



Transformação Digital: Eliminação do Uso de Papel em Organizações Através da Integração de Sistemas de BPM, Gestão de Processos Eletrônicos e Gerenciamento de Recursos Humanos

**Relatório Técnico relativo ao Trabalho de Conclusão Curso
do Bacharelado em Sistemas de Informação na modalidade Empresa**

Aluno

Gutenberg Duarte Neves de Barros

Orientador

Guilherme Vilar
DEINFO/UFRPE

17 de outubro de 2022

Gutenberg Duarte Neves de Barros

**Transformação Digital: Eliminação do Uso de Papel em Organizações
Através da Integração de Sistemas de BPM, Gestão de Processos
Eletrônicos e Gerenciamento de Recursos Humanos**

Relatório Técnico apresentado ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Universidade Federal Rural de Pernambuco – UFRPE

Departamento de Estatística e Informática

Curso de Bacharelado em Sistemas de Informação

Orientador: Guilherme Vilar

Recife

Outubro de 2022

GUTENBERG DUARTE NEVES DE BARROS

Transformação Digital: Eliminação do Uso de Papel em Organizações Através da Integração de Sistemas de BPM, Gestão de Processos Eletrônicos e Gerenciamento de Recursos Humanos

Relatório Técnico apresentado ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Aprovada em: 13 de Outubro de 2022.

BANCA EXAMINADORA

Guilherme Vilar (Orientador)
Departamento de Estatística e Informática
Universidade Federal Rural de Pernambuco

Rodrigo Elia Assad
Departamento de Estatística e Informática
Universidade Federal Rural de Pernambuco

Resumo

A transformação digital nas organizações está demandando cada vez mais que soluções de software distintas sejam capazes de compartilhar e trocar informações entre si para apoiar as empresas e seus gestores na tomada de decisões e também impulsionar a diminuição de trabalhos manuais e repetitivos por parte de seus colaboradores com a automação de processos. Neste trabalho foi desenvolvido soluções de software que possibilitaram integrações de sistemas BPM, gestão de processo eletrônicos, e gerenciamento de recursos humanos para atender uma demanda específica do cliente Pitang na automação do compartilhamento de informações entre estes sistemas.

Palavra-chave: transformação digital, automação, integração, BPM, processo eletrônico, recursos humanos.

Sumário

1	Introdução	1
1.1	Objetivo	1
1.2	Contexto	2
2	A Empresa e sua Atuação	2
2.1	Pitang	2
2.1.1	Setor	3
2.1.2	Produtos	3
2.2	Atuação Profissional	4
2.2.1	Ingresso e Papéis	4
2.2.2	Projeto Atual	4
3	Desenvolvimento Realizado na Empresa	4
3.1	Problemática e a Solução Proposta	5
3.1.1	Metadados P2F	5
3.1.2	Processo: Benefício Vida	6
3.2	Tecnologias Utilizadas	6
3.2.1	Pitang to Flow - P2F	7
3.2.2	Bizagi	8
3.2.3	Nuxeo	9
3.2.4	Docker	9
3.2.5	Web Service	10
3.3	Desenvolvimento	10
3.3.1	Arquitetura	11
3.3.2	Solução 1: Metadados P2F	12
3.3.3	Solução 2: Benefício Vida	12
3.4	Contribuição	15
3.4.1	Contribuição do Autor	17
3.4.2	Benefício ao Cliente	20
4	Dificuldades Encontradas	21

4.1 Tecnologias	21
4.2 P2F	21
4.3 Comunicação	22
4.4 Priorização	22
5 Impactos da Formação no Trabalho	22
6 Conclusão	23

1 Introdução

Transformação digital é um meio pelo qual organizações usam tecnologia para solucionar problemas rotineiros do dia a dia. O conceito pode parecer futurista, mas na prática é algo pelo qual as empresas e seus gestores precisam pensar diariamente para se manterem relevantes no mercado [1].

Nos últimos anos, em decorrência da pandemia da COVID-19, as mudanças no formato de trabalho impulsionaram a corrida pela transformação digital. Em outras palavras, as instituições se viram impelidas a repensar como a empresa e seus processos funcionavam.

Os serviços e/ou produtos não podiam deixar de ser oferecidos, mas a forma como eles eram desenvolvidos precisava de mudanças para atender a nova realidade trazida pela COVID.

A necessidade de melhoria faz parte do ciclo de toda e qualquer organização. Negócios que não se reinventam tendem a perder sua relevância.

Partindo desse conceito, a digitalização e automação de processos é uma das possíveis alternativas para impulsionar a transformação digital dentro das organizações. O processo de digitalização de processos traz ganhos de velocidade, identificação de gargalos, eliminação da geração de papel, diminuição dos canais de comunicação, além proporcionar uma maior transparência para os interessados em acompanhar essas requisições.

Este trabalho abordou o conceito de transformação digital no âmbito da digitalização e automação de atividade e processos digitais. Nas seções a seguir serão apresentados os objetivos gerais e específicos que o trabalho atendeu.

1.1 Objetivo

Este trabalho está associado à área de desenvolvimento de software e sustentabilidade. O trabalho teve como objetivo geral desenvolver uma solução para as necessidades dos clientes da Pitang, empresa voltada para o desenvolvimento de soluções digitais, no quesito relacionado à digitalização de processos eletrônicos e integrações desses processos com sistemas de terceiros.

Para atendimento das necessidades do cliente, os seguintes objetivos específicos foram atendidos:

1. Entendimento da necessidade de negócio do cliente;
2. Desenvolvimento de novos metadados na aplicação Pitang to Flow (P2F) para integração dos processos modelados no Bizagi;
3. Disponibilizar nova versão com os metadados em ambiente de Produção do cliente;
4. Desenvolvimento de Scripts Groovy: requisitarBeneficioVida, listarTodosOperadoresBeneficio, listarOperadorBeneficio para integração com o Sistema de Recursos Humanos (SRH); e
5. Homologar o funcionamento dos scripts no processo Beneficio Vida junto ao cliente.

1.2 Contexto

Um cliente da Pitang, que neste trabalho chamaremos de Empresa X, possui licença de uso e serviços do produto Pitang to Flow (P2F), detalhado no item [3.2.1](#), que é um sistema de gestão de processos eletrônicos (SGPEs).

A Empresa X no final do ano de 2020 solicitou à Pitang melhorias e serviços de integração para sanar uma dor de negócio.

O problema e solução proposta tratados neste trabalho são apresentados com mais riquezas de detalhes na seção [3.1](#). Nas seções seguintes são abordados detalhes de como a solução foi implementada e quais tecnologias foram utilizadas.

Os problemas e soluções apresentados no presente relatório foram baseados nas demandas específicas de um cliente real da Pitang. Por questões de confidencialidade, seus dados, como nome e informações sensíveis, serão ocultadas ou trocadas.

2 A Empresa e sua Atuação

2.1 Pitang

Originada através da cisão da área de Projetos Comerciais do Centro de Estudos e Sistemas Avançado do Recife (C.E.S.A.R), a Pitang, foi fundada no ano de 2004, contando inicialmente com 120 colaboradores e sua primeira sede localizada no prédio da empresa Votorantim, no bairro do Recife Antigo.

Hoje, 18 anos depois de sua fundação, sua sede continua localizada no bairro do Recife Antigo, porém agora no prédio do Porto Digital, e conta com um escritório em São Paulo e outro em Brasília. E possui aproximadamente 400 colaboradores espalhados pelo Brasil, trabalhando em formato de teletrabalho, nos escritórios ou alocados em clientes.



Figura 1: Mapa de Colaboradores
Fonte: [\[2\]](#)

2.1.1 Setor

A Pitang atua em todo o território nacional atendendo empresas do setor público e privado, de médio a grande porte, em diversas áreas de atuação, tais como: finanças e serviços financeiros, energia, governo, indústria, serviços e varejo. Desenvolvendo e entregando soluções inovadoras através do desenvolvimento de projetos que visam a transformação digital de seus clientes utilizando competências como:

- Computação Cognitiva;
- Plataformas de Experiências Digitais;
- Analytics;
- Computação em Nuvem;
- Aplicativos Móveis;
- UX;
- Agilidade;
- DevOps; e
- Testes.

2.1.2 Produtos

Um dos principais serviços oferecidos pela Pitang é o de Consultoria, e por isso ela investe em colaboradores com os mais diversos *skills* e experiências, permitindo assim que ela consiga entregar junto ao cliente um resultado personalizado e de qualidade, seja este uma consultoria especializada ou o desenvolvimento de produtos inovadores.

Além do investimento em pessoas, a Pitang desenvolveu e/ou adquiriu soluções de softwares que disponibiliza a seus clientes via licenças de uso. São estes:

- Voxia: é uma solução inteligente para gestão de conteúdos extraídos no ambiente jurídico;
- Merit Coin: Plataforma gamificada, simples e divertida que permite o reconhecimento e valorização dos colaboradores por meio do feedback em todos os níveis hierárquicos.
- PROMPT: Solução para gestão de projetos de Pesquisa e Desenvolvimento. Gestão 100% digital e transparente para as organizações que possuem projetos de apoio à pesquisa;
- Consensus: Uma solução ágil e inteligente que acelera acordos extrajudiciais;
- Atende em Casa: Plataforma de teleorientação, telemonitoramento e acolhimento de pessoas.
- P2F: Solução de gestão de documentos que auxilia empresas no mapeamento e automatização de processos a fim de erradicar o desperdício de energia, papel e retrabalho de equipes.

2.2 Atuação Profissional

2.2.1 Ingresso e Papéis

Ingressei na Pitang em setembro de 2019, contratado como Consultor de TI. Inicialmente fiquei alocado em um projeto da Secretaria de Educação e Esportes de Pernambuco (SEE-PE), onde dávamos suporte e desenvolvíamos novos sistemas para Gerência de Informações e Sistemas Aplicativos (GISA). Após a finalização do projeto junto a SEE, em outubro de 2020, ingressei como Consultor de TI no Projeto Pitang to Flow (P2F), onde estou até o momento.

Por a Pitang ter uma cultura que incentiva os seus consultores a desenvolver suas habilidades e fazerem o que gosta, mesmo tendo passado apenas em dois projetos, tive a oportunidade de atuar com diferentes papéis durante esse período. Realizando assim atividades de consultoria de processos e negócios, qualidade de software, gestão de equipe, e desenvolvimento de soluções.

2.2.2 Projeto Atual

O projeto P2F está dividido em duas células, uma de inovação e outra de sustentação.

Na célula de inovação a equipe é formada por 1 - Product Owner (P.O), 1 - Scrum Master (SM) e 4 - Desenvolvedores. Onde temos como objetivo a modernização da aplicação com tecnologias mais atuais, uma melhor experiência do usuário e novas funcionalidades que hoje não são suportadas no produto atual.

Na célula de sustentação a equipe é formada por 2 - Analista de Processo e 2 - Solucionadores. Onde temos como objetivo garantir que os problemas e necessidade de negócio do cliente sejam solucionados.

Minhas horas de atuação no P2F estão rateadas nas duas frentes. Atuando como SM na célula de inovação da aplicação, facilitando e apoiando no planejamento do projeto a fim de garantir que as entregas sejam realizadas dentro do prazo e escopo acordado junto ao P.O e que a equipe não tenha impedimentos. E como solucionador na célula de sustentação, garantindo que as necessidades de negócio, erros, e melhorias sejam desenvolvidas e disponibilizadas conforme solicitação dos clientes e/ou analistas de processo.

3 Desenvolvimento Realizado na Empresa

Esta seção descreve o desenvolvimento de software e de processos realizados para entregar a necessidade de negócio do cliente. Adicionalmente apresenta os resultados de minha atuação como solucionador na aplicação do P2F célula de sustentação, e por questão de confidencialidade, alguns dados podem ser ocultados ou trocados por informações fictícias para não infringir a Lei Geral de Proteção de Dados (LGPD).

3.1 Problemática e a Solução Proposta

A problemática e solução desenvolvida apresentada neste trabalho foi realizada para atender necessidades negociais específicas da Empresa X, cliente da Pitang, que possui licença de uso do produto P2F, detalhado na seção [3.2.1](#)

No final do ano de 2020, o cliente solicitou a Pitang, através do setor de recursos humanos (RH), que fosse modelado um novo processo eletrônico no P2F, denominado Benefício Vida (BV).

O BV tem como objetivo oferecer aos colaboradores da organização benefícios adicionais ligados à saúde, que são pagos em folha a cada colaborador. Todas as solicitações são analisadas pelo departamento de RH. Uma vez aprovada, as informações devem ser inseridas no sistema de recursos humanos (SRH) para que o pagamento seja realizado no mês subsequente da aprovação.

Visando agilizar a análise das solicitações e liberação do benefício aos colaboradores, a Empresa solicitou que a equipe do P2F, analista de processos e solucionador, juntamente com a do SRH realizasse um estudo de viabilidade para que as informações inseridas nos processos eletrônicos do BV, após a aprovação no RH, fossem inseridas de forma automática no SRH. Eliminando assim o retrabalho que a equipe teria de passar as informações de um sistema para o outro.

Até então, o P2F se comunicava com o SRH apenas para consumo de dados, não realizando o envio de informações a serem integradas no SRH. Esse cenário fazia com que muitos processos, já implantados na aplicação, demorassem muito tempo para terem seus dados inseridos ou atualizados no histórico do colaborador no SRH por serem feitos de forma manual.

Esse estudo de viabilidade da integração, por mais que nesse momento fosse para um cenário específico do processo do cliente, abriria um precedente para possíveis melhorias e automação de processos já existentes ou para novos a serem implantados no P2F.

Nas subseções [3.1.1](#) e [3.1.2](#) são detalhados os resultados do estudo de viabilidade e as customizações proposta a serem desenvolvidas para atender o resultado esperado pelo cliente.

3.1.1 Matadados P2F

Um ponto relevante, e impeditivo para continuidade da demanda de integração, inicialmente o escopo do trabalho, levantando durante o estudo de viabilidade foi a identificação da não aderência entre os campos dos processos do P2F com os campos que seriam alimentados via a integração no SRH.

O SRH possui campos com restrições de tamanho e/ou compostos exclusivamente por caracteres maiúsculos ou minúsculos. Enquanto que o P2F, essas limitações não existiam.

Esta incompatibilidade dos formatos dos dados entre um sistema e outro poderia causar problemas durante as tentativas de integração dos dados.

Visando eliminar esse bloqueio, conversas técnicas entre o solucionador do P2F e o desenvolvedor do SRH foram realizadas para levantar o melhor cenário técnico de atendimento da demanda.

As possibilidades levantadas foram de converter os dados enviados pelo P2F no SRH, converter

quando os dados tivessem no web service ou restringir os campos no mesmo formado no SRH no P2F. A solução escolhida foi a de restringir os campos no P2F. Essa alternativa traria benefícios não apenas para o processo específico do cliente, mas poderia ser utilizada em todo o produto P2F e incorporada a outros clientes.

A proposta de solução foi o desenvolvimento de metadados de restrição de campo que limitasse o número de caracteres, e formatação do texto em caixa alta ou baixa. Esses três novos metadados seriam desenvolvidos e implementados na aplicação. Possibilitando assim que a demanda de integração entre P2F e SRH no processo BV pudesse ser continuada.

3.1.2 Processo: Benefício Vida

Com a conclusão do desenvolvimento dos Metadados P2F, a demanda de levantamento do BV foi retomada. Abaixo apresento os critérios de aceitação levantando para que a integração atendesse às necessidade de automação do processo BV no envio e recebimento de dados entre P2F e SRH.

O objetivo do processo BV é proporcionar a todos os colaboradores da organização uma ajuda de custo adicional referente aos seus custos com operadores de benefícios de saúde (OB).

Para que os colaboradores possam abrir um processo de solicitação do benefício no P2F é necessário que os dados dos OB apresentados no processo aberto no P2F estejam cadastrados no SRH ou sejam enviados para cadastro antes que a solicitação de implantação seja realizada.

Dado essa premissa, foram definidos três requisitos base para desenvolvimento da integração. Para todos eles é necessário que o sistema retorne ao usuário com uma mensagem de sucesso e fracasso sobre a implantação da solicitação de benefício no SRH.

Requisito 1: Operadores de Benefício cadastradas no SRH e código do operador no P2F. Neste cenário o sistema deverá enviar para o SRH implantar os dados do processo.

Requisito 2: Operadores de Benefício cadastrados no SRH e sem código do operador no P2F. Neste cenário o sistema deverá trazer as informações do Operador para o P2F e o P2F retornar os dados para implantar no SRH os dados do processo.

Requisito 3: Operadores de Benefício não cadastrados no SRH e sem código do operador no P2F. Neste cenário o sistema deverá enviar ao SRH as informações do novo Operador de Benefício, o SRH realizar o cadastro, retornar ao P2F a confirmação do cadastro do operador, o P2F retorna os dados para implantar no SRH os dados do processo.

Toda a problemática e solução apresentada acima, assim como as tecnologias utilizadas para o seu desenvolvimento serão descritas nas seções seguintes.

3.2 Tecnologias Utilizadas

Nesta seção serão descritas as ferramentas, aplicações e serviços utilizados para desenvolvimento da solução de software entregue ao cliente. O desenvolvimento foi baseado utilizando a linguagem de programação Java 6 e Apache Groovy. O *Tech Stack* utilizado será melhor descrito nas subseções a seguir.

3.2.1 Pitang to Flow - P2F

A plataforma Pitang To Flow (P2F) é um sistema de gerenciamento de processos eletrônicos e documentos digitais, desenvolvido em Java6 e AngularJS 1.5. A aplicação foi idealizada com o objetivo de erradicar o desperdício de energia e o retrabalho no dia-a-dia da organização, times e equipes [3].

O P2F possui três módulos principais:

- Processos;
- Organograma; e
- Arquivo Digital.



Figura 2: P2F - tela principal
Fonte: o autor

O módulo de processos é baseado na metodologia de gestão de processos de negócio (BPM), e possibilita duas formas de implementação dentro de sua aplicação: uma Ad-Hoc e outra sequencial.

No modelo ad-hoc [3] sua implementação é estruturada de maneira que as atividades do processo possam ser executadas de forma não sequencial, a depender do usuário que esteja atuando no processo, sendo ele todo modelado na própria interface do P2F. Enquanto que no modelo sequencial [3] a ordem de execução é determinada no momento da modelagem do processo e seguirá sempre o mesmo fluxo de acordo com as regras e condicionais definidas no seu desenvolvimento. Para o modelo sequencial, a modelagem do processo é feita a ferramenta Bizagi [3.2.2], e após a modelagem o fluxo é exportado para o formato de arquivo XPDL e importado para o P2F.

Os módulos de organograma e arquivo digital são gerenciados dentro do P2F mas administrados pelo sistema de gestão de conteúdo e documentos, Nuxeo [3.2.3].

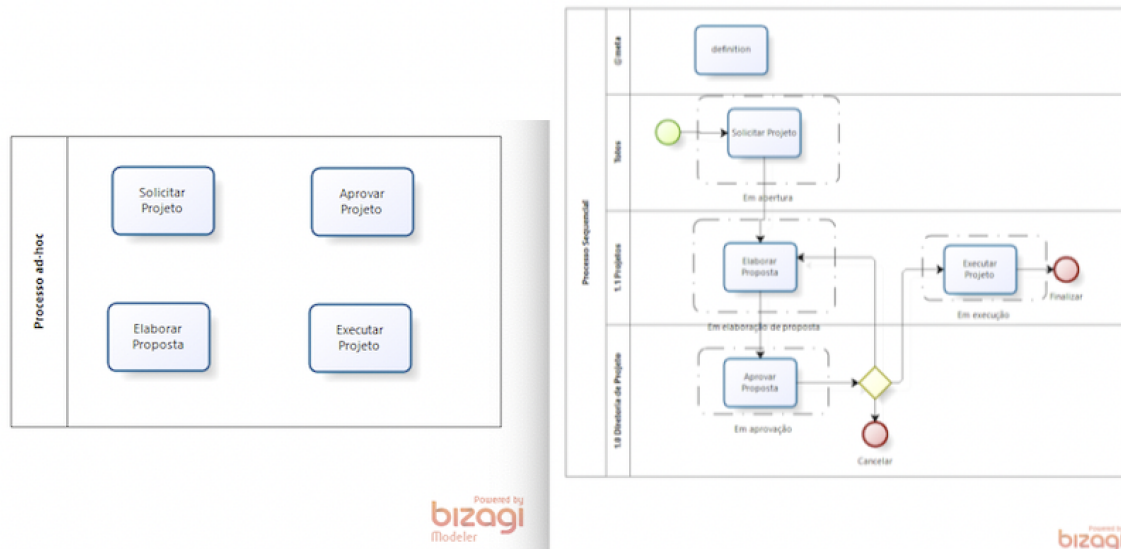


Figura 3: Modelo de processo ad-hoc a esquerda e sequencial a direita
 Fonte: o autor

3.2.2 Bizagi

Bizagi Modeler é um programa de software desenvolvido em Java por empresa de mesmo nome fundada em 1989. A aplicação é gratuita e apesar de compatível apenas com sistema operacional Windows é bastante utilizada mundialmente por organizações de vários portes e segmentos [4].

O Bizagi é uma aplicação que possibilita a diagramação, documentação e simulação gráfica de processos utilizando a notação global de processos, denominada Business Process Model and Notation (BPMN) [5].

As modelagens implementadas no Bizagi podem ser publicadas em diversos formatos como Word e PDF ou exportadas em formato de imagem e XPD. L.

Por ser uma aplicação desenvolvida em java e possibilitando que a modelagem seja exportada para XPD. L. O P2F utiliza sua base para a criação dos processos sequenciais utilizados pelos clientes. Possibilitando uma melhor validação do fluxo do processo, já que trata-se de uma ferramenta visual.

Para integração dos processos implementados no Bizagi ao P2F, foram e continuam sendo desenvolvidos metadados de integração. Esses metadados são incorporados às propriedades das tarefas do Bizagi no momento da modelagem, definindo as características e regras de cada campo do processo, como por exemplo se o campo será no formato numérico, textual, se o campo será obrigatório, opcional ou somente leitura e qualquer outra regra que seja necessária ao atendimento do negócio e processos dos clientes.

Todos os metadados utilizados na aplicação são desenvolvidos pela equipe da Pitang e sempre que surge uma nova necessidade de negócio, um novo desenvolvimento é implementado.

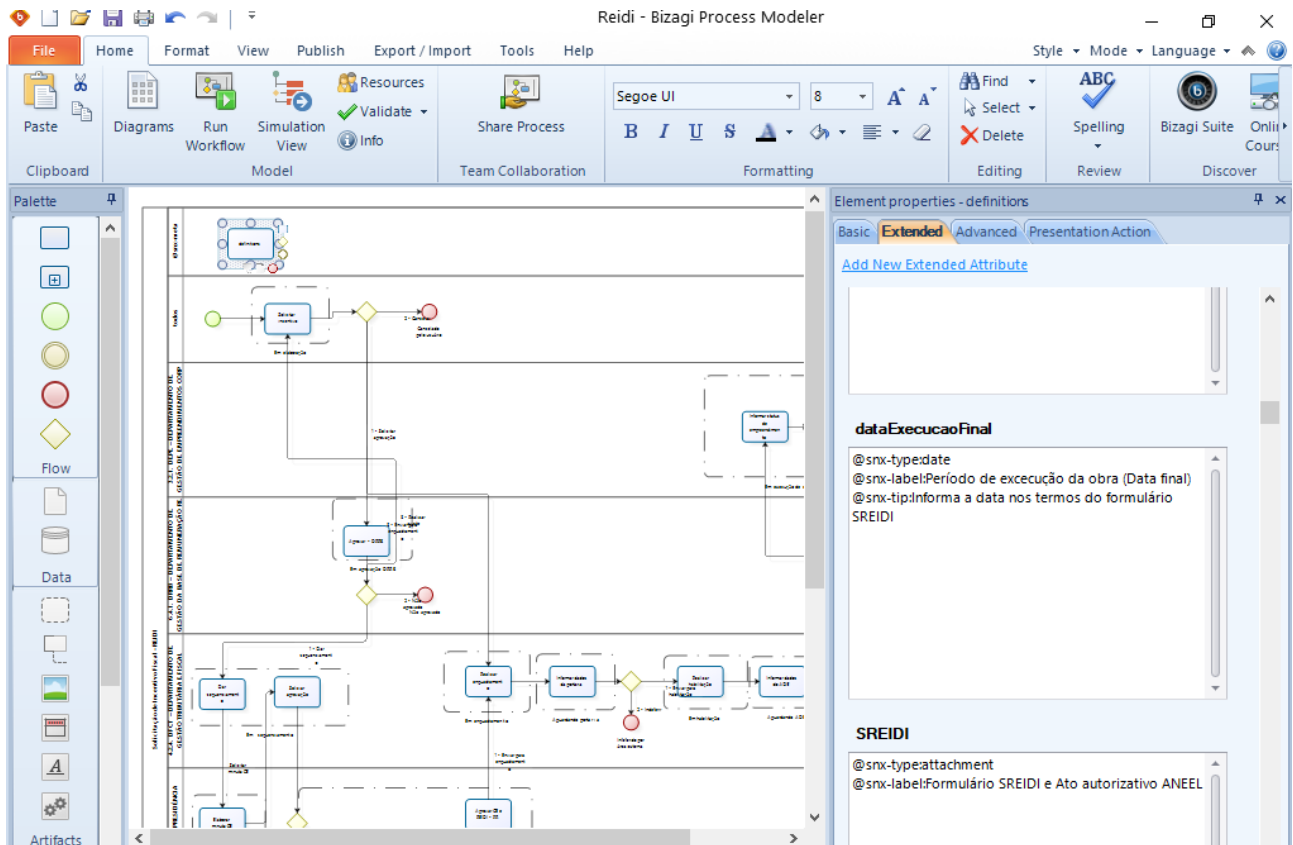


Figura 4: Exemplo de Metadado
Fonte: o autor

3.2.3 Nuxeo

Nuxeo é uma ferramenta de gerenciamento de conteúdo corporativo (ECM) Open Source, baseada em Java. A plataforma foi desenvolvida após uma re-arquitetura tecnológica que a empresa Nuxeo passou em 2005, porém o lançamento oficial do produto só se deu em fevereiro de 2010.

O Nuxeo possui uma arquitetura flexível, desenvolvida no modelo de camadas e que permite o gerenciamento de documentos de forma estruturada, realizando versões fluxos de trabalho associados aos documentos, busca avançada e controle de acesso [6].

A plataforma oferece versão gratuita, sem suporte e manutenção por parte da empresa Nuxeo e a versão *enterprise* e com vários serviços associados.

No projeto, utilizamos a versão *Open Source*, conforme imagem [5], para controle de acesso dos usuários e armazenamento de conteúdo da aplicação P2F.

3.2.4 Docker

Docker é uma plataforma *Open Source* desenvolvida com o objetivo de facilitar a criação e manutenção de ambientes isolados. A plataforma possibilita que uma aplicação ou serviço dela seja empacotado em uma caixa, denominadas de container, possibilitando a portabilidade desse container para qualquer outro hospedeiro que também seja dockerizado. Em linhas gerais, o Docker possibilita que em apenas uma única máquina, servidor, seja possível rodar N aplicações sem que

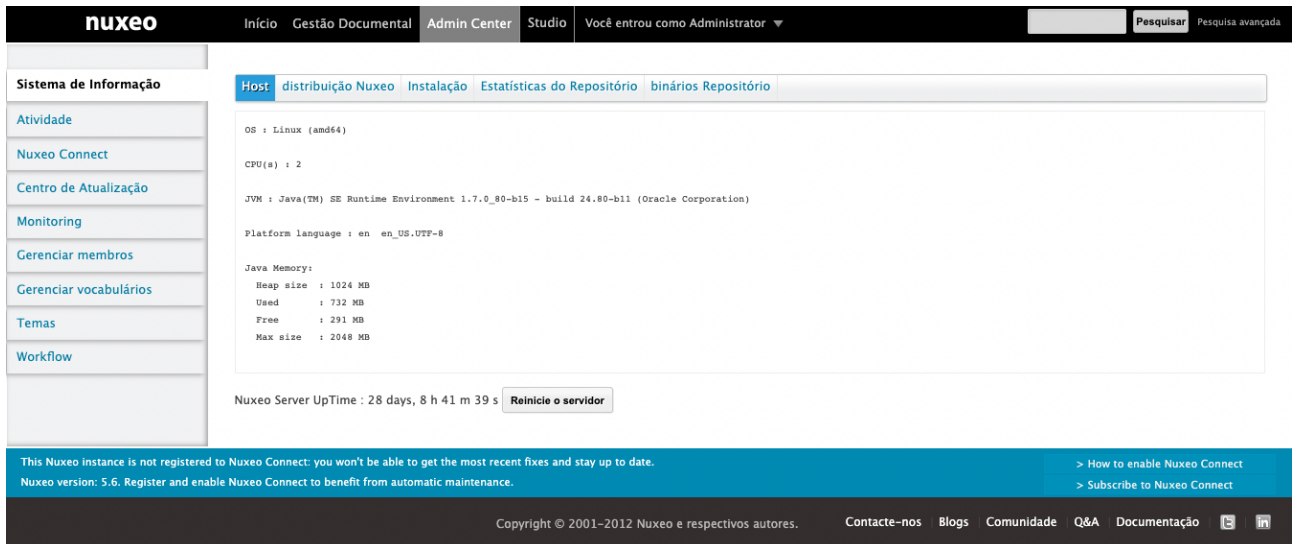


Figura 5: Nuxeo P2F
Fonte: o autor

estas tenham impacto entre si [7].

No projeto P2F o Docker é utilizado para isolar as aplicações P2F e Nuxeo em container distintos, permitindo os serviços funcionem de forma isolada. Facilitando na atuação e manutenção de ambas as aplicações sempre que necessário.

3.2.5 Web Service

Web services são funções de software que possibilitam a comunicação por meio de redes entre aplicações distintas mesmo que tais aplicações não tenham sido desenvolvidas com a mesma linguagem de software.

O termo web service é considerado um modelo abstrato, por não possuir uma interface de comunicação com o usuário. Por meio dele, sistemas conseguem compartilhar informações de forma controlada com outras aplicações. O modelo é baseado em padrões da internet definidos pelo World Wide Web (W3C) e por isso não precisam de configurações adicionais nos firewalls da rede, dado que o transporte é realizado via protocolo de comunicação, como HTTP por exemplo [8].

A comunicação é feita obedecendo padrões pré-definidos de como a transmissão será realizada, e existem diversos padrões, como por exemplo REST, SOAP, XML, JSON, dentre outros. A tomada de decisão sobre qual deles utilizar, dependerá das características de cada projeto [9].

3.3 Desenvolvimento

Nesta seção será descrito a solução desenvolvida utilizando o *Tech Stach* descrito na seção [3.2]

3.3.1 Arquitetura

A arquitetura deste trabalho tem como objeto central o P2F, que é o sistema utilizado para o desenvolvimento e modelagem dos processos eletrônicos disponibilizados aos clientes da Pitang e que encontra-se inserido dentro de um docker container, juntamente com o serviço de ECM da aplicação.

Conforme representação gráfica apresentando a arquitetura na imagem [6](#), o Bizagi é utilizado para que os processos sequenciais utilizados na aplicação possam ser modelados e importados ao P2F via upload de arquivo XPDL gerado pelo Bizagi.

Também é possível que o P2F se comunique com sistemas externos, via Web Service, seja apenas para enviar, receber ou enviar e receber dados. Essa comunicação é possível com o desenvolvimento de scripts de integrações, desenvolvidos na linguagem Groovy, que são incorporados na aplicação sem a necessidade de geração de nova versão.

Com uma clareza melhor sobre a arquitetura deste trabalho, nas seções seguintes serão apresentadas as soluções desenvolvidas.

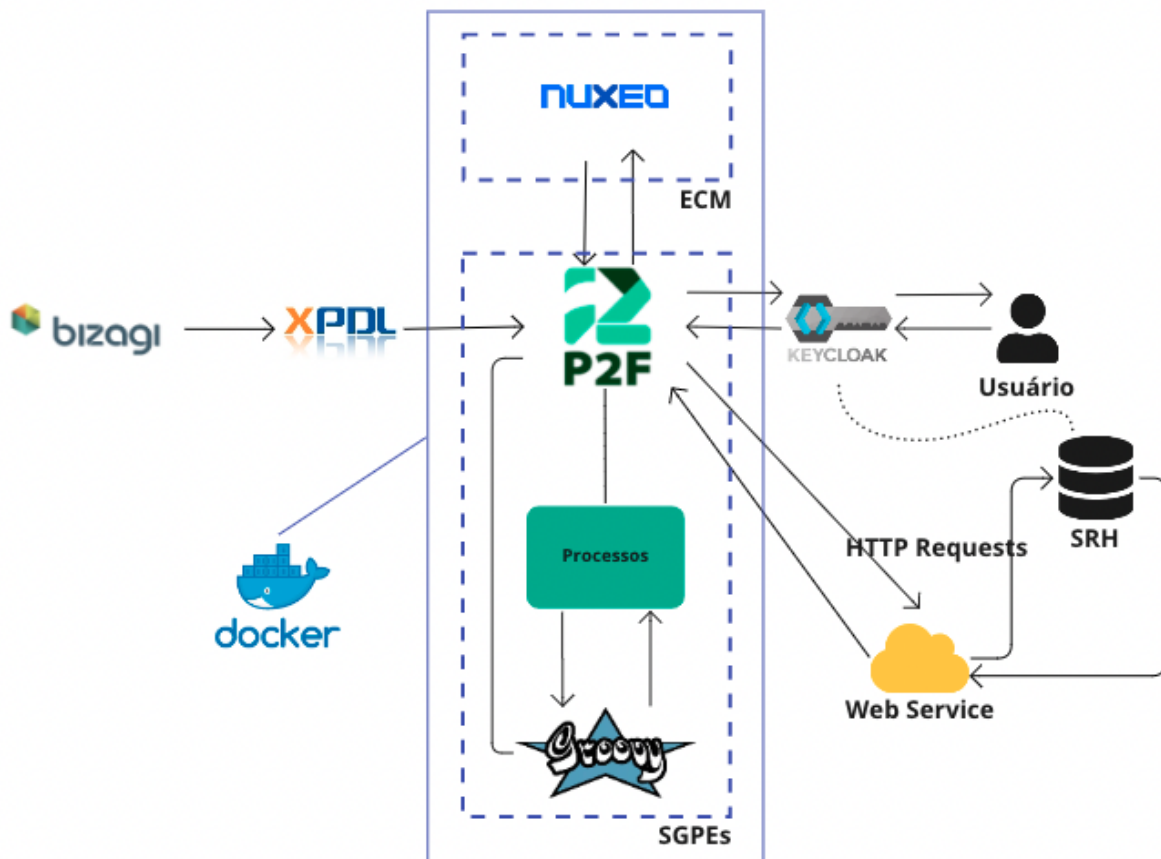


Figura 6: Estrutura arquitetural
Fonte: o autor

3.3.2 Solução 1: Metadados P2F

Para atender a solução apresentada no [3.1.1](#), foram criados e implementados na aplicação P2F três novos metadados de campo, conforme descrição abaixo:

- @snx-text-size: através desse comando será possível limitar o número de caracteres do campo texto para o número de caracteres informados na modelagem
- @snx-text-upercase: este metadado definirá que o campo texto terá apenas letras maiúsculas
- @snx-text-lowercase: este metadado definirá que o campo terá apenas letras minúsculas.

Após o desenvolvimento da solução uma nova versão foi gerada [7](#) para que a atualização fosse disponibilizada. Para geração da nova versão foi realizado o seguinte passo a passo:

1. Realizar o merge com as branches desenvolvidas;
2. Na ferramenta IntelliJ foi executado o comando Replace In Files e alterado a tag <version> do pom.xml presente em 39 arquivos para a nova versão;
3. Abrir a aba Maven no IntelliJ, ir em profiles e deixar marcado os profiles ao3-sonarqube, assemble-static-content-to-docker, depoy-on-nexus e nexus-pitang-repositories;
4. Ainda na aba Maven ir no módulo ao3-bmps, marcar as opções clean e deploy no lifecycle e dar play para geração de versão;
5. Entrar no servidor do cliente via SSH;
6. Executar o comando `sudo su - e_fds`;
7. Ir para o caminho `/opt/docker/app` e alterar o `docker-compose` para o número da versão gerada;
8. Descer e subir os containers através do comando: `docker-compose down docker-compose up -d docker-compose logs -f`

Após a subida da nova versão do P2F em homologação, foram realizados testes, e a solução foi homologada pelo cliente. Com a homologação, os mesmos procedimentos executados acima foram feitos para gerar e disponibilizar a nova versão em ambiente de produção do cliente.

3.3.3 Solução 2: Benefício Vida

Para atender as necessidades levantadas na seção [3.1.2](#), foi definido pela equipe de negócio junto ao cliente que as informações do processo deveriam ser disponibilizadas para um web service a partir do momento que o usuário finalizasse o processo na etapa de “Validar os dados” e o campo “Dados do(a) Colaborador(a) válidos para implantação no SRH?” estivesse preenchido com “Sim”. Ficando a empresa responsável pela aplicação do SRH a cargo da disponibilização do web service e informar como as informações enviadas seriam consumidas para que o desenvolvimento fosse realizado conforme esperado.

```

PITANG-253: /opt/docker/apps/bpms-new
bpms-dev-nginx | 192.168.48.1 - - [05/Aug/2022:15:24:21 -0300] "GET /assets/js/vendor/angular/angular.min.js.map HTTP/1.1" 200 437854 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.0.0 Safari/537.36"
bpms-dev-nginx | 2022/08/05 15:24:21 [error] 88#88: *6 open() /opt/www/assets/js/vendor/angular/angular-idle.map failed (2: No such file or directory), client: 192.168.48.1, server: , request: "GET /assets/js/vendor/angular/angular-idle.map HTTP/1.1", host: "localhost:8000"
bpms-dev-nginx | 192.168.48.1 - - [05/Aug/2022:15:24:21 -0300] "GET /assets/js/vendor/angular/angular-resource.min.js.map HTTP/1.1" 200 11037 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.0.0 Safari/537.36"
bpms-dev-nginx | 192.168.48.1 - - [05/Aug/2022:15:24:21 -0300] "GET /assets/js/vendor/angular/angular-sanitize.min.js.map HTTP/1.1" 200 10302 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.0.0 Safari/537.36"
bpms-dev-nginx | 192.168.48.1 - - [05/Aug/2022:15:24:21 -0300] "GET /assets/js/vendor/ui-select/select.min.js.map HTTP/1.1" 200 144307 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.0.0 Safari/537.36"
bpms-dev-nginx | 192.168.48.1 - - [05/Aug/2022:15:24:21 -0300] "GET /assets/js/vendor/angularjs-currency-input-mask/angularjs-currency-input-mask.js.map HTTP/1.1" 200 18035 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.0.0 Safari/537.36"
bpms-dev-nginx | 192.168.48.1 - - [05/Aug/2022:15:24:21 -0300] "GET /assets/js/vendor/angular-idle.map HTTP/1.1" 404 772 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.0.0 Safari/537.36"
bpms-dev-nginx | 192.168.48.1 - - [05/Aug/2022:15:24:21 -0300] "GET /assets/js/vendor/angular/angular-route.min.js.map HTTP/1.1" 200 12216 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.0.0 Safari/537.36"
bpms-dev-nginx | 192.168.48.1 - - [05/Aug/2022:15:24:21 -0300] "GET /assets/css/select.min.css.map HTTP/1.1" 200 11221 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.0.0 Safari/537.36"
bpms-dev | 15:24:27.288 [INFO] b.c.s.c.auth.NuxeoAuthenticationFilter : VERIFICANDO A SENHA DO USUÁRIO: Gerente
bpms-dev | 15:24:44.984 [INFO] b.c.s.c.auth.NuxeoAuthenticationFilter : VERIFICANDO A SENHA DO USUÁRIO: Gerente
bpms-dev | 15:25:30.930 [INFO] b.c.s.c.auth.NuxeoAuthenticationFilter : VERIFICANDO A SENHA DO USUÁRIO: Gerente
bpms-dev | 15:26:16.948 [INFO] b.c.s.c.auth.NuxeoAuthenticationFilter : VERIFICANDO A SENHA DO USUÁRIO: Gerente
bpms-dev | 15:27:02.922 [INFO] b.c.s.c.auth.NuxeoAuthenticationFilter : VERIFICANDO A SENHA DO USUÁRIO: Gerente
bpms-dev | 15:27:48.921 [INFO] b.c.s.c.auth.NuxeoAuthenticationFilter : VERIFICANDO A SENHA DO USUÁRIO: Gerente
^CERROR: Aborting.
PITANG-253: /opt/docker/apps/bpms-new$ dcdown
Stopping bpms-dev-nginx ... done
Stopping bpms-dev ... done
Stopping bpms-dev-nuxeo ... done
Stopping bpms-dev-mongodb ... done
Stopping bpms-dev-postgres ... done
Removing bpms-dev-nginx ... done
Removing bpms-dev ... done
Removing bpms-dev-nuxeo ... done
Removing bpms-dev-mongodb ... done
Removing bpms-dev-postgres ... done
Removing network bpms-new_bpms-dev-net
PITANG-253: /opt/docker/apps/bpms-new$

```

Figura 7: Geração de Versão
Fonte: o autor

O desenvolvimento da solução foi baseado no atendimento dos três requisitos referenciados na seção [3.1.2](#)

Em caso do **Requisito 1** apresentados na seção [3.1.2](#) acontecer, os dados [8](#) serão enviados para o SRH consumir via web service com a execução do script requisitarBeneficioVida [9](#), conforme modelo abaixo:

```

<requisitarBeneficioVida>
<matricula> <!-- Matrícula do colaborador --> </matricula>
<peessoaFi- sica> <!-- Número do pfis do colaborador --> </peessoaFisica>
<dataProcesso> <!-- Data do processo P2F --> </dataProcesso>
<codigoP2F> <!-- Código reservado para P2F --> </codigoP2F>
<peessoa- Juridica> <!-- Código (PJUR) da operadora --> </peessoaJuridica>
<valor> <!-- Valor do operador de benefício --> </valor>
</requisitarBeneficioVida>

```

Ao finalizar a execução o sistema do SRH deverá retornar informações ao P2F sobre o sucesso ou falha na implantação para que o usuário tome ciência. Em caso de sucesso, o usuário poderá fechar o processo, e em caso de falha poderá tentar enviar novamente os dados para implantação.

Em caso do **Requisito 2** acontecer o P2F irá chamar o script provider listarTodosOperadoresBeneficio [10](#), que tem a função de realizar uma requisição ao SRH para receber a lista de OB cadastradas no sistema, retornando um JSON com os dados de CNPJ, Nome do operador de benefício, Código ANS, Código de Pessoa Jurídica conforme modelo abaixo:

P2F	Regra	SRH - Resposta da Serviço	Campo do Banco
Dados do(a) Colaborador(a) válidos para implantação no SRH?	Enviar para o SRH as informações do processo se neste campo estiver preenchida a opção "Sim"		
Número do Processo	Enviar para o SRH o número do Processo que se encontra preenchido neste campo	codigoP2F	
Matrícula	Enviar para o SRH a matrícula que se encontra preenchida neste campo	matricula	RTIN_REMP_CHAPA
Pessoa	Enviar para o SRH o número da Pessoa Física	pessoaFisica	PFIS_NUMERO
Data da Conclusão da Abertura do Processo	Enviar para o SRH a última Data da Conclusão da Abertura do Processo. Esta informação consta no histórico resumido do Processo. Precisa ser a última data da abertura.	dataProcesso	DT_INI_VIG
Código da Operadora	Enviar para o Código da Operadora que se encontra preenchido neste campo	pessoaJuridica	DBHRM.PESSOAS_JURIDICAS.COD
CNPJ da Operadora do Plano de Saúde do(a) Colaborador(a)	Enviar para o SRH o CNPJ que se encontra preenchido neste campo	cnpj	DBHRM.ESTABELECIMENTOS.CGC
Nome da Operadora do Plano de Saúde do(a) Colaborador(a)	Enviar para o SRH o Nome da Operadora do Plano de Saúde que se encontra preenchido neste campo	nome	DBHRM.PESSOAS_JURIDICAS.NOME
Número do Reconhecimento pela ANS	Enviar para o SRH o Número do Reconhecimento pela ANS que se encontra preenchido neste campo	numero	
Valor do Plano de Saúde do(a) Colaborador(a) no momento do cadastro (DESCONSIDERAR QUALQUER VALOR DE COPARTICIPAÇÃO)	Enviar para o SRH o Valor do plano de saúde que se encontra preenchido neste campo	valor	

Figura 8: Campos enviados para implantação no SRH
Fonte: o autor

```

<resposta>
<sucesso>>true</sucesso> <!-- FLAG DE SUCESSO NA CONSULTA -->
<operadoras> listarOperadorBeneficioRequest
<operadora>
<cnpj> <!-- CNPJ do operador de benefício--> </cnpj>
<nome> <!-- Nome da operador de benefício --> <nome>
<codigoAns> <!-- Código da ANS --> </codigoAns>
<pessoaJuridica> <!-- Código da Pessoa Jurídica --> </pessoaJuridica>
</operador>
<operador>
...
</operador>
</resposta>

```

Caso o OB esteja cadastrado no SRH, o P2F então executará o script listarOperadorBeneficio 11 para recuperar os dados específicos do operador em questão.

```

30 ▼      } catch (SOAPFaultException sfe) {
31          log.error(" MESSAGE: ${sfe.message}")
32          log.error(" TEXT: ${sfe.text}")
33          log.error(" STATUS CODE: ${sfe.httpResponse.statusCode}")
34          log.error(" DETAIL: ${sfe.fault.detail.text()}")
35          throw new SOAPFaultException(sfe)
36
37 ▼      } catch (Exception e) {
38          log.error(e)
39          throw new Exception(e)
40      }
41
42      def map = returnMap(response.listarOperadorBeneficioResponse)
43      log.info(map)
44
45      def resultMap = [:]
46      resultMap["sucesso"] = map["resposta"]["sucesso"]
47      resultMap["operadoras"] = returnOperadora(map)
48
49 ▼      try {
50          resultMap["mensagem"] = map["resposta"]["erros"]["erro"]["mensagem"]
51 ▼          if (resultMap["mensagem"] == null) {
52              resultMap["mensagem"] = map["resposta"]["message"]
53          }
54 ▼      } catch (Exception ignored) {
55          resultMap["mensagem"] = null
56      }
57
58      log.info("Finalizou ListarTodosPlanoSaude")
59      return [resultMap]

```

Figura 9: Script requisitarBeneficioVida

Fonte: o autor

Após essa execução, mais uma vez o serviço requisitarBeneficioVida [9](#) será chamado pelo P2F e enviará os dados [8](#) para o SRH que deve retornar informando sobre o sucesso ou falha da implantação no sistema.

Em caso do **Requisito 3** ocorrer o P2F irá chamar o script provider listarTodosOperadorBeneficio [10](#) e não encontrando o OB o sistema irá enviar os dados do CNPJ, Nome do Operador e Código ANS do novo operado para que seja inserido no SRH através do script provider inserirOperadorBeneficio [12](#). O SRH retornará o sucesso ou falha da solicitação de serviço.

Ocorrendo falha no serviço de cadastro do novo Operador de Benefício, o usuário poderá tentar novamente.

Em caso de sucesso no cadastro, o sistema irá executar o script requisitarBeneficioVida [9](#) que deverá enviar os dados [8](#) para implantação no SRH, retornando ao usuário o resultado positivo ou negativo da execução.

3.4 Contribuição

Na célula de sustentação do P2F utilizamos a *framework* de gestão ágil Kanban para controlar as demandas que chegam para equipe atender. Todas as demandas, sejam de melhoria ou correção, são levantadas juntos aos clientes pelos analistas de processos e cadastradas como *Backlog* a ser desenvolvido por um dos solucionadores.


```

1 package scripts.providers
2
3 import org.apache.commons.logging.Log
4 import org.apache.commons.logging.LogFactory
5 import wslite.soap.SOAPClient
6 import wslite.soap.SOAPFaultException
7
8 class clienteListarTodosOperadorBeneficio {
9
10     private static final Log log =
11         LogFactory.getLog(clienteListarTodosOperadorBeneficio.class)
12     public static final String WSDL_HML =
13         "http://homologacao.cliente.com.br/servicorequerimento/ServicoProcesso?wsdl"
14     public static final String WSDL_PRD =
15         "http://www.cliente.com.br/servicorequerimento/ServicoProcesso?wsdl"
16
17     Object call(args) {
18         log.info("Iniciou clienteListarTodosOperadorBeneficio")
19
20         def soapClient = new SOAPClient(WSDL_HML)
21         log.debug(">>cnpj: " + args['CNPJ'])
22
23         def response
24         try {
25             response = soapClient.send(SOAPAction: "listarOperadorBeneficio") {
26                 envelopeAttributes "xmlns:ser":
27                     "http://servicorequerimento/v01/ServicoProcesso"
28                 body {
29                     "ser:listarOperadorBeneficioRequest" {
30
31                     }
32                 }
33             }
34         } catch (SOAPFaultException sfe) {
35             log.error(" MESSAGE: ${sfe.message}")
36             log.error(" TEXT: ${sfe.text}")
37             log.error(" STATUS CODE: ${sfe.httpResponse.statusCode}")
38             log.error(" DETAIL: ${sfe.fault.detail.text()}")
39             throw new SOAPFaultException(sfe)
40         }
41     }
42 }

```

Figura 10: Script listarTodosOperadorBeneficio
Fonte: o autor

As tarefas de correção e melhoria seguem fluxos diferentes antes de entrarem na coluna de A fazer do quadro Kanban do projeto. Para as tarefas do tipo correção são detalhados apenas os erros encontrados e que resultados ou comportamentos são esperados como retorno do sistema. Para atividades do tipo melhoria, os analistas escrevem um documento de requisitos com todos os cenários e critérios de aceitação negociais levantados junto ao cliente.

Este trabalho apresentou as soluções, do tipo melhoria, desenvolvidas para atender necessidades da Empresa X, detalhadas na seção 3.3.2 e 3.3.3. Nas subseções abaixo são detalhadas as contribuições do autor para que a solução fosse implementada e uma análise dos benefícios gerados para o cliente com a entrega da solução em ambiente de produção.

```

1 package scripts.providers
2
3 import org.apache.commons.logging.Log
4 import org.apache.commons.logging.LogFactory
5 import wslite.soap.SOAPClient
6 import wslite.soap.SOAPFaultException
7
8 class ClienteListarOperador {
9
10     private static final Log log = LogFactory.getLog(ClienteListarOperador.class)
11     public static final String WSDL_HML =
12         "http://homologacao.cliente.com.br/servicorequerimento/ServicoRequerimento?wsdl"
13     public static final String WSDL_PRD =
14         "http://www.cliente.com.br/servicorequerimento/ServicoRequerimento?wsdl"
15
16     Object call(args) {
17         log.info("Iniciou ClienteListarOperador")
18
19         def soapClient = new SOAPClient(WSDL_HML)
20         log.debug(">>cnpj: " + args['CNPJ'])
21
22         def response
23         try {
24             response = soapClient.send(SOAPAction: "listarOperadorBeneficio") {
25                 envelopeAttributes "xmlns:ser":
26                     "http://servicorequerimento.cliente.com.br/v01/ServicoRequerimento"
27                 body {
28                     "ser:listarOperadorBeneficioRequest" {
29                         "cnpj"(args['CNPJ'])
30                     }
31                 }
32             }
33         } catch (SOAPFaultException sfe) {
34             log.error(" MESSAGE: ${sfe.message}")
35             log.error(" TEXT: ${sfe.text}")
36             log.error(" STATUS CODE: ${sfe.httpResponse.statusCode}")
37             log.error(" DETAIL: ${sfe.fault.detail.text()}")
38             throw new SOAPFaultException(sfe)
39         }
40     }
41 }

```

Figura 11: Script listarOperadorBeneficio
Fonte: o autor

3.4.1 Contribuição do Autor

Apesar de todo o trabalho apresentado ter sido construído em equipe, o analista e solucionador da Pitang, departamento de RH da Empresa X, e equipe de desenvolvimento do SRH, cada membro do projeto contribuiu com sua área de expertise para que a necessidade de negócio levantada fossem implementadas e disponibilizadas em produção para os colaboradores da Empresa X utilizarem o BV. Nesta seção, descrevo as principais contribuições que produzi para o trabalho.

Análise de risco

No P2F, toda melhoria a ser implementada passa por uma análise de risco que a nova funcionalidade pode vir a causar no funcionamento da aplicação e segue pela estimativa de desenvolvimento. Neste trabalho, essa atividade foi realizada tanto para o desenvolvimento dos metadados de customização quanto para a implementação dos scripts de integração.

Foi criada uma matriz contendo as principais ameaças que poderiam afetar o desenvolvimento, a probabilidade de impacto, o tipo de resposta ao risco e a ação de contorno. Abaixo apresento a imagem [13](#) com alguns dos riscos levantados de ambas as soluções apresentadas neste trabalho.

```

1 package scripts.providers
2
3 import org.apache.commons.logging.Log
4 import org.apache.commons.logging.LogFactory
5 import wslite.soap.SOAPClient
6 import wslite.soap.SOAPFaultException
7
8 class inserirOperadorBeneficio {
9
10     private static final Log log = LogFactory.getLog(inserirOperadorBeneficio.class)
11     public static final String WSDL_HML =
12         "http://homologacao.cliente.com.br/servicorequerimento/ServicoRequerimento?wsdl"
13     public static final String WSDL_PRD =
14         "http://www.cliente.com.br/servicorequerimento/ServicoRequerimento?wsdl"
15
16     Object call(args) {
17         log.info("Iniciou inserirOperadorBeneficio")
18
19         def soapClient = new SOAPClient(WSDL_HML)
20         log.debug(">>cnpj: " + args['CNPJ'])
21         log.debug(">>nome: " + args['NOME'])
22         log.debug(">>ans: " + args['ANS'])
23
24         def response
25         try {
26             response = soapClient.send(SOAPAction: "inserirOperadorBeneficio") {
27                 envelopeAttributes "xmlns:ser":
28                     "http://servicorequerimento.cliente.com.br/v01/ServicoRequerimento"
29                 body {
30                     "ser:inserirOperadorBeneficioRequest" {
31                         "cnpj"(args['CNPJ'])
32                         "nome"(args['NOME'])
33                         "codigoAns"(args['ANS'])
34                     }
35                 }
36             }
37         } catch (SOAPFaultException sfe) {
38             log.error(" MESSAGE: ${sfe.message}")
39             log.error(" TEXT: ${sfe.text}")
40             log.error(" STATUS_CODE: ${sfe.httpResponse.statusCode}")
41         }
42     }
43 }

```

Figura 12: Script inserirOperadorBeneficio
 Fonte: o autor

Matriz de risco						
Descrição	Probabilidade	Impacto	Nível	Resposta	Responsável	Ação de Contorno
Falha na subida de versão em homologação e produção devido o deploy ser realizado manualmente	3	5	15	Mitigar	Solucionador	Detalhar o passo a passo na SMP. Acompanhar a subida com equipe de infra da Empresa X
Tempo elavado para o desenvolvimento dos metadados por falta de documentação da arquitetura dos metadados já existentes na aplicação	3	3	9	Aceitar	Solucionador	Estudar o código fonte e mapear as classes e funções ligadas aos metadados
Dificuldade do solucionador desenvolver os scripts de integração devido ao desconhecimento do negócio do cliente.	4	5	20	Mitigar	Analista	Envolver o Solucionador nas reuniões de alinhamento com o cliente.
Falha na integração devido a mudanças não comunicadas por parte do SRH	3	4	12	Transferir	Solucionador	Alinhar com cliente e SRH que qualquer mudança no escopo ou alinhamentos definidos em reunião impactarão a entrega da solução no prazo acordado. Reforçar necessidade de comunicação constante.

Figura 13: Matriz de Risco
 Fonte: o autor

Metadados

Conforme matriz dos riscos, apresentadas na imagem [13](#), o principal risco associado ao desenvolvimento dos novos metadados estava na falta de documentação e conhecimento da arquitetura dos metadados já existentes na aplicação. Para esta atividade, o plano de mitigação ao risco foi tirar algumas horas do dia para estudar o código e fazer o levantamento de todas as classes e funções que os metadados estavam associados, enquanto aguardávamos aprovação do cliente para iniciar.

Uma vez que todas as classes e funções foram mapeadas no código, o desenvolvimento da funcionalidade foi realizado de acordo com o tempo estimado.

Versão e Deploy

Com o desenvolvimento concluído o passo seguinte foi gerar a nova versão em ambiente de desenvolvimento para testes e validação do analista de processos. A geração de versão é realizada na ferramenta IntelliJ, onde alterado as tag de versão e dou início a geração da nova versão. No container docker realizo a mudança do docker-compose e subo a aplicação no ambiente de desenvolvimento para que o analista de processos possa iniciar os testes de negócio.

Como a equipe de sustentação é enxuta e não possui testador, todos os testes são realizados pelo analista de processos. Não existe um processo de code review do código gerado. Nesse ponto, toda homologação é feita apenas no âmbito comercial da aplicação.

Uma vez homologada a solução, o próximo passo é elaborar um documento técnico, denominado SMP, para que seja subida a nova versão nos ambientes de Homologação e Produção do cliente. A SMP serve como um manual para apoiar a equipe de infraestrutura da Empresa X com a descrição de tudo que está subindo na nova release e como essa subida deve ser feita. Esse documento é necessário pois no projeto não temos uma esteira de deploy, e além de ser feito manualmente, a equipe da Pitang não tem permissão de acesso aos ambientes de homologação e produção do cliente.

Durante a subida nos ambientes, o solucionador ficava disponível em *call* com equipe de infraestrutura do cliente para dar suporte caso algum problema na subida ou geração da versão ocorresse e para garantir que todo o roteiro da SMP fosse seguindo conforme instrução.

Scripts de Integração

Conforme matriz dos riscos, apresentadas na imagem [13](#), o principal risco técnico associado ao desenvolvimento da integração do P2F com o SRH estava relacionado a inexistência de uma API de integração já existente e que o desenvolvimento seria realizado separadamente pelas duas empresas ao mesmo tempo. Com o objetivo de mitigar a possibilidade de erros e retrabalho, realizamos algumas reuniões em conjunto para definir o escopo de entrega por parte da Pitang e por parte do SRH.

O resultado dos alinhamentos foi que o SRH ficaria responsável pelo levantamento do web service que consumiria os dados enviados pelo P2F e também os do SRH. O solucionador da Pitang ficou a cargo do desenvolvimento dos scripts para requisição e envio de dados do P2F.

Inicialmente a ideia seria criar uma integração que pudesse ser utilizada por todos os processos da Empresa X utilizados no P2F que fizesse integração com o SRH, e que poderia vir a ser utilizado por outros clientes que utilizavam o SRH. Contudo, na evolução do entendimento, o escopo da in-

tegração seria apenas para o processo BV, que era exclusivo a realidade da Empresa X, e devido essa especificação de customização, optei pelo desenvolvimento da integração via script Groovy, que é uma linguagem baseada em Java e que o código poderia ser anexado no código de scripts da aplicação sem a necessidade de geração uma nova versão. Eliminando assim a necessidade de geração de SMP para subidas dos ambientes.

A maior dificuldade encontrada durante a execução da demanda de integração foi no entendimento das regras de negócio e que informações deveriam ser enviadas e consumidas pelo serviço.

Para clarear o entendimento do negócio, o analista de processos modelou todo o fluxo e cenários no Bizagi. Esses modelos ajudaram na definição e criação dos scripts. Com o entendimento do que precisava ser feito. Desenvolvi os seguintes scripts:

- requisitarBeneficioVida
- listarTodosOperadoresBeneficio
- listarOperadorBeneficio
- inserirOperadorbenefio

Todos os scripts foram desenvolvidos usando a linguagem Groovy. Inicialmente foi inserido no ambiente de homologação, para que os teste fossem realizados pelo analista de processo e verificar se as integrações estão sendo realizadas e as novas informações inseridas no perfil de cada colaborador.

Durante os testes, ajustes no script com relação ao formato do envio dos dados de data foram realizados para poderem ser consumidos corretamente pelo SRH.

Após a homologação, os scripts foram disponibilizados em produção para que os colaboradores pudessem abrir os processos e requisitar o Benefício Vida.

3.4.2 Benefício ao Cliente

A solução desenvolvida para atender ao processo BV teve o início dos testes de homologação junto ao cliente em março de 2021. Esses testes foram realizados no ambiente de homologação do cliente.

Durante as validações necessidades de ajustes pontuais foram mapeadas tanto da parte do P2F quanto do SRH. Após a realização dos ajustes, nova rodada de teste foi agendada com o cliente e em maio de 2019 a solução foi publicada em ambiente de produção e comunicada aos usuários do sistema.

Desde a publicação do processo, no dia 19/05/2021, até a data de 16/09/2022 foram abertos 4.723 processos do tipo Benefício Vida. Tendo sido 73% desde já finalizados e grande parte implantando com sucesso no SRH. Os 27% restante destes processos, 1.272, estão em andamento, conforme [14](#).

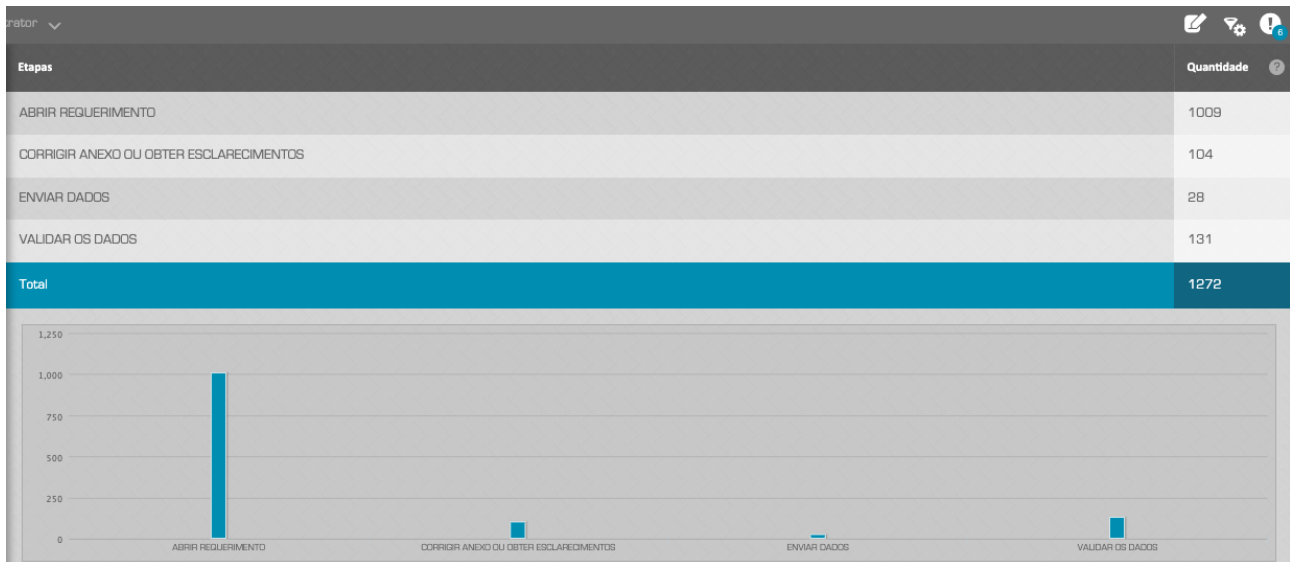


Figura 14: Processos BV em andamento
 Fonte: o autor

A entrega de valor com esse entrega para o cliente foi muito celebrada não apenas pelos colaboradores que ganharam mais um benefício, mas para o setor de gestão de pessoas que iria executar manualmente o cadastro dessas informações no SRH.

4 Dificuldades Encontradas

Durante o projeto alguns desafios foram postos no caminho. A seguir estão as maiores dificuldades mapeadas durante a execução.

4.1 Tecnologias

A principal dificuldade enfrentada foi precisar trabalhar com tecnologias que não estava familiarizado. Docker, Nuxeo, ECM, eram aplicações que não havia trabalhado antes e, como tudo novo, precisei correr atrás para aprender os conceitos e para que eram utilizados dentro do projeto e contexto de negócio do P2F.

Além das aplicações, as linguagens base do produto P2F são antigas, Java 6 no *Backend* e Angular JS 1.5 no *Frontend*, o que causaram uma certa estranheza no início do projeto devido grande parte da atuação na frente de sustentação ser restrita a correção e manutenção de algo que não foi desenvolvido pela pessoa.

4.2 P2F

Entrar na em um projeto para atuar na sustentação de um produto complexo e que já está no mercado a um tempo é um desafio enorme. A necessidade de revisar os conceitos e padrões das

tecnologias e linguagens utilizadas e ao mesmo tempo aprender sobre as regras de negócio da aplicação para atendimentos dos bugs e melhorias foram os principais empecilhos no início do projeto.

Contudo, estar em um projeto organizado, documentado e ter uma equipe de negócio disponível para apoiar e sanar as dúvidas foram pontos fundamentais para que em pouco tempo a problemática fosse amenizada.

4.3 Comunicação

De acordo com a pesquisa realizadas pelo PMI em 2014, comunicação é um dos principais problemas em projetos [10]. E durante a execução deste não foi diferente.

É fundamental que as pessoas que estão em contato direto com o cliente transpareçam para os demais membros da equipe as expectativas, dificuldades e prazos acordados para que todos estejam na mesma página e focados em garantir a entrega dentro do que foi pactuado.

As falhas de comunicação ocorridas no início da execução do projeto foram pontos dificultadores, mas uma vez que passei a ser incluído em reuniões e acompanhar mais de perto os levantamentos de requisitos essas dificuldades foram diminuindo.

4.4 Priorização

Outro dos grandes problema de quem trabalha na sustentação de produtos legados é a priorização das demandas e a urgência de todo novo item de backlog que é cadastrado na fila. A falta de um padrão claro para categorização sobre a criticidade desses itens deixa tudo na mão do analista que tiver recebendo a demanda e que em muitos casos não observa o backlog já existente.

Isso impacta diretamente a equipe que está na ponta, atuando na execução das demandas pois constantemente precisam parar o desenvolvimento de determina solução para iniciar outra ou resolver um problema que não era tão crítico assim e poderia ter sido apenas cadastrado para resolução futura.

5 Impactos da Formação no Trabalho

A academia teve e continuará tendo um impacto muito relevante sobre minha atuação profissional. Todas as experiências vivenciadas em cada disciplina cursada no decorrer da graduação, sejam elas obrigatórias ou optativas, foram extremamente importantes e me proporcionaram ingressar no mercado de trabalho mais preparado para os desafios que diariamente são colocados.

No trabalho, diariamente surgem problemas a serem solucionados, sejam eles no desenvolvimento de uma solução de software para um cliente ou projeto interno da empresa, treinamento de novos colaboradores, apresentação técnica nas inúmeras comunidade de práticas que tem na empresa, e posso dizer que durante a graduação todos esses *soft* e *hard skills* foram sendo cultivados em cada projeto que os professores passavam, em cada necessidade de passar a noite estudando para realizar uma avaliação da disciplina ou apresentar um projeto executado.

Em especial, o principal impacto que a universidade me proporcionou foi poder ver que minha formação não estava sendo guiada apenas para continuar na academia, mas sim para contribuir com a sociedade e retornar de alguma maneira o privilégio de estar cursando gratuitamente um curso superior em uma universidade pública e de qualidade através do projeto de extensão que participei na disciplina de IHC feito no Movimento Pró-Criança.

6 Conclusão

A necessidade que sistemas se comuniquem e compartilhem dados entre si para que empresas e gestores possam tomar melhores decisões ou que seus colaboradores possam atuar em atividades mais complexas e relevantes as suas funções ao invés de se preocuparem com atividades manuais e repetitivas é crescente. Este trabalho apresentou um cenário muito pontual de um cliente da Pitang, mas que é comum a muitas organizações.

A entrega de valor realizada ao cliente demandante, que foi a área de Gestão de Pessoas da Empresa X, foi de extrema importância aos seus colaboradores. Após a implementação do processo BV, outros processos entram na fila de levantamento da equipe de negócios. Sendo o primeiro, a expansão do benefício para os dependentes dos colaboradores e para possibilidade de realização alteração e exclusão de dados cadastrados no processo e implantados no SRH. Além desses, outros processos estão sendo levantando para entrarem no Backlog da equipe de negócio e consequentemente novas necessidades de integrações serão necessárias para que os processos sejam totalmente automatizados, liberando os colaboradores para atuarem em outras demandas que não podem ser automatizadas via processo eletrônico.

Durante a execução do projeto alguns problemas de comunicação, priorização e falta de conhecimento dos negócios e das tecnologias utilizadas foram impactantes para demora do desenvolvimento, porém estes foram desafios superados.

Por fim, é importante salientar que a contribuição deste projeto vai além dos números apresentados na seção [3.4.2](#).

Com a informatização dos processos para o formato digital e automatizado, considerando por exemplo apenas o processo BV, que em seu formato físico teria em média 4 folhas, pelo menos 2 árvores deixaram de ser utilizadas para produção de papel e cerca de 188.920L de água deixaram de ser utilizados para a produção desse papel apenas com um a digitalização e automação de um processo da organização.

No ano de 2021 foram abertos, no P2F, 58.492 processos eletrônicos dos mais diversos tipos. Esse valor representa uma economia ao meio ambiente enorme, e só foi possível por meio da transformação digital mediante a informatização e automação dos processos utilizados na Empresa X.

Referências Bibliográficas

- [1] A. Rabelo, "Transformação digital: o que é e quais os seus impactos na sociedade," 2020. [Online]. Available: <https://rockcontent.com/br/blog/transformacao-digital/>

- [2] Pitang, “Transformação digital com soluções inovadoras,” 2020. [Online]. Available: <https://www.pitang.com/a-pitang>
- [3] —, “Solução para gestão de documentos,” 2020. [Online]. Available: <https://www.pitang.com/a-pitang>
- [4] Bizagi, “Plataforma low-code de automatização de processos.”
- [5] techtudo, “Bizagi modeler: modele processos de negócio.” [Online]. Available: <https://www.techtudo.com.br/tudo-sobre/bizagi-modeler/>
- [6] Nuxeo, “Sobre a plataforma nuxeo,” 2022. [Online]. Available: <https://www.nuxeo.com/about/>
- [7] M. Guedes, “No final das contas: o que é o docker e como ele funciona?” 2019. [Online]. Available: <https://www.treinaweb.com.br/blog/no-final-das-contas-o-que-e-o-docker-e-como-ele-funciona>
- [8] G. Polo, “O que é um web service?”
- [9] M. Preti, “Entenda o que são web services e como eles podem reduzir os custos do seu site.” [Online]. Available: <https://c2ti.com.br/blog/entenda-o-que-sao-web-services-e-como-eles-podem-reduzir-os-custos-do-seu-site-tecnologia>
- [10] C. M. Xavier, “Comunicação é o principal projeto em projetos,” 2018. [Online]. Available: <https://beware.com.br/blog/comunicacao-principal-problema-em-projetos/>