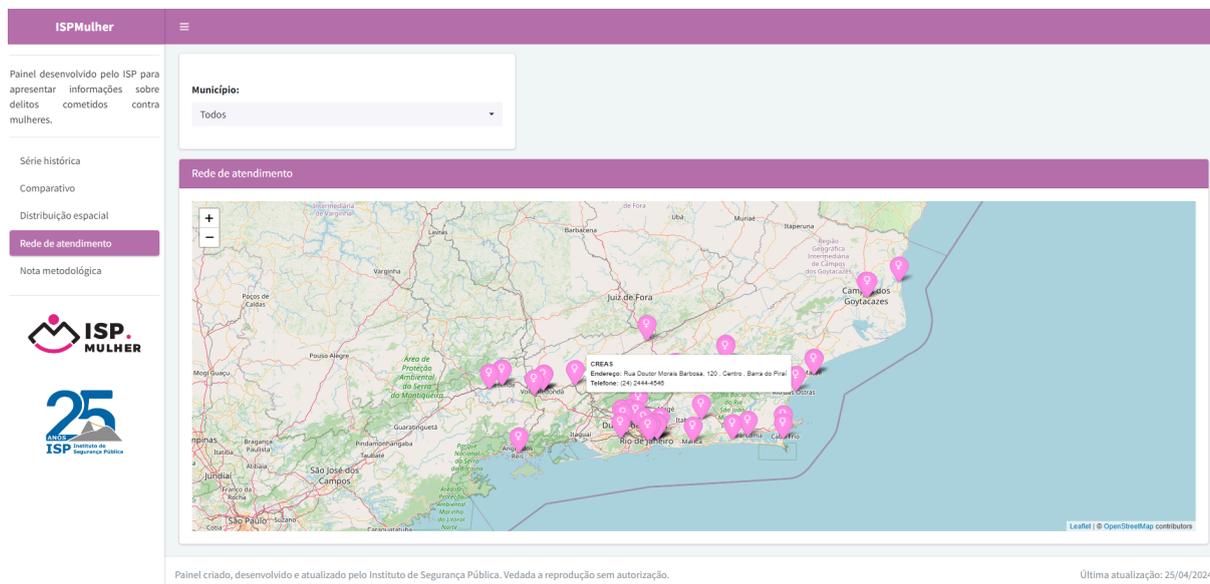
A Figura 5 a seguir apresenta a aba da rede de atendimento para mulheres vítimas de violência. Os dados da rede foram extraídos do Dossiê Mulher, uma publicação realizada anualmente pelo ISP que apresenta os dados mais atualizados sobre vitimização feminina, georreferenciados no pacote *tidygeocoder* e visualizados em um mapa de pontos utilizando o *leaflet* (Cambon et al, 2021). As opções de interatividade do pacote permitem que, ao passar o mouse sobre um dos pontos, uma caixa de informações seja exibida, contendo o nome da unidade da rede, o telefone para contato e o endereço. Também foi incluído um filtro de município, permitindo reduzir o número de pontos apresentados para o número selecionado de municípios.

Todos os painéis produzidos pelo ISP são finalizados com a apresentação de sua nota metodológica. Aqui são apresentados detalhes sobre os agrupamentos de delitos incluídos no painel, especificações sobre a maneira com que as contagens são feitas, visto que determinados delitos são contados por casos e outros, por vítimas e também são apresentadas outras referências relevantes que sejam necessárias.

Além do painel Mulher, o ISPConecta foi lançado com outros *dashboards* públicos: o Série Histórica e o Armas. O primeiro apresenta informações sobre os números observados de vítimas ou casos sobre cerca de 50 agrupamentos de delitos, incluindo comparativos de área e visualizações espaciais dos mesmos. Já o segundo apresenta os dados de apreensão de armas ocorridos, permitindo filtrar por diferentes categorias de armas (ex:

arma de fogo, arma branca, entre outras) e por espécie (pistola, revólver, entre outros dentro da categoria de armas de fogo). Ambos os *dashboards* são atualizações de outros mais antigos desenvolvidos em *Tableau*, porém trazendo modernizações e maiores possibilidades para os usuários.

Figura 5: Aba da rede de atendimento do painel mulher



Fonte: Elaborado pelo autor.

Conclusão

Ao longo dos anos, em sua missão de dar mais transparência aos dados de segurança pública do Rio de Janeiro, o Instituto de Segurança Pública vem desenvolvendo diversos produtos para tornar estas informações mais acessíveis para a população. O ISPConecta surge como uma alternativa de modernização de uma dessas iniciativas, porém baseado em soluções de *software* livre.

Tendo o *R* e o *Shiny* como pilares de sua construção, os novos painéis buscam não apenas repetir os papéis da infraestrutura anterior, mas sim expandir o que era possível ser feito. O servidor que hospeda o *Shiny* também foi configurado para armazenar um banco de dados que alimenta os painéis, estes foram construídos de modo que todos os seus menus consumam as informações do banco e se atualizem automaticamente uma vez que o painel seja carregado e consuma os dados do banco. Esta estrutura de atualização garante maior celeridade ao processo de divulgação de dados, pois uma vez que o banco seja atualizado todos os painéis também serão.

Deste modo, a adoção do *Shiny* garante não apenas maior economia de recursos uma vez que são utilizados apenas *softwares* livres e a própria infraestrutura do ISP, mas



também de tempo, uma vez que se tornou possível simplificar grande parte dos processos envolvidos na atualização e automatizá-los. A integração com o *Google Analytics* permite maior monitoramento dos painéis, permitindo saber quando, de onde e como foram acessados, permitindo gerar resultados de utilização que não estavam disponíveis na infraestrutura anterior.

Os três painéis lançados até então foram apenas o passo inicial de um projeto mais abrangente. Ainda existem painéis em *Tableau* que deverão ser atualizados para o *Shiny*, além de novos produtos que agora poderão ser lançados. Entre estes estão planejados painéis temáticos para delitos de trânsito, para crimes de letalidade violenta, voltados para grupos vulneráveis, entre outros, e também um *dashboard* para substituir a atual estrutura de pedidos de dados no qual os usuários poderão realizar consultas no banco de dados público do ISP de modo similar ao que é feito nos sistemas do DATASUS.

Referências

R CORE TEAM. **R: a language and environment for statistical computing**. R Foundation for Statistical Computing, Vienna, Austria: 2021.

WICKHAM, Hadley. FRANÇOIS, Romain. HENRY, Lionel. MULLER, Kirill & VAUGHAN, Davis. **dplyr: a grammar of data manipulation**. R package version 1.12: 2023.

WICKHAM, Hadley. VAUGHAN, Davis. GIRLICH, Maximillian. **tidyr: tidy messy data**. R package version 1.3.0: 2023.

GROLEMUND, Garrett. WICKHAM, Hadley. **Dates and times made easy with lubridate**. Journal of Statistical Software, 40 (3): 2011.

CHANG, Winston. CHENG, Joe. ALLAIRE, JJ. SIEVERT, Carson. SCHLOERKE, Barret. XIE, Yihui, ALLEN, Jeff. MCPHERSON, Jonathan. DIPERT, Alan. BORGES, Barbara. **Shiny: web application framework for R**. R package version 1.7.4: 2022.

WICKHAM, Hadley. **stringr: simple, consistent wrappers for common string operations**. R package version 1.5.0: 2022.

ALLAIRE, Joe et al. **rmarkdown: dynamic documents for R**. R package version 2.21: 2023.

R SPECIAL INTEREST GROUP ON DATABASES (R-SIG-DB). WICKHAM, Hadley. MULLER, Kirill. **DBI: R database interface**. R package version 1.1.1: 2021.

WICKHAM, Hadley. **httr: tools for working with urls and http**. R package version 1.4.7: 2023.



OOMS, Jeroen. **The jsonlite package: a practical and consistent mapping between json data and r objects**: 2014.

RIPLEY, Brian. LAPSLEY, Michael. **RODBC: odbc database access**. R package version 1.3-19: 2021.

CHENG, Joe. KARAMBELKAR, Bhaskar. XIE, Yuhui. **leaflet: create interactive web maps with the javascript leaflet library**. R package version 2.1.1: 2021.

GRANJON, David. **bs4Dash: a bootstrap4 version of shinydashboard**. R package version 2.1.0: 2022.

XIE, Yihui. CHENG, Joe. TAN, Xianying. **DT: a wrapper of the javascript library datatables**. R package version 0.1.9: 2021.

CAMBON, Jesse. HERNANGÓMEZ, Diego. BELANGER, Christopher. POSSENRIEDE, Daniel. **tidygeocoder: an R package for geocoding**. The Journal of Open Source Software, 6(65). 2021.

COENE, John. **echarts4r: Create Interactive Graphs with 'Echarts JavaScript' Version 5**. R package: 2022.

Instituto de Segurança Pública. 2024. **Institucional - Quem somos**. Disponível em: <<https://www.isp.rj.gov.br/node/649>>. Acesso em: 30 abr. 2024.

Instituto de Segurança Pública]. 2024. **ISP Dados Visualização**. Disponível em: <<https://www.ispvisualizacao.rj.gov.br/index.html>>. Acesso em 20 mar. 2024.

Anexo 1: Código do script server.R

```
shinyServer(function(input, output) {  
  
  base_sh = reactive({  
  
    if(input$sh_tipo_area == 'Estado'){  
  
      base %>%  
        filter(delito %in% input$sh_delitos) %>%  
        group_by(ano, mes) %>%  
        tally(n, name = 'Total')  
  
    } else  
  
    if(input$sh_tipo_area == 'Município'){
```



```
base %>%
```

```
  filter(delito %in% input$sh_delitos) %>%
```

```
  filter(fmun %in% input$sh_fmun) %>%
```

```
  group_by(ano, mes) %>%
```

```
  tally(n, name = 'Total')
```

```
} else
```

```
if(input$sh_tipo_area == 'AISP'){
```

```
  base %>%
```

```
    filter(delito %in% input$sh_delitos) %>%
```

```
    filter(aisp %in% input$sh_aisp) %>%
```

```
    group_by(ano, mes) %>%
```

```
    tally(n, name = 'Total')
```

```
} else
```

```
if(input$sh_tipo_area == 'RISP'){
```

```
  base %>%
```

```
    filter(delito %in% input$sh_delitos) %>%
```

```
    filter(risp %in% input$sh_risp) %>%
```

```
    group_by(ano, mes) %>%
```

```
    tally(n, name = 'Total')
```

```
} else
```

```
if(input$sh_tipo_area == 'CISP'){
```

```
  base %>%
```

```
    filter(delito %in% input$sh_delitos) %>%
```

```
    filter(circ %in% input$sh_cisp) %>%
```

```
    group_by(ano, mes) %>%
```

```
    tally(n, name = 'Total')
```

```
} else
```



```
if(input$sh_tipo_area == 'Região administrativa'){

  base %>%
    filter(delito %in% input$sh_delitos) %>%
    filter(regiao %in% input$sh_regiao) %>%
    group_by(ano, mes) %>%
    tally(n, name = 'Total')

}

})

output$sh_serie = renderEcharts4r({

  base_sh() %>%
  mutate(mes_ano = paste0(mes,'-',ano),
         mes_ano = my(mes_ano)) %>%
  ungroup() %>%
  select(mes_ano, Total) %>%
  e_chart(mes_ano) %>%
  e_line(Total) %>%
  e_tooltip(trigger = 'axis',
           formatter = e_tooltip_pointer_formatter(style = 'decimal', locale = 'pt-br')) %>%
  e_legend(show = F) %>%
  e_mark_point("Total", data = list(type = "min")) %>%
  e_mark_point("Total", data = list(type = "max")) %>%
  e_tooltip(trigger = 'axis',
           formatter = e_tooltip_pointer_formatter(style = 'decimal', locale = 'pt-br')) %>%
  e_y_axis(
    index = 1,
    axisLabel = list(formatter = e_axis_formatter(style = "decimal", locale = "pt-br"))
  ) %>%
  e_theme_custom('{"color":["#794f74"]}')

})
```



```
output$sh_total_mes = renderbs4InfoBox({

  if(input$sh_tipo_area == 'RISP'){

    validate(

      need(length(input$sh_risp) >= 1, "Selecione pelo menos uma RISP.")

    )

  } else if(input$sh_tipo_area == 'AISP'){

    validate(

      need(length(input$sh_aisp) >= 1, "Selecione pelo menos uma AISP.")

    )

  } else if(input$sh_tipo_area == 'CISP'){

    validate(

      need(length(input$sh_cisp) >= 1, "Selecione pelo menos uma CISP.")

    )

  } else if(input$sh_tipo_area == 'Município'){

    validate(

      need(length(input$sh_fmun) >= 1, "Selecione pelo menos um município.")

    )

  } else if(input$sh_tipo_area == 'Região administrativa'){

    validate(
```



```
    need(length(input$sh_regiao) >= 1, "Selecione pelo menos uma região.")

  )

}

a = base_sh() %>%
  mutate(data = ymd(paste(ano,'-',mes,'-01'))) %>%
  group_by(data) %>%
  tally(Total, name = 'Total') %>%
  filter(data == max(data))

mes = case_when(
  month(a$data) == 1 ~ 'Janeiro',
  month(a$data) == 2 ~ 'Fevereiro',
  month(a$data) == 3 ~ 'Março',
  month(a$data) == 4 ~ 'Abril',
  month(a$data) == 5 ~ 'Maio',
  month(a$data) == 6 ~ 'Junho',
  month(a$data) == 7 ~ 'Julho',
  month(a$data) == 8 ~ 'Agosto',
  month(a$data) == 9 ~ 'Setembro',
  month(a$data) == 10 ~ 'Outubro',
  month(a$data) == 11 ~ 'Novembro',
  month(a$data) == 12 ~ 'Dezembro',
)

texto = paste0('Vítimas em ', mes[1], ' de ', year(a$data[1]), ':')

bs4InfoBox(texto, prettyNum(a$Total[1], decimal.mark = ',', big.mark = '.', digits = 2,
                             scientific = F), width = 12,
            color = 'primary', fill = T, elevation = 1,
            icon = icon("venus"))
```



})

```
output$sh_total_mes_passado = renderbs4InfoBox({
```

```
  if(input$sh_tipo_area == 'RISP'){
```

```
    validate(
```

```
      need(length(input$sh_risp) >= 1, "Selecione pelo menos uma RISP.")
```

```
    )
```

```
  } else if(input$sh_tipo_area == 'AISP'){
```

```
    validate(
```

```
      need(length(input$sh_aisp) >= 1, "Selecione pelo menos uma AISP.")
```

```
    )
```

```
  } else if(input$sh_tipo_area == 'CISP'){
```

```
    validate(
```

```
      need(length(input$sh_cisp) >= 1, "Selecione pelo menos uma CISP.")
```

```
    )
```

```
  } else if(input$sh_tipo_area == 'Município'){
```

```
    validate(
```

```
      need(length(input$sh_fmun) >= 1, "Selecione pelo menos um município.")
```

```
    )
```



```
} else if(input$sh_tipo_area == 'Região administrativa'){

  validate(

    need(length(input$sh_regiao) >= 1, "Selecione pelo menos uma região.")

  )

}

a = base_sh() %>%
  mutate(data = ymd(paste(ano,'-',mes,'-01'))) %>%
  group_by(data) %>%
  tally(Total, name = 'Total') %>%
  filter(data == rollback(max(data - 365), roll_to_first = T) )

mes = case_when(
  month(a$data) == 1 ~ 'Janeiro',
  month(a$data) == 2 ~ 'Fevereiro',
  month(a$data) == 3 ~ 'Março',
  month(a$data) == 4 ~ 'Abril',
  month(a$data) == 5 ~ 'Maio',
  month(a$data) == 6 ~ 'Junho',
  month(a$data) == 7 ~ 'Julho',
  month(a$data) == 8 ~ 'Agosto',
  month(a$data) == 9 ~ 'Setembro',
  month(a$data) == 10 ~ 'Outubro',
  month(a$data) == 11 ~ 'Novembro',
  month(a$data) == 12 ~ 'Dezembro',
)

texto = paste0('Vítimas em ',mes[1],' de ',year(a$data[1]),":")

bs4InfoBox(texto,
  prettyNum(a$Total[1], decimal.mark = ',', big.mark = '.', digits = 2,
            scientific = F), width = 12,
  color = 'primary', fill = T, elevation = 1,
```



```
        icon = icon("venus"))

    })

    output$sh_total_mes_12 = renderbs4InfoBox({

    if(input$sh_tipo_area == 'RISP'){

        validate(

            need(length(input$sh_risp) >= 1, "Selecione pelo menos uma RISP.")

        )

    } else if(input$sh_tipo_area == 'AISP'){

        validate(

            need(length(input$sh_aisp) >= 1, "Selecione pelo menos uma AISP.")

        )

    } else if(input$sh_tipo_area == 'CISP'){

        validate(

            need(length(input$sh_cisp) >= 1, "Selecione pelo menos uma CISP.")

        )

    } else if(input$sh_tipo_area == 'Município'){

        validate(

            need(length(input$sh_fmun) >= 1, "Selecione pelo menos um município.")

        )

    }

    })
```



```
)

} else if(input$sh_tipo_area == 'Região administrativa'){

  validate(

    need(length(input$sh_regiao) >= 1, "Selecione pelo menos uma região.")

  )

}

a = base_sh() %>%
  mutate(data = ymd(paste(ano,'-',mes,'-01'))) %>%
  filter( (ano == year(max(.$data)) & mes == month(max(.$data))) | (ano ==
(year(max(.$data)) - 1) & mes == month(max(.$data)))) %>%
  group_by(data) %>%
  tally(Total, name = 'Total') %>%
  mutate(var = 100 * (Total/lag(Total) - 1),
    var = round(var, 1))

if(is.infinite(a$var[2]) == T){
  bs4InfoBox("Variação percentual entre os meses:", '-', width = 12,
    color = 'primary', fill = T, elevation = 1,
    icon = icon("calendar"))
} else {
  if(is.nan(a$var[2]) == T){
    bs4InfoBox("Variação percentual entre os meses:", '-', width = 12,
      color = 'primary', fill = T, elevation = 1,
      icon = icon("calendar"))
  } else {
    if( (((is.infinite(a$var[2]) == F & is.nan(a$var[2])) == F) & ( (a$Total[1] >= 50) == T ) &
((a$Total[2] >= 50) == T) ) == T) == T ){
      bs4InfoBox("Variação percentual entre os meses:", paste0(prettyNum(a$var[2],
big.mark = '.', decimal.mark = ','), "%"), width = 12,
        color = 'primary', fill = T, elevation = 1,
        icon = icon("calendar"))
    }
  }
}
```



```
    } else {
      bs4InfoBox("Variação percentual entre os meses:", '-', width = 12,
        color = 'primary', fill = T, elevation = 1,
        icon = icon("calendar"))
    }
  })

base_comp = reactive({

  if(input$comp_tipo_area == 'Estado'){

    base %>%
      filter(delito %in% input$comp_delitos) %>%
      mutate(periodo = case_when(

        data >= input$comp_data1[1] & data <= input$comp_data1[2] ~ 'Período 1',
        data >= input$comp_data2[1] & data <= input$comp_data2[2] ~ 'Período 2'

      )) %>%
      filter(!is.na(periodo)) %>%
      group_by(periodo, delito) %>%
      tally(n, name = 'Total')

  } else

  if(input$comp_tipo_area == 'Município'){

    base %>%
      filter(delito %in% input$comp_delitos) %>%
      filter(fmun %in% input$comp_fmfun) %>%
      mutate(periodo = case_when(

        data >= input$comp_data1[1] & data <= input$comp_data1[2] ~ 'Período 1',
```



```
data >= input$comp_data2[1] & data <= input$comp_data2[2] ~ 'Período 2'

)) %>%
filter(!is.na(periodo)) %>%
group_by(periodo, delito, fmun) %>%
tally(n, name = 'Total')

} else

if(input$comp_tipo_area == 'AISP'){

base %>%
filter(delito %in% input$comp_delitos) %>%
filter(aisp %in% input$comp_aisp) %>%
mutate(periodo = case_when(

data >= input$comp_data1[1] & data <= input$comp_data1[2] ~ 'Período 1',
data >= input$comp_data2[1] & data <= input$comp_data2[2] ~ 'Período 2'

)) %>%
filter(!is.na(periodo)) %>%
group_by(periodo, delito, aisp) %>%
tally(n, name = 'Total')

} else

if(input$comp_tipo_area == 'RISP'){

base %>%
filter(delito %in% input$comp_delitos) %>%
filter(risp %in% input$comp_risp) %>%
mutate(periodo = case_when(

data >= input$comp_data1[1] & data <= input$comp_data1[2] ~ 'Período 1',
data >= input$comp_data2[1] & data <= input$comp_data2[2] ~ 'Período 2'

)) %>%
```



```
filter(!is.na(periodo)) %>%
group_by(periodo, delito, risp) %>%
tally(n, name = 'Total')

} else

if(input$comp_tipo_area == 'CISP'){

base %>%
filter(delito %in% input$comp_delitos) %>%
filter(circ %in% input$comp_cisp) %>%
mutate(periodo = case_when(

data >= input$comp_data1[1] & data <= input$comp_data1[2] ~ 'Período 1',
data >= input$comp_data2[1] & data <= input$comp_data2[2] ~ 'Período 2'

)) %>%
filter(!is.na(periodo)) %>%
group_by(periodo, delito, circ) %>%
tally(n, name = 'Total')

} else

if(input$comp_tipo_area == 'Região administrativa'){

base %>%
filter(delito %in% input$comp_delitos) %>%
filter(regiao %in% input$comp_regiao) %>%
mutate(periodo = case_when(

data >= input$comp_data1[1] & data <= input$comp_data1[2] ~ 'Período 1',
data >= input$comp_data2[1] & data <= input$comp_data2[2] ~ 'Período 2'

)) %>%
filter(!is.na(periodo)) %>%
group_by(periodo, delito, regiao) %>%
tally(n, name = 'Total')
```



```
    }

  })

  output$comp_tabela = renderDataTable({

    validate(

      need(input$comp_delitos, "Selecione pelo menos um delito.")

    )

    validate(

      need(int_overlaps(interval(ymd(input$comp_data1[1]), ymd(input$comp_data1[2])),
interval(ymd(input$comp_data2[1]), ymd(input$comp_data2[2]))) == F, 'Os períodos
comparados não podem se sobrepor.')

    )

    if(input$comp_tipo_area == 'RISP'){

      validate(

        need(length(input$comp_risp) >= 1, "Selecione pelo menos uma RISP.")

      )

    } else if(input$comp_tipo_area == 'AISP'){

      validate(

        need(length(input$comp_aisp) >= 1, "Selecione pelo menos uma AISP.")

      )

    }

  })

}
```



```
} else if(input$comp_tipo_area == 'CISP'){

  validate(

    need(length(input$comp_cisp) >= 1, "Selecione pelo menos uma CISP.")

  )

} else if(input$comp_tipo_area == 'Município'){

  validate(

    need(length(input$comp_fmun) >= 1, "Selecione pelo menos um município.")

  )

} else if(input$comp_tipo_area == 'Região administrativa'){

  validate(

    need(length(input$comp_regiao) >= 1, "Selecione pelo menos uma região.")

  )

}

if(input$comp_tipo_area == 'Estado'){

  base_comp() %>%
  pivot_wider(id_cols = c('delito'), names_from = periodo, values_from = Total) %>%
  mutate(dif = `Período 2` - `Período 1`,
         var = `Período 2` / `Período 1` - 1) %>%
  mutate(var = ifelse(`Período 2` >= 50 & `Período 1` >= 50, var, NA)) %>%
  arrange(desc(`Período 2`)) %>%
  datatable(rownames = F,
            options = list(dom = 'tp',
```



```
language = list(url =
'//cdn.datatables.net/plug-ins/1.10.11/i18n/Portuguese.json')),
colnames = c('Delito', 'Período 1', 'Período 2', 'Diferença Absoluta', 'Diferença
Percentual')) %>%
formatPercentage(5, digits = 1, mark = '.', dec.mark = ',')

} else

if(input$comp_tipo_area == 'Município'){

base_comp() %>%
pivot_wider(id_cols = c('fmun', 'delito'), names_from = periodo, values_from = Total)
%>%
mutate(dif = `Período 2` - `Período 1`,
var = `Período 2` / `Período 1` - 1) %>%
mutate(var = ifelse(`Período 2` >= 50 & `Período 1` >= 50, var, NA)) %>%
arrange(desc(`Período 2`)) %>%
datatable(rownames = F,
options = list(dom = 'tp',

language = list(url =
'//cdn.datatables.net/plug-ins/1.10.11/i18n/Portuguese.json')),
colnames = c('Município', 'Delito', 'Período 1', 'Período 2', 'Diferença Absoluta',
'Diferença Percentual')) %>%
formatPercentage(6, digits = 1, mark = '.', dec.mark = ',')

} else

if(input$comp_tipo_area == 'AISP'){

base_comp() %>%
pivot_wider(id_cols = c('aisp', 'delito'), names_from = periodo, values_from = Total)
%>%
mutate(dif = `Período 2` - `Período 1`,
var = `Período 2` / `Período 1` - 1) %>%
mutate(var = ifelse(`Período 2` >= 50 & `Período 1` >= 50, var, NA)) %>%
```



```
    arrange(desc(`Período 2`)) %>%
    datatable(rownames = F,
              options = list(dom = 'tp',
                              language = list(url =
//cdn.datatables.net/plug-ins/1.10.11/i18n/Portuguese.json')),
              colnames = c('AISP', 'Delito', 'Período 1', 'Período 2', 'Diferença Absoluta',
'Diferença Percentual')) %>%
    formatPercentage(6, digits = 1, mark = '.', dec.mark = ',')

} else

if(input$comp_tipo_area == 'RISP'){

    base_comp() %>%
    pivot_wider(id_cols = c('risp', 'delito'), names_from = periodo, values_from = Total)
%>%
    mutate(dif = `Período 2` - `Período 1`,
           var = `Período 2` / `Período 1` - 1) %>%
    mutate(var = ifelse(`Período 2` >= 50 & `Período 1` >= 50, var, NA)) %>%
    arrange(desc(`Período 2`)) %>%
    datatable(rownames = F,
              options = list(dom = 'tp',
                              language = list(url =
//cdn.datatables.net/plug-ins/1.10.11/i18n/Portuguese.json')),
              colnames = c('RISP', 'Delito', 'Período 1', 'Período 2', 'Diferença Absoluta',
'Diferença Percentual')) %>%
    formatPercentage(6, digits = 1, mark = '.', dec.mark = ',')

} else

if(input$comp_tipo_area == 'CISP'){

    base_comp() %>%
```



```
        pivot_wider(id_cols = c('circ', 'delito'), names_from = periodo, values_from = Total)
%>%
mutate(dif = `Período 2` - `Período 1`,
       var = `Período 2`/`Período 1` - 1) %>%
mutate(var = ifelse(`Período 2` >= 50 & `Período 1` >= 50, var, NA)) %>%
arrange(desc(`Período 2`)) %>%
datatable(rownames = F,
          options = list(dom = 'tp',
                        language = list(url =
//cdn.datatables.net/plug-ins/1.10.11/i18n/Portuguese.json')),
          colnames = c('CISP', 'Delito', 'Período 1', 'Período 2', 'Diferença Absoluta',
'Diferença Percentual')) %>%
formatPercentage(6, digits = 1, mark = '.', dec.mark = ',')

} else

if(input$comp_tipo_area == 'Região administrativa'){

base_comp() %>%
pivot_wider(id_cols = c('regiao', 'delito'), names_from = periodo, values_from =
Total) %>%
mutate(dif = `Período 2` - `Período 1`,
       var = `Período 2`/`Período 1` - 1) %>%
mutate(var = ifelse(`Período 2` >= 50 & `Período 1` >= 50, var, NA)) %>%
arrange(desc(`Período 2`)) %>%
datatable(rownames = F,
          options = list(dom = 'tp',
                        language = list(url =
//cdn.datatables.net/plug-ins/1.10.11/i18n/Portuguese.json')),
          colnames = c('Região', 'Delito', 'Período 1', 'Período 2', 'Diferença Absoluta',
'Diferença Percentual')) %>%
formatPercentage(6, digits = 1, mark = '.', dec.mark = ',')

}
```



})

output\$mapa_cisp = renderLeaflet({

if(input\$mapa_tipo_area == 'CISP'){

teste = base %>%

mutate(data = ymd(paste(ano,'-',mes,'-01')) %>%

filter(data >= input\$mapa_data[1] & data <= input\$mapa_data[2]) %>%

filter(delito == input\$mapa_delitos) %>%

dplyr::select(circ, n) %>%

group_by(circ) %>%

tally(n, name = 'total')

shp_cisp\$dp = factor(shp_cisp\$dp)

shp_cisp = left_join(shp_cisp, teste, by = c('dp' = 'circ'))

pal <- colorNumeric(

palette = c(

"#f1eef6",

"#d7b5d8",

"#df65b0",

"#dd1c77",

"#980043"),

domain = shp_cisp\$total)

labels <- sprintf(

"CISP %s:
%g ",

shp_cisp\$dp, shp_cisp\$total

) %>% lapply(htmltools::HTML)

leaflet(shp_cisp) %>%

addProviderTiles(providers\$CartoDB.Positron,

options = providerTileOptions(minZoom = 8, maxZoom = 13)) %>%

addPolygons(

fillColor = ~pal(total),



```
weight = 2,
opacity = 1,
color = "black",
dashArray = "3",
fillOpacity = 0.6,
label = labels) %>%
addLegend(pal = pal, values = ~total, opacity = 0.7, title = NULL,
position = "bottomright")

} else

if(input$mapa_tipo_area == 'AISP'){

teste = base %>%
mutate(data = ymd(paste(ano,'-',mes,'-01'))) %>%
filter(data >= input$mapa_data[1] & data <= input$mapa_data[2]) %>%
filter(delito == input$mapa_delitos) %>%
dplyr::select(aisp, n) %>%
group_by(aisp) %>%
tally(n, name = 'total')

shp_aisp = left_join(shp_aisp, teste, by = c('aisp' = 'aisp'))

pal <- colorNumeric(
palette = c(
"#f1eef6",
"#d7b5d8",
"#df65b0",
"#dd1c77",
"#980043"),
domain = shp_aisp$total)

labels <- sprintf(
"<strong>AISP %s:</strong><br/>%g ",
shp_aisp$aisp, shp_aisp$total
) %>% lapply(htmltools::HTML)
```



```
leaflet(shp_aisp) %>%
  addProviderTiles(providers$CartoDB.Positron,
    options = providerTileOptions(minZoom = 8, maxZoom = 13)) %>%
  addPolygons(
    fillColor = ~pal(total),
    weight = 2,
    opacity = 1,
    color = "black",
    dashArray = "3",
    fillOpacity = 0.6,
    label = labels) %>%
  addLegend(pal = pal, values = ~total, opacity = 0.7, title = NULL,
    position = "bottomright")

} else

if(input$mapa_tipo_area == 'RISP'){

  teste = base %>%
    mutate(data = ymd(paste(ano,'-',mes,'-01'))) %>%
    filter(data >= input$mapa_data[1] & data <= input$mapa_data[2]) %>%
    filter(delito == input$mapa_delitos) %>%
    dplyr::select(risp, n) %>%
    group_by(risp) %>%
    tally(n, name = 'total')

  shp_risp$risp = as.numeric(shp_risp$risp)

  shp_risp = left_join(shp_risp, teste, by = c('risp' = 'risp'))

  pal <- colorNumeric(
    palette = c(
      "#f1eef6",
      "#d7b5d8",
      "#df65b0",
      "#dd1c77",
      "#980043"),
```



```
domain = shp_risp$total)

labels <- sprintf(
  "<strong>RISP %s:</strong><br/>%g ",
  shp_risp$risp, shp_risp$total
) %>% lapply(htmltools::HTML)

leaflet(shp_risp) %>%
  addProviderTiles(providers$CartoDB.Positron,
    options = providerTileOptions(minZoom = 8, maxZoom = 13)) %>%
  addPolygons(
    fillColor = ~pal(total),
    weight = 2,
    opacity = 1,
    color = "black",
    dashArray = "3",
    fillOpacity = 0.6,
    label = labels) %>%
  addLegend(pal = pal, values = ~total, opacity = 0.7, title = NULL,
    position = "bottomright")

} else

if(input$mapa_tipo_area == 'Município'){

  teste = base %>%
    mutate(data = ymd(paste(ano,'-',mes,'-01')))) %>%
    filter(data >= input$mapa_data[1] & data <= input$mapa_data[2]) %>%
    filter(delito == input$mapa_delitos) %>%
    dplyr::select(fmun_cod, n) %>%
    group_by(fmun_cod) %>%
    tally(n, name = 'total')

  shp_fmuni = left_join(shp_fmuni, teste, by = c('code_muni' = 'fmuni_cod'))

  pal <- colorNumeric(
```



```
palette = c(
  "#f1eef6",
  "#d7b5d8",
  "#df65b0",
  "#dd1c77",
  "#980043"),
domain = log(shp_fmun$total+0.1))

labels <- sprintf(
  "<strong>%s:</strong><br/>%g ",
  shp_fmun$name_muni, shp_fmun$total
) %>% lapply(htmltools::HTML)

leaflet(shp_fmun) %>%
  addProviderTiles(providers$CartoDB.Positron,
    options = providerTileOptions(minZoom = 8, maxZoom = 13)) %>%
  addPolygons(
    fillColor = ~pal(log(total+0.1)),
    weight = 2,
    opacity = 1,
    color = "black",
    dashArray = "3",
    fillOpacity = 0.6,
    label = labels) %>%
  addLegend(pal = pal, values = ~log(total+0.1), opacity = 0.7, title = NULL,
    position = "bottomright", bins = 10,
    labFormat = labelFormat(transform = function(x) unique(round(exp(x),0))))

}

})

output$mapa_rede = renderLeaflet({

  validate(
```



```
    need(length(input$cref_fmfun) >= 1, 'Selecione pelo menos um município.')
```

)

```
    cref %>%
    filter(fmfun %in% input$cref_fmfun) %>%
    leaflet() %>%
    addProviderTiles(
      "OpenStreetMap",
      group = "OpenStreetMap"
    ) %>%
    addAwesomeMarkers( lng = ~longitude, lat = ~latitude,
      label = ~label,
      icon = awesomeIcons(
        icon = 'venus',
        library = 'fa',
        markerColor = "pink",
        iconColor = "#fff"
      )
    )
  })
})
```

Anexo 2: Código do script ui.R

```
addResourcePath(prefix = "www", directoryPath = "www")

ui = dashboardPage(
  dark = NULL,
  title = "Mulher",
  header = dashboardHeader(title = dashboardBrand(title = 'ISPMulher'), status = 'primary',
border = F),
  sidebar = bs4DashSidebar(minified = F, skin = 'light', elevation = 1,
    collapsed = F, width = 300, status = 'primary',
    tags$hr()),
```



```
tags$div('Painel desenvolvido pelo ISP para apresentar informações sobre
delitos cometidos contra mulheres.', style = 'text-align:justify'),
tags$hr(),
sidebarMenu(
  menuItem("Série histórica", tabName = "sh"),
  menuItem("Comparativo", tabName = 'comp'),
  menuItem("Distribuição espacial", tabName = 'espacial'),
  menuItem("Rede de atendimento", tabName = 'rede'),
  menuItem("Nota metodológica", tabName = 'nota')
),
tags$hr(),
tags$div(tags$img(src='www/mulhisp_2.png', align = "center", width
="200px"),
align = "center", position = "absolute", bottom = 0)),
controlbar = NULL,
footer = bs4DashFooter(left = "Painel criado, desenvolvido e atualizado pelo Instituto de
Segurança Pública. Vedada a reprodução sem autorização.",
right = data_att, fixed = T),
body = dashboardBody(
  use_theme(create_theme(
    bs4dash_status(
      primary = "#B46EAB"
    )
  )),
tags$head(includeHTML("google_analytics.html")),
tags$head(
  tags$style(HTML('
    .brand-text { font-weight:700 !important;
    font-size:20px !important;
    color: #FFFFFF;
  })
  .layout-fixed .brand-link {
    text-align: center;
    background-color: #B46EAB;
  }
  .info-box .info-box-number{
```



```
font-size: 1.25rem;
}
))),
tabItems(
  tabItem(tabName = 'sh',
    fluidRow(
      box(width = 4, collapsible = F, solidHeader = F,
        headerBorder = F,
        pickerInput(
          "sh_delitos",
          'Delitos:',
          choices = lista_delitos,
          selected = 'Feminicídio',
          multiple = F,
          options = list(`dropdown-align-right` = TRUE,
            `container` = "body",
            `live-search` = T,
            `actions-box` = TRUE,
            `count-selected-text` = "Todos")
        )
      ),
      box(width = 4, collapsible = F, solidHeader = F,
        headerBorder = F,
        pickerInput('sh_tipo_area',
          'Tipo de região:',
          choices = c('Estado', 'Município', 'RISP', 'AISP', 'CISP', 'Região
administrativa'),
          selected = 'Estado', multiple = F)
      ),
      box(width = 4, collapsible = F, solidHeader = F,
        headerBorder = F, height=126,
        conditionalPanel(
          condition = "input.sh_tipo_area == 'Região administrativa'",
          pickerInput(
            "sh_regiao",
            label = 'Região administrativa:',
            choices = lista_reg,
```



```
selected = lista_reg,
multiple = T,
options = list(`dropdown-align-right` = TRUE,
              `container` = "body",
              `live-search` = T,
              `actions-box` = TRUE,
              `select-all-text` = "Todas",
              `deselect-all-text` = 'Nenhuma',
              `selected-text-format` = paste0("count > ", 6),
              `count-selected-text` = "Todas")
)
),
conditionalPanel(
  condition = "input.sh_tipo_area == 'RISP'",
  pickerInput(
    "sh_risp",
    label = 'RISP:',
    choices = 1:7,
    selected = 1:7,
    multiple = T,
    options = list(`dropdown-align-right` = TRUE,
                  `container` = "body",
                  `live-search` = T,
                  `actions-box` = TRUE,
                  `select-all-text` = "Todas",
                  `deselect-all-text` = 'Nenhuma',
                  `selected-text-format` = paste0("count > ", 6),
                  `count-selected-text` = "Todas")
  )
),
conditionalPanel(
  condition = "input.sh_tipo_area == 'AISP'",
  pickerInput(
    "sh_aisp",
    label = 'AISP:',
    choices = lista_aisp,
    selected = lista_aisp,
```



```
multiple = T,
options = list(`dropdown-align-right` = TRUE,
              `container` = "body",
              `live-search` = T,
              `actions-box` = TRUE,
              `select-all-text` = "Todas",
              `deselect-all-text` = 'Nenhuma',
              `selected-text-format` = paste0("count > ", 38),
              `count-selected-text` = "Todas")
)
),
conditionalPanel(
  condition = "input.sh_tipo_area == 'CISP'",
  pickerInput(
    "sh_cisp",
    label = 'CISP:',
    choices = lista_cisp,
    selected = lista_cisp,
    multiple = T,
    options = list(`dropdown-align-right` = TRUE,
                  `container` = "body",
                  `live-search` = T,
                  `actions-box` = TRUE,
                  `select-all-text` = "Todas",
                  `deselect-all-text` = 'Nenhuma',
                  `selected-text-format` = paste0("count > ", 137),
                  `count-selected-text` = "Todas")
  )
),
conditionalPanel(
  condition = "input.sh_tipo_area == 'Município'",
  pickerInput(
    "sh_fmum",
    label = 'Município:',
    choices = lista_fmum,
    selected = lista_fmum,
    multiple = T,
```



```
options = list(`dropdown-align-right` = TRUE,
              `container` = "body",
              `live-search` = T,
              `actions-box` = TRUE,
              `select-all-text` = "Todas",
              `deselect-all-text` = 'Nenhuma',
              `selected-text-format` = paste0("count > ", 91),
              `count-selected-text` = "Todos")
)
)
),
fluidRow(
  bs4InfoBoxOutput("sh_total_mes"),
  bs4InfoBoxOutput("sh_total_mes_passado"),
  bs4InfoBoxOutput("sh_total_mes_12")
)
,
fluidRow(
  box(collapsible = T, solidHeader = T,
      headerBorder = F, status = 'primary',
      width = 12, height = 450,
      title = 'Série histórica',
      echarts4rOutput("sh_serie"))
)),
tabItem(tabName = "comp",
  fluidRow(
    box(width = 2, collapsible = F, solidHeader = F,
      headerBorder = F,
      pickerInput(
        "comp_delitos",
        'Delitos:',
        choices = lista_delitos,
        selected = lista_delitos,
        multiple = T,
        options = list(`dropdown-align-right` = TRUE,
                      `container` = "body",
```




```
condition = "input.comp_tipo_area == 'RISP'",
pickerInput(
  "comp_risp",
  label = 'RISP:',
  choices = 1:7,
  selected = 1:7,
  multiple = T,
  options = list(`dropdown-align-right` = TRUE,
    `container` = "body",
    `live-search` = T,
    `actions-box` = TRUE,
    `select-all-text` = "Todas",
    `deselect-all-text` = 'Nenhuma',
    `selected-text-format` = paste0("count > ", 6),
    `count-selected-text` = "Todas")
)
),
conditionalPanel(
  condition = "input.comp_tipo_area == 'AISP'",
  pickerInput(
    "comp_aisp",
    label = 'AISP:',
    choices = lista_aisp,
    selected = lista_aisp,
    multiple = T,
    options = list(`dropdown-align-right` = TRUE,
      `container` = "body",
      `live-search` = T,
      `actions-box` = TRUE,
      `select-all-text` = "Todas",
      `deselect-all-text` = 'Nenhuma',
      `selected-text-format` = paste0("count > ", 38),
      `count-selected-text` = "Todas")
    )
  ),
conditionalPanel(
  condition = "input.comp_tipo_area == 'CISP'",
```



```
pickerInput(  
  "comp_cisp",  
  label = 'CISP:',  
  choices = lista_cisp,  
  selected = lista_cisp,  
  multiple = T,  
  options = list(`dropdown-align-right` = TRUE,  
    `container` = "body",  
    `live-search` = T,  
    `actions-box` = TRUE,  
    `select-all-text` = "Todas",  
    `deselect-all-text` = 'Nenhuma',  
    `selected-text-format` = paste0("count > ", 137),  
    `count-selected-text` = "Todas")  
  )  
,  
conditionalPanel(  
  condition = "input.comp_tipo_area == 'Município'",  
  pickerInput(  
    "comp_fmfun",  
    label = 'Município:',  
    choices = lista_fmfun,  
    selected = lista_fmfun,  
    multiple = T,  
    options = list(`dropdown-align-right` = TRUE,  
      `container` = "body",  
      `live-search` = T,  
      `actions-box` = TRUE,  
      `select-all-text` = "Todos",  
      `deselect-all-text` = 'Nenhuma',  
      `selected-text-format` = paste0("count > ", 91),  
      `count-selected-text` = "Todos")  
    )  
  )  
,  
box(collapsible = F, solidHeader = F,  
  headerBorder = F,
```



```
width = 3,
airMonthpickerInput("comp_data1", "Período 1:",
                    value = c("2023-01-01", "2023-02-01"),
                    language = 'pt-BR', clearButton = T,
                    update_on = 'close', autoClose = T,
                    toggleSelected = F, range = T,
                    view = "months",
                    minView = "months",
                    dateFormat = "MM-yyyy",
                    minDate = '2018-01-01', maxDate = '2024-02-01')
),
box(collapsible = F, solidHeader = F,
    headerBorder = F,
    width = 3,
    airMonthpickerInput("comp_data2", "Período 2:", range = T,
                        value = c("2024-01-01", "2024-02-01"),
                        language = 'pt-BR', clearButton = T,
                        update_on = 'close', autoClose = T,
                        toggleSelected = F, range = T,
                        view = "months",
                        minView = "months",
                        dateFormat = "MM-yyyy",
                        minDate = '2018-01-01', maxDate = '2024-02-01')
    )
),
fluidRow(
  box(collapsible = T, solidHeader = T,
      headerBorder = F, status = 'primary',
      width = 12,
      title = 'Comparativo',
      dataTableOutput("comp_tabela"))
  ),
tabItem(tabName = "especial",
  fluidRow(
    box(width = 4, collapsible = F, solidHeader = F,
        headerBorder = F,
```



```
pickerInput(  
  "mapa_delitos",  
  'Delitos:',  
  choices = lista_delitos,  
  selected = 'Feminicídio',  
  multiple = F,  
  options = list(`dropdown-align-right` = TRUE,  
    `container` = "body",  
    `live-search` = T,  
    `actions-box` = TRUE,  
    `select-all-text` = "Todos",  
    `deselect-all-text` = 'Nenhum',  
    `selected-text-format` = paste0("count > ", 16),  
    `count-selected-text` = "Todos")  
)  
,  
box(collapsible = F, solidHeader = F,  
  headerBorder = F,  
  width = 4,  
  airMonthpickerInput("mapa_data", "Período:", range = T,  
    value = c('2024-01-01', '2024-02-01'),  
    language = 'pt-BR', clearButton = T,  
    update_on = 'close', autoClose = T,  
    toggleSelected = F,  
    minDate = '2018-01-01',  
    maxDate = '2024-02-01')  
),  
box(collapsible = F, solidHeader = F,  
  headerBorder = F,  
  width = 4,  
  pickerInput(  
    "mapa_tipo_area",  
    'Área:',  
    choices = c('Município', 'RISP', "AISP", 'CISP'),  
    selected = 'Município'  
  ))  
,
```



```
fluidRow(  
  box(  
    collapsible = F, solidHeader = T,  
    headerBorder = T, status = "primary",  
    width = 12, title = "Distribuição espacial",  
    withSpinner(leafletOutput("mapa_cisp", height = 550), type = 4, color =  
"#B46EAB", size = 1)  
  )  
)  
,  
tabItem(tabName = "rede",  
  fluidRow(  
    box(collapsible = F, solidHeader = F,  
      headerBorder = F,  
      width = 4,  
      pickerInput(  
        "cref_fmum",  
        'Município:',  
        choices = lista_fmum_cref,  
        selected = lista_fmum_cref,  
        multiple = T,  
        options = list(`dropdown-align-right` = TRUE,  
          `container` = "body",  
          `live-search` = T,  
          `actions-box` = TRUE,  
          `select-all-text` = "Todos",  
          `deselect-all-text` = 'Nenhum',  
          `selected-text-format` = paste0("count > ", 26),  
          `count-selected-text` = "Todos")  
        ))  
  ),  
  fluidRow(  
    box(  
      collapsible = F, solidHeader = T,  
      headerBorder = T, status = "primary",  
      width = 12, title = "Rede de atendimento",
```



```
withSpinner(leafletOutput("mapa_rede", height = 520), type = 4, color = "#B46EAB",
size = 1)
)
),
tabItem(tabName = "nota",
box(collapsible = F, solidHeader = F,
headerBorder = F,
width = 12,
tags$div("Ao longo dos seus 25 anos, o Instituto de Segurança Pública vem
dedicando esforços para a publicização ininterrupta dos dados referentes à violência contra
a mulher no estado do Rio de Janeiro. Dessa forma, esperamos colaborar para a
estruturação de políticas públicas transversais e a ampliação da visibilidade e do
entendimento deste grave fenômeno social: a violência contra a mulher.", style =
'text-align:justify'),
tags$br(),
tags$div("A primeira aba do painel apresenta as séries históricas dos delitos
desde o nível estadual a desagregações por município, região administrativa, RISP, AISP e
CISP. Na segunda aba é possível estabelecer os comparativos entre dois períodos distintos
para os delitos e regiões selecionadas. Em seguida, a terceira aba traz a distribuição
espacial dos delitos com a possibilidade de visualização de um mapa coroplético nas mais
diversas divisões administrativas do estado. Por fim, o painel apresenta um mapa que exhibe
a rede de atendimento para mulheres vítimas de violência, trazendo a posição aproximada
destes locais, seu endereço e seu contato.
", style = 'text-align:justify'),
tags$br(),
tags$div("Nota 1: As contagens apresentadas neste painel foram obtidas através
dos registros de ocorrência provenientes da Secretaria de Estado de Polícia Civil (SEPOL)
do Rio de Janeiro, conforme previsão do art. 3º Decreto nº 36.872, de 17 de janeiro de
2005.", style = 'text-align:justify'),
tags$br(),
tags$div("Nota 2: Para a contabilização das titulações foram utilizados os
números de vítimas do sexo feminino para os seguintes agrupamentos de delitos: homicídio
doloso, tentativa de homicídio, constrangimento ilegal, feminicídio, tentativa de feminicídio,
assédio sexual, difamação e violação de domicílio. A definição destes delitos compete a
autoridade policial da SEPOL.", style = 'text-align:justify'),
tags$br(),
```



tags\$div("Nota 3: A Rede de Atendimentos exibida no painel foi extraída do
Dossiê Mulher 2023. Acessível em:
isp.rj.gov.br/sites/default/files/2023-11/DossieMulher2023.pdf", style = 'text-align:justify'),

tags\$br(),

tags\$div("Nota 4: A periodicidade dos dados é trimestral, após serem feitas as
erratas de acordo com a metodologia do ISP. Atualmente os dados possuem erratas até
setembro de 2023.", style = 'text-align:justify'),

tags\$br(),

tags\$div("Nota 5: As desagregações administrativas de Região Integrada de
Segurança Pública (RISP), Área Integrada de Segurança Pública (AISP) e Circunscrição
Integrada de Segurança Pública (CISP), estão definidas no seguinte endereço:
ispdados.rj.gov.br/divisaoTerritorial.html", style = 'text-align:justify'))

)

)

)

)