

Uma Proposta de Integração para um RDA de Baixo Custo para Prevenção de Fraudes em Operações de Crédito

Felipe Nathan S. dos Santos^{1,2}, André Marcos Silva¹

¹Centro Universitário Campo Limpo Paulista (UNIFACCAMP)
Jardim América, CEP 13231-230, Campo Limpo Paulista, SP – Brasil

²Fidelity National Serviços e Contact Center LTDA
Processadora e Serviços, Vila Santana II, CEP 13219-001 Jundiaí, SP – Brasil

felipenathan29@gmail.com, andre@faccamp.br

Abstract. *This work proposes the development of an RDA (Robotic Desktop Automation) that unifies multiple interfaces into an integrated tool. The solution aims to provide customer service employees with a centralized application, bringing together all the necessary information for analysis and decision-making in the credit fraud identification process. Currently, hundreds of employees dedicate significant time to this task, but with the implementation of this unified portal, it is expected that user tasks and views will be centralized in a more intuitive, secure, productive, and less exhausting manner.*

Resumo. *Este trabalho propõe o desenvolvimento de um RDA (Robotic Desktop Automation) que unifica várias interfaces em uma ferramenta integrada. A solução visa fornecer aos colaboradores de atendimento uma aplicação centralizada, reunindo todas as informações necessárias para a análise e tomada de decisão no processo de identificação de fraudes em operações de crédito. Atualmente, centenas de colaboradores dedicam um tempo significativo a essa tarefa, mas espera-se que, com a implementação desse portal único, as tarefas e visualizações dos usuários sejam centralizadas de forma mais intuitiva, segura, produtiva e menos exaustiva.*

1. Introdução

A análise de fraude em cartões de crédito tem como objetivo identificar e prevenir transações fraudulentas, protegendo tanto os consumidores quanto as instituições financeiras contra perdas financeiras. Esse processo visa detectar padrões anômalos em transações, que possam indicar atividades fraudulentas, como o uso não autorizado de cartões, clonagem, ou transações suspeitas. O objetivo final é minimizar os impactos financeiros da fraude e melhorar a segurança das transações eletrônicas (Martins e Glegale, 2022).

Por várias outras dificuldades, entre elas, constante evolução das técnicas de engenharia social, mudanças frequentes de leis de privacidade e evolução de meios tecnológicos (Souza e Bordin, 2023), temos também as questões relacionadas à características analógicas do processo (Silva e Lauers, 2015). Por questões legais e comportamento subjetivo do comportamento humano, é necessário que este processo tenha uma etapa de intervenção humana inserido no processo decisório. Este cenário faz com que este tipo de sistema também herde desafios relacionados ao usuário operador do sistema, tais como, usabilidade, satisfação, fadiga, segurança e produtividade.

De outra forma, no campo da tecnologia, um RDA (do inglês, Robotic Desktop Automation), ou automação robótica de processos no nível de *desktop*, é uma maneira acessível de ajudar sua organização a passar por uma transformação digital e aproveitar os benefícios dos processos de negócios automatizados (Automation Anywhere, 2024).

Segundo IBM (2021), uma solução baseada em RDA pode impulsionar a produtividade, expandir negócios, reduzir o *time to value*, aumentar rendimentos e reduzir erros humanos; ou seja, esta tecnologia pode permitir a melhoria direta do processo de identificação de fraudes em operações enquanto ainda permite a interferência humana na camada de decisão do ciclo (Wattenberg, 2019).

2. Objetivos e Metodologia

Este artigo apresenta uma proposta e o desenvolvimento de um sistema baseado em Robotic Desktop Automation (RDA) nomeado como Tela Única, projetado para integrar três sistemas distintos em um portal unificado. O estudo de caso, tomado como laboratório real para o experimento, concentra suas atividades na área financeira de pagamento de créditos, mais especificamente, no setor de identificação e tratamento de possíveis fraudes. Basicamente, a proposta é uma ferramenta que permite aos colaboradores acessar rapidamente as informações necessárias para tomar decisões, realizar manutenções no sistema e categorizar os atendimentos.

A metodologia adotada envolveu a integração com processos de *mainframe* através da troca de arquivos do tipo texto, interação com sistemas web fechados via *scraping*, e comunicação humana através de plataformas de mensageria. Para este cenário proposto, o projeto utiliza uma combinação de tecnologias modernas: Python para o backend, Selenium para *web scraping*, e Bootstrap, HTML, Javascript e CSS para a construção do *frontend*. O banco de dados é gerido pelo SQL Server, garantindo robustez e escalabilidade.

Além da problemática envolvida no domínio tomado como estudo e caso, este trabalho se concentra nos desafios técnicos uma vez que se propõe encontrar uma solução pertinente ao contexto mas sem lançar mão da inovação e criatividade técnica alinhado com a obtenção de um produto confiável e produtivo para uma linha de operação real, por exemplo, dependência de integração com sistemas, legados, fechados e de terceiros, envolvendo principalmente limitações técnicas, disponibilidade de calendário, falta de interesse e alto custo envolvido para mudanças.

3. Projeto

O setor responsável pelo processo de prevenção a fraudes é composto por aproximadamente 135 colaboradores diretos, que juntos contribuem com um volume médio de 83 mil ligações por mês, e cada atendimento possuindo com um tempo médio de conexão de 400 segundos (por volta de 6.5 minutos por contato telefônico).

Após a etapa de análise (Tabela 1) e estudo de viabilidade junto dos especialistas da área, foi mapeado que com a automação das tarefas previstas no projeto, o processo ganharia teria um ganho previsto de redução de até 10% no total de colaboradores, já convertendo os benefícios de produtividade por quantidade de funcionários necessários, por tempo de atendimento. Outro benefício seria relacionado com o processo de

treinamento, uma única ferramenta simplificaria drasticamente o volume de informações necessárias para serem aprendidas pelos novos colaboradores.

Tabela 1. Mapeamento do Processo Atual

ESPECIFICAÇÃO	QUANTIDADE	DESCRIÇÃO
processo (caso de uso)	1	prevenção a fraudes
atores	2	usuário direto do sistema e cliente do banco (em chamada telefônica)
usuários (colaboradores)	135	funcionários colaboradores contratados e terceirizados
funcionalidade de software	9	buscar histórico de operações, CRUD de informações de cliente, alterar dados, bloqueio de cartão, enviar código confirmação e encerrar processo
integrações	3	mainframe, sistema web fechado, e sistema de mensagem (confirmações)
interferências manuais externas	1	cliente bancário por chamada de telefone

A figura 1.a demonstra o fluxo básico das principais etapas que compõem o processo de validação e tomada de decisão em uma possível análise de suspeita de fraude. E mesmo no projeto do sistema unificador, este fluxo deve ser mantido, uma vez que respeita um protocolo já definido e aprovado no domínio do sistema.

A proposta do novo sistema deve estar focado no desenvolvimento do portal que vai agrupar estas etapas deste processo. Ou seja, ao invés do colaborador acessar diretamente cada um dos sistemas para a realização do processo, conforme ator “colaborador 1” apresentado na figura 1.b, ele precisará, acessar somente uma única interface, que além de automatizar as diferentes autenticações nos devidos sistemas, ele também terá, de forma transparente, a automação das consultas, preparação dos dados, atualizações e algumas tomadas de decisão.

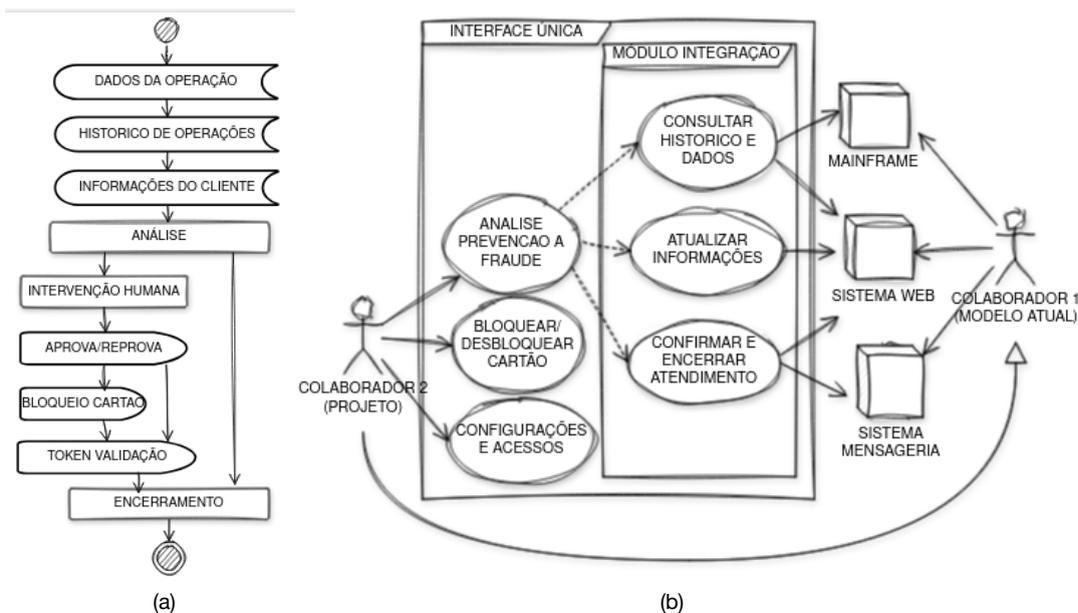


Figura 1. Diagrama de atividade das etapas do processo automatizado e (b) diagrama de caso de uso mostrando os envolvidos e sistemas externos na integração

Vale lembrar que, de acordo com o projeto, por questões burocráticas, histórico, permissões e de controle, os colaboradores deverão continuar possuindo as distintas credenciais de acesso nos sistemas externos participantes. Através de alguma página de configuração e definições o usuário colaborador poderá fazer a manutenção de suas credenciais, de acordo com cada política de segurança. Quanto a isso, somente o processo de *login* é automatizado pelo sistema de tela única.

4. Resultado Obtido

4.1. Apresentação funcional

A principal característica do Tela Única é consolidar em uma única ferramenta as informações necessárias para o usuário fazer seu atendimento e a tomada de decisão em um menor tempo sem ter que navegar em diversas outras ferramentas, cada uma com sua etapa de autenticação, padrão de navegação, identidade visual (Button, 2007) e padronização de apresentação (Figura 2).

A interface principal do sistema é organizada em painéis. No topo, há uma barra de navegação com links para 'Manual', 'Configurações' e 'Ferramentas', além de ícones de busca e acesso. O painel principal é dividido em seções:

- Informações de Entrada:** Campos para CPF (000.000.000-00), Cartão ou Final Cartão (0000 0000 0000 0000 / 0000) e Telefone de Bico ((00) 0 0000-0000). Botões para 'Enviar', 'Cartões' e 'Ligação'.
- Cliente:** Campos para Nome (placeholder) e CPF (000.000.000-00). Campos para 'Cliente VIP - UHV' (placeholder), 'Status Cartão' (placeholder) e 'Usar Monto Fixo?' (botão 'NÃO').
- Despesa Alertada:** Tabela com cabeçalho: 'Fila Alertada', 'Data e Hora', 'Valor', 'Nome Do Estabelecimento', 'POS', 'Codigo Do Indicador', 'Ramo' e 'Status'. O conteúdo da tabela é 'Sem Registros'.
- Q Procura:** Campos para 'Telefone' ((00) 0 0000-0000), 'Data Incluída' (dd/mm/yyyy) e 'Status' (botão 'Fraude'). Campos para 'Tipo de Fraude' e 'Fonte Informante'.
- Dados:** Campos para 'Abertura da conta' (placeholder), 'Data de Emissão' (placeholder) e 'Última alteração de bloqueio' (dd/mm/yyyy). Campos para 'Última alteração de endereço', 'Última alteração de telefone' e 'Última alteração de e-mail PJ'.

Figura 2. Tela principal do sistema

Por se tratar de uma operação que presta atendimento de prevenção de fraudes, normalmente a análise é realizada com base em uma transação iniciada pelo cliente (pessoa física) através de seu cartão de crédito. O sistema Tela Única (Figura 2) faz a captura dessa transação através de *web scraping*, realiza a validação da operação no *mainframe* se a transação faz parte do perfil do cliente. Caso a análise seja positiva, o sistema informa o usuário que se trata de uma transação segura; em caso negativo, o operador, usuário do sistema, faz uma intervenção para confirmação de forma verbal com o cliente do cartão.

Entre os possíveis inconsistências nesta análise, ou seja, as possíveis situações de alerta durante este processo (Tabela 2), podem ser destacadas como anomalias no comportamento do cliente portador do cartão de crédito, por exemplo: (a) *compra não esperada pela Internet* ou (b) *compra online muito acima do valor médio das transações já realizadas*.

Nestas situações, através do *mainframe*, o sistema faz a captura de informações cadastrais, histórico de operações, tela de autorizações e informativos das últimas 12 faturas. A automação realiza as validações técnicas necessárias, por exemplo, se o telefone que iniciou o contato consta em *blacklist* (Levine, 2010) das operadoras de telefonia ou trata-se de um telefone seguro. Esta análise é imediata, completamente automática e a possibilidade de fraude é sinalizada através de cores diretamente na interface do usuário atendente.

Tabela 2. Recurso funcional do sistema desenvolvido

RECURSO FUNCIONAL	DESCRIÇÃO	INTEGRAÇÃO
possibilidade de fraude por comportamento da operação	anomalia de operação (meio da transação)	consulta mainframe
	discrepância de valor de compra por média de operações	consulta mainframe
possibilidade de fraude por análise de cadastro	modificações recentes em dados de cadastro do cliente	consulta sistema web
possibilidade de fraude por análise de protocolo	consideração de matriz de risco exclusivo por operadora ou instituição bancária	protocolo institucional
botões de ações rápidas (<i>toolbar</i>)	bloqueio e desbloqueio de cartão (no <i>mainframe</i>)	<i>input</i> no mainframe
	liberação de serviço <i>bypass</i> (no <i>mainframe</i>)	<i>input</i> no mainframe
	flexpan (no sistema web, por <i>web scraping</i>)	<i>input</i> no sistema web
encerramento	solicitação de confirmação por intervenção humana	confirmação por whatsapp
	<i>status</i> e categorização do atendimento por <i>web scraping</i> no sistema web	<i>input</i> no sistema web

Além dos alertas sobre o comportamento de compra realizado, o sistema também indica outras anomalias. Por exemplo, indicando possibilidade de alteração recente no cadastro do cliente ou se houve tentativas ou bloqueios recentes no cartão.

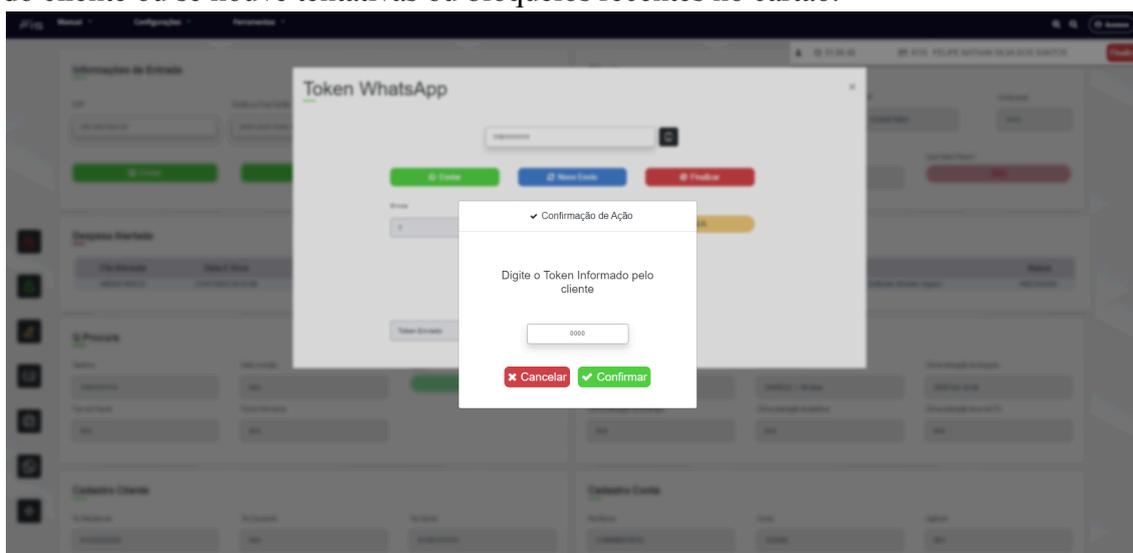


Figura 3. Tela de validação por envio de *token* e finalização do atendimento

4.2. Apresentação técnica do protótipo

O sistema desenvolvido é basicamente um centralizador de operações no nível web que integra 3 grandes serviços, fornecendo para os usuários uma tela única segura e de fácil manipulação. Para desenvolvimento do *backend* do sistema, foi utilizado, o python foi utilizado como principal linguagem de desenvolvimento.

Na camada de apresentação (*front-end*), basicamente o sistema utiliza HTML e CSS e ainda explora algumas facilidades do *framework* Bootstrap e biblioteca de javascript e jQuery.

- **jQuery** (2024): para renderização de tabela, máscaras e auto-preenchimento de dados, melhorando a performance e a experiência do usuário;
- **Bootstrap** (2024): biblioteca para manipulação (apresentação e controle) de componentes HTML.
- **Javascript**: para lógica de interface HTML, incluindo, validação de *inputs*, construção e chamadas de formulários, controle e contagem de tempo do usuário em tela de operação.

Na camada de dados o sistema tem o *sqlite3* como gerenciador central dos dados, mas também propõe uma combinação de diversas soluções integradas para facilitar as tarefas necessárias com os dados. Principalmente para conexões, renderizações e manipulação de dados, o sistema utiliza diversas bibliotecas como apoio:

- **Pyobdc** - biblioteca python para utilização do SQL Server utilizada para incluir o total de chamadas que cada colaborador realizou através da automação do Tela Única. Também facilitou nas tabulações realizadas no fim de cada contato ou atendimento. Para efeito de performance, esta conexão é diária;
- **Sqlite3** - biblioteca python para bancos de dados SQLite, utilizada para manipulação de dados em repositório de dados local para dinamismo e automação, e manipulação de arquivos de dados tipo *.db* em base de dados na rede;
- **Win32com**: utilizada para integração com o ambiente *mainframe* e para manipulação como objeto no código. Também foi utilizado para tornar a integração com *mainframe* uma camada transparente para o usuário final;
- **Pandas**: utilizada basicamente nas funções de manipulação de dados, principalmente para manipulação de tabelas e realização de filtros que resultaram em tomadas decisão incorporadas no sistema, dando a sensação de automatismo de algumas operações por parte dos usuários;
- **Requests**: biblioteca python para requisições HTTP (Requests, 2024). Foi como uma API para envio de mensagem tipo *token* através do sistema de mensageria WhatsApp.
- **Selenium**: biblioteca para automatizar testes funcionais de páginas web (Selenium, 2024). Foi adaptada como uma estratégia para captura de dados e repetição de ações do usuário.
- **Eel framework**: biblioteca python para desenvolvimento de páginas facilitando conexão com o funções javascript para manipulação de dados e componentes web (Python-eel, 2024).

4.3. Problemas enfrentados

Entre as dificuldades do projeto, no que se refere, os problemas técnicos que implicaram em alguma mudança estratégica ou alguma solução mais criativa, pode-se apontar: o *problema de concorrência* e a *falta de uma API por parte dos sistemas terceiros*.

- *controle de concorrência no servidor de dados*: o modelo de implementação em produção na empresa, por questões de performance e segurança possui um processo que não permite múltiplas consultas, limitando a quantidade de acesso e tempo de conexão durante as requisições de dados. Por este motivo não foi possível o uso direto do SQL Server na implementação da Tela Única. Porém, considerando que o sistema em desenvolvimento contempla um processo onde uma média de 135 funcionários podem utilizar de forma acesso e bloqueios no SGBD, foi necessário a implementação de uma estratégia de espelhamento e sincronismo de dados na parte do cliente-usuário através do `sqlite3`.
- *falta de uma API por parte dos sistemas terceiros*: um dos maiores problemas durante o desenvolvimento, foi quanto a disponibilização dos dados disponíveis no sistema web. A falta de uma API que facilitasse a recuperação de informações, e principalmente o não interesse em disponibilização destas informações por parte do sistema terceirizado, forçou o desenvolvimento de uma estratégia de recuperação forçada dos dados necessários para o processo. Uma saída para este inconveniente foi o desenvolvimento de uma camada baseada em recuperação de dados por *web scraping*, diretamente nas páginas web do sistema, no nível de apresentação.

5. Conclusão

O resultado alcançado até o momento do projeto está de acordo com o esperado. O sistema foi apresentado para os envolvidos no projeto e responsáveis pela futura área usuária, e os mesmos demonstraram muita satisfação com a apresentação da versão, que inclusive já está liberada para implantação na produção. No momento a Tela Única segue, ainda no ambiente de desenvolvimento, por questões dos testes de aceitação, que ainda estão em andamento. Entre os principais benefícios do projeto estão a facilidade e os mecanismos de integração realmente muito pertinentes para o processo e ao mesmo tempo de baixo custo de desenvolvimento.

Do ponto de vista acadêmico, o artigo contribui com a comunidade de engenharia de software, usabilidade e integração, uma vez que detalha o processo de desenvolvimento, os desafios encontrados durante a integração dos sistemas e os resultados obtidos, oferecendo *insights* valiosos sobre a aplicação prática do RDA em um estudo de caso real em um ambiente caótico envolvido com transações financeiras.

Quanto à possibilidade de trabalhos futuros, uma ótima linha de pesquisa segue na linha de um projeto que pudesse concentrar-se nas estratégias ou padrões de projeto inseridos na camada de integração promovendo proteção ou facilidades de manutenção desta camada diante de modificações nos sistemas clientes.

Agradecimentos especiais à organização, seus especialistas e colaboradores que permitiram-se envolver como estudo de caso em todas as fases do projeto, um laboratório real para análise, desenvolvimento e validação dos resultados.

Referências Bibliográficas

- Automation Anywhere (2024), "Automation Now and Next: State of Intelligent Automation Report". Automation Anywhere Report. 4a. ed, [online] Disponível em <https://www.automationanywhere.com/rpa/desktop-automation>; acessado em 08/2024.
- Bootstrap (2024). "Javascript MDN Web Documentation". [online] Disponível em <https://getbootstrap.com/>, acessado em agosto de 2024.
- Butow, E. (2007). "User Interface Design for Mere Mortals". 1a. ed., Indianapolis, IN: Addison-Wesley, 2007.
- IBM (2021). "Automação e RPA na empresa: Como as empresas estão aumentando a produtividade e acelerando a geração de valor". IBM eBook. Automação e automação robótica de processos (RPA). IBM Market Development & Insights. [online] Disponível em <https://www.ibm.com/downloads/cas/M4GG0KXV>. 2021.
- Levine, J. (2010). "DNS Blacklists and Whitelists". RFC 5782, Internet Research Task Force (IRTF), ISSN: 2070-1721. <https://www.rfc-editor.org/rfc/rfc5782.txt>. 2010.
- jQuery (2024) "Get started with Bootstrap", [online] Disponível em <https://jquery.com/>, acessado em agosto de 2024.
- Martins, E. e Glegale, N. V. (2022) "Detecção de fraudes no segmento de crédito financeiro utilizando aprendizado de máquina: uma revisão da literatura". Revista e-TECH: Tecnologias para Competitividade Industrial. e-TECH, Florianópolis, v. 15 n. 3. DOI: <https://doi.org/10.18624/etech.v15i3.1198>. 2022
- Python-eel (2024). "Biblioteca Python para HTML GUI apps". Repositório github do projeto Eel. [online] disponível em <https://github.com/python-eel/Eel>, acessado em julho de 2024.
- Requests (2024). "Requests HTTP for Humans". [online] disponível em <https://requests.readthedocs.io/en/latest/>, acessado em julho de 2024.
- Selenium (2024). "Selenium Browser Automation Project". [online] disponível em <https://www.selenium.dev/>, acessado em agosto de 2024.
- Silva, A. M. e Lauers S., M. I. (2015). "Abstraction of Process Modeling Like Support to Requirements Engineering to Complex Systems". DOI 10.14684/ICECE.9.2015.30-32. COPEC IX International Conference on Engineering and Computer Education, May 10 - 13, 2015, Žilina, SLOVAKIA, 2015.
- Souza, D. H. M. e Bordin Jr., C. J. (2023). "Detecção de fraude de cartão de crédito por meio de algoritmos de aprendizado de máquina". Revista Brasileira de Computação Aplicada, Abril. DOI: 10.5335/rbca.v15i1.13790. Vol. 15, N o 1, pp. 1–11. 2023.
- Wattenberg, F. de M. (2019). "Robotic Process Automation: aplicações e resultados do uso da tecnologia". Monografia de Bacharelado em Administração; Departamento de Ciências Administrativas; Universidade Federal do Rio Grande do Sul. Rio Grande do Sul, 2019.