

SÃO PAULO TECH SCHOOL

EDUARDO GOMES DE CAMARGO – RA: 01251062
FLÁVIA CHRISTINE HIDALGO PEREIRA – RA: 01251105
GABRIEL OLIVEIRA SILVA DE PIETRO – RA: 01251048
LAIZA TAVARES GOMES – RA: 04242054
SANDRO GUEDES JÚNIOR – RA: 01251016

Professor(a): **Fernanda Caramico**

Fraux – Grupo 10

Fraudes em e-commerce

São Paulo – SP

2025

Sumário

1. Contexto	3
2. Justificativa	3
3. Objetivo	4
3.1 Objetivo geral	4
3.2 Objetivos específicos	4
3.3 Pesquisa de Campo.....	5
3.4 Histórias do usuário	5
3.5 Lean UX:.....	5
4. Premissas.....	6
5. Restrições e Exclusões	6
5.1 Restrições	6
5.2 Exclusões.....	6
6. Arquitetura da solução	7
6.1 Fluxo do processo	7
6.2 Componente JAR	7
6.3 Site institucional e dashboard.....	7
7. Requisitos	8
8. Evolução por Sprint (Visão Geral)	9
9. Conclusão	10
10. Fontes Utilizadas.....	10

1. Contexto

O e-commerce no Brasil apresenta um crescimento expressivo, mas também enfrenta um desafio significativo: as perdas financeiras causadas por fraudes e chargebacks.

Chargeback é a contestação de uma transação feita pelo titular do cartão, que pode ocorrer por diversos motivos, desde erro operacional até uso fraudulento ou a chamada *fraude amigável*, quando o cliente age de má-fé. Esse problema impacta diretamente a saúde financeira dos lojistas virtuais.

Segundo pesquisas recentes, as tentativas de fraude no e-commerce brasileiro representaram cerca de **4,09% das transações em 2023**, resultando em um potencial prejuízo de bilhões de reais para o setor. Para cada transação fraudulenta, uma loja virtual precisa realizar várias novas vendas para compensar a perda, o que pode levar pequenos negócios à falência. A América Latina, em particular, é a região com o maior índice de chargeback do mundo, superando a média global de 3%.

A proprietária da loja **Arte Petrichor** relata a dificuldade de controlar a quantidade de chargebacks, que são um risco constante ao negócio. A falta de visibilidade e controle sobre essas ocorrências impede um planejamento adequado e a tomada de ações preventivas.

2. Justificativa

A solução proposta se justifica pela necessidade urgente de mitigar as perdas financeiras e operacionais causadas pelas fraudes no e-commerce. A redução dos custos com fraudes pode aumentar o **EBITDA** em mais de 30%, demonstrando o impacto direto na rentabilidade do negócio.

Atualmente, os lojistas não perdem apenas dinheiro, mas muitas vezes também o produto já enviado. Além disso, altos índices de chargeback podem resultar em:

- Penalidades de operadoras de cartão;
- Aumento das taxas de processamento;
- Dano à reputação da marca;
- Perda de credibilidade junto a bancos e consumidores.

O projeto **Fraux** visa fornecer ao usuário uma ferramenta que permita monitorar e analisar chargebacks, identificando tendências e padrões. Ao capacitar os lojistas com informações e indicadores claros, a solução permite uma atuação **proativa**, reduzindo riscos em vez de reagir a cada problema individualmente.

Com a evolução até a Sprint 3, a solução passa a:

- Ler e tratar arquivos de transações automaticamente a partir de um bucket S3;
- Persistir esses dados em banco de dados;
- Gerar logs e notificações em canais de Slack;
- Exibir dashboards atualizados a cada processamento;
- Enviar e gerenciar alertas configuráveis;
- Oferecer um chatbot de IA (Frabot) para uso do usuário;
- Permitir interação do usuário com o sistema por meio de avaliações e edição de perfil.

Isso ajuda o lojista monitorar padrões para se proteger contra fraudes de cartões clonados e clientes mal-intencionados, aumentando a continuidade e a sustentabilidade financeira do negócio.

3. Objetivo

3.1 Objetivo geral

Criar uma **plataforma de monitoramento de fraudes** para lojistas de e-commerce em geral, entregando indicadores e insights para a tomada de decisões estratégicas.

3.2 Objetivos específicos

- Desenvolver uma solução que monitore fraudes e forneça indicadores de perdas financeiras, identificando períodos de maior ocorrência e os principais motivos.
- Disponibilizar **dashboards** que sejam atualizados automaticamente a cada execução do JAR, refletindo a situação mais recente do negócio.
- Manter o usuário constantemente alerta sobre as métricas e possíveis anomalias nas transações.
- Implementar uma **estrutura de ingestão de dados** via arquivos no Amazon S3, processamento em JAR executável e persistência no MySQL em ambiente Docker em instância EC2.
- Permitir a **parametrização de alertas**, de forma que o lojista possa configurar regras como:
 - valor mínimo ou máximo da transação;
 - cidade/estado de origem;
 - método de pagamento;
 - status/indícios de fraude.
- Integrar um **chatbot de IA (Frabot)** desenvolvido em Python para o usuário.

- Disponibilizar uma página de **avaliações**, permitindo que usuários registrem feedback sobre a solução.
- Permitir que o usuário edite seu **perfil**, mantendo seus dados atualizados na plataforma.

3.3 Pesquisa de Campo

Para a um melhor entendimento das dores e necessidades dos clientes, foi realizada uma visita com uma pessoa do ramo de e-commerce. Contatamos a Karine Gomes, uma gerente de e-commerce, que nos forneceu informações valiosas para o desenvolvimento completo deste projeto.

Evidência: [evidenciaVisita.jpg](#)

Entrevista: [Ata-Visita 24-08.pdf](#)

3.4 Histórias do usuário

Com base na entrevista, fizemos Histórias do Usuário, que auxiliaram a identificar as dores e necessidades do cliente.

Histórias do usuário: [User stories.docx](#)

3.5 Lean UX:

Por meio destas Histórias do usuário, foram mapeados seus problemas, soluções e benefícios em uma visualização geral, para isso foi feito um Lean UX, que reúne todos os tópicos mencionados.

Lean UX: [Lean Ux Fraux \(1\).jpg](#)

4. Premissas

- Haverá acesso a dados de transações financeiras contendo, no mínimo:
 - id da transação;
 - data/hora;
 - valor;
 - método de pagamento;
 - status/indício de fraude;
 - identificadores de cliente e empresa.
- O modelo de dados segue o **MER** definido (usuários, compras, alertas etc.), permitindo a criação de relatórios, dashboards e alertas.
- A instância **EC2** hospeda containers Docker separados para:
 - MySQL (banco da aplicação);
 - Site institucional (front/servidor);
 - JAR executável responsável por ler arquivos do S3, tratar e inserir no MySQL, além de gerar logs e enviar notificações para o Slack.

5. Restrições e Exclusões

5.1 Restrições

- O projeto estará limitado à **análise e monitoramento** de dados de chargebacks e fraudes.
- No desenvolvimento do site institucional, serão utilizadas apenas as tecnologias aprendidas em sala de aula:
 - **HTML, CSS, JavaScript e Node.js**.
- O deploy do site institucional é feito utilizando **Docker**, isolando um ambiente próprio para execução da aplicação.
- A infraestrutura deve ser de **baixo custo e simples**, com uma única instância EC2 e containers isolados.

5.2 Exclusões

- A solução **não terá como objetivo realizar a contestação automática** de fraudes junto ao banco ou operadoras de cartão.
- O foco é fornecer **informações, indicadores e alertas** para que o lojista realize a contestação por conta própria de forma mais embasada.

6. Arquitetura da solução

A arquitetura do Fraux é baseada em uma estrutura de baixo custo, utilizando serviços em nuvem e containers:

- **Amazon EC2**
 - Hospeda múltiplos containers Docker:
 - MySQL (banco de dados da aplicação);
 - Site institucional (front-end e back-end em Node.js);
 - JAR executável (pipeline de ingestão e processamento).
- **Amazon S3**
 - Armazena arquivos Excel contendo os dados de transações financeiras.

Arquitetura: [Arquitetura técnica \(1\).svg](#)

6.1 Fluxo do processo

Para mapear o processo que será feito pelo software, foi criado um BPMN, que mostra o fluxo de todos os processos realizados.

BPMN: [diagram.svg](#)

6.2 Componente JAR

O JAR é responsável por:

- Ler um arquivo **Excel** em um bucket S3;
- Tratar, validar e transformar os dados da planilha;
- Inserir os dados tratados no banco **MySQL**;
- Gerar **logs** de processamento (sucesso, falhas, linhas inválidas etc.);
- Enviar **notificações para o Slack**, informando status de execução, possíveis erros ou alertas relevantes.

6.3 Site institucional e dashboard

O site institucional, desenvolvido com **HTML, CSS, JavaScript e Node.js**, oferece:

- Uma **dashboard** que consome os dados inseridos pelo JAR no MySQL e plota gráficos e indicadores a cada nova execução do JAR;
- Funções para **ligar e desligar alertas** diretamente na página do usuário;
- Tela de **parametrização de alertas**, onde o usuário define condições como valor mínimo, cidade/estado, método de pagamento e outros filtros relacionados a fraudes;

- Página do **Frabot**, o chatbot de IA feito em Python, integrado ao sistema para auxiliar o usuário com suas dúvidas;
- Página de **avaliações**, permitindo que o usuário deixe comentários e notas sobre a experiência com a ferramenta;
- Funcionalidade para o usuário **editar seu perfil**, atualizando informações pessoais.

7. Requisitos

Para iniciar o desenvolvimento do projeto, é necessário mapear todos os requisitos, para isso foi feito um backlog listando todos os requisitos.

No Backlog é possível identificar as seguintes colunas

- Requisito: Descreve a atividade a ser feita
- Categoria: A qual conjunto de atividades este requisito pertence
- Código: Cada requisito possui um código para identificar o requisito
- Task: Nome da atividade a ser feita
- Descrição: Descreve a task indicada
- Tipo: Identifica se o requisito é funcional ou não funcional
 - Requisitos funcionais: são todos os problemas e necessidades que devem ser atendidos e resolvidos pelo software por meio de funções ou serviços.
 - Requisitos não funcionais: requisitos não funcionais são todos aqueles relacionados à forma como o software tornará realidade o que está sendo planejado.
- Importância: Classifica a importância da atividade, visando sua conclusão em cada sprint.
- Status: Como a atividade está e se já foi desenvolvida
- Tamanho: Categorização com base na escala de Fibonacci
- Urgência: Priorização da atividade
- Custo: Monetário
- Sprint: Em qual etapa do projeto será concluída tal atividade

As colunas seguintes, são relacionadas à Matriz de rastreabilidade, onde é possível verificar de onde surgiu aquele requisito e seu critério de avaliação.

Backlog: [ProductBacklog-01.xlsx.url](#)

8. Wireframes

Com base nas necessidades

8. Evolução por Sprint (Visão Geral)

- **Sprint 1**
 -
- **Sprint 2**
 - Definição do contexto, problema e justificativa;
 - Modelagem de dados (MER);
 - Estrutura básica da aplicação web;
 - Concepção do fluxo de ingestão (S3 → JAR → MySQL).
- **Sprint 3**
 - Implementação do JAR com:
 - leitura do Excel no S3,
 - tratamento e inserção dos dados no MySQL,
 - geração de logs,
 - envio de notificações ao Slack;
 - Implementação da **dashboard dinâmica**, atualizada a cada execução do JAR;
 - Funcionalidade de **liga/desliga de alertas**;
 - **Parametrização de alertas** (valor, região, método de pagamento etc.);
 - Integração do **Frabot (chatbot em Python)** na plataforma;
 - Criação da **página de avaliações**;
 - Funcionalidade de **edição de perfil** do usuário.

9. Conclusão

A implementação do Fraux proporciona aos lojistas virtuais uma ferramenta essencial para a **gestão de riscos** relacionados a fraudes e chargebacks.

Ao transformar dados brutos de transações em **insights açãoáveis** por meio de dashboards, alertas e notificações, a solução permite que os empreendedores:

- tomem decisões mais informadas,
- reduzam perdas financeiras,
- aumentem o controle sobre chargebacks,
- fortaleçam a reputação e a sustentabilidade do negócio.

Com a evolução até a Sprint 3, o projeto deixa de ser apenas uma ideia de monitoramento e passa a ser uma **plataforma funcional**, com:

- pipeline automatizado de ingestão de dados (S3 → JAR → MySQL),
- notificações integradas ao Slack,
- dashboards interativos,
- alertas configuráveis,
- suporte por IA via Frabot,
- interação ativa do usuário por meio de avaliações e edição de perfil.

Dessa forma, o Fraux se posiciona como uma solução de apoio estratégico para lojistas digitais que desejam enfrentar um dos maiores desafios do e-commerce moderno: a **fraude por chargeback**.

10. Fontes Utilizadas

<https://www.kaggle.com/datasets/shriyashjagtap/fraudulent-e-commerce-transactions>

<https://www.kaggle.com/datasets/bhadramohit/credit-card-fraud-detection>

<https://ipnews.com.br/perdas-com-fraudes-podem-consumir-cerca-de-2-da-receita-de-lojas-virtuais/>

<https://www.migalhas.com.br/depeso/377163/de-quem-e-a-responsabilidade-pelo-chargeback>

<https://www.terra.com.br/noticias/chargeback-contestacao-de-compras-pode-implicar-em-prejuizo>

<https://www.ecommercebrasil.com.br/noticias/em-2023-409-das-transacoes-nos-e-commerce-brasileiros-foram-tentativas-de-fraude>

<https://www.meioemensagem.com.br/proxima/golpes-e-fraudes-impactam-e->

[commerce-no-brasil-em-r-85-milhoes](#)

<https://www.abras.com.br/clipping/tecnologia/114412/e-commerces-tem-muito-a-perder-com-os-chargebacks>

<https://www.zoop.com.br/blog/pagamento/o-que-e-chageback>

<https://blog.cielo.com.br/dicas-e-historias-de-sucesso/o-que-e-chageback/>