

\_\_\_\_\_

PÓS-GRADUAÇÃO

**XP Educação**

**Relatório do Projeto Aplicado**

Modelo Arquitetural para solução refatorada do Sistema de Gestão do Mercado

Roger Kayque Primiani Freitas

Orientador(a): Dr. Lucas Wilman da Silva Crispim

Setembro de 2024

****

Roger Kayque Primiani Freitas

**XP EDUCAÇÃO**RELATÓRIO DO PROJETO APLICADO

Modelo Arquitetural para solução refatorada do Sistema de Gestão do Mercado

Relatório de Projeto Aplicado desenvolvido para fins de conclusão do curso Arquitetura de software e soluções.  
  
Orientador (a): Dr. Lucas Wilman da Silva Crispim

**São José dos Campos  
setembro de 2024**

**Sumário**

[1. CANVAS do Projeto Aplicado 4](#_Toc181398097)

[1.1 Desafio 5](#_Toc181398098)

[1.1.1 Análise de Contexto 5](#_Toc181398099)

[1.1.2 Personas 7](#_Toc181398100)

[1.1.3 Benefícios e Justificativas 10](#_Toc181398101)

[1.1.4 Hipóteses 12](#_Toc181398102)

[1.2 Solução 13](#_Toc181398103)

[1.2.1 Objetivo SMART 13](#_Toc181398104)

[1.2.2 Premissas e Restrições 13](#_Toc181398105)

[1.2.3 Backlog de Produto 15](#_Toc181398106)

[2. Área de Experimentação 16](#_Toc181398107)

[2.1 Sprint 1 16](#_Toc181398108)

[2.1.1 Solução 16](#_Toc181398109)

[● Evidência do planejamento: 16](#_Toc181398110)

[● Evidência da execução de cada requisito: 16](#_Toc181398111)

[● Evidência dos resultados: 18](#_Toc181398112)

[2.1.2 Lições Aprendidas 19](#_Toc181398113)

[2.2 Sprint 2 22](#_Toc181398114)

[2.2.1 Solução 22](#_Toc181398115)

[● Evidência do planejamento: 22](#_Toc181398116)

[● Evidência da execução de cada requisito: 22](#_Toc181398117)

[● Evidência dos resultados: 24](#_Toc181398118)

[2.2.2 Lições Aprendidas 26](#_Toc181398119)

[2.3 Sprint 3 27](#_Toc181398120)

[2.3.1 Solução 27](#_Toc181398121)

[● Evidência do planejamento: 27](#_Toc181398122)

[● Evidência da execução de cada requisito: 27](#_Toc181398123)

[● Evidência dos resultados: 31](#_Toc181398124)

[2.3.2 Lições Aprendidas 33](#_Toc181398125)

[3. Considerações Finais 36](#_Toc181398126)

[3.1 Resultados 36](#_Toc181398127)

[3.2 Contribuições 38](#_Toc181398128)

[3.3 Próximos passos 39](#_Toc181398129)

## 

## 1. CANVAS do Projeto Aplicado

Figura conceitual, que representa todas as etapas do Projeto Aplicado.

Uma imagem contendo Interface gráfica do usuário

Descrição gerada automaticamente

## Desafio

### 1.1.1 Análise de Contexto

A rede de mercados FictMarket do Brasil, que iniciou sua jornada como um pequeno mercado local, e devido seu sucesso, teve uma expansão significativamente grande, hoje constando com mais de 100 mercados espalhados pelo sudeste do Brasil, não expandiu sua solução tecnológica na mesma proporção.

Hoje, essa rede lida com sistemas descentralizados, onde as informações de estoque e financeiros são assíncronas e muitas vezes desatualizadas.

Parte dos dados ainda é registrada em planilhas por um sistema legado em VB, e posteriormente consolidadas manualmente, cujos arquivos são enviados por FTP ao time financeiro.

Essa falta de integração e automação tem gerado perdas significativas, desde problemas de estoque até erros financeiros, comprometendo a operação e a tomada de decisões estratégicas.

Foram identificadas as verdadeiras causas dos problemas enfrentados, através das seguintes percepções e informações baseada em reuniões com as personas utilizadoras dos sistemas de gestão da empresa, sendo elas:

* Falta de integração dos sistemas, por serem descentralizados e com atualizações assíncronas, o que compromete a precisão e eficácia da operação;
  + A falta de integração entre os sistemas, ocasiona a desatualização e consistência dos dados, prejudicando a tomada de decisões.
* Dependência de processos manuais, onde parte dos dados possui registros em planilhas e posteriormente é consolidade manualmente;
  + Processos manuais são suscetíveis a erros humanos e ineficiências, causando problemas de precisão e atrasos na gestão dos dados.
* Transmissão de dados por FTP externo, riscos de segurança, possibilidade de erros na transferência, podendo ser considerado, para essa situação um método antiquado;
  + Provedores externos FTP para transmissão e armazenamento de dados, pode resultar em problemas de segurança e confiabilidade no processo.
* Falta de automação, nos processos de integração e consolidação dos dados, impacta diretamente a eficiência da operação, e mais suscetível a erros humanos.
  + A automação poderia reduzir a carga de trabalho manual e melhorar a eficiência dos processos.

Matriz CSD:

Uma imagem contendo Calendário

Descrição gerada automaticamente

Observação do tipo POEMS:

Tela de celular

Descrição gerada automaticamente com confiança média

### 1.1.2 Personas

|  |  |
| --- | --- |
| Mulher com cabelo curto  Descrição gerada automaticamente com confiança média | Ana Paula Souza  **Idade:** 42 anos  **Cargo:** Analista Contábil Sênior  **Área de Trabalho:** Contabilidade no FictMarket  **Formação:** Graduada em Contabilidade. |
| **Experiência Profissional:**   * Possui mais de 10 anos de experiência na área contábil, com uma carreira sólida em empresas de médio e grande porte. | |
| **Características Comportamentais:**   * Responsável, detalhista, focado na conformidade e eficiência. Tem uma abordagem prática para resolver problemas e valoriza a precisão. | |
| **O que fala e faz:**   * No FictMarket, ela é responsável pela conciliação financeira, preparação de relatórios contábeis e análise de indicadores financeiros. | |
|  | |
|  | |

|  |  |
| --- | --- |
| Homem de óculos sorrindo  Descrição gerada automaticamente | Felipe Martins  **Idade:** 34 anos  **Cargo:** Analista de Sistemas  **Área de Trabalho:** TI  **Formação:** Ciência da Computação |
| **Experiência Profissional:**   * Análise, desenvolvimento e implementação de sistemas de informação, melhoria da infraestrutura tecnológica. | |
| **Características Comportamentais:**   * Proativo, curioso e orientado para solução de problemas. Interessado em novas tecnologias e na resolução de desafios técnicos. | |
| **O que fala e faz:**   * No FictMarket, trabalha na manutenção das aplicações legadas buscando entregar melhorias contínuas que auxiliem nas necessidades especificadas naquele momento pelo negócio. | |
|  | |
|  | |

|  |  |
| --- | --- |
| Mulher sorrindo posando para foto  Descrição gerada automaticamente | Mariana Costa  **Idade: 38 anos**  **Cargo: Administradora**  **Área de Trabalho: Administrativo**  **Formação: Administração de Empresas** |
| **Experiência Profissional:**   * Coordenação das operações diárias do escritório, gerenciamento de recursos humanos e apoio aos gestores na tomada de decisões administrativas. | |
| **Características Comportamentais:**   * Organizada, comunicativa e focada em eficiência. Valoriza a gestão eficaz de recursos e o equilíbrio entre trabalho e vida pessoal. | |
| **O que fala e faz:**   * No FictMarket, Mariana Costa é responsável, pela operação administrativa e pela gestão de recursos humanos. | |
|  | |
|  | |

### 1.1.3 Benefícios e Justificativas

A proposta de valor do projeto de refatoração da solução sistêmica do FictMarket, consiste na transformação da infraestrutura tecnológica da empresa, gerando um ambiente e trabalho eficiente e integrado. Como justificativa para a execução deste projeto e resolução do desafio/problema apresentado, enfatizamos:

* Eficiência operacional: Com a implementação de uma solução integrada e automatizada, o processo contará com a centralização das informações, resultando em uma operação mais eficiente e com menor propensão a erros.
* Redução de custos: A modernização dos sistemas e a automação dos processos, tendem a reduzir custos de “toils” operacionais, e, com a diminuição do suporte técnico extensivo nos sistemas legados. É esperado também, ter uma redução de erros financeiros e no estoque, que atualmente são considerados problemas contábeis.
* Melhoria na receita: Com a integração dos sistemas e centralização dos dados atualizados, a empresa terá capacidade de identificação de padrões e tendencias de consumo, dado, a empresa continua crescendo e que cada mercado, está em uma região, e cada região possui um padrão de venda de produtos.
* Impactos sociais e ambientais: A melhora na eficiência operacional tende a permitir melhoria na alocação dos recursos, gerando impacto positivo na responsabilidade social corporativa. Utilizando um sistema completamente integrado, é possível que os escritórios administrativos da empresa, passem a utilizar menos papel, o que a logo prazo trará um impacto ambiental positivo.
* Melhoria na tomada de decisões: A integração de sistemas e a redução de processos manuais, proporcionarão dados atualizados e relatórios mais precisos, permitindo aos gestores tomarem decisões informadas com base em números.

Blueprint:

Tabela

Descrição gerada automaticamente

Proposta de Valor:

Diagrama

Descrição gerada automaticamente com confiança média

### 1.1.4 Hipóteses

Concluindo esta análise detalhada, e a coleta das informações relevantes, foi possível organizar um conjunto de ideias para guiar o desenvolvimento do projeto. A tabela de observações das hipóteses, sintetizará o material acumulado até o momento, facilitando a priorização das atividades que exigem atenção imediata.

Matriz de hipóteses:

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Priorização de Ideias.

Calendário

Descrição gerada automaticamente

## 1.2 Solução

### 1.2.1 Objetivo SMART

Elaborar um Documento de Arquitetura de Software (DAS) durante 3 sprints, totalizando 45 dias, para orientar a construção de uma solução em micro serviços hospedados em um provedor em nuvem (AWS). A solução deve ser escalável e fácil de usar e manter, aumentando a eficiência operacional dos processos, tornando uma geração de relatório com dados históricos que hoje levam dias, em um processo executado em menos de 3 hora, e garantir a integridade dos dados em 100%, pois não haverá mais processos manuais. O projeto contará com uma equipe de arquitetura e desenvolvimento composta por membros do time de TI internos, que serão responsáveis pela entrega dos serviços. Com base no planejamento e na capacidade atual da equipe, o tempo de 45 dias foi definido como realista. A integração com AWS, um provedor familiar para a equipe, bem como a linguagem .NET contribuirá para garantir a viabilidade da solução.  
O objetivo é garantir que a tecnologia facilite o crescimento da empresa.

### 1.2.2 Premissas e Restrições

Premissas:

* O sistema legado deve manter a operação até que todas as validações e transições para o novo sistema estejam concluídas.
* Dados financeiros e administrativos devem ser confiáveis e acessíveis;
* A infraestrutura de TI deve ser capaz de suportar a nova arquitetura bem como sua escalabilidade;

Restrições:

* A complexidade de gestão de micro serviços pode aumentar a complexidade na manutenção do sistema;
* A integração dos dados gerados pelo legado pode ser um desafio;
* A implementação do data mesh deve ser eficiente;
* O custo da infraestrutura em nuvem, deve ser avaliado considerando a escalabilidade do processo.

Matriz de Riscos:

Uma imagem contendo Interface gráfica do usuário

Descrição gerada automaticamente

### 1.2.3 Backlog de Produto

Com o objetivo SMART formado e alinhado, foram criados o Backlog e refinamento para execução nas próximas sprints. [Acesso ao Backlog.](https://trello.com/invite/b/66e768cbcfcda6d81b3e3dc9/ATTIe510e4b3bea3fe7893d456d5370bd86dA40D0AB3/das-fictmarket)

Tela de celular com fundo preto

Descrição gerada automaticamente

# 2. Área de Experimentação

## 2.1 Sprint 1

### 2.1.1 Solução

#### Evidência do planejamento:

A sprint 1 consiste na identificação e mapeamento dos requisitos e processos, a fim de garantir que a nova solução entregue, supere as expectativas das personas envolvidas, e agregue valor a empresa.

Tela de celular com fundo preto

Descrição gerada automaticamente

#### Evidência da execução de cada requisito:

O mapeamento dos processos operacionais da rede de supermercados FictMarket foi elucidado com a ferramenta Draw.io, visando documentar detalhadamente o fluxo de atividades e as interações entre os setores. Esse mapeamento permitiu a visualização clara e precisa dos processos, facilitando a identificação de lacunas operacionais e ineficiências no sistema atual. Com a documentação estruturada, foi possível propor soluções para essas lacunas, atendendo a gama de requisitos funcionais e não funcionais que visam otimizar as operações e garantir maior eficiência nos processos internos.

Mapeamento em Draw IO:

Uma imagem contendo Texto

Descrição gerada automaticamente

Com o mapeamento dos processos, foi possível trazer uma visão mais clara dos requisitos funcionais e não funcionais para atender a expectativa das personas supracitadas neste projeto.

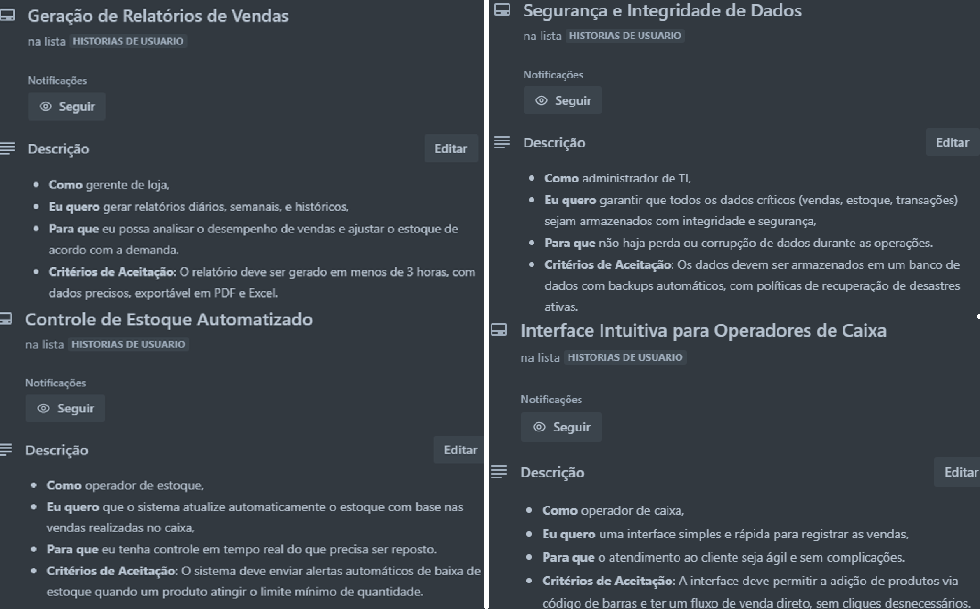
Requisitos:

Uma imagem contendo Tabela

Descrição gerada automaticamente

#### Evidência dos resultados:

Ao entender o processo e coletar os requisitos, analisando as necessidades, foram geradas no [Trello](https://trello.com/invite/b/66f4875c2a44b644aa82f71f/ATTI74adf31868c184c0ed1f5098724c1f017B4C30E3/sprint-1), as histórias de Usuário para elucidar as expectativas do cliente:



Também foi possível mapear o novo processo operacional, descrito através do diagrama abaixo:

Diagrama

Descrição gerada automaticamente

### 2.1.2 Lições Aprendidas

Durante a sprint 1, como planejado, realizamos o mapeamento dos processos operacionais da rede de supermercados FictMarket, e uma análise detalhada dos requisitos funcionais e não funcionais para desenvolvimento da solução. Esse processo foi fundamental para estabelecermos a base do projeto. Como lições aprendidas da sprint, conseguimos ver alguns tópicos e seus respectivos sucessos e pontos a serem melhorados:

* Comunicação com Stakeholders:
  + O que deu certo:
    - O envolvimento contínuo dos stakeholders do projeto, permitiu a compreensão dos processos críticos;
    - O envolvimento das personas citadas, ajudaram a identificar os principais desafios e melhorias esperadas no processo atual.
  + O que podemos melhorar:
    - Reuniões precisam de otimização de tempo, e evitar dispersões.
* Mapeamento dos processos:
  + O que deu certo:
    - O mapeamento de processos foi abrangente e identificou as áreas que necessitam de automação e melhorias, como controle de estoque, problemas na reposição de produtos e geração de relatórios financeiros.
    - A identificação dos gargalos operacionais foi clara e bem documentada.
  + O que podemos melhorar:
    - Alguns processos, como a interação com os sistemas de terceiros (ERP, gateways de pagamento) não puderam ser mapeados na sprint 1 devido a conflitos de agenda. Seria interessante abordar os processos adjacentes desde o início para evitar retrabalho nas próximas sprints.
* Análise de requisitos Funcionais e não funcionais:
  + O que deu certo:
    - A identificação dos requisitos funcionais, como controle de estoque, vendas em tempo real e integração com sistemas administrativos, foi eficiente.
    - Os requisitos não funcionais relacionados à escalabilidade, segurança de dados, e performance do sistema foram bem estabelecidos, alinhando-se às melhores práticas de AWS e micro serviços.
    - Conseguimos criar um backlog inicial claro e priorizado, alinhado com as necessidades operacionais.
    - A análise de requisitos de usabilidade foi bem definida, especialmente no que diz respeito à facilidade de uso para os usuários finais.
  + O que podemos melhorar:
    - Requisitos de disponibilidade e recuperação de desastres precisaram ser revisados mais de uma vez, pois nem todos os stakeholders estavam alinhados com as implicações de tempo de inatividade.
    - Não existe uma documentação de requisitos de performance e tempo de resposta, especialmente em relação aos sistemas de relatórios em tempo real.

## 2.2 Sprint 2

### 2.2.1 Solução

#### Evidência do planejamento:

A sprint 2 envolve a elaboração do design da nova solução, alinhada aos requisitos e processos identificados. O objetivo é assegurar que a nova solução atenda, e supere as expectativas das personas envolvidas.

Tela de celular com texto preto sobre fundo branco

Descrição gerada automaticamente

#### Evidência da execução de cada requisito:

Durante a abordagem de priorização das necessidades do cliente em relação ao produto oferecido, foi utilizado o Kano Model, por se tratar de uma abordagem eficaz que nos ajuda a entender melhor as expectativas e traz ao desenvolvimento mais que as necessidades básicas, as experiencias que superam as expectativas das personas.

Kano Model:

Tabela

Descrição gerada automaticamente

A solução de data mesh para o consumo das planilhas armazenadas no FTP foi reavaliada, assim como o uso de arquiteturas baseadas na utilização do AWS GLUE.  
Devido ao grande volume de arquivos a serem processados, o custo dessas alternativas seria elevado.

Como o processo não será executado de forma recorrente e não haverá reaproveitamento contínuo, optamos por implementar uma solução utilizando o AWS Batch e um script na linguagem Python, com foco em otimizar os custos (FinOps) no pós projeto.

O desenho da arquitetura de dados da solução proposta:

Interface gráfica do usuário

Descrição gerada automaticamente

Já a nova solução do sistema hospedada na AWS, terá uma arquitetura em micro serviços na qual os serviços se comunicarão via SQS para que ambos tenham seu relacionamento em tempo real. A estrutura de dados escolhida foi um banco de dados RDS e um bucket S3 para manutenção de arquivos.

Os serviços utilizados na nuvem foram definidos por custo X benefício, sendo assim, utilizamos o máximo possível de serviços serverless e com melhor custo, por este motivo, não criamos uma arquitetura Cloud Agnostic:  
  
Tela de celular com aplicativos

Descrição gerada automaticamente

#### Evidência dos resultados:

Ao iniciar o desenho da nova arquitetura, pudemos elucidar o processo de consumo dos arquivos XLS gravados no FTP. Analisando as necessidades das personas envolvidas, identificamos que a criação de um data mesh traria um custo e um tempo de projeto ainda maior, onde possibilitamos a carga dos arquivos para um RDS através de um script em Python e uma arquitetura com AWS Batch, trazendo o resultado esperado e garantindo a eficiência financeira ao projeto.

Com a visão simplificada abaixo, é possível entender o resultado da arquitetura dos dados, a forma que o usuário vai acessá-lo e, os recursos que não terão custo recorrente, bem como aqueles que podem ser desativados:

Diagrama

Descrição gerada automaticamente

Com a elucidação da nova arquitetura, é possível também apresentar a funcionalidade do novo serviço, onde, temos um único sistema, trabalhando em módulos, garantindo que os trabalhos que são manuais, estão sendo realizados pelos próprios micros serviços através da comunicação assíncrona gerida pelo SQS.

O Nível 1 do C4 Model da nova solução, é capaz de demonstrar as particularidades do projeto:

Diagrama

Descrição gerada automaticamente

### 2.2.2 Lições Aprendidas

* Arquitetura da solução e Otimização de custos:
  + O que deu certo:
    - A troca do Data Mesh pelo AWS Batch foi uma decisão acertada para otimizar os custos, reduzindo o custo de infraestrutura com processamentos esporádicos e volumosos;
    - O AWS Batch se mostrou uma solução eficiente e escalável para cargas pontuais, o que atende a necessidade, sem a complexidade de uma arquitetura mais robusta.
    - A flexibilidade de trabalhar em nuvem, permitiu o ajuste da arquitetura conforme necessidade do projeto, sem comprometer o orçamento, a performance e a escalabilidade.
  + O que podemos melhorar:
    - Um ponto de atenção ao dimensionamento dos recursos computacionais no AWS Batch, para melhoria do desempenho no processamento de planilhas extensas sem impactar o orçamento do projeto;
    - Visto o ponto anterior, podemos melhorar o planejamento de custos futuros para termos mais detalhes, que considerem diferentes cenários de uso, otimizando ainda mais as despesas a longo prazo.
* Automação de processos e integração dos dados:
  + O que deu certo:
    - A migração dos processos manuais para uma solução automatizada com AWS batch melhorou significativamente a eficiência dos processos operacionais e reduziu a possibilidade de erros humanos;
    - Tal automatização, que antes envolvia a manipulação manual, foi um grande avanço na integridade e confiabilidade dos dados.
  + O que podemos melhorar:
    - Com o fim da utilização do FTP na nova arquitetura, o processo de geração dos relatórios será em tela. Os arquivos XLS gerados até o momento, serão armazenados no S3. O custo pode ser otimizado com a utilização do Glacier. Para isso, é necessário termos bem definido na regra de negócio, quais os períodos históricos que precisamos ter para recuperação rápida dos arquivos, e se realmente isso será necessário, visto que os dados estão disponíveis em um RDS.

## 2.3 Sprint 3

### 2.3.1 Solução

#### Evidência do planejamento:

A sprint 3 possui como objetivo a revisão da arquitetura e código bem como a elaboração da documentação técnica da nova solução.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

#### Evidência da execução de cada requisito:

Revisão do script que realizará a carga dos arquivos salvos na nova estrutura criada, trazendo os dados para o ambiente RDS e salvando o backup no bucket S3.

A linguagem escolhida para a carga no AWS Batch foi o Python por oferecer robustez e bibliotecas que facilitam a manipulação de arquivos XLS/CSV e fácil integração com os serviços da AWS via boto3, o que simplifica a operação de escrita no RDS e upload no S3. A curva de aprendizado é baixa e a eficiência em processamento, mostra que é ideal para o objetivo. Além do mais, Python apoia os princípios de FinOps, oferecendo uma solução econômica, escalável e alinhada ao modelo pay-as-you-go do AWS Batch.

A solução completa, com o código, imagens e os diagramas podem ser visualizados em <https://github.com/Rogerprm/XPEPosARS>.

Texto

Descrição gerada automaticamente

Durante a revisão da arquitetura, realizamos a validação através do C4 model – Nível 3, visto que no mesmo temos o detalhamento dos componentes individualmente da solução, com clareza na estrutura, com isso, além de validar o design da arquitetura, podemos garantir que esta alinhada com os objetivos.

Tela de computador com fundo preto

Descrição gerada automaticamente

Após validar a arquitetura, o próximo passo é gerar a infraestrutura como código (IaC) na AWS, automatizando a criação e manutenção dos serviços descritos na arquitetura, garantindo consistência e agilidade. Utilizamos o Terraform dado os benefícios entregues, sendo os principais, dentre eles, a portabilidade (Cloud Agnostic), controle de versionamento e escalabilidade.

A solução completa, com o código, imagens e os diagramas podem ser visualizados em <https://github.com/Rogerprm/XPEPosARS>.

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Com isso, iniciamos o processo de documentar tecnicamente a nova solução com o intuito de termos as etapas detalhadas e compreensíveis para as futuras manutenções e melhorias.

Texto

Descrição gerada automaticamente

A Documentação completa, solução, com o código, imagens e os diagramas podem ser visualizados em <https://github.com/Rogerprm/XPEPosARS>.

#### Evidência dos resultados:

Durante a sprint realizamos a revisão e a documentação. As atividades foram cruciais para garantir que a nova arquitetura da solução tivesse validada, documentada e pronta para rodar em ambiente produtivo.

A arquitetura foi cuidadosamente revisada e ajustada com base no custo X benefício da solução. A principal modificação foi a substituição do mesh pelo AWS Batch, visando a melhor estratégia FinOps e eficiência na entrega. Conforme o detalhamento executado no C4 Model nível 3 evidenciou, as integrações foram validadas e otimizadas.

C4 Nível 3 com foco nas integrações:  
Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Foi gerado também a documentação técnica detalhada da solução e disponibilizada no repositório do projeto. Possuindo a descrição da infraestrutura AWS utilizada e os micro serviços que compõem a solução.

Texto

Descrição gerada automaticamente

A infraestrutura foi automatizada, utilizando o terraform para provisionar toda a infraestrutura na aws:

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Foi criado um repositório para armazenar toda a solução, organizado com os códigos de infra, scripts e a documentação do projeto.

Tela de celular com publicação numa rede social

Descrição gerada automaticamente

### 2.3.2 Lições Aprendidas

* **Revisão e Validação da Arquitetura:**
  + **O que deu certo:**
* A substituição da arquitetura inicial para uso do **AWS Batch** foi uma escolha acertada, garantindo redução de custos ao evitar o **AWS Glue** e **Data Mesh**. A solução atendeu às necessidades do projeto e trouxe maior eficiência.
* A validação da arquitetura garantiu uma implementação eficiente, o que contribuiu com o cronograma previsto.
* A integração com o **FTP** pode ser eliminada, garantindo maior segurança e consistência ao fluxo de dados, com a migração total para o **S3**.
* **O que podemos melhorar:**
* Ainda que o **AWS Batch** tenha sido eficiente, uma análise mais aprofundada de outras soluções para processamentos esporádicos poderiam ser feitas, caso futuramente, seja necessário novos processamentos recorrentes, visando ainda mais otimizações de custo.
* **Documentação Técnica:**
* **O que deu certo:**
* A documentação da nova arquitetura clara e estruturada, permitindo que as futuras manutenções e melhorias corram sem dificuldades e garante uma transição suave para o uso do **IaC** com **Terraform**.
* **O que podemos melhorar:**
* Uma maior automatização no processo de geração da documentação e controle de versões poderia ter sido explorada para ganhar tempo e manter a padronização.
* **Execução dos Processos:**
* **O que deu certo:**
* A execução dos processos definidos foi bem conduzida, especialmente a transição dos arquivos de **FTP** para **S3**, garantindo que o fluxo fosse mantido sem interrupções para as áreas envolvidas.
* A integração entre **AWS Batch**, **RDS**, e **S3** funcionou conforme esperado, garantindo a confiabilidade no processamento e armazenamento de dados.
* **O que podemos melhorar:**
* Uma maior robustez nos testes de carga de dados extensos seria ideal para prever gargalos. O escalonamento das instâncias **EC2** em determinados momentos podem exigir ajustes manuais, algo que poderia ter sido previsto, para que não houvesse travamentos durante a carga.
* A automação do processo CI-CD poderia ser melhorada, garantindo que futuros ajustes no pipeline sejam implementados de maneira mais simples, rápida e sem interferências no ambiente de produção.

# 3. Considerações Finais

## 3.1 Resultados

Ao longo da execução do projeto aplicado, resultados significativos foram alcançados, no que diz respeito à eficiência técnica e amadurecimento das práticas arquiteturais da solução. Este projeto, foi focado na implementação de uma solução baseada em micro serviços na AWS, e nos permitiu atender os requisitos previamente estabelecidos, também superar os desafios, que contribuíram para um refinamento do processo da solução.

Temos como os principais resultados positivos:

* **Economia de custos com a utilização do AWS Batch:** A estratégia ao optarmos pelo AWS Batch em vez do AWS Glue, foi eficiente na redução de custos do projeto, para a realização do processamento dos arquivos armazenados no FTP, mantendo a integridade e eficiência, considerando, especialmente que esta carga de dados não será um processo recorrente.
* **Arquitetura modular e escalável:** A adoção da arquitetura baseada em micro serviços com ECS garante uma solução escalável e tecnicamente fácil de ser mantida. O API Gateway e o Cognito cuidam do roteamento e autenticação, enquanto o ALB e ECS distribuem as requisições e executam os serviços de forma eficiente, mesmo em um cenário de cargas altas.
* **Melhoria em eficiência operacional:** A implementação de micro serviços bem estruturados com comunicação assíncrona gerida pelo SQS, reduz o tempo de geração dos relatórios históricos, que antes levavam dias, para horas, gerando impacto positivo na eficiência operacional.
* **Robustez na segurança e autenticação:** A integração com o serviço Amazon Cognito, pode garantir um controle robusto de autenticação e autorização para as APIs, protegendo acesso aos serviços e garantindo conformidade com requisitos de segurança importantes e que a empresa deve manter.

Como pontos negativos e/ou dificuldades enfrentadas, podemos mencionar:

* **Integração com o FTP:** A integração inicial com o FTP, para consumo e ingestão das planilhas enfrentou desafios de confiabilidade e segurança, resultando em um carregamento mais lento devido a latência limitante do protocolo. Com a migração para o S3, este é um gargalo superado.
* **Complexidade no gerenciamento inicial dos recursos na AWS:** O gerenciamento dos recursos da AWS apresenta complexidade, especialmente com o Terraform, que exige um esforço extra na configuração dos recursos corretamente a fim de evitar, redundâncias, configurações errôneas, falhas de deploy e custos desnecessários.

A implementação do projeto proporcionou vivenciar diversas experiencias tecnológicas enriquecedoras, abrangendo avanços no que diz respeito a arquitetura de soluções, dentre elas:

* **FinOps é um trabalho contínuo:** a avaliação contínua dos custos (princípios de FinOps), assegura decisões arquiteturais. A escolha do AWS Batch ao invés do Glue, foi um aprendizado crucial sobre otimização de recursos em nuvem, sem sacrificar a qualidade da solução.
* **Melhores práticas em arquitetura:** A adoção do C4 Model ajudou a visualizar as interações entre os sistemas, garantindo assim, a consistência e coesão do design proposto. Foi de grande importância para facilitar a comunicação entre arquitetura, desenvolvimento e demais stakeholders envolvidos no projeto.
* **Documentação e processo transparente:** A documentação do projeto foi essencial para facilitar as futuras manutenções do projeto**; e o uso do terraform no provisionamento da infraestrutura (IaC) garante a transparência e a rastreabilidade em todas as etapas. Cada mudança na infra, é devidamente registrada, o que facilita no rastreamento das alterações e no FinOps.**

## 3.2 Contribuições

O PA (Projeto Aplicado) trouxe soluções essenciais no enfrentamento do desafio, focando em eficiência, escalabilidade e custo-benefício, que a abordagem anterior não permitia. As principais contribuições foram:

* **Inovação arquitetural:** A arquitetura baseada em micro serviços permitiu que cada módulo funcione de forma independente e coordenada, facilitando atualizações sistêmicas e integrações a novos módulos sem interrupções. O uso do SQS para troca de mensagens entre os micros serviços, evita gargalos naturais em estruturas monolíticas, permitindo processamento assíncrono. Essa modularidade e desacoplamento sistêmico, são a longo prazo, benéficos, pois oferecem maior flexibilidade para atualizações.
* **Eficiência Operacional:** O AWS Batch em conjunto com os scripts em Python, gerados para processar o grande volume de dados eficientemente, para o RDS, economiza tempo das personas envolvidas nos processos de consolidação, bem como, garante confiabilidade nos dados, por não necessitar de ajustes manuais.
* **Redução de Custos com AWS Batch:** A escolha do batch para processar os arquivos foi inovadora, dado o contexto do processamento, onde ao invés de utilizarmos uma ferramenta própria para o objetivo, pudemos, considerando a ingestão esporádica dos dados, utilizar outros meios alinhando-se aos princípios de FinOps, o que mostrou, a compreensão e o adequação dos requisitos financeiros do projeto.
* **Solução no armazenamento dos dados:** A combinação do RDS para os dados relacionais e o S3 para armazenamento de arquivos traz eficiência no uso de diferentes tipos de dados. O S3 permite armazenamento a longo prazo de grandes volumes de dados com baixo custo, podendo ser ainda menor, para arquivos armazenados apenas como histórico (S3 Glacier), enquanto o RDS mantem os dados transacionais em um ambiente de alto desempenho. Com a desativação do FTP, que inicialmente atuava como repositório da empresa, elimina-se a dependência externa, o que torna o processo mais ágil e seguro.
* **Vantagens e diferenciais da nuvem:** O projeto utilizou uma combinação de serviços oferecidos pela plataforma AWS, para criar uma estrutura ágil e escalável, permitindo a empresa lidar com aumentos de demanda sem comprometer a performance dos sistemas. A arquitetura flexível e modular, permite a adição de novos micros serviços e APIs conforme demanda, mantendo a estrutura base intacta, que se alinham aos objetivos de crescimento da empresa.

## 3.3 Próximos passos

Para o aprimoramento da solução proposta, os próximos passos importantes podem ser adotados:

* **Monitoramento e alertas:** Aprimorar o monitoramento dos recursos com alertas granulares para garantir a identificação precoce de problemas no ambiente, permitindo uma resposta proativa. Isso reduziria os riscos de interrupção e manteria o sistema funcionando de maneira otimizada. Como ferramenta de observabilidade de baixo custo, pode ser implementado utilizando o AWS CloudWatch, que permite configurar alertas detalhados para o monitoramento dos recursos.
* **Planejamento de redundância:** Mesmo estando em nuvem, para garantir a resiliência da solução a longo prazo, é recomendável a utilização de zonas de disponibilidade adicionais. O uso de múltiplas AZs permite que a aplicação continue funcionando mesmo que algum datacenter do provedor entre em falha.
* **Melhoria no processo de deploy:** Para melhorar o processo de deploy da solução, a utilização de uma estratégia de publicação Canary seria uma adição valiosa ao projeto. Esse método permite a implantação de novas versões de maneira gradual, testando em frações do tráfego, e uma vez que versão se mostra estável, é expandido para todos os usuários.
* **Melhorias em segurança e compliance:** A nova solução contempla segurança na comunicação e autenticação de usuários e serviços, porém, melhorias na criptografia dos dados e a implementação de padrões mais rígidos, como o LGPD, aumentam a proteção dos dados, e conformidade as regulamentações de privacidade. O uso de ferramentas para gerenciamento de credenciais, também pode ser avaliado para trazer ainda mais segurança ao ambiente. O provedor de nuvem AWS oferece ferramentas que auxiliam nisso, como AWS KMS e o Secret manager.
* **Criar o MVP do projeto:** Desenvolver um MVP (Minimum Viable Product) com o menor investimento de tempo e recurso, para validar a proposta do projeto com uma versão simplificada, porém funcional. O MVP terá as funcionalidades essenciais para atender as principais dores e necessidades das personas envolvidas validando o valor da solução.