



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
CURSO DE ENGENHARIA ELÉTRICA

HENRIQUE SPENCER ALBUQUERQUE

RELATÓRIO DE ESTÁGIO OBRIGATÓRIO

Natal/RN
2023

Henrique Spencer Albuquerque

Relatório de Estágio Obrigatório

Este relatório tem finalidade de descrever as atividades realizadas no estágio curricular supervisionado, realizado na empresa BTG, como requisito necessário para obter o título de Engenheiro Eletricista.

Orientador: Prof. Dr. Valentin Obac Roda

Orientador:
Prof. Dr Valentin Obac Roda

Supervisor:
NOME

Natal/RN
2023

FOLHA DE APROVAÇÃO

Relatório de estágio supervisionado, desenvolvido na empresa BTG, apresentado em DATA de julho de 20223, na cidade de Natal, no estado do Rio Grande do Norte, examinado por:

Professora Doutor Valentin Obac Roda
Orientador de Estágio
Universidade Federal do Rio Grande do Norte

NOME
Supervisor de Estágio
CARGO BTG

LISTA DE FIGURAS

Figura 1: Visualização da estrutura de um html	13
Figura 2: Ilustração do papel do react.	16
Figura 3: Ilustração do funcionamento geral de um CI/CD.....	19
Figura 4: Tela de aprovação de documentos do Docmatch de sinistros.....	27
Figura 5: Fluxo antigo de funcionamento do pagamento de Peritos.	29
Figura 6: Fluxo novo de funcionamento do pagamento de Peritos.	29
Figura 7: Tela gerencial do pagamento de perito.	30
Figura 8: Exemplo de relatorio enviado pelo projeto de Auditoria de Operações no FTS.	31

LISTA DE ABREVEATURAS E SIGLAS

CI -- Continuous Integration

CD -- Continuous Deployment

JS -- Java Script

SQL -- Structured Query Language

SQS -- Simple Queue Service

SNS -- Simple Notification Service

HTML -- HyperText Markup Language

CSS -- Cascading Style Sheets

S3 -- Simple Storage Service

SUMÁRIO

1. IDENTIFICAÇÃO	7
2. RESPONSABILIDADE E COMPROMISSO	8
2.1. Termo do Aluno.....	8
2.2. Termo do Supervisor	9
3. INTRODUÇÃO.....	10
4. BTG Pactual	11
5. APROFUNDAMENTAÇÃO TEÓRICA.....	12
5.1. Python.....	12
5.2. Java Script (Js).....	12
5.3. HyperText Markup Language (HTML)	13
5.4. Cascading Style Sheets (CSS)	14
5.5. Structured Query Language (SQL).....	14
5.6. React	15
5.7. Flask.....	16
5.8. Fast API	17
5.9. Continuous Integration/Continuous Deployment (CI/ CD).....	18
5.10. Micro serviços AWS	19
5.10.1. Amazon Simple Storage Service (S3)	20
5.10.2. Amazon Simple Queue Service (SQS).....	21
5.10.3. Amazon Simple Notification Service (SNS).....	22
5.10.4. Lambda	23
5.10.5. DynamoDB	23
6. ATIVIDADES DESENVOLVIDAS.....	25
6.1. Projeto de extração e batimento digital assets	25
6.2. Projeto de validação de documentos para pagamento de indenização de seguro rural ...	26
6.3. Relatório Custódia de Criptomoedas(DRU - MYNT).....	28
6.4. Projeto de validação de documentos do pagamento de vistorias aos peritos	28
6.5. Auditoria de Operações no FTS	30
7. CONCLUSÃO.....	33
8. REFERÊNCIAS	34

1. IDENTIFICAÇÃO

Nome: Henrique Spencer Albuquerque

Av. Amintas Barros, 3735

Lagoa Nova

Telefone: (84) 9 9910-0137

Email: henriquespencer11@gmail.com

Empresa: BTG.

Av. Brigadeiro Faria Lima, 3477

Itaim Bibi, São Paulo

Telefone: (11) 4007-2511

Supervisor: NOME

Engenheiro Eletricista

CARGO da BTG.

Telefone: TELEFONE

Email: EMAIL

2. RESPONSABILIDADE E COMPROMISSO

2.1. Termo do Aluno

Eu, Henrique Spencer Albuquerque, portador do RG número 002.962.925 SSP/RN, domiciliado na Amintas Barros, 3735 – Lagoa Nova, Natal/RN, responsabilizo-me pela veracidade das informações contidas neste relatório e autorizo ao representante legal da Universidade Federal do Rio Grande do Norte a fazer uso de qualquer meio legal aplicável para comprová-las.

Henrique Spencer Albuquerque

Estagiário

2.2. Termo do Supervisor

Eu, **NOME, CARGO** na empresa BTG, responsabilizo-me pela veracidade das informações contidas neste relatório e autorizo ao representante legal da Universidade Federal do Rio Grande do Norte a fazer uso de qualquer meio legal aplicável para comprová-las.

NOME

Supervisor de Estágio

CARGO

3. INTRODUÇÃO

Este relatório foi desenvolvido com o objetivo de descrever as principais atividades realizadas no estágio curricular obrigatório do curso de Engenharia Elétrica que ocorreu na empresa BTG Pactual. Os trabalhos que foram executados durante o período de estágio proporcionaram uma aprendizagem e aplicação dos conteúdos ministrados na UFRN, sendo realizados em obras em andamentos e obras a serem executadas.

O estágio foi realizado por um período de 18(dezoito) meses, durante Outubro de 2022 a Julho de 2023, obedecendo a carga horária de 6(seis) horas diárias, acompanhado e supervisionado pelo NOME e cargo BTG Pactual e orientado pela professor Doutor Valentin Obac Roda, do Departamento de Engenharia Elétrica da Universidade Federal do Rio Grande do Norte.

Este relatório será apresentado as principais atividades realizadas durante o período de estágio, comentadas de maneira direta e concisa.

4. BTG Pactual

A BTG Pactual é uma empresa financeira global especializada em serviços de banco de investimento e gestão de ativos. Fundado em 1983, o BTG Pactual se tornou um dos maiores bancos de investimento da América Latina, com presença em 13 países e mais de 3 mil colaboradores.

A empresa tem uma história marcada por uma abordagem inovadora e uma ampla gama de serviços financeiros. Ao longo dos anos, a BTG estabeleceu-se como uma das principais instituições financeiras da América Latina, com uma reputação de excelência e expertise em mercados emergentes.

A BTG oferece uma série de serviços para clientes institucionais e corporativos, incluindo assessoria em fusões e aquisições, finanças corporativas, gestão de ativos, gestão de recursos e mercado de capitais. A empresa também possui uma divisão de wealth management, fornecendo serviços personalizados para clientes de alta renda.

Ao longo de sua história, a BTG alcançou marcos significativos. Em 2007, a empresa realizou sua oferta pública inicial (IPO), fortalecendo sua posição como uma das principais instituições financeiras do Brasil. Em 2012, a BTG expandiu sua presença globalmente com aquisições estratégicas e parcerias em diferentes regiões.

No entanto, a BTG também enfrentou desafios em sua trajetória. Em 2015, a empresa passou por um período de turbulência devido à prisão de seu fundador e então presidente, André Esteves. Após essa fase conturbada, a BTG passou por uma reestruturação interna e implementou medidas para fortalecer sua governança corporativa e gestão de riscos.

Nos últimos anos, a BTG tem se consolidado como uma empresa resiliente, focada em oferecer soluções inovadoras e adaptar-se às demandas do mercado financeiro. Com uma equipe experiente e uma abordagem ágil, a BTG continua a fornecer serviços financeiros de qualidade e a buscar oportunidades de crescimento em mercados domésticos e internacionais.

Em resumo, a BTG é uma empresa financeira global com uma história notável, oferecendo serviços de banco de investimento, gestão de ativos e wealth management. Apesar dos desafios enfrentados, a BTG continua a se posicionar como uma das principais instituições financeiras do Brasil e a buscar oportunidades de sucesso em âmbito internacional.

5. APROFUNDAMENTAÇÃO TEÓRICA

5.1. Python

Python é uma linguagem de programação de alto nível amplamente utilizada na área de ciência de dados, desenvolvimento web e automação de tarefas. No contexto da fundamentação teórica de um relatório de estágio, Python pode desempenhar um papel fundamental ao oferecer uma base sólida para a análise e processamento de dados.

Python é conhecido por sua sintaxe simples e legibilidade, tornando-o acessível para iniciantes e profissionais. Com uma ampla gama de bibliotecas e ferramentas disponíveis, como NumPy, Pandas e Matplotlib, Python permite manipular, visualizar e extrair insights de dados de forma eficiente.

5.2. Java Script (Js)

JavaScript é uma linguagem de programação amplamente utilizada no desenvolvimento web. Ela permite adicionar interatividade e dinamismo às páginas da web, tornando-as mais funcionais e envolventes para os usuários.

Ao contrário do HTML e do CSS, que são usados para criar a estrutura e o estilo visual das páginas, respectivamente, o JavaScript é responsável por fornecer funcionalidades interativas. Isso inclui recursos como validação de formulários, criação de animações, manipulação de eventos, requisições assíncronas a servidores, manipulação do DOM (Document Object Model) e muito mais.

Uma das principais características do JavaScript é a sua capacidade de ser executado diretamente no navegador do usuário, sem a necessidade de instalação de software adicional. Isso significa que, ao visitar um site que utiliza JavaScript, o código é interpretado e executado pelo próprio navegador, permitindo que a página responda dinamicamente aos eventos e interações do usuário.

JavaScript é uma linguagem de fácil aprendizado e possui uma ampla variedade de frameworks e bibliotecas, como React, Angular e Vue.js, que facilitam o desenvolvimento de aplicações web mais complexas e escaláveis.

Em resumo, o JavaScript é uma linguagem essencial para o desenvolvimento web moderno, permitindo a criação de sites interativos e dinâmicos, proporcionando uma melhor experiência ao usuário e tornando a web mais funcional e poderosa.

5.3. HyperText Markup Language (HTML)

HTML (HyperText Markup Language) é a linguagem padrão utilizada para estruturar e organizar o conteúdo das páginas da web. Ele fornece a base estrutural para a criação de elementos e componentes que compõem uma página, como texto, imagens, links, formulários e muito mais.

Com o HTML, é possível definir a hierarquia e a disposição dos elementos em uma página da web, por meio do uso de tags ou etiquetas. Cada tag possui uma função específica e é colocada entre colchetes angulares (< e >) para indicar seu início e fim. Por exemplo, a tag <h1> é usada para criar um cabeçalho de nível 1, enquanto a tag <p> é usada para definir um parágrafo de texto. Além disso, o HTML permite a criação de links usando a tag <a>, que conecta diferentes páginas da web entre si. Ele também possibilita a inserção de imagens através da tag , tornando as páginas visualmente mais atraentes e informativas. Essa estrutura é exemplificada na figura 1.

Figura 1: Visualização da estrutura de um html

```
<!DOCTYPE html>
<html>

  <head>
    <title>Page Title</title>
  </head>

  <body>
    <h2>Heading Content</h2>
    <p>Paragraph Content</p>
  </body>

</html>
```

Fonte: Elaborado Pelo Autor.

Embora o HTML seja uma linguagem de marcação e não de programação, ele é fundamental para a construção de páginas da web, sendo a base sobre a qual outras tecnologias, como CSS (Cascading Style Sheets) e JavaScript, são aplicadas para adicionar estilo e interatividade.

Em suma, o HTML desempenha um papel fundamental na criação de páginas da web, fornecendo a estrutura e a organização necessárias para exibir conteúdo de forma clara e coerente. É a linguagem base que permite a construção de páginas estáticas, que podem ser estilizadas e tornadas interativas por meio de outras tecnologias complementares.

5.4. Cascading Style Sheets (CSS)

CSS (Cascading Style Sheets) é uma linguagem de estilo utilizada para definir a aparência e o layout de elementos em páginas da web. Enquanto o HTML estrutura o conteúdo, o CSS é responsável por controlar o design e a apresentação desse conteúdo.

Com o CSS, é possível definir propriedades como cor, fonte, tamanho, espaçamento, posição e muito mais para os elementos HTML. Essas propriedades são aplicadas aos elementos por meio de seletores, que especificam quais elementos devem receber os estilos definidos.

Uma das principais vantagens do CSS é a sua capacidade de separar a estrutura do conteúdo da sua apresentação visual. Isso significa que é possível alterar a aparência de um site inteiro fazendo alterações em um único arquivo CSS, em vez de precisar modificar cada elemento individualmente.

Além disso, o CSS permite criar estilos reutilizáveis, por meio da definição de classes e IDs. As classes podem ser aplicadas a vários elementos, permitindo uma estilização consistente em várias partes do site. Os IDs, por outro lado, são únicos e geralmente são usados para estilizar elementos específicos.

Com a evolução do CSS, também surgiram técnicas avançadas, como flexbox e grid, que facilitam o posicionamento e o alinhamento de elementos na página. Isso oferece maior controle sobre o layout e permite criar designs responsivos, adaptáveis a diferentes tamanhos de tela.

Em resumo, o CSS desempenha um papel crucial no design de páginas da web, permitindo a personalização da aparência e do layout dos elementos. Ele ajuda a criar sites visualmente atraentes, coerentes e responsivos, melhorando a experiência do usuário e possibilitando a criação de interfaces modernas e estilizadas.

5.5. Structured Query Language (SQL)

SQL (Structured Query Language) é uma linguagem de programação usada para gerenciar e manipular bancos de dados relacionais. Ela oferece uma forma padronizada e eficiente de armazenar, recuperar e modificar dados em um formato estruturado.

O SQL permite que os usuários definam a estrutura de um banco de dados por meio da criação de tabelas, que consistem em linhas e colunas. Essas tabelas representam entidades e seus atributos, como clientes, produtos ou pedidos. Por meio de comandos SQL, é possível realizar várias operações nos dados armazenados nessas tabelas.

O SQL oferece uma ampla gama de comandos para interagir com bancos de dados. Os mais comuns incluem:

- SELECT: Recupera dados de uma ou mais tabelas com base em critérios específicos.
- INSERT: Adiciona novos dados a uma tabela.
- UPDATE: Modifica dados existentes dentro de uma tabela.
- DELETE: Remove dados de uma tabela.
- JOIN: Combina dados de várias tabelas com base em colunas relacionadas.
- CREATE: Cria novas tabelas, visões ou outros objetos de banco de dados.
- ALTER: Modifica a estrutura de tabelas ou objetos existentes.
- DROP: Exclui tabelas, visões ou outros objetos do banco de dados.

O SQL é uma linguagem declarativa, o que significa que você especifica quais dados deseja ou quais operações deseja realizar, e o sistema de gerenciamento de banco de dados cuida do "como". Ele fornece uma maneira poderosa e eficiente de interagir com bancos de dados, sendo uma habilidade fundamental para administradores de bancos de dados, analistas de dados e desenvolvedores de software.

Em resumo, o SQL é uma linguagem projetada para gerenciar bancos de dados relacionais. Ele permite que você defina, recupere, modifique e manipule dados de maneira organizada e eficiente. Compreender o SQL é essencial para trabalhar com bancos de dados e executar tarefas relacionadas a dados de forma eficaz.

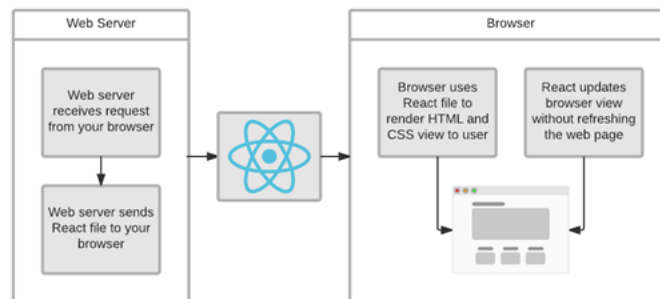
5.6. React

React é uma biblioteca JavaScript de código aberto utilizada para criar interfaces de usuário interativas e reativas. Desenvolvida pelo Facebook, o React permite a construção eficiente de componentes reutilizáveis que atualizam e renderizam automaticamente quando os dados mudam.

Uma das principais características do React é o seu modelo de programação baseado em componentes. Os componentes são peças isoladas de código que encapsulam a lógica e a apresentação de elementos da interface. Com o React, é possível criar hierarquias complexas de componentes, facilitando a organização e a manutenção do código.

O React utiliza uma abordagem chamada "Virtual DOM" (Documento Objeto Virtual), que é uma representação leve e eficiente do DOM (Modelo de Objeto de Documento) real. O Virtual DOM permite que o React atualize somente as partes necessárias da interface, otimizando o desempenho e proporcionando uma experiência de usuário mais fluida, como pode ser bem observado na figura 2.

Figura 2: Ilustração do papel do react.



Fonte: Elaborado Por Ben Nadel, retirado em <https://www.bennadel.com>.

Além disso, o React pode ser combinado com outras bibliotecas ou frameworks, como o React Router para gerenciamento de rotas e o Redux para gerenciamento de estado global da aplicação. Essas combinações tornam o desenvolvimento de aplicações web mais robusto, escalável e modular.

O React também oferece suporte a recursos avançados, como renderização do lado do servidor (Server-side Rendering - SSR) e criação de aplicativos móveis com o React Native. Isso amplia a versatilidade do React, permitindo que ele seja usado em diferentes contextos de desenvolvimento.

Em resumo, o React é uma biblioteca JavaScript popular e poderosa para a criação de interfaces de usuário reativas. Sua abordagem baseada em componentes, Virtual DOM e suporte a recursos avançados o tornam uma escolha popular para o desenvolvimento de aplicações web interativas e escaláveis.

5.7. Flask

Flask é um framework leve de desenvolvimento web em Python. Ele oferece uma abordagem minimalista e simples para a construção de aplicativos web, permitindo que os desenvolvedores criem rapidamente aplicações com funcionalidades básicas.

Com o Flask, é possível criar rotas e manipuladores de solicitações (request handlers) para lidar com diferentes URLs e métodos HTTP. Isso permite que os desenvolvedores definam como o aplicativo deve responder a solicitações específicas, como exibir uma página, processar um formulário ou fornecer dados de uma API.

Uma das características distintivas do Flask é a sua filosofia "sem opiniões" (unopinionated). Isso significa que o Flask não impõe uma estrutura rígida ou um conjunto

específico de ferramentas, dando aos desenvolvedores mais liberdade para escolher as tecnologias e a organização do projeto.

Embora seja considerado um framework minimalista, o Flask possui uma ampla gama de extensões disponíveis que podem ser adicionadas conforme necessário. Essas extensões fornecem recursos adicionais, como autenticação de usuário, integração de banco de dados, suporte a APIs RESTful e muito mais. Isso permite que os desenvolvedores ampliem a funcionalidade do Flask de acordo com os requisitos do projeto.

Outra vantagem do Flask é a sua curva de aprendizado acessível, tornando-o uma ótima opção para iniciantes em desenvolvimento web. A sintaxe clara e a documentação abrangente facilitam a compreensão e o uso do framework, permitindo que os desenvolvedores criem rapidamente aplicativos web funcionais.

Em resumo, o Flask é um framework de desenvolvimento web em Python que prioriza a simplicidade e a flexibilidade. Com ele, é possível construir rapidamente aplicativos web básicos, mantendo a liberdade de escolha das tecnologias e ferramentas. Sua abordagem minimalista e curva de aprendizado acessível o tornam uma opção popular para desenvolvedores que desejam criar aplicativos web com eficiência.

5.8. Fast API

FastAPI é um framework moderno, rápido e altamente eficiente para construir APIs com Python. Ele foi projetado para oferecer alta performance e produtividade, tornando-se uma escolha popular para o desenvolvimento de aplicações web com endpoints de API.

Uma das características marcantes do FastAPI é o seu desempenho excepcional. Isso é possível através do uso de programação assíncrona e da framework subjacente chamada Starlette. Ao aproveitar o poder das operações assíncronas, o FastAPI consegue lidar com altas cargas e requisições simultâneas de forma eficiente, resultando em tempos de resposta extremamente rápidos.

O FastAPI também é conhecido pela sua interface intuitiva e fácil de usar. Ele utiliza os recursos de dicas de tipo (type hints) do Python, permitindo que os desenvolvedores definam endpoints de API claros e autoexplicativos. Essa funcionalidade possibilita a validação automática de dados, serialização e geração de documentação interativa da API usando os padrões OpenAPI e JSON Schema. Como resultado, os desenvolvedores podem criar rapidamente APIs robustas e bem documentadas com pouco esforço.

Além disso, o FastAPI integra-se perfeitamente a outras bibliotecas e frameworks populares do Python, como SQLAlchemy para suporte a banco de dados e Pydantic para

validação e serialização de dados. Isso torna o FastAPI uma escolha versátil para construir aplicações complexas que requerem interações com banco de dados e processamento de dados.

Além disso, o FastAPI possui suporte incorporado para autenticação, injeção de dependências, validação de requisições e outras funcionalidades comuns no desenvolvimento web. Ele segue a filosofia de "baterias incluídas, mas opcionais", permitindo que os desenvolvedores escolham e habilitem apenas as funcionalidades necessárias para seu caso de uso específico.

Em resumo, o FastAPI é um framework web moderno e de alta performance para construir APIs com Python. Ele combina o poder da programação assíncrona, dicas de tipo e geração automática de documentação para oferecer APIs eficientes e bem documentadas. Com sua simplicidade, desempenho e conjunto de recursos abrangente, o FastAPI tem ganhado popularidade entre os desenvolvedores que buscam um framework rápido e produtivo para o desenvolvimento de APIs.

5.9. Continuous Integration/Continuous Deployment (CI/ CD)

CI/CD (Continuous Integration/Continuous Deployment) é uma prática de desenvolvimento de software que visa automatizar e agilizar o processo de entrega e implantação de aplicações.

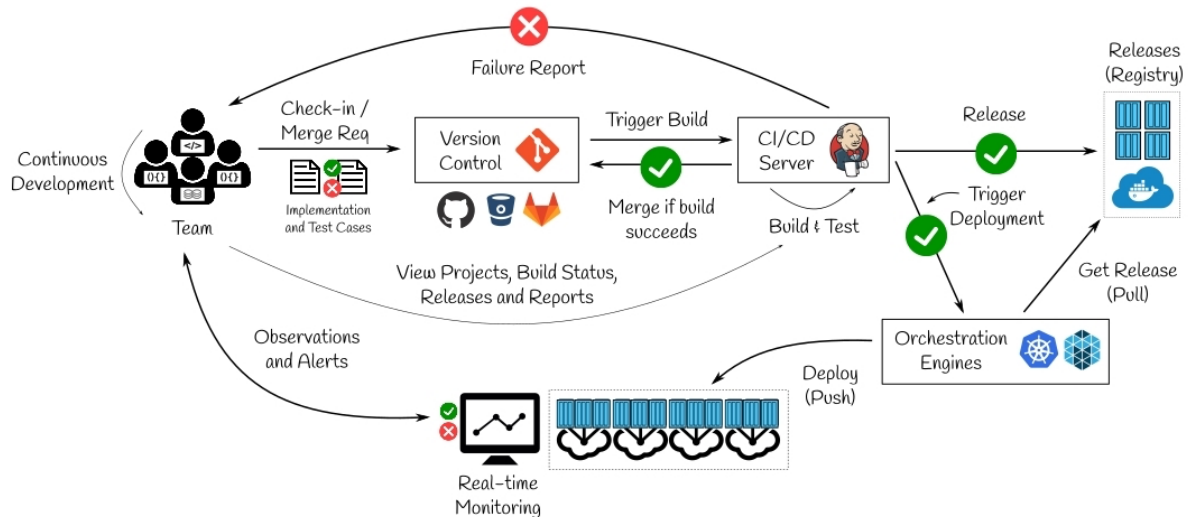
A integração contínua (CI) refere-se à prática de combinar o trabalho de vários desenvolvedores em um repositório central, onde é realizado um processo automatizado de compilação, teste e verificação de qualidade do código. Esse processo ocorre regularmente, assim que as alterações são enviadas ao repositório, permitindo a detecção precoce de erros e conflitos de integração. A CI busca garantir que o código seja sempre funcional e esteja alinhado com as melhores práticas de desenvolvimento.

A implantação contínua (CD), por sua vez, é o processo automatizado de entrega e implantação do software em ambientes de produção. Isso envolve a automação de tarefas como compilação, empacotamento, teste, aprovação e implantação propriamente dita. Com o CD, é possível eliminar o trabalho manual e demorado de implantação, reduzindo assim a possibilidade de erros humanos e aumentando a eficiência do processo.

Ao adotar CI/CD, as equipes de desenvolvimento podem garantir um ciclo de desenvolvimento mais rápido, confiável e consistente. Isso permite que as alterações sejam entregues aos usuários finais em um ritmo acelerado, promovendo maior agilidade e melhorias contínuas na qualidade do software. Além disso, o CI/CD facilita a colaboração entre os membros da equipe, uma vez que o código é integrado e testado de forma constante. Isso

proporciona uma visibilidade clara do status do projeto e permite a identificação rápida de problemas, facilitando a resolução e a implementação de melhorias. Tudo esse funcionamento pode ser bem observado na figura 3.

Figura 3: Ilustração do funcionamento geral de um CI/CD.



Fonte: retirado de <https://www.opsmx.com/blog/what-is-a-ci-cd-pipeline/>.

Em resumo, CI/CD é uma prática que visa automatizar e acelerar o processo de entrega e implantação de software. Através da integração contínua e implantação contínua, é possível garantir a qualidade do código, reduzir erros e entregar alterações aos usuários finais de forma rápida e eficiente.

5.10. Micro serviços AWS

Microserviços AWS são uma abordagem arquitetural para o desenvolvimento de aplicativos distribuídos na nuvem utilizando serviços da Amazon Web Services (AWS). Eles são uma maneira de projetar aplicativos dividindo-os em componentes independentes, conhecidos como microserviços, que podem ser desenvolvidos, implantados e escalados de forma independente.

Cada microserviço é responsável por uma funcionalidade específica do aplicativo e pode ser desenvolvido usando a linguagem de programação e o conjunto de tecnologias mais adequados para aquela funcionalidade. Esses microserviços são executados como processos independentes e se comunicam entre si por meio de APIs.

Os serviços da AWS fornecem um conjunto abrangente de ferramentas e recursos para ajudar no desenvolvimento, implantação e gerenciamento de microserviços. Por exemplo, o

Amazon Elastic Container Service (ECS) e o Amazon Elastic Kubernetes Service (EKS) são serviços que permitem implantar e orquestrar contêineres para executar os microserviços. O AWS Lambda é outro serviço que permite executar código sem precisar provisionar ou gerenciar servidores, sendo útil para implementar microserviços serverless.

A arquitetura de microserviços AWS traz diversos benefícios. Ela permite que as equipes de desenvolvimento trabalhem de forma independente em diferentes partes do aplicativo, tornando o desenvolvimento mais ágil. Além disso, os microserviços podem ser escalados individualmente, o que possibilita uma melhor utilização dos recursos e uma maior capacidade de resposta aos picos de tráfego.

Outro benefício importante dos microserviços AWS é a resiliência e a tolerância a falhas. Como os microserviços são independentes, a falha em um deles não afeta o funcionamento dos outros. Além disso, a AWS oferece recursos de monitoramento e gerenciamento que facilitam a detecção e o diagnóstico de problemas em tempo real.

Em resumo, os microserviços AWS são uma abordagem arquitetural para o desenvolvimento de aplicativos distribuídos na nuvem. Eles permitem a criação de aplicativos divididos em componentes independentes, que podem ser desenvolvidos e implantados de forma independente, aproveitando os serviços e recursos da AWS para garantir escalabilidade, resiliência e agilidade no desenvolvimento de aplicativos na nuvem.

5.10.1. Amazon Simple Storage Service (S3)

O Amazon S3 (Simple Storage Service) é um serviço de armazenamento em nuvem oferecido pela Amazon Web Services (AWS). Ele fornece um armazenamento durável, escalável e seguro para armazenar e recuperar dados de maneira eficiente.

O S3 é projetado para oferecer uma solução simples e confiável para armazenamento de dados na nuvem. Ele permite que os usuários armazenem e recuperem qualquer quantidade de dados de forma rápida e segura. O serviço é altamente escalável, o que significa que ele pode lidar com qualquer volume de dados, desde algumas imagens até petabytes de informações.

Uma das principais características do S3 é a sua durabilidade. Os dados armazenados no S3 são distribuídos em várias zonas de disponibilidade para garantir redundância e resiliência. Isso significa que mesmo em caso de falha de hardware ou problemas em um local específico, os dados permanecem acessíveis e protegidos.

Além disso, o S3 oferece recursos avançados de segurança e controle de acesso. Os usuários podem definir políticas de acesso granulares, criptografar os dados em repouso e em

trânsito, e usar recursos como o AWS Identity and Access Management (IAM) para gerenciar as permissões de acesso.

Outra vantagem do S3 é a sua integração com outros serviços da AWS. É possível utilizar o S3 como um repositório de dados para outros serviços, como o Amazon EC2 para armazenamento de backups ou o Amazon Athena para análise de dados. Além disso, o S3 é compatível com uma ampla variedade de ferramentas e frameworks, facilitando a integração com aplicativos e fluxos de trabalho existentes.

Em resumo, o Amazon S3 é um serviço de armazenamento em nuvem que oferece escalabilidade, durabilidade e segurança para armazenar e recuperar dados de forma eficiente. Com recursos avançados de controle de acesso e integração com outros serviços da AWS, o S3 é uma solução versátil para o armazenamento de dados na nuvem, atendendo às necessidades de empresas de todos os tamanhos.

5.10.2. Amazon Simple Queue Service (SQS)

O Amazon SQS (Simple Queue Service) é um serviço de fila de mensagens oferecido pela Amazon Web Services (AWS). Ele fornece uma maneira simples e escalável de trocar mensagens entre diferentes componentes de aplicativos ou sistemas distribuídos.

O SQS é projetado para facilitar a comunicação assíncrona entre componentes, permitindo que eles se comuniquem de forma eficiente e confiável. Ele atua como um intermediário entre os remetentes das mensagens e os destinatários, garantindo a entrega confiável das mensagens em uma variedade de cenários.

As filas do SQS armazenam as mensagens enviadas pelos remetentes até que os destinatários as recebam e as processem. Isso permite que os componentes de um sistema trabalhem em seu próprio ritmo e sejam desacoplados uns dos outros. O SQS suporta uma variedade de padrões de mensagens, como fila única, fila FIFO (First-In-First-Out) e tópicos de publicação/assinatura.

Uma das principais vantagens do SQS é a sua alta escalabilidade. Ele pode lidar com um grande volume de mensagens e dimensionar automaticamente a capacidade de processamento com base na demanda. Isso garante que os sistemas possam lidar com picos de tráfego e manter um fluxo contínuo de mensagens, mesmo em situações de alto volume.

O SQS também oferece recursos avançados de segurança e durabilidade. Ele replica automaticamente as mensagens em várias zonas de disponibilidade para garantir a resiliência e a disponibilidade dos dados. Além disso, as mensagens podem ser criptografadas em repouso e em trânsito para garantir a proteção dos dados confidenciais.

Outra vantagem do SQS é a sua integração com outros serviços da AWS. É possível combinar o SQS com serviços como o Amazon Lambda, o Amazon EC2 e o Amazon S3 para criar arquiteturas de aplicativos altamente escaláveis e resilientes. Além disso, o SQS é compatível com uma variedade de linguagens de programação e oferece SDKs para facilitar a integração com aplicativos existentes.

Em resumo, o Amazon SQS é um serviço de fila de mensagens que fornece uma maneira confiável e escalável de trocar mensagens entre componentes de aplicativos ou sistemas distribuídos. Com recursos avançados de escalabilidade, segurança e integração, o SQS é uma solução poderosa para facilitar a comunicação assíncrona e o desacoplamento entre os componentes de um sistema distribuído.

5.10.3. Amazon Simple Notification Service (SNS)

O Amazon SNS (Simple Notification Service) é um serviço de mensagens e notificações em tempo real fornecido pela Amazon Web Services (AWS). Ele permite que os desenvolvedores enviem mensagens para uma variedade de destinos, como aplicativos móveis, dispositivos IoT, e-mails, mensagens de texto (SMS) e até mesmo serviços da AWS. Com o SNS, os usuários podem receber informações importantes, atualizações, alertas e notificações relevantes de forma rápida e confiável.

Uma das principais características do SNS é sua flexibilidade e escalabilidade. Ele permite o envio de mensagens para um único destinatário ou para um grande número de assinantes simultaneamente. Além disso, o SNS suporta tópicos de publicação/assinatura, onde os remetentes publicam mensagens em tópicos específicos e os assinantes recebem essas mensagens. Isso permite a criação de sistemas distribuídos e escaláveis, onde os assinantes podem se inscrever nos tópicos de seu interesse e receber apenas as notificações relevantes.

O SNS também oferece recursos avançados de segurança e monitoramento. Ele garante a entrega confiável das mensagens, utilizando a redundância em várias zonas de disponibilidade para evitar a perda de dados. Além disso, os usuários têm controle total sobre as permissões de acesso e podem configurar alertas e monitorar a atividade do SNS por meio de métricas e logs detalhados.

Em resumo, o Amazon SNS é uma solução versátil para enviar mensagens e notificações em tempo real para uma variedade de destinos. Com sua flexibilidade, escalabilidade e recursos de segurança, o SNS é uma escolha popular para desenvolvedores que desejam manter seus usuários atualizados e envolvidos por meio de comunicações rápidas e confiáveis.

5.10.4. Lambda

O AWS Lambda é um serviço de computação em nuvem oferecido pela Amazon Web Services (AWS). Ele permite que os desenvolvedores executem código sem a necessidade de provisionar ou gerenciar servidores. Com o Lambda, os desenvolvedores podem escrever funções e carregá-las na plataforma, que cuida da infraestrutura necessária para executar o código de maneira escalável e altamente disponível.

As funções do Lambda são acionadas por eventos, como a chegada de uma nova mensagem em uma fila do Amazon Simple Queue Service (SQS), o upload de um arquivo para o Amazon S3 ou a ocorrência de um evento do Amazon CloudWatch. Quando um evento ocorre, o Lambda inicia a execução da função correspondente, proporcionando uma resposta rápida e elástica.

Uma das principais vantagens do AWS Lambda é a sua capacidade de dimensionamento automático. À medida que a carga de trabalho aumenta ou diminui, o Lambda provisiona ou desativa automaticamente a capacidade de computação necessária para lidar com a demanda. Isso garante uma execução eficiente e otimizada do código, sem a preocupação com infraestrutura subjacente.

Em resumo, o AWS Lambda é um serviço de computação sem servidor que permite aos desenvolvedores executar código em resposta a eventos específicos. Com sua escalabilidade automática e gerenciamento simplificado da infraestrutura, o Lambda permite que os desenvolvedores se concentrem na lógica do aplicativo, sem se preocuparem com provisionamento ou administração de servidores.

5.10.5. DynamoDB

O Amazon DynamoDB é um serviço de banco de dados NoSQL totalmente gerenciado oferecido pela Amazon Web Services (AWS). Ele foi projetado para fornecer um armazenamento de dados escalável, de alta disponibilidade e baixa latência para aplicativos que precisam lidar com grandes volumes de dados e requisitos de desempenho exigentes.

Uma das principais características do DynamoDB é sua capacidade de dimensionamento automático. Ele pode lidar com cargas de trabalho variáveis, ajustando dinamicamente a capacidade de armazenamento e o desempenho do banco de dados conforme necessário. Isso significa que os aplicativos podem escalar facilmente sem a necessidade de ajustes manuais ou interrupções.

O DynamoDB também oferece uma estrutura de dados flexível, permitindo que os desenvolvedores armazenem e consultem dados de forma eficiente. Ele suporta a modelagem de dados em formato de documento ou de chave-valor, e os desenvolvedores podem criar tabelas com campos-chave primários simples ou compostos para atender às necessidades específicas de seus aplicativos. Além disso, o DynamoDB fornece recursos de segurança, como criptografia de dados em repouso e em trânsito, controle de acesso granular e integração com outros serviços de segurança da AWS. Isso garante a proteção dos dados armazenados no banco de dados.

Em resumo, o Amazon DynamoDB é um serviço de banco de dados NoSQL altamente escalável, projetado para fornecer armazenamento de dados rápido, flexível e de alta disponibilidade. Com sua capacidade de dimensionamento automático, estrutura de dados versátil e recursos de segurança avançados, o DynamoDB é uma escolha popular para aplicativos que precisam de um banco de dados eficiente e confiável na nuvem.

6. ATIVIDADES DESENVOLVIDAS

Devido a abrangente área de atuação assumida pela área de Automation na empresa BTGPactual com diversos projetos e de áreas diferentes internas do banco. É prática rotineira o estudo das principais das mais novas tecnologias utilizadas no mercado, e há sempre uma pressão para cada projeto seu ser melhor que o anterior, sempre utilizando novas tecnologias com intuito de redução de custo e velocidade de processamento.

As atividades desempenhadas no período de estágio eram voltadas, quase que exclusivamente, para o projeto de automatização de processos internos na empresa. Os projetos iam desde o levantamento das necessidades dos clientes e definições de projeto até o fornecimento da solução.

O estudo das ferramentas utilizadas para projetos também foi algo marcante no período de estágio. Além da utilização da linguagem Python, linguagem bastante conhecida por sua curva de aprendizado rápido e sua grande versatilidade, tecnologias como Excel, SQL, JotForm, Flask, HTML, JS, CSS, React, FastAPI, Micro serviços da AWS, CI/CD e outros fizeram-se presentes no desenvolvimento de projetos.

Além das atividades ligadas a programação em si dos projetos, o estágio também proporcionou a experiência muito enriquecedora de ter contato com diversas pessoas de áreas completamente diferentes da minha(clientes), possibilitando entender o funcionamento de diversas áreas necessárias no funcionamento de um banco de investimentos.

Participei de diversos projetos durante o estágio, sendo os principais listados abaixo:

6.1. Projeto de extração e batimento digital assets

O projeto teve como objetivo otimizar o processo de obtenção e reconciliação de dados das corretoras Gemini, Bitstamp e FTX, na área de OTC Crypto. Este relatório descreverá a ferramenta desenvolvida, os benefícios alcançados e a melhoria na eficiência operacional.

A ferramenta desenvolvida consiste em um sistema automatizado para extração, consolidação, geração de relatórios diários e reconciliação das operações realizadas nas corretoras mencionadas. Anteriormente, esse processo era realizado manualmente por meio de uma planilha no Excel, exigindo consultas frequentes aos sites das corretoras. Com a automação implementada, foi possível reduzir significativamente o tempo necessário para executar essas tarefas.

O principal benefício alcançado com a implementação da ferramenta foi a melhoria na eficiência operacional. A analista responsável pelo trabalho relatou uma economia de tempo

estimada em 1 hora e 30 minutos por dia, em comparação com o método manual anteriormente utilizado. Essa economia de tempo permitiu que a analista se concentrasse em outras tarefas de maior valor agregado.

Além disso, a automação contribuiu para aumentar a confiança nos resultados diários, reduzindo erros humanos que poderiam ocorrer durante a operação manual. A ferramenta garantiu maior precisão e consistência nos dados extraídos e reconciliados, melhorando a qualidade do fechamento diário das operações.

O projeto de extração e batimento de ativos digitais nas corretoras Gemini, Bitstamp e FTX proporcionou uma melhoria significativa na eficiência operacional da área de OTC Crypto. A ferramenta desenvolvida automatizou tarefas anteriormente realizadas manualmente, resultando em ganhos de tempo consideráveis e redução de erros. A implementação bem-sucedida desse projeto demonstra a importância da automação e tecnologia na otimização de processos e na melhoria dos resultados organizacionais.

6.2. Projeto de validação de documentos para pagamento de indenização de seguro rural

O projeto teve como objetivo otimizar o processo de validação de documentos para o pagamento de indenização de seguro rural, substituindo o método manual anteriormente utilizado por uma solução automatizada.

O projeto consistiu no desenvolvimento de uma tela de validação de documentos (figura 4) enviados pelos corretores para efetuação do pagamento de sinistros. Anteriormente, todo o processo era realizado manualmente por meio de envios de e-mails, cadastros manuais no sistema I4Pro e utilização de diversas planilhas para controle do fluxo. A nova solução automatizada simplificou e agilizou o processo de validação, proporcionando maior eficiência e redução de erros.

Figura 4: Tela de aprovação de documentos do Docmatch de sinistros.

A interface de aprovação de documentos do DocMatch apresenta uma barra superior com o logo 'btg pactual automação' e 'DocMatch', e o nome de usuário 'Henrique Spencer'. O formulário principal é dividido em seções: 'Dados do Sinistro' (com campos para Segurado, Corretor, Beneficiário, Favorecido e Dados de Pagamento), uma coluna 'Pendentes' com três cartões de documentos para revisão, uma coluna 'Aprovados' com um cartão de 'Aviso de Término de Coberto', e uma coluna 'Reprovados'. Um botão 'CONFIRMAR' está no topo direito.

Fonte: Elaborado Pelo Autor.

A automação do processo de validação de documentos trouxe os seguintes benefícios:

- **Eficiência:** A implementação da solução automatizada permitiu uma alavancagem de 3 horas por dia para o analista responsável. O tempo economizado possibilitou o cumprimento da SLA (Service Level Agreement) definida para o processo e proporcionou uma maior capacidade de atendimento a outros fluxos de trabalho.
- **Comunicação:** A automação mitigou diversos casos de má comunicação entre as partes envolvidas, que antes dependiam de e-mails manuais. A nova solução facilitou o compartilhamento de informações e a comunicação eficiente entre os corretores e a equipe responsável pela validação.
- **Protótipo para o DocMatch:** Esse projeto serviu como o primeiro protótipo para o desenvolvimento do DocMatch, uma solução mais ampla para validar documentos independentemente do fluxo específico. Os aprendizados obtidos com esse projeto foram fundamentais para o desenvolvimento de um validador de documentos e informações mais flexível para futuros fluxos de trabalho.

O projeto de automação da validação de documentos para pagamento de indenizações foi bem-sucedido e trouxe benefícios significativos para a área de Operations - Insurance Rural. A substituição do processo manual por uma solução automatizada melhorou a eficiência operacional, a comunicação entre as partes envolvidas e contribuiu para o desenvolvimento de soluções mais amplas, como o DocMatch.

6.3. Relatório Custódia de Criptomoedas(DRU - MYNT)

O projeto foi feito para área DRU - MYNT e teve como objetivo criar relatórios para a Receita Federal, exibindo todas as transações dos clientes que excederam o valor de R\$30.000 em movimentações no mês. Esses relatórios visavam cumprir com as obrigações regulatórias relacionadas à prestação de contas das transações realizadas pelos clientes.

A implementação desse projeto trouxe os seguintes benefícios:

- **Eficiência Regulatória:** Os relatórios criados permitiram atender às exigências da Receita Federal em relação à prestação de contas das movimentações de criptomoedas dos clientes. Com a automação desse processo, foi possível reduzir o tempo necessário para gerar os relatórios, otimizando a eficiência regulatória.
- **Cumprimento das Obrigações:** A criação dos relatórios facilitou o cumprimento das obrigações legais relacionadas à prestação de informações à Receita Federal. Isso ajudou a evitar problemas regulatórios e a manter a conformidade com as normas vigentes.

O projeto de custódia de criptomoedas no âmbito da área DRU - MYNT foi um sucesso em termos de eficiência regulatória. A criação dos relatórios para a Receita Federal, que detalhavam as movimentações dos clientes que excediam R\$30.000 no mês, permitiu cumprir as obrigações legais de prestação de contas.

A automação desse processo resultou em uma melhoria significativa na eficiência operacional, reduzindo o tempo necessário para gerar os relatórios e garantindo o cumprimento das obrigações regulatórias de forma precisa e eficiente.

Assim, essa experiência proporcionou uma compreensão mais aprofundada das demandas regulatórias e das medidas necessárias para garantir a conformidade em um ambiente que envolve criptomoedas.

6.4. Projeto de validação de documentos do pagamento de vistorias aos peritos

Automatiza todo o processo de validação de documentos para o ressarcimento de despesas dos peritos feitas durante a realização de vistorias.

Era realizado de maneira manual pela cliente (esse fluxo está bem descrito na imagem abaixo), tendo que disparar diversos emails ao dia e utilizava-se diversas planilhas para o controle do processo. Além disso, o banco tinha uma postura mais passiva no processo, esperando que a cobrança dos serviços vinha por parte das reguladoras. Todo esse fluxo de como funcionava antes está desenhado na figura 5.

```
graph LR; Regulatora1[Regulatora] -- "Vistorias  
Notas Fiscais Perito  
Notas Fiscais Reguladora" --> BO[Back Office]; BO -- "Validação" --> PA[Pagamento Perito Atualizada]; PA --> PP[Portal de Pagamentos]; PP --> CP[Controle Pagamentos Peritos- Consolidado]; CP --> ANP[ANP Mes]; ANP --> HD[Honorários e Despesas sinistro]; HD --> BO; BO -- "Reenvio" --> Regulatora2[Regulatora];
```

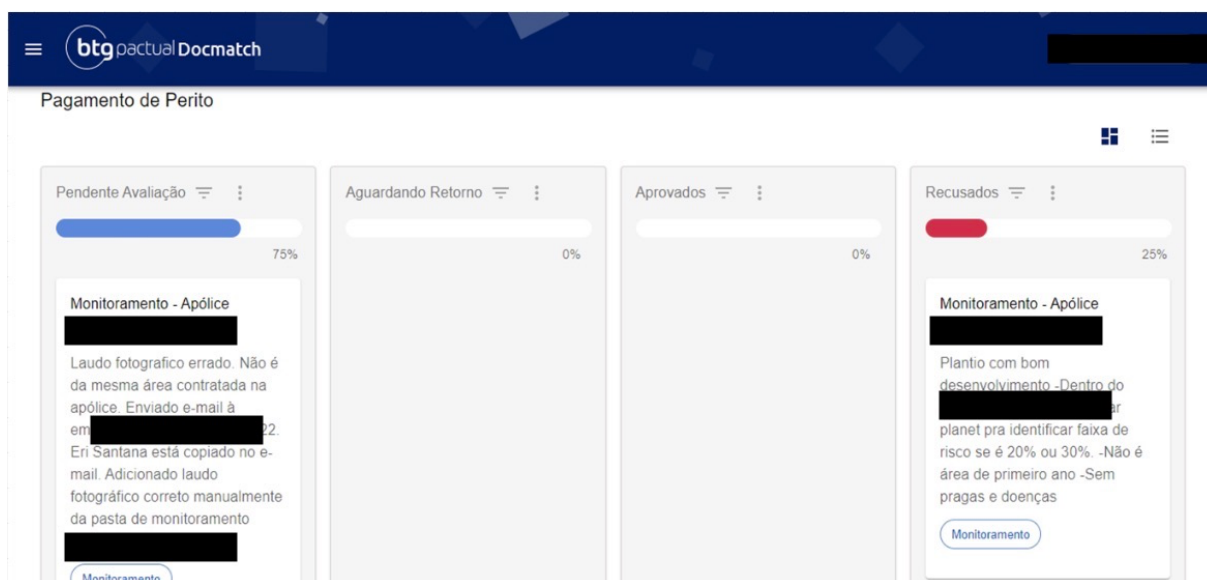
Esse fluxo tem uma grande vazão, cerca de 600 casos ao mês. Assim, nossa automação trouxe um bom ganho de eficiência para o analista, estimado em 3h diárias, além disso ele está sendo responsável por mitigar os erros operacionais que ocorriam no processo e demonstrar uma postura mais ativa por parte do nosso banco, onde, agora nos que pedimos as reguladoras as notas fiscais e os dados para o devido pagamento das vistorias, assim, não ficamos mais de maneira passiva esperando a cobrança das reguladoras. Esse novo fluxo esta bem descrito na figura 6.

Fonte: Elaborado Pelo Autor.

Outro ponto a ser destacado era como era feito o cálculo do custo da gasolina, ninguém tinha percebido que no calculo estava sendo considerado um carro com um consumo muito baixo, na casa dos 3Km/L, tivemos uma call sobre o assunto com a head da área de seguro rural que isso acarretava diversas distorções entre as vistorias longe e as mais perto, onde o perito preferia vistorias mais longes e também se deslocar de carro do que ir de avião. Infelizmente, não conseguimos mudar a maneira que é feito esse calculo, mais estamos gerando um relatório mensal mostrando as diferenças dos custos pagos e os reais, para que posteriormente possa ser feita uma reestruturação da distribuição do pagamento de um laudo.

Por fim, esse projeto foi um meio para desenvolvermos um docmatch mais intercambiável que irá servir para validação de documentos/informações em outros fluxos. A tela gerencial do projeto pode ser visualizada na figura 7.

Figura 7: Tela gerencial do pagamento de perito.



Fonte: Elaborado Pelo Autor.

6.5. Auditoria de Operações no FTS

O projeto foi desenvolvido para área FICC & EQUITIES/ALM e teve como objetivo garantir que as operações aprovadas pela equipe de ALM estivessem alinhadas com a realidade. Foi desenvolvido um relatório diário que reportava todas as operações falhas cadastradas no FTS (Sistema de Gerenciamento de Operações) um exemplo desse reporte esta na figura 8.

Figura 8: Exemplo de relatório enviado pelo projeto de Auditoria de Operações no FTS.

Batimento operações FTS - 30/06

BTG Pactual | Automation 2

To: [Redacted]
Cc: [Redacted]

Retention Policy: Brazil – Email Retention Policy 5 Years - 8 (5 years) Expires: 28/06/2028

If there are problems with how this message is displayed, click here to view it in a web browser.

erros_grid_fts.xlsx
23 KB

Translate message to: English | Never translate from: Portuguese (Brazil) | Translation preferences

btg pactual automation **Acompanhamento Operações ALM**

Prezado(a), segue abaixo os erros encontrados no **batimento da FTS**:

Arquivos avaliados no **batimento**
FTS: Grid **FTS** 20230630 20h.xlsx
 Entuba: EntubaPre20230630.xlsx

Boletas revisadas: 5165
 Boletas com erros: [Redacted]
 Erros encontrados: [Redacted]

TradeID	Categoria	Taxa FTS	Taxa Entuba	Financeiro	Operador	Produto	Indexador	Data Venc.
6894655	10- Erro taxa simples	[Redacted]	[Redacted]	[Redacted]	[Redacted]	LFSN	IPCA	01/07/2033
6870259	10- Erro taxa simples 11- Sem taxa agente	[Redacted]	[Redacted]	[Redacted]	[Redacted]	LFSN	IPCA/IPCA	15/07/2033 15/07/2033
.....	10- Erro taxa simples 11- Sem	[Redacted]	[Redacted]	[Redacted]	[Redacted]	[Redacted]	[Redacted]	[Redacted]

Fonte: Elaborado Pelo Autor.

O projeto consistiu no desenvolvimento de um relatório diário que identificava e reportava todas as operações falhas cadastradas no FTS. Essas operações passavam por uma análise do time de ALM e era necessário garantir que estivessem aderentes à realidade. Durante o desenvolvimento do projeto, uma das principais mudanças foi a inclusão de uma tela que permitia à usuária modificar diversas regras utilizadas na verificação das operações.

A implementação desse projeto trouxe os seguintes benefícios:

- **Auditoria de Operações:** O relatório diário possibilitou uma auditoria eficiente das operações cadastradas no FTS. Com a identificação das operações falhas, foi possível realizar uma análise minuciosa e garantir que estivessem em conformidade com as regras estabelecidas pelo time de ALM. Isso proporcionou uma melhor governança e confiabilidade dos dados cadastrados no sistema.
- **Mitigação de Riscos Operacionais:** A auditoria das operações no FTS ajudou a mitigar riscos operacionais. Ao identificar e corrigir operações falhas ou inconsistentes, foi possível evitar problemas futuros, garantindo a integridade e a qualidade dos dados utilizados nas atividades da área.

O projeto de auditoria de operações no FTS foi concluído com sucesso, entregando o relatório diário e a tela para modificação de regras. Essa solução proporcionou uma auditoria eficiente das operações cadastradas, melhorando a governança e a confiabilidade dos dados no sistema.

Durante o desenvolvimento do projeto, foram enfrentadas diversas mudanças, mas a inclusão da tela de modificação de regras foi uma das principais. Essa funcionalidade permitiu maior flexibilidade e controle para a usuária, contribuindo para uma auditoria mais precisa das operações.

Os ganhos obtidos com o projeto foram significativos, pois proporcionaram maior confiabilidade dos dados, mitigação de riscos operacionais e uma melhor governança das operações no FTS.

7. CONCLUSÃO

O estágio obrigatório realizado na empresa BTG Pactual foi fundamental para meu desenvolvimento profissional, acarretando um aprofundamento de conhecimentos teóricos obtidos em sala de aula e em novos aprendizados adquiridos através do cotidiano da empresa.

Destaca-se a importância de três disciplinas do curso de Engenharia Elétrica para o desenvolvimento das atividades do estágio: Algoritmos e Lógica de Programação, Programação avançada e Estatística Aplicada a Engenharia Elétrica. A primeira é fundamental ensina vários conceitos básicos logica de programação necessários para ser ter bom pensamento lógico. A segunda concentra principalmente nos paradigmas da Programação Orientada a Objetos (POO), explorando os conceitos fundamentais da POO, como classes, objetos, herança, polimorfismo e encapsulamento que foram de extrema importância para escrever códigos claros e assertivos. Já a terceira é imprescindível por ser uma matéria cuja ementa permite ao aluno desenvolver um pensamento crítico e mais embasado ao analisar grandes massas de dados, esses conhecimentos são extremamente valiosos na análise de dados usando programação, pois permitem que os programadores compreendam a natureza dos dados, identifiquem padrões, realizem testes estatísticos e tomem decisões embasadas em evidências estatísticas

Em conclusão, posso afirmar que minha experiência durante o estágio foi extremamente proveitosa. Fui exposto a desafios complexos, que me incentivaram a buscar soluções inovadoras e aprimorar minhas habilidades técnicas. Além disso, trabalhar em equipe com profissionais experientes proporcionou um ambiente de aprendizado enriquecedor, no qual pude aprender com suas experiências e receber feedback construtivo. Essa experiência me proporcionou um crescimento profissional significativo, ao mesmo tempo em que me permitiu contribuir para os objetivos e resultados da organização. Estou grato pela oportunidade de estagiar e confiante de que as lições aprendidas serão valiosas para minha carreira futura.

8. REFERÊNCIAS

AMARAL JÚNIOR, Odravison. Arquitetura de micro serviços: uma comparação com sistemas monolíticos. 2017.

ROSTAMI MAZRAE, Pooya et al. On the usage, co-usage and migration of CI/CD tools: A qualitative analysis. Empirical Software Engineering, v. 28, n. 2, p. 52, 2023.