PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

PUC Minas Virtual

Pós-graduação *Lato Sensu* em Arquitetura de *Software* Distribuído

Projeto Integrado

Relatório Técnico

SCIAC Sistema de Coleta, Integração e Análises de Informação de Custos

Bárbara Bruna Machado Vilela

Belo Horizonte
Janeiro de 2022

Sumário

1. Introdução

Nos últimos anos, a aplicação das Tecnologias de Informação e Comunicação (TIC) para melhor governança organizacional, ou e-governança (CUNHA e MIRANDA, 2013) tem crescido substancialmente na esfera pública brasileira. Entretanto, em se tratar de gestão de custos, embora afirme SILVA et al (2007 apud COUTO et al, 2017) que haja, uma busca constante de sistemas de informações mais acurados e com ênfase na apuração de custos dos serviços prestados, além de uma tendência dos órgãos utilizarem informações gerenciais nas tomadas de decisões internas à administração, as entidades governamentais ainda enfrentam grandes desafios, pois os projetos de adoção de técnicas de gerenciamento de custos são dificultados pela não integração com os demais sistemas estruturantes organizacionais e com o planejamento estratégico; escassez de recursos para viabilização e manutenção dos projetos; alta burocratização e normatização características do setor público; além de prejuízos causados pela busca por legitimação social por parte dos gestores (MESSIAS, FERREIRA e SOUTES , 2018). Nas instâncias municipais brasileiras, a ineficiência na utilização dos recursos públicos é escancarada anualmente pelos dados do IFGF (Índice FIRJAN de Gestão Fiscal) que, em 2020, revelaram que 1704 prefeituras não geram receita suficiente para a manutenção da estrutura administrativa; e que Mais de 1/3 das prefeituras analisadas gastam mais de 54% da receita com pessoal. Enquanto a pesquisa TIC-Governo Eletrônico de 2019 aponta que, naquele ano, embora mais de 75% dos municípios brasileiros tenham utilizado Sistemas de Informação para fins financeiros, contábeis e/ou orçamentários, apenas 16% afirmam utilizar de TIC para apoio à decisão. Pois, , a despeito da legislação em vigor, que obriga os órgãos públicos brasileiros a implementar sistemas de gerenciamento de custos (Brasil, 2000; Franco et al, 2013 apud MESSIAS FERREIRA eSOUTES, 2018), ainda existem municípios brasileiros que não promoveram ensaios visando à adoção de sistema de custos e não dispõem de estrutura organizacional para tal (COUTO et al, 2017).

Dentro desse contexto, este projeto visa apresentar uma descrição da proposta arquitetural desenvolvida para atender a um Sistema de Coleta, Integração e Análise de informações de custos (SCIAC), responsável por coletar dados de custos relativas à um órgão do executivo municipal brasileiro (cuja identificação real será omitida, por restrições da entidade), provenientes de diferentes fontes dentro e fora da organização, tal como a construção e manutenção do conceito de Centro-de-custos, com base na sua estrutura organizacional, integrar todos esses dados em uma base unificada e, assim, possibilitar a geração de informações consistentes e robustas, matéria prima para diversas análises de cunho gerencial, financeiro e orçamentário. Seus principais desafios são: orquestração das coletas em suas especificidades, o que envolve acesso direto à base de outros sistemas, importação de arquivos em diferentes formatos e obtenção de dados via APIs Rest; e a consistente integração entre todos esses dados, garantindo a correta alocação dos custos coletados.

Desenvolvido em PHP 5, uma versão já descontinuada da linguagem, e caracterizado por forte acoplamento de código, o projeto tem demandado considerável esforço da equipe de desenvolvimento para sua manutenção. Além disso, requer urgente

atualização afim de adequar-se à recentes mudanças na infraestrutura de implantação padrão utilizada pelo município. Assim, considerou-se mais apropriado a reconstrução da aplicação, utilizando uma arquitetura mais moderna, que possibilite uma melhor manutenabilidade do código, economia de recursos de TI – com a diminuição de demandas para correção de erros, melhoria dos processos de trabalho - com a extinção de diversas tarefas manuais provenientes das falhas do sistema, além de melhores condições para expansões futuras. Assim, o projeto tem como objetivos específicos:

- Realizar um estudo minucioso do software, descrevendo, de maneira clara e objetiva, os requisitos arquiteturais relevantes;
- Modelar toda a arquitetura proposta para a reconstrução da aplicação;
- Realizar a prova de conceito (POC) da arquitetura proposta;
- Desenvolver protótipo da solução proposta, levando-se em conta a prioridade dos requisitos levantados, o tempo disponível e o acompanhamento da equipe, de modo a garantir a continuidade do desenvolvimento pela mesma a posteriori.

Um vídeo introdutório com a apresentação desse projeto pode ser encontrado em:
https://youtu.be/ZRrzl9-Iz0I

2. Cronograma do Trabalho

A seguir é apresentado o cronograma proposto para as etapas deste trabalho.

Datas		**Atividade / Tarefa**	**Produto / Resultado**
De	**Até**		
01/01/2022	15/02/2022	1. Estudo detalhado do sistema em produção, sua arquitetura, bases de dados, principais funcionalidades e integrações	Aquisição dos conhecimentos necessários para continuidade dos trabalhos
04/02/2022	08/02/2022	2. Construção do conteúdo introdutório deste documento e Cronograma de atividades do projeto	Introdução e Cronograma de Atividades
09/02/2022	10/02/2022	3. Levantamento das restrições arquiteturais da aplicação	Lista de Restrições Arquiteturais
10/02/2022	12/02/2022	4. Levantamento de Requisitos	Listas dos Requisitos Funcionais (RF) e não Funcionais (RNF)
14/02/2022	15/02/2022	5. Documentar os mecanismos arquiteturais	Descrição dos mecanismos arquiteturais
16/02/2022	18/02/2022	6. Construção do Diagrama de Contexto	Diagrama de Contexto
21/02/2022	21/02/2022	7. Apresentação da arquitetura proposta à equipe responsável pelo projeto	Obtenção da opinião crítica da equipe.
22/02/2022	15/03/2022	8. Desenvolvimento da Prova de Conceito (POC) da arquitetura proposta	Prova de Conceito (POC)
16/03/2022	12/10/2022	9. Construção do protótipo da aplicação	Protótipo dos principais requisitos identificados
05/04/2022	07/04/2022	10. Gravação e edição do video de apresentação do projeto	Vídeo 1: Apresentação do projeto
08/04/2022	11/04/2022	11. Construção do Diagrama de Containers	Diagrama de Containers
24/08/2022	31/08/2022	12. Construção do Diagrama de Componentes	Diagrama de Componentes
15/09/2022	10/10/2022	13. Alterações na documentação devido à mudanças nas tecnologias propostas para o *Backend* da Aplicação.	Artefatos atualizados
10/10/2022	14/10/2022	14. Documentação do Código da aplicação	Descrição do código, conforme artefato “Código do modelo C4
17/10/2022	19/10/2022	15. Análise das abordagens arquiteturais utilizadas	Descrição das Análises Arquiteturais
20/10/2022	27/10/2022	16. Construção dos cenários com base nas RNFs definidas	Cenários contemplando cada RNF.
24/10/2022	31/10/2022	17. Teste dos cenários e documentação dos resultados	Documentação dos resultados obtidos
01/11/2022	04/11/2022	18. Crítica quanto às evidências resultantes do processo de análise	Descrição das evidências obtidas
07/11/2022	10/11/2022	19. Construção de uma avaliação crítica da arquitetura proposta	Avaliação Crítica e Conclusão
11/11/2022	18/11/2022	20. Gravação e edição do vídeo de apresentação final do projeto	

3. Especificação Arquitetural da solução

Esta seção apresenta a especificação básica da arquitetura da solução a ser desenvolvida, incluindo diagramas, restrições e requisitos definidos pelo autor, tal que permitem visualizar a macroarquitetura da solução.

3.1 Restrições Arquiteturais

R1: O software deve ser compatível com a esteira padrão de implantação (DevOps) utilizada pela organização, que contempla o uso de containers Docker, o orquestrador de containers, Kubernetes, e a ferramenta GitLab para controle de versionamento de código;

R2: A equipe de desenvolvimento responsável pela aplicação possui características singulares. Trata-se de uma equipe de 06 (seis) profissionais em estágio profissional, cuja permanência na equipe permeia o período de 2 (dois) anos apenas. São profissionais dedicados e empenhados, porém, possuem, em sua maioria, pouca ou nenhuma experiência em TI. A equipe conta ainda, com um Analista de Dados dedicado ao projeto;

R3: O processo de desenvolvimento deve ser compatível com a metodologia ágil de desenvolvimento de softwares Scrum;

R4: Por se tratar de uma migração, é imprescindível que o novo desenvolvimento permita o “estrangulamento parcial” do sistema legado, ou seja, ambas as soluções devem trabalhar juntas até que a nova substitua completamente a antiga;

R5: O sistema deve ter seus dados armazenados em uma base compatível com ferramentas de análises, geração de dashboards e afins.

3.2 Requisitos Funcionais

ID	Descrição Resumida	Dificuldade (B/M/A)*	Prioridade (B/M/A)*
RF01	O sistema deve ter seu acesso restrito aos usuários por ele responsáveis	B	A
RF02	O sistema deve permitir o agendamento das coletas de dados nas diferentes fontes	A	A
RF03	O sistema deve permitir a deleção de cargas mal sucedidas	A	A
RF04	O sistema deve permitir o disparo automático de tarefas de carga e transformação de dados, seja por agendamento ou eventos previamente definidos	A	A
RF05	O sistema deverá coletar informações sobre a estrutura organizacional do município e, com base nelas, manter atualizados os registros de Centros de Custos	A	A
RF06	O sistema deve permitir o cadastro das unidades físicas, ou seja, os locais onde são estabelecidos fisicamente os órgãos do município	M	A
RF07	O sistema deve permitir a criação e manutenção dos relacionamentos entre os centros de custos e as unidades onde os mesmos operam	A	A
RF08	Centros de custos desativados devem ser mantidos na base e sua desativação registrada, para fins históricos	B	A
RF09	O sistema deve prover meios de gerenciar o histórico das alterações nos relacionamentos entre centros de custos e unidades	M	A
RF10	O sistema deve possibilitar a coleta dos dados referentes aos custos com Pessoal	M	A
RF11	O sistema deve possibilitar a coleta dos dados referentes aos custos com Materiais de Consumo	M	A
RF12	O sistema deve prover meios de alocar cada tipo de custo aos respectivos centros de custos e/ou unidades aos quais estão relacionados	A	A
RF13	O sistema deve prover meios de identificar cada unidade através da utilização da base de endereços geo-referenciados do município	A	M
RF14	O sistema deve comter meios de ingresso de informações de custos de forma manual, sempre que não houver como coletá-los de uma fonte informatizada	M	M
RF15	O sistema deve permitir a importação de arquivos em diferentes formatos para os dados provenientes de fontes indisponíveis	M	M

	para acesso direto		
RF16	O sistema deve possibilitar a coleta dos dados referentes aos custos com Energia	M	M
RF17	O sistema deve possibilitar a coleta dos dados referentes aos custos com Água e Esgotamento	M	M
RF18	O sistema deve possibilitar a coleta dos dados referentes aos custos com Telefonia	M	M
RF19	O sistema deve possibilitar a integração com dados orçamentários	A	M
RF20	O sistema deve possibilitar a coleta de dados de "Produção", relativos às quantidades de serviços prestados a fim de que possam ser mensurados seus custos	A	M
RF21	O sistema deve permitir o cadastro e a gestão de usuários e seus perfis	B	B
RF22	Após cada processo de carga, o sistema deverá enviar alertas por e-mail contendo o status da carga aos usuários interessados	B	B

Tabela 1: ***Lista dos Requisitos Funcionais (RFs) do Sistema***
Considerar: B=Baixa, M=Média, A=Alta.

3.3 Requisitos Não-funcionais

ID	Descrição	Prioridade B/M/A
RNF01	*O sistema deve permitir portabilidade do código entre instâncias de clusters do Docker*	A
RNF02	A segurança deve ser considerada durante os processos de integração entre os módulos, de modo a impedir que terceiros acessem funcionalidades inapropriadamente	A
RNF03	As entregas parciais do sistema devem levar em conta a continuidade do uso do sistema legado, atualmente em produção, e permitir a continuidade dos trabalhos com a cooperação entre as duas soluções	A
RNF04	O sistema deve permitir a integração com diversos sistemas, seja via requisições HTTP-Rest ou acesso direto às bases Oracle	A
RNF05	Os módulos de interação com o usuário devem ser acessíveis, seguindo as normas de acessibilidade definidas pela W3C	A
RNF06	As tecnologias a serem utilizadas devem levar em conta a curva de aprendizagem, devido às características da equipe de desenvolvimento envolvida	M
RNF07	O sistema (módulos de coleta e carga de dados) deve apresentar disponibilidade mínima de 76%, estando disponível 18h/dia, 6 dias/semana	B
RNF08	A base de dados deve apresentar, pelo menos, 90% de disponibilidade para leitura dos dados e análises	B

Tabela 2: ***Lista dos Requisitos Não Funcionais (RNFs) do Sistema***
Considerar: B=Baixa, M=Média, A=Alta.

3.4 Mecanismos Arquiteturais

Análise	***Design***	**Implementação**
Persistência	NoSQL – Orientado à Grafos	Driver Neo4j
Back end	Orquestração de pipelines Workflows (ETLs) modularizados/API REST	Apache Airflow Python/FastAPI
Autorização e Autenticação	Json Web Token – JWT	Python-jose
Integração	ORM (Oracle) APIs Rest	SQLAlchemy HTTPs/Json
Log do sistema	Local (servidor) E-mail (Status de cargas)	Python Logging Apache Airflow
Teste de Software	Test Driven Development – TDD	Robot
Front end	Single Page Application – SPA	React/typescript
Deploy	Conteinerização/versionamento (Padrão da organização)	Kubernetes, Docker/GitLab

Tabela 3: ***Mecanismos Arquiteturais propostos para o Sistema***

4. Modelagem Arquitetural

Esta seção apresenta a modelagem arquitetural da solução proposta, de forma a permitir seu completo entendimento visando à implementação da prova de conceito (seção 5).

Para esta modelagem arquitetural optou-se por utilizar o modelo C4 para documentação de arquitetura de software. Mais informações a respeito podem ser encontradas aqui: *https://c4model.com/* e aqui: *https://www.infoq.com/br/articles/C4-architecture-model/*. Dos quatro nível que compõem o modelo C4 três serão apresentados aqui e somente o Código será apresentado na próxima seção (5).

4.1 Diagrama de Contexto

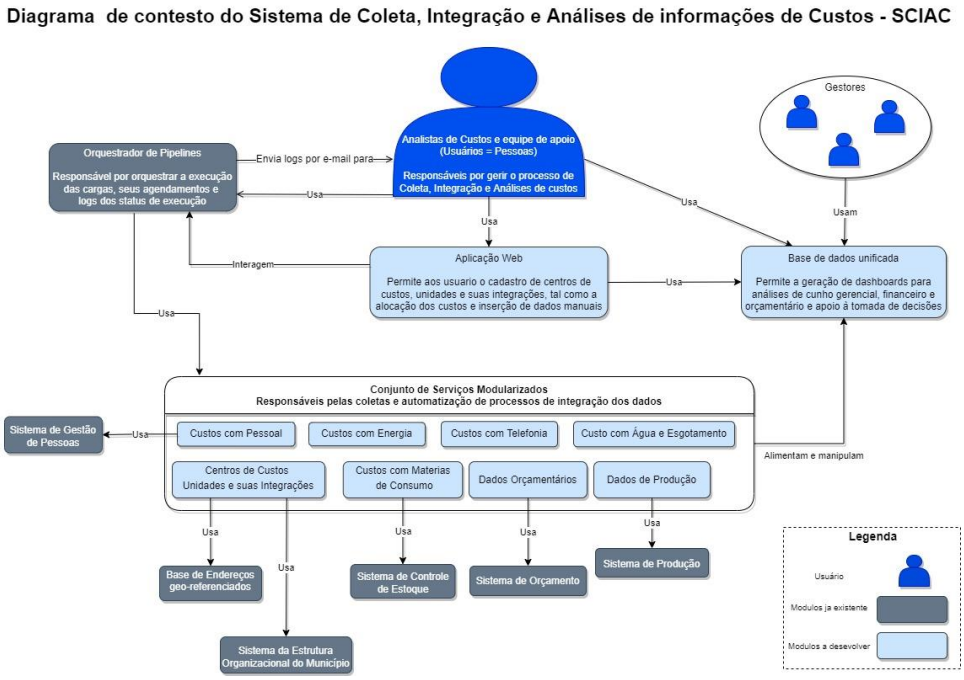

Figura 1 – Diagrama de contexto- Disponível em:
https://drive.google.com/file/d/1pXFW3lh89rzLkfweX0LgZiQ2LSm6CrTt/view?usp=sharing

A Figura 1 apresenta a Visão Geral da solução proposta. Nela, é possível visualizar um Diagrama de Contexto, que, segundo o modelo C4, deve apresentar o contexto onde a solução está inserida, contemplando as pessoas e demais softwares com os quais ele interage.

No diagrama, podemos visualizar a Equipe de Custos, responsável pela gestão das coletas, integração, validação e análises preliminares dos dados. Ela utilizará do software através de duas "pontas": a Aplicação Web, que contará com uma interface gráfica específica para o sistema (a ser desenvolvida); e da ferramenta Orquestrador de Pipelines (um software já existente), que permitirá o agendamento de tarefas, envio de alertas, dentre outros. Ambas poderão se comunicar, possibilitando maior reuso dos componentes envolvidos.

A ferramenta de orquestração de pipelines utilizará de um conjunto de Serviços Modularizados, onde cada módulo será responsável pelos processos (ETLs) relativos a um assunto específico do sistema.

Esses, assim como a aplicação web, alimentarão e manipularão uma Base de Dados Unificada, que, nada mais é, do que o objetivo principal da solução. Essa base de dados será a fonte valiosa de informações, a serem geradas por Gestores de distintos setores dentro da organização, além da própria Equipe de Custos, com base nos dados nela contidos, a fim de obter insumos para suas tomadas de decisão.

Para que os dados sejam coletados e integrados, o Conjunto de Serviços Modularizados necessitará se comunicar com diversos sistemas já existentes, dentre eles, inclusive, sistemas externos à organização.

4.2 Diagrama de Container

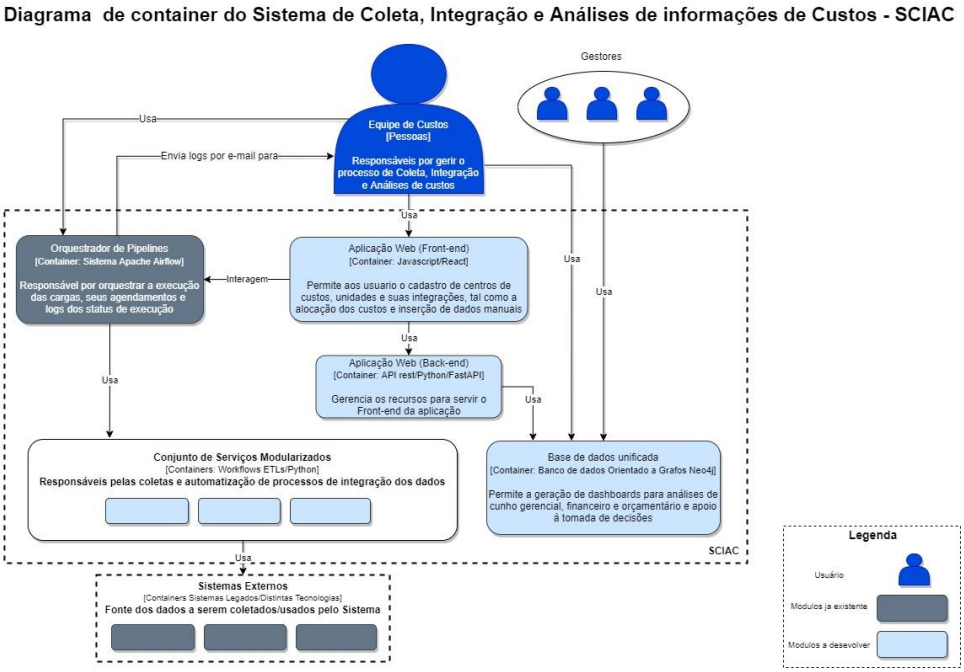

Figura 2 – Diagrama de container- Disponível em: https://drive.google.com/file/d/1RB3URIa5ea4NF6g0E0ducdTHd2k1VH4U/view?usp=sharing

A figura 2 apresenta os *containers* da aplicação...

4.3 Diagrama de Componentes

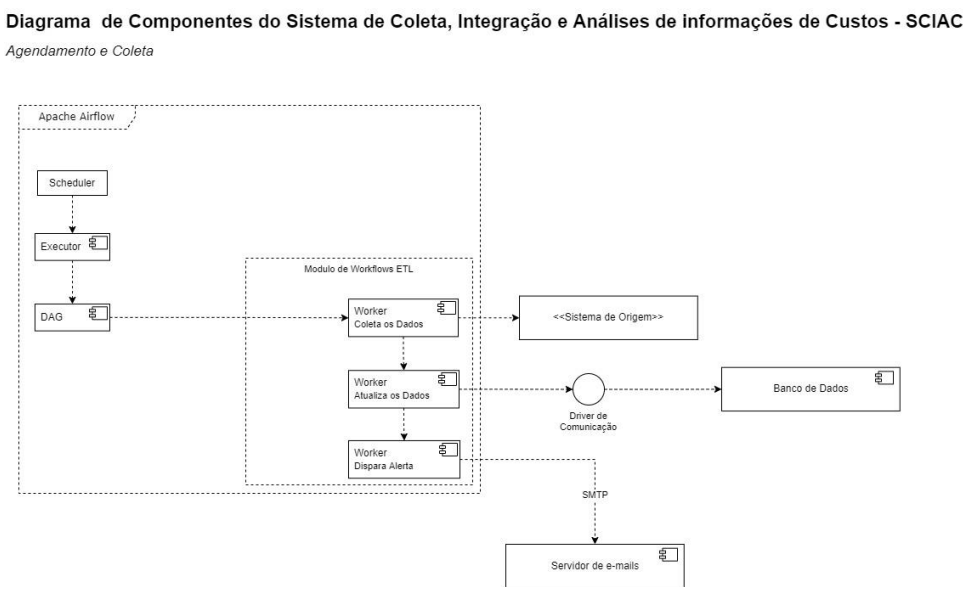

Figura 3 – Diagrama de Componentes - Disponívem em: *https://drive.google.com/file/d/1XTZ-z8ETJzAsxuzjeHhAwg6GQeYqRznx/view?usp=sharing*

Na Figura 3 é possível observar as entidades participantes de um módulo específico da solução proposta. Ela contempla uma visão detalhada do processo de Extração, Tratamento e Carga (ETL) dos dados provenientes de um sistema externo, que requer integração com o SCIAC. São elas:

- *Apache Airflow: Trata-se de uma ferramenta Open Source, configurada através de código em linguagem Python. Ela é composta por inúmeros componentes próprios, que possibilitam o agendamento de tarefas, sua execução e monitoramento. Uma descrição da arquitetura do Apache Airflow pode ser encontrada em:* https://joaoarthurbm.github.io/arqsoft-blog/posts/airflow/.
  Dentre os componentes internos do Apache airflow, temos:
  - Sheduler: responsável por monitorar todas as tasks e DAGs, e enviá-las para a fila de execução quando suas dependências são concluídas. Nesse contexto, foi considerado o ator do processo, pois é o responsável por iniciá-lo;
  - Executor: Responsável por iniciar cada tarefa, conforme determinação do Sheduler;
  - DAG (Directed Acyclic Graph): Conceito principal do Apache Airflow. Indica um grafo de tarefas, onde cada nó é uma tarefa (Worker) e necessariamente precisa ter um início e um fim (não pode ser cíclico). No contexto em questão, tanto as DAGs como os Workers serão construídos customizados para atender à aplicação;
  - Workers: Cada worker representa uma tarefa a ser executada. O conjunto de tarefas relativo a um módulo do sistema deve ter suas responsabilidades isoladas dos demais, a fim de garantir a independência entre eles.

- *Sistema origem: Interface de comunicação com um sistema/fonte de dados externa à aplicação;*

- *Banco de Dados: A solução propõe o uso do banco de dados Neo4j, que será atualizado por meio do uso de um drive de comunicação (biblioteca python);*

- *Servidor de Emails: O servidor de e-mails da organização, a comunicação da aplicação com ele se fará através do protocolo SMTP/SSL.*

-

5. Prova de Conceito (PoC)

Nessa sessão, será apresentada a Prova de Conceito (POC) desenvolvida para a solução proposta que, para sua construção, levou-se em conta as seguintes considerações:

- Mínimo Produto Viável (MPV): Considerou-se que a entrega mais relevante para a aplicação se refere aos processos de ETL relativos aos Centros de Custos, visto se tratar do módulo que requer maior esforço de manutenção por parte da equipe de TI responsável pelo sistema legado;
- Maior complexidade no processo de integração: Dentre as diferentes camadas da solução, os módulos de ETL são os mais complexos, dado utilizarem de um ferramental desconhecido pela equipe de TI envolvida (Apache Airflow) e envolver a integração entre a aplicação, os sistemas de origem dos dados e a base de dados unificada;
- O repasse para a equipe de desenvolvimento: Considerou-se mais apropriado explorar a ferramenta Apache Airflow e suas propriedades básicas no intuito de facilitar a compreensão da equipe de desenvolvimento acerca de todo o processo, tornando-o mais visual e factível a todos os envolvidos.

Uma descrição detalhada dos componentes implementados e suas integrações será apresentada a seguir.

5.1 Integração entre Componentes

Para a POC e prototipagem, com o intuito de apresentar os principais módulos da arquitetura proposta, conforme considerações antes apresentadas, foram desenvolvidos:

- Uma imagem Docker, denominada “barbaravilela/sciac-airflow”, pública e disponível no DockerHub, baseada na imagem “puckel/docker-airflow” (também disponível publicamente*), contendo a ferramenta Apache Airflow e os programas* de ETLs, aqui denominados como “serviços modularizados” relativos ao módulo de Centros de Custos;
- Uma API fake, responsável por simular o serviço web oferecido pelo Sistema da Estrutura Organizacional do Município, fonte dos dados a serem extraídos, tratados e carregados na base de dados do SCIAC.

Ao manipular a ferramenta Apache Airflow, é possível visualizar as execuções da DAG denominada “dag_carga_centros_custos”, que tem por funcionalidade acessar a API fake, relativa à Estrutura Organizacional do Município, salvar os dados nela obtidos em um arquivo temporário, manipular esses dados e armazená-los em uma base do banco de dados Neo4j, também disponível em nuvem. Por fim, a tarefa final envia um e-mail de alerta, informando o status da execução.

5.2 Código da Aplicação

Como preconiza o modeloC4, optou-se por não documentar o nível de código, por se considerar desnecessário para a compreensão da solução apresentada.

Todo código implementado para o protótipo, tais como os links e credenciais para acesso à cada ferramenta envolvida estão disponíveis no seguinte repositório, no GitHub: https://github.com/BarbaraVilela/SCIACArchitecture

6. Avaliação da Arquitetura (ATAM)

A avaliação da arquitetura desenvolvida neste trabalho é abordada nesta seção visando avaliar se ela atende ao que foi solicitado pelo cliente, segundo o método ATAM.

6.1. Análise das abordagens arquiteturais

Apresente aqui um breve resumo das principais características da proposta arquitetural. Para isto, utilize o método Architecture Tradeoff Analysis Method (ATAM), no qual são utilizados cenários para fazer essa análise.

Exemplo:

Atributos de Quailidade	Cenários	Importância	Complexidade
Interoperabilidade	Cenário 1: O sistema deve se comunicar com sistemas de outras tecnologias.	A	M
Usabilidade	Cenário 2: O sistema deve prover boa usabilidade.	M	B
Manutenibilidade	Cenário 3: O sistema deve ter a manutenção facilitada.	M	M

6.2. Cenários

Mostre os cenários utilizados na realização dos testes da sua aplicação. Escolha cenários de testes que demonstrem os requisitos não funcionais (atributos de qualidade) sendo satisfeitos. Priorize os cenários para a avaliação segundo critérios quantitativos ou qualitativos.

Exemplos de cenários:

Cenário 1 - Interoperabilidade: Ao acessar a URL do serviço de informações gerenciais via HTTP GET, o mesmo deve retornar as informações no formato JSON.

Cenário 2 - Usabilidade: Ao navegar na tela, o sistema deve apresentar boa usabilidade. A navegação deve apresentar facilidade e o acesso as funcionalidades deve ser bem objetivo para a função que precisar ser realizada, o usuário deve ser capaz de efetuar uma compra em no máximo 5 minutos, assim garantindo a agilidade e a usabilidade para ficar de acordo com um dos requisitos não funcionais.

Cenário 3 - Manutenibilidade: Havendo a necessidade de alterar o gateway de pagamento somente será necessário fazer alteração no broker da funiconalidade de pagamento, facilitando a manutenção e os testes.

6.3. Evidências da Avaliação

Apresente as medidas registradas na coleta de dados. Para o que não for possível quantificar apresente uma justificativa baseada em evidências qualitativas que suportem o atendimento ao requisito não-funcional.

Atributo de Qualidade:	Interoperabilidade
Requisito de Qualidade:	O sistema deve se comunicar com outras tecnologias.
Preocupação:	
O sistema deve ter como resposta a uma requisição uma saída de fácil leitura por outro componente.	
Cenário(s):	
Cenário 1	
Ambiente:	
Sistema em operação normal	
Estímulo:	
O sistema de monitoramento envia uma requisição para o serviço REST do módulo de informações gerenciais.	
Mecanismo:	
Criar um serviço REST para atender às requisições do sistema de monitoramento	
Medida de resposta:	
Retornar os dados requisitados no formato JSON	
Considerações sobre a arquitetura:	
Riscos:	Alguma instabilidade na rede pode deixar a conexão lenta ou mesmo a perda de pacotes.
Pontos de Sensibilidade:	Não há
Tradeoff:	Não há

Acrescente imagens e descreva os testes realizados, de tal forma que se comprove a realização da avaliação.

Faça isto para todos os cenários apresentados no tópico 6.1.

6.4. Resultados Obtidos

Apresente os resultados da arquitetura produzida, indicando seus pontos fortes e suas limitações. A título de sugestão construa uma tabela apresentando esses resultados, como no exemplo que segue:

Requisitos Não Funcionais	Teste	Homologação
RNF01: O sistema deve ...	OK	OK
RNF02: O sistema deve ...	OK	N.A.
RNF03: ...	OK	N.A.

Obs: N.A.: não se aplica.

7. Avaliação Crítica dos Resultados

Apresente aqui, de forma resumida, os principais pontos positivos e negativos da arquitetura proposta. Adote uma postura crítica que permita entender as limitações arquiturais, incluindo os prós e contras das tecnologias. Você pode utilizar o formato textual ou produzir um quadro resumo.

Ex. de quadro resumo:

Ponto avaliado	**Descrição**
xxxxxxxxxxxxxxx	xxxxxxxxxxxxxxxxxxxx

8. Conclusão

Descreva, de forma sucinta, quais foram as lições aprendidas na execução do seu projeto arquitetural. Procure apresentá-las de tal forma que fiquem configurados os *trade-offs* da arquitetura produzida, como por exemplo, Segurança X Desempenho, Granularidade X Manutenibilidade, etc.

Aqui deve ser apresentado também tudo que se aprendeu com esse projeto, de modo a servir como ajuda para outros profissionais.

Também se faz necessário evidenciar as possibilidades de melhoria do projeto, caso se deseje dar continuidade a ele. Nesse sentido, indique possíveis ajustes ou melhorias arquiteturais, que possam vir a ser realizados futuramente.

Lições aprendidas (ex.):

1. xxxxxxxxxxxxxxxxx
2. xxxxxxxxxxxxxxxxx
3. xxxxxxxxxxxxxxxxx

Referências

CUNHA, Maria Alexandra Viegas Cortez da; MIRANDA, Paulo Roberto de Mello. O uso de TIC pelos governos: uma proposta de agenda de pesquisa a partir da produção acadêmica e da prática nacional. Organizações & Sociedade, V.20, N.66, 2013. Disponível em: https://www.scielo.br/j/osoc/a/gDHX66twKTVV6SD3VJnKSWL/?lang=pt

Couto, M.H.A.; Yoshitake, M.; Fraga, M.S. e Tinoco, J.E.P. Sistema de informação de custos: uma experiência de implantação na secretaria de educação do município de Itaguari/GO. Revista de Tecnologia Aplicada (RTA), v.6, n.1, jan-abr 2017, p.16-33. ISSN: 2237-3713 . Disponível em: http://dx.doi.org/10.21714/2237-3713rta2017v6n1p16

MESSIAS, Diego; FERREIRA, Júlio César; SOUTES, Dione Olesczuk. Gestão de custos no setor público: um panorama de experiências internacionais. Revista Serv. Público Brasília, v.69, n.3, p.585-604. Brasília, 2018

IFGF - Índice Firjan de Gestão Fiscal
Disponível em: https://www.firjan.com.br/ifgf/consulta-ao-indice/

TIC-GovernoEletrônico
Disponível em: https://cetic.br/pt/pesquisa/governo-eletronico/