|  |
| --- |
| uNIVERSIDADE REGIONAL DE BLUMENAU  CENTRO DE CIÊNCIAS EXATAS E NATURAIS  CURsO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO |
| BEDEVOPS: FERRAMENTA WEB DE APOIO A IMPLANTAÇÃO DA CULTURA DEVOPS  Ewerthon Ricardo Just |
| bLUMENAU  2020 |

|  |
| --- |
| Ewerthon Ricardo Just  BEDEVOPS: FERRAMENTA WEB DE APOIO A IMPLANTAÇÃO DA CULTURA DEVOPS  Trabalho de Conclusão de Curso apresentado ao curso de graduação em Sistemas de Informação do Centro de Ciências Exatas e Naturais da Universidade Regional de Blumenau como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação.  Profa. Simone Erbs da Costa, Mestre - Orientadora |
| bLUMENAU  2020 |
| Esta página deverá ser substituída pela folha de assinaturas entregue na Banca.  Digitalize a folha e cole aqui para a entrega da versão final do TCC.  Atenção: não ultrapasse as margens! |
|  |

Dedico este trabalho aos meus pais e professores que sempre me incentivaram nos estudos, fazendo com que eu alcançasse o objetivo de bacharel em Sistemas de Informação. Dedico também à minha namorada e a minha família, que sempre estiveram ao meu lado.

AGRADECIMENTOS

[Colocar menções a quem tenha contribuído, de alguma forma, para a realização do trabalho.]

“Tente uma, duas, três vezes e se possível tente a quarta, a quinta e quantas vezes for necessário. Só não desista nas primeiras tentativas, a persistência é amiga da conquista. Se você quer chegar aonde a maioria não chega, faça o que a maioria não faz”.

Bill Gates

RESUMO

O resumo é uma apresentação concisa dos pontos relevantes de um texto. Informa suficientemente ao leitor, para que este possa decidir sobre a conveniência da leitura do texto inteiro. Deve conter OBRIGATORIAMENTE o **OBJETIVO**, **METODOLOGIA**, **RESULTADOS** e **CONCLUSÕES**. O resumo deve conter de 150 a 500 palavras e deve ser composto de uma sequência corrente de frases concisas e não de uma enumeração de tópicos. O resumo deve ser escrito em um único texto corrido (sem parágrafos). Deve-se usar a terceira pessoa do singular e verbo na voz ativa (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2003).

Palavras-chave: Sistemas de informação. Monografia. Resumo. Formato.

[Palavras-chave são separadas por ponto, com a primeira letra maiúscula. Caso uma palavra-chave seja composta por mais de uma palavra, somente a primeira deve ser escrita com letra maiúscula, sendo que as demais iniciam com letra minúscula, desde que não sejam nomes próprios.]

ABSTRACT

*Abstract* é o resumo traduzido para o inglês. *Abstract* vem em uma nova folha, logo após o resumo. Escrever com letra normal (sem itálico).

Key-words: Computer science. Monograph. Abstract. Format.

[*Key-words* são separadas por ponto, com a primeira letra maiúscula. Caso uma *key-word* seja composta por mais de uma palavra, somente a primeira deve ser escrita com letra maiúscula, sendo que as demais iniciam com letra minúscula, desde que não sejam nomes próprios.]

LISTA DE Figuras

[Figura 1- Puppet Enterprise Status Dashboard 17](#_Toc54985765)

[Figura 2- Utilização do Puppet pelo software Ovenbird 20](#_Toc54985766)

[Figura 3- Tela de pipeline de implantação 21](#_Toc54985767)

[Figura 4 - Interface gráfica e exemplos de Jobs da ferramenta Jenkins 22](#_Toc54985768)

LISTA DE Quadros

[Quadro 1- Comparativo de resultados em caso de falhas no ambiente produtivo 18](#_Toc54985776)

Lista de tabelas

[Tabela 1 – Trabalhos finais realizados no Curso de Ciência da Computação 20](#_Toc48830092)

LISTA DE ABREVIATURAS E SIGLAS

[Deve conter as abreviaturas e siglas utilizadas mais de uma vez ao longo do texto em ordem alfabética. A seguir estão dois exemplos de forma de apresentação.]

ABNT – Associação Brasileira de Normas Técnicas

API – Application Programming Interface

SUMÁRIO

[1 Introdução 14](#_Toc56532935)

[1.1 OBJETIVOS 15](#_Toc56532936)

[1.2 estrutura 15](#_Toc56532937)

[2 FUNDAMENTAÇÃO TEÓRICA 16](#_Toc56532938)

[2.1 DEVOPS 16](#_Toc56532939)

[2.2 FERRAMENTAS DE AUTOMATAÇÃO 17](#_Toc56532940)

[2.3 ADIANTI BUILDER 17](#_Toc56532941)

[2.4 TRABALHOS CORRELATOS 17](#_Toc56532942)

[2.4.1 ESTUDO DE CASO DE UTILIZAÇÃO DA METODOLOGIA DEVOPS PARA ATENDER AO PROCESSO DE CONTINUIDADE DE SERVIÇOS CONFORME O FRAMEWORK ITIL 17](#_Toc56532943)

[2.4.2 AUTOMATIZAÇÃO DE ENTREGA DE SOFTWARE EM AMBIENTE ÁGIL DE DESENVOLVIMENTO 20](#_Toc56532944)

[2.4.3 UM PROCESSO DE INTRODUÇÃO DE DEVOPS EM SISTEMAS LEGADOS 23](#_Toc56532945)

[3 DESENVOLVIMENTO DA FERRAMENTA 25](#_Toc56532946)

[3.1 LEVANTAMENTO DE REQUISITOS 25](#_Toc56532947)

[3.2 ESPECIFICAÇÃO 25](#_Toc56532948)

[3.2.1 Diagrama de Caso de Uso 26](#_Toc56532949)

[3.2.2 Matriz de rastreabilidade dos RFs e sua relação com os Caso de Uso 26](#_Toc56532950)

[3.2.3 Modelo Entidade Relacionamento 26](#_Toc56532951)

[3.2.4 Diagrama de Classe 26](#_Toc56532952)

[3.2.5 Diagrama de atividades 26](#_Toc56532953)

[3.2.6 Diagrama de componentes 26](#_Toc56532954)

[3.2.7 Diagrama de implantação 26](#_Toc56532955)

[3.2.8 Esquema de tecnologias 26](#_Toc56532956)

[3.3 IMPLEMENTAÇÃO 26](#_Toc56532957)

[3.3.1 Técnicas e ferramentas utilizadas 26](#_Toc56532958)

[3.3.2 Operacionalidade da implementação 26](#_Toc56532959)

[3.4 RESULTADOS E DISCUSSões 27](#_Toc56532960)

[4 CONCLUSÕES 28](#_Toc56532961)

[4.1 EXTENSÕES 28](#_Toc56532962)

[Referências 29](#_Toc56532963)

[APÊNDICE A – Descrição dos Casos de Uso 33](#_Toc56532964)

[APÊNDICE B – Dicionário de Dados 34](#_Toc56532965)

[ANEXO A – Exemplo 35](#_Toc56532966)

# Introdução

As empresas de Tecnologia da Informação (TI), focadas no desenvolvimento de software, precisam buscar constantemente aprimorar a gestão de seus projetos (VARAJÃO, 2018). Neste contexto surgiu o fenômeno do desenvolvimento ágil, que busca envolver o cliente no trabalho da equipe de desenvolvimento, entregar com maior rapidez e de forma iterativa (NUOTTILA; AALTONEN; KUJALA, 2016). Cabe destacar que as empresas são motivadas pela necessidade de manter seus produtos e/ou serviços em um patamar de qualidade superior e assim inseri-los no mercado com mais rapidez (VARAJÃO, 2018).

Conforme a demanda por TI cresceu, métodos ágeis de desenvolvimento de software foram desenvolvidos no final da década de 90, trazendo uma nova abordagem para a organização, implementando a cultura de múltiplas tarefas (SATO, 2014). Com isso, as equipes operacionais tiveram que aumentar a frequência com que realizavam *deploys*, acarretando em gargalos no processo e as demandas geradas eram inalcançáveis, devido a não utilização de práticas ágeis (SILVA, 2016). De acordo com Humble e Farley (2014), essa evolução ocorreu apenas nas equipes de desenvolvimento de software, estabeleceu-se assim uma diferença de agilidade perante as equipes operacionais.

Sem a utilização destes métodos, as equipes operacionais tiveram que se desenvolver e começar a trabalhar de forma automatizada e dinâmica, para aumentar a velocidade de entrega de software em produção (SILVA, 2016). Humble e Farley (2010) complementam que com o aumento da demanda vinda por parte do desenvolvimento, a taxa de erros na implantação causados pelas equipes operacionais acabou se elevando. Com a iniciativa da automatização, começou-se a trabalhar em com mais colaboração entre as equipes, engajando-se no fornecimento de uma plataforma para realização de testes, implantação e execução de serviços em produção com padronização, segurança e rapidez (KIM *et al.*, 2016).

Surgiu então, um novo fenômeno a partir da união das palavras Desenvolvimento e Operações (DevOps). Diante deste cenário, propõe-se o desenvolvimento de uma ferramenta que auxiliará o usuário na implantação do DevOps, a partir de um diagnóstico inicial e de recomendações apresentadas pela ferramenta. Conjectura-se assim auxiliar na remoção das barreiras existentes entre as equipes de desenvolvimento e operações, direcionando o uso constante da colaboração entre os times, não apenas na implantação do software, mas em todo seu ciclo de vida (MEDRADO, 2015).

## OBJETIVOS

O objetivo deste trabalho é desenvolver uma ferramenta web de apoio à implantação de melhores práticas e processos do DevOps. Sendo os objetivos específicos:

1. disponibilizar para as organizações, por meio da aplicação de um questionário em uma interface web, um diagnóstico para implantação da cultura DevOps;
2. fornecer aos usuários um diagnóstico acompanhado de soluções e direcionamentos, tais como sugestões de ferramentas que auxiliam na implantação da automatização;
3. analisar e avaliar a usabilidade e a experiência de usuário das interfaces desenvolvidas pelo método Relationship of M3C with User Requirements and Usability and Communicability Assessment in groupware (RURUCAg), visando avaliar, de maneira simples, a usabilidade, a experiência de uso e os requisitos do sistema.

## estrutura

Este trabalho está dividido em quatro capítulos. O primeiro capítulo apresenta a introdução do trabalho desenvolvido, seus objetivos e sua estrutura do trabalho. No segundo capítulo, são abordados os conceitos e fundamentos de maior importância para o desenvolvimento da ferramenta, sendo eles: os conceitos da cultura DevOps, as ferramentas focadas em automação e os trabalhos correlatos. O terceiro capítulo traz os levantamentos de informações, os requisitos necessários para a implementação da ferramenta, as técnicas e as ferramentas utilizadas, bem como a operacionalidade da implementação e os resultados e discussões são apresentados. Por fim, são apresentadas as conclusões, bem como sugeridas as extensões para serem implementadas no futuro.

# FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os conceitos e fundamentos mais relevantes para elaboração do projeto desenvolvido. Na seção 2.1 é abordado o que é o DevOps. A seção 2.2 apresenta ferramentas para automação. Por fim, a seção 2.3 apresenta os trabalhos correlatos.

## DEVOPS

O termo DevOps surgiu pela unificação das palavras “desenvolvimento” e “operações”, que segundo Silva (2016) é um movimento cultural focado na melhoria da entrega de soluções ao cliente. Segundo Gene Kim *et al.* (2018) o movimento se tornou popular em 2009, quando Patrick Debois criou o primeiro *DevOpsDays* inspirado pela palestra ”10+ Deploys per Day: Dev and Ops Cooperation at Flickr” apresentada na conferência Velocity de 2009 por John Allspaw e Paul Hammond. Porém, segundo Correa (2017), as raízes do DevOps formaram-se em 2008, quando Patrick Debois publicou o artigo “*Agile and Operations Infrastrucuture: How Infra-gile Are You?*” (DEBOIS, 2008).

Segundo Ribeiro (2019), as equipes de desenvolvimento e operações possuem motivações distintas, sendo o desenvolvimento responsável pela evolução do produto, enquanto a equipe de operações é o responsável pela implantação e manutenção da infraestrutura. Fato é que as equipes tendem a seguir caminhos diferentes, principalmente por terem objetivos distintos (GAEA, 2020). Por isso DevOps apresenta em seu conceito práticas destinadas a integrar estas equipes por meio da comunicação e colaboração, simplificando processos, melhorando a qualidade e reduzindo o tempo de entrega (RIBEIRO, 2019).

Silva (2016) coloca que práticas como automação e uso de ferramentas constituem os pilares dessa nova cultura. DevOps é uma combinação de conceitos culturais, práticas e ferramentas que podem melhorar a capacidade da empresa de distribuir aplicativos e serviços com maior rapidez (AMAZON, 2020). Segundo Ribeiro (2019), com a adoção do DevOps nas organizações, as equipes de desenvolvimento e operações passam a atuar com os mesmos objetivos, deixando de trabalhar de forma isolada.

Ainda segundo Ribeiro (2019) na cultura DevOps, assim como em todas as culturas existentes, existem variações em sua adoção, sendo necessária a adaptação a realidade estrutural de cada orgaização. Entretanto, alguns aspectos assemelham-se dentre todas estas variações, sendo algumas delas a integração contínua, a automatização de processos, a colaboração e a comunicação (AMAZON, 2020). Alem destes Sharma (2014) coloca que outras semelhanças são: ambientes de desenvolvimento e teste equiparados ao de produção; rotinas de implantação padronizadas para que sejam repetíveis e confiáveis; monitoramento e validação da qualidade operacional; e o aumento dos feedbacks entre todas as partes.

## FERRAMENTAS DE AUTOMATAÇÃO

## TRABALHOS CORRELATOS

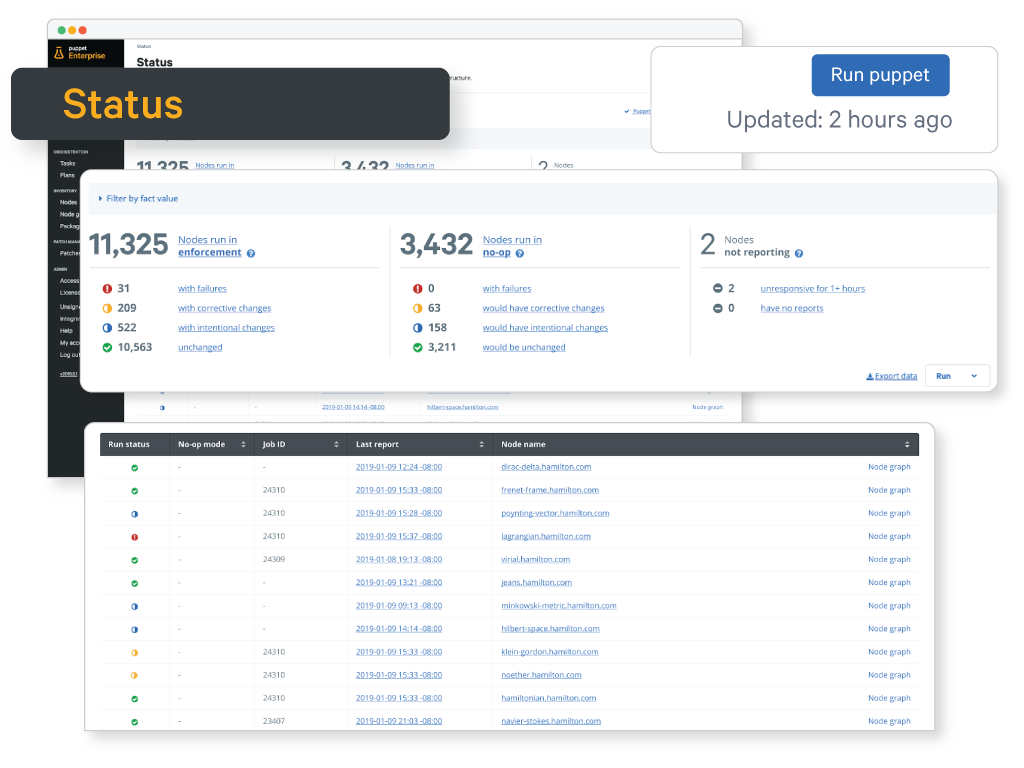
Nesta seção são apresentados trabalhos que apresentam semelhança com os principais objetivos do trabalho proposto. A subseção 2.1 aborda a implantação do DevOps em conformidade com a Biblioteca de Infraestrutura de Tecnologia da Informação, traduzida do inglês Information Technology Infrastructure Library (ITIL) (SILVA; GOMES, 2016). A subseção 2.2 apresenta a construção de um software capaz de integrar ferramentas, técnicas e boas práticas para implementar padrões passíveis de automatização do desenvolvimento de sistemas (NUNES, 2014). Por fim, a subseção 2.3 traz a aplicabilidade do DevOps em sistemas legados (CRUZ, 2018).

### ESTUDO DE CASO DE UTILIZAÇÃO DA METODOLOGIA DEVOPS PARA ATENDER AO PROCESSO DE CONTINUIDADE DE SERVIÇOS CONFORME O FRAMEWORK ITIL

Silva e Gomes (2016) tem como principal objetivo provar que as ferramentas e técnicas ágeis observadas no conceito de DevOps podem realmente garantir a continuidade dos serviços de Tecnologia da Informação (TI) de forma eficaz. Este estudo teve como foco a continuidade de serviço, a garantia da eficiência e flexibilidade de infraestrutura de Tecnologia da Informação (TI) e a avaliação da automatização de um processo antes feito manualmente (SILVA; GOMES, 2016). Para uma avaliação da implantação do DevOps, Silva e Gomes (2016) utilizaram duas Virtual Machines (VM), instalando Puppet Master e o Puppet Agent, para a realização de gerenciamento de configurações (PUPPET, 2020). As principais características identificadas são: apresenta indicadores da importância do DevOps; auxiliar a automatização por meio de ferramentas; instiga a automatização de processos e rotinas; avalia os ganhos com a implantação do DevOps (SILVA; GOMES, 2016).

Silva e Gomes (2016) ainda listaram três ferramentas que seguem o proposito DevOps, focadas em suportar a automatização da infraestrutura de Tecnologia da Informação (TI). Porém, para que o estudo não se tornasse muito grande, Silva e Gomes (2016) decidiram focar seus esforços na busca por uma ferramenta que auxiliasse na gestão de configuração, pois ela demonstra maiores impactos na continuidade de serviços. Sendo elas: Ansible (ANSIBLE, 2020); Chef (CHEF, 2020); e Puppet (PUPPET, 2020). A Figura 1 apresenta a ferramenta Puppet.

Figura 1- Puppet Enterprise Status Dashboard



Fonte: Puppet (2020).

A ferramenta Puppet (Figura 1) foi aplicada com a utilização de suas funcionalidades em uma instituição municipal governamental que presta serviços técnicos a outras agências vinculadas a sua organização (SILVA; GOMES, 2016). Silva e Gomes (2016) ainda colocam que o estudo foi composto por oito fases e que ele só se tornou possível com a concepção de governança e DevOps, sendo governança sobre “o que fazer” e para o DevOps a responsabilidade de “como fazer”. O ponto crucial do estudo foi a colaboração entre as equipes, pois a equipe de operações forneceu informações sobre segurança da informação e os componentes responsáveis ​​pela operação e o setor de desenvolvimento os detalhes técnicos e peculiares (SILVA; GOMES; 2016).

Como resultado do estudo, Silva e Gomes (2016) pontuam que com o auxílio da automatização na recuperação de um ambiente, não houve interrupções durante o trabalho. Em comparativo ao procedimento manual, além de gerar interrupções, foi necessário a concentração de recursos para que o serviço fosse reestabelecido, acarretando variações de solução devido a possibilidade de má-interpretação dos procedimentos necessários (SILVA; GOMES, 2016). Silva e Gomes (2016) apresentam também conforme disposto no Quadro 1, um resumo dos resultados obtidos no estudo de caso realizado.

Quadro 1- Comparativo de resultados em caso de falhas no ambiente produtivo

|  |  |  |
| --- | --- | --- |
| Descrição | Antes da implementação  (Processo Manual) | Após implementação  (Processo DevOps) |
| Rastreabilidade e Versionamento | As mudanças não podem ser rastreadas nem versionadas. | As mudanças nas configurações são versionadas o que possibilita a restauração do serviço a partir do ponto desejado  (infraestrutura como código). |
| Tempo de recuperação  Muito Baixo: < 30min;  Baixo: < 60min;  Moderado: 61min. < 120min;  Alto: 121min. < 360min.;  Muito Alto: > 361min. | Alto podendo chegar a Muito Alto se o especialista não estiver presente. | Muito baixo. Possibilidade de redução do *Service Level Agreement* (SLA). |
| Conhecimento da equipe (Membros que sabem restabelecer o serviço)  Muito ruim: 0% até 25%;  Ruim: 26% 49%;  Bom: 51%. < 75%;  Excelente 76% < 100%. | Muito ruim. Poucos sabem restabelecer o ambiente. Na ausência dos especialistas o trabalho é interrompido. | Bom. Mais membros conhecem o processo de restauração, já que o processo é simplificado com auxílio de uma ferramenta. |
| Padronização  Padronização Ausente: processo anual realizado aleatoriamente;  Pouca: processo manual que segue passos descritos em um modelo;  Moderada: processo automatizado, assistido por um operador;  Alta: sem a intervenção humana. | Pouca Padronização. Mesmo os passos sendo descritos por um *script* detalhado, o processo não tem garantias. Pode haver modificações dos passos por quem implementa. | Padronização Moderada. Só existe uma maneira de executar a restauração, mas o processo necessita ser iniciado e acompanhado por um operador. |
| Confiabilidade do ambiente  Baixo Risco: apresenta sempre os mesmos resultados.  Alto Risco: pode apresentar resultados diferentes. | Alto Risco. Os resultados podem não ser os mesmos desejados pelo negócio por se tratar de um processo manual. | Baixo risco. Os resultados são previamente testados e garantidos por um processo padronizado e automatizado. |

Fonte: Adaptado de Silva e Gomes (2016).

### AUTOMATIZAÇÃO DE ENTREGA DE SOFTWARE EM AMBIENTE ÁGIL DE DESENVOLVIMENTO

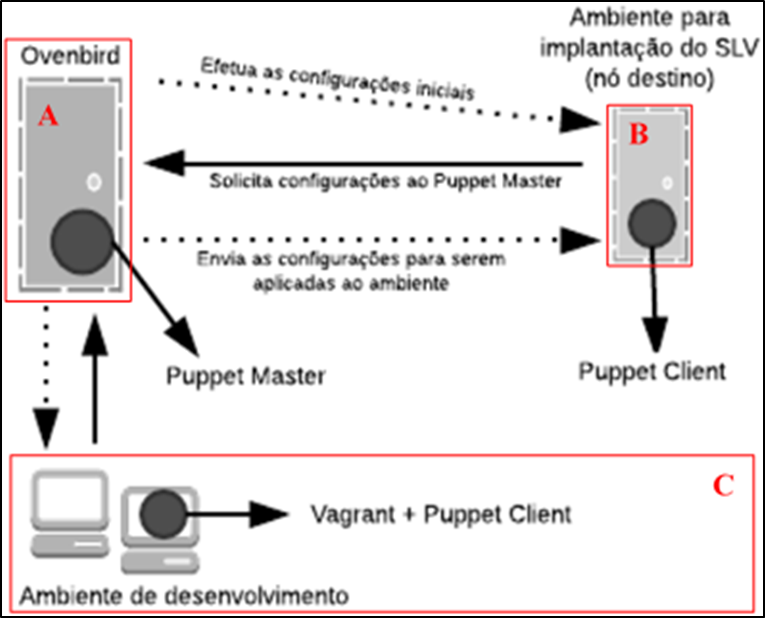
Nunes (2014) visou a construção de um software que pudesse integrar ferramentas, tecnologias e boas práticas para ajudar a realizar padronizações de automatização no desenvolvimento de sistemas web. Em seu desenvolvimento foram utilizadas como ferramentas web, a biblioteca Bootstrap[[1]](#footnote-1), Hypertext Markup Language (HTML) e da Cascading Style Sheets (CSS), tendo seu foco voltado a disponibilizar uma interface que permitisse acesso ao sistema via internet (NUNES, 2014). As principais características trazidas por Nunes (2014) são: auxilia na automatização por meio de ferramentas; instiga a automatização de processos e rotinas; e desenvolvimento de software auxiliador.

Nunes (2014) propôs o desenvolvimento do sistema Ovenbird com o objetivo de gerenciar a transição da solicitação desde todas as fases de desenvolvimento até a entrega ao usuário final. Por meio de uma interface amigável, abstraindo a complexidade das ferramentas disponíveis no mercado, de forma que seja fácil de usar por qualquer membro da equipe responsável pelas Soluções de Lojas Virtuais (SLV) (NUNES, 2014). Nunes (2014) queria que cada loja virtual fosse representada no Overnbird como um projeto devidamente nomeado a um cliente, sendo cada projeto composto pelos ambientes de implementação e requisitos que devem ser desenvolvidos (NUNES, 2014).

Os processos de disponibilização dos ambientes foram executados por meio de uma ferramenta de automatização de infraestrutura chamado Puppet[[2]](#footnote-2) para que fosse possível estabelecer e restabelecer o estado de um ambiente (NUNES, 2014). Motivado por grande parte das mudanças acontecerem em dados de configuração, já que para cada projeto deveriam ser possíveis inserir configurações dos ambientes. Na Figura 2 é demonstrado como a ferramenta foi utilizada (NUNES, 2014).

O retângulo A, da Figura 2, apresenta o servidor hospedeiro junto a um serviço Puppet Master, que fornece as configurações para os ambientes. No retângulo B é representado um ambiente hospedeiro de um SLV junto a um serviço Puppet Client, sendo responsável por obter as configurações do serviço Puppet Master do servidor hospedeiro. O retângulo C apresenta dois ambientes de desenvolvimento disponibilizados pelo Vagrant[[3]](#footnote-3) com configuração gerenciada também pelo Puppet (NUNES, 2014).

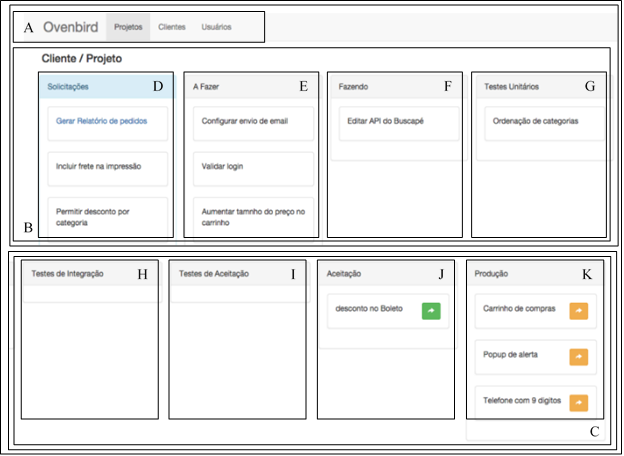
Figura 2- Utilização do Puppet pelo software Ovenbird



Fonte: Adaptada de Nunes (2014).

No software de Nunes (2014), para cada SLV, um projeto é criado no Ovenbird para agrupar solicitações e em cada novo projeto cadastrado é criado um espaço de trabalho na qual as solicitações são apresentadas. Além do cadastro de projetos ainda são possíveis cadastrar usuários, clientes, especialistas, ambientes e solicitações de funcionalidades (NUNES; 2014). Nunes (2014) também desenvolveu uma pipeline de implantação, compondo além das colunas que refletem ao quadro Kanban, outras que permitem tornar todas as etapas de desenvolvimento visíveis.

A Figura 3 demonstra a tela desta pipeline. O retângulo A apresenta os botões de navegação da tela principal do software desenvolvido por Nunes (2014). Os retângulos B e C apresentam as colunas do quadro com cada etapa do desenvolvimento; o retângulo D traz as Solicitações feitas pelos clientes no projeto; o retângulo E centraliza as tarefas pendentes, às então classificadas como A Fazer e no retângulo F o pipeline lista as tarefas que estão sendo realizadas; no retângulo G é possível visualizar a lista com as tarefas em fase de Testes Unitários; no retângulo H é possível visualizar as tarefas que estão realizando Testes de Integração, no próximo passo, demostrado pelo retângulo I, é apresentado a lista das tarefas que estão passando pelos Testes de Aceitação; por fim no Retângulo J é possível ver a fase de Aceitação das tarefas e no retângulo K a última fase de sua pipeline, a qual lista as tarefas predispostas a Produção.

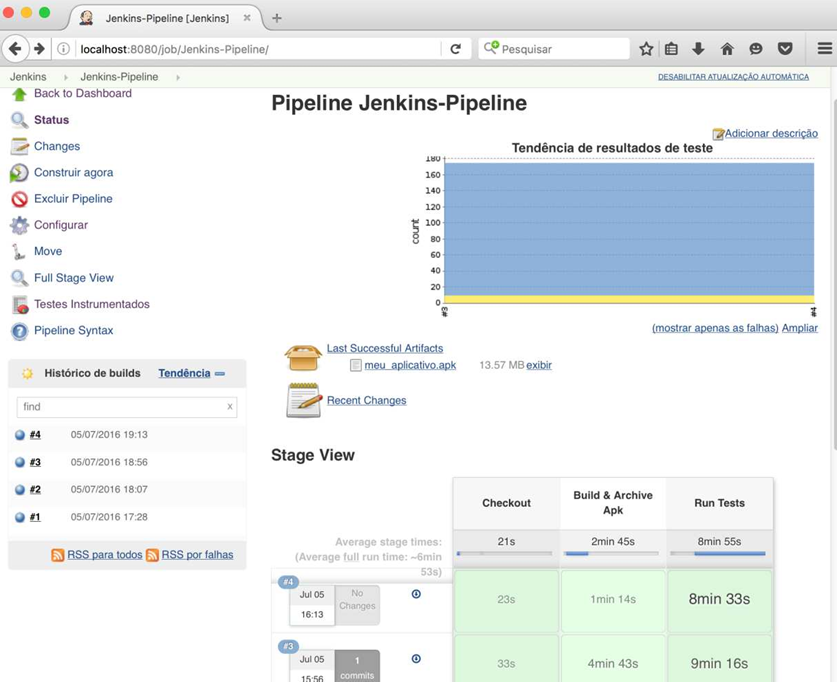
Figura 3- Tela de pipeline de implantação

Fonte: Adaptada de Nunes (2014).

### UM PROCESSO DE INTRODUÇÃO DE DEVOPS EM SISTEMAS LEGADOS

O objetivo da pesquisa de Cruz (2018) foi definir um processo com métodos, técnicas e tipos de ferramentas que auxiliasse a introduzir DevOps e Entrega Contínua para empresas que ainda possuam softwares legados. Cruz (2018) concentra grande parte de seu estudo de caso na utilização de ferramentas de integração contínua e automatização da Implantação de software, com a aplicação da ferramenta Jenkins[[4]](#footnote-4) em seu estudo de caso. As principais características do trabalho são: apresenta indicadores da importância do DevOps; auxilia a automatização por meio de ferramentas; apresenta dificuldades na implantação do DevOps; e instiga a automatização de processos e rotinas. A Figura 4 representa a interface gráfica do Jenkins apresentando a pipeline de implantação já criada por Cruz (2018). No retângulo A são apresentados por Cruz (2018) os três Jobs configurados em seu estudo de caso: *checkout*, *build* *and archive Android Package* (APK) e *run tests*.

Figura 4 - Interface gráfica e exemplos de Jobs da ferramenta Jenkins



A

Fonte: Adaptada de Cruz (2018).

Cruz (2018) aponta que os principais desafios enfrentados por empresas que tentam implantar DevOps e Entrega Contínua estão relacionados à mudança de cultura, pois todas as equipes envolvidas participam de todo o ciclo de vida de entrega do software. A principal mudança baseia-se na colaboração entre a equipe de desenvolvimento e de operações, além de que terão que ser adotadas novas ferramentas, novos processos e alterar a forma com que as equipes executam suas tarefas (CRUZ, 2018).

Cruz (2018) coloca ser fundamental o apoio da parte gerencial para uma inserção gradual dos processos, ferramentas e treinamentos, podendo ainda ocorrer resistência por membros das equipes. Outro desafio colocado por Cruz (2018) está relacionado as ferramentas que serão adotadas, pois a gama disponível no mercado é imensa, o que torna uma tomada de decisão muito mais difícil. Além de ser necessário definir se serão inseridas ferramentas pagas ou gratuitas, Cruz (2018) coloca ser importante verificar os prós e contras de cada uma das ferramentas utilizadas.

# DESENVOLVIMENTO DA FERRAMENTA

Neste capítulo, são abordados assuntos referentes ao desenvolvimento do sistema. A seção 3.1 apresenta o levantamento de requisitos. A seção 3.2 apresenta a especificação. A seção 3.3 detalha a implementação das técnicas e ferramentas utilizadas, bem como a operacionalidade da implementação. A seção 3.4 aborda sobre os resultados e discussões, trazendo a comparação entre os trabalhos correlatos e o trabalho desenvolvido, bem como traz o resultado da avaliação aplicada pelo Método RURUCAg.

## LEVANTAMENTO DE REQUISITOS

Nesta subseção, são elencados os principais Requisitos Funcionais (RF), disponibilizados no Quadro 2; Requisitos Não Funcionais (RNF), dispostos no Quadro 3 e as Regras de Negócios (RN), que podem ser vistos no Quadro 4. A ferramenta BeDevOps deverá:

Quadro 2 - Requisitos Funcionais

|  |
| --- |
| **Requisitos Funcionais** |
| RF01: manter o cadastro de usuários (Create, Read, Update and Delete - CRUD). |
| RF02: manter o cadastro de perguntas do questionário (CRUD). |
| RF03: manter o cadastro de categorias (CRUD). |
| RF04: manter o cadastro de ferramentas (CRUD). |
| RF05: manter o cadastro de relatórios (CRUD). |
| RF06: registrar as respostas do questionário elaborado (CRUD). |
| RF07: gerar relatórios de status em automatização e devidas orientações. |
| RF08: permitir consultar ferramentas. |
| RF09: permitir consultar perguntas. |
| RF10: permitir consultar relatórios. |
| RF11: permitir que o usuário possa realizar o log in no sistema. |
| RF12: permitir que o usuário altere sua senha de acesso. |
| RF13: permitir que o usuário se cadastre no sistema |

Fonte: elaborado pelo autor.

Quadro 3 - Requisitos Não Funcionais

|  |
| --- |
| **Requisitos Não Funcionais** |
| RNF01: ser desenvolvido utilizando a plataforma Adiati Builder. |
| RNF02: ser desenvolvido utilizando o MYSQL para a persistência de dados. |
| RNF03: possuir um domínio registrado pela Hostinger. |
| RNF04: ser hospedado totalmente na nuvem utilizando a AWS. |
| RNF05: manter a autenticação do usuário. |
| RNF06: ser responsivo nos navegadores Google Chrome versão 86.0.4240, Mozilla Firefox versão 82 ou superior. |
| RNF07: gerar relatórios em formato Portable Document Format (PDF) |
| RNF08: utilizar o serviço Simple Mail Transfer Protocol (SMTP) da Google. |
| RNF09: utilizar o método Relationship of M3C with User Requirements and Usability and Communicability Assessment in groupware (RURUCAg), para validar o uso da aplicação com profissionais da área de TI |

Fonte: elaborado pelo autor.

## ESPECIFICAÇÃO

Nesta seção é apresentada a especificação da ferramenta desenvolvida, contendo seus detalhes e diagramas. Na subseção 3.2.1 é mostrado o Diagrama de Caso de Uso (DCU) (Figura 5). A subseção 3.2.2 exibe a Matriz de rastreabilidade entre os RF (Quadro 2) e os Use Case (UC), bem como a relação dos RF aplicados ao Modelo 3C de Colaboração (Quadro 6). Na subseção 3.2.3 é apresentado o Modelo Entidade Relacionamento (MER), Diagrama de componentes na subseção 3.2.4 e o Diagrama de tecnologias na subseção 3.2.5.

### Diagrama de Caso de Uso

Esta subseção apresenta o DCU da ferramenta desenvolvida, conforme exibido na Figura 5, contendo os atores Administrador, Gerente e Colaborador. O UC que se refere à tela de cadastro de novo usuário é o UC01 – Manter usuários (CRUD), cuja responsabilidade é do usuário (Administrador). Além desta, outras reponsabilidades atribuídas ao usuário (administrador) são: cadastrar as ferramentas ilustrado pelo UC02 – Manter ferramentas; cadastrar categorias conforme apresentado no UC03 – Manter categorias; e cadastrar perguntas como demonstrado no UC04 – Manter perguntas.

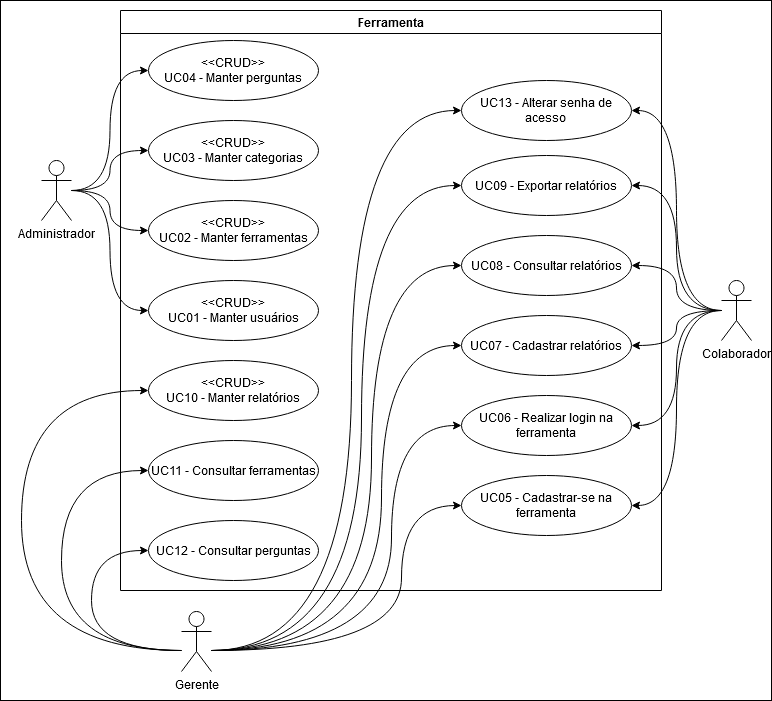
Os usuários (Gerente) e (Colaborador) podem realizar a ação de cadastrar-se, demonstrado por meio do UC05 – Cadastrar-se na ferramenta. Após o cadastro, eles podem realizar o log in, explicado por meio do UC06 – Realizar log in para ter acesso às demais funcionalidades da ferramenta. Um usuário (Gerente) será inicialmente um (colaborador) até que o mesmo solicite ao (Administrador) para que seja concedida a permissão de (Gerente).

Após o acesso, o usuário poderá responder ao questionário da ferramenta, ilustrado por meio do UC07 - Cadastrar relatórios. Para cadastrar um relatório o usuário deve informar um título, uma breve descrição e clicar em prosseguir para que o questionário com as perguntas seja apresentado. Após responder e comentar cada uma das perguntas e salvar o relatório, o usuário poderá apenas visualizar o relatório gerado em forma de gráfico demonstrado no UC08 – Consultar relatórios.

Após selecionar um dos relatórios cadastrados, ele poderá exportar o resultado detalhados de um relatório em forma de documento por meio do UC09 – Exportar relatórios. Além de visualizar e exportar, a permissão de excluir um relatório aplica-se apenas ao usuário (Gerente) conforme UC10 – Manter relatórios. Já a permissão de editar relatórios não se faz presente, uma vez que estaria infringindo a ideia de manter o histórico de evolução da implantação das ferramentas.

O usuário (gerente) pode realizar consulta das ferramentas cadastradas como ilustrado pelo UC11 – Consultar ferramentas. A consulta de perguntas demostrada pelo UC12 – Consultar perguntas também pode ser realizada elo usuário (Gerente). O usuário também pode alterar sua senha pela tela de perfil do usuário, ou também redefinir sua senha na tela de redefinir senha por meio do UC13 – Alterar senha de acesso.

Figura 5 – Diagrama de Caso de Uso



Fonte: elaborado pelo autor.

### Matriz de rastreabilidade dos RFs e sua relação com os Casos de Uso

Nesta subseção, é apresentado a matriz de rastreabilidade dos RF, bem como os UC, que podem ser vistos pelo Quadro 4.

Quadro 4 - Matriz de rastreabilidade dos RFs e sua relação com os UC

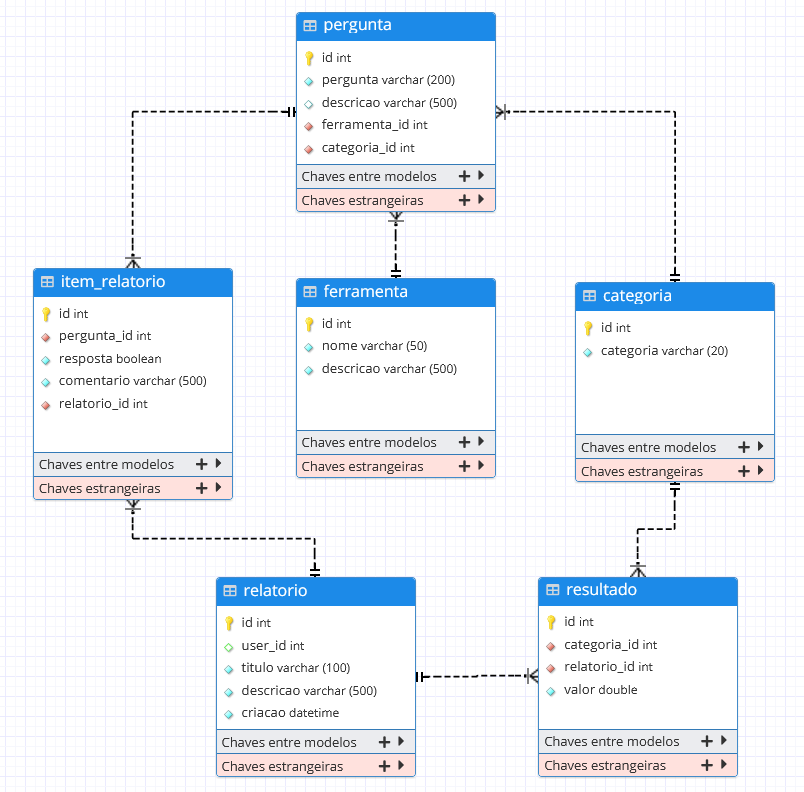
|  |  |
| --- | --- |
| **Requisitos Funcionais** | **UC** |
| RF01: manter o cadastro de usuários (Create, Read, Update and Delete - CRUD). | UC01 |
| RF02: manter o cadastro de perguntas do questionário (CRUD). | UC04 |
| RF03: manter o cadastro de categorias (CRUD). | UC03 |
| RF04: manter o cadastro de ferramentas (CRUD). | UC02 |
| RF05: manter o cadastro de relatórios (CRUD). | UC10 |
| RF06: registrar as respostas do questionário elaborado (CRUD). | UC07 |
| RF07: gerar relatórios de status em automatização e devidas orientações. | UC09 |
| RF08: permitir consultar ferramentas. | UC11 |
| RF09: permitir consultar perguntas. | UC12 |
| RF10: permitir consultar relatórios. | UC08 |
| RF11: permitir que o usuário possa realizar o log in no sistema. | UC06 |
| RF12: permitir que o usuário altere sua senha de acesso. | UC13 |
| RF13: permitir que o usuário se cadastre no sistema | UC05 |

Fonte: elaborado pelo autor.

### Modelo Entidade Relacionamento

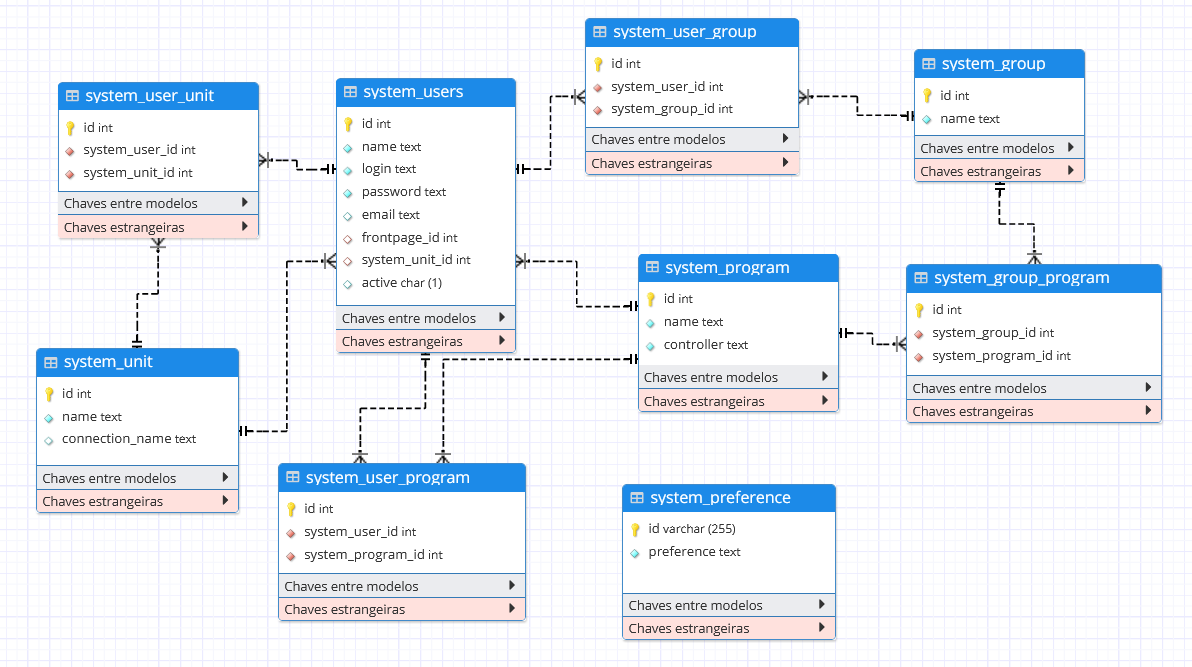
Esta subseção apresenta o Modelo Entidade Relacionamento (MER) da ferramenta desenvolvida. A Figura 6 mostra o modelo da estrutura do banco de dados desenvolvida pelo autor nomeada bedevops, em que são persistidas as informações da ferramenta. Neste modelo podem ser observadas as entidades da ferramenta que são representadas por tabelas, além dos seus relacionamentos e dependências com as outras tabelas. A ferramenta utilizada para realizar o desenvolvimento do trabalho pré-dispõem de modelos de estrutura do banco de dados para o funcionamento base da ferramenta, sendo elas de: permission (Figura 7); communication (Figura 8); e log (Figura 9). Já no Apêndice A pode ser encontrado o dicionário de dados destas tabelas. No MER também são apresentados os tipos de dados de cada atributo, assim como são diferenciadas as chaves primárias pela chave amarela, as chaves estrangeiras pelo losango vermelho, e as chaves estrangeiras entre modelos pelo losango verde.

Figura 6 – Modelo Entidade Relacionamento (bedevops)



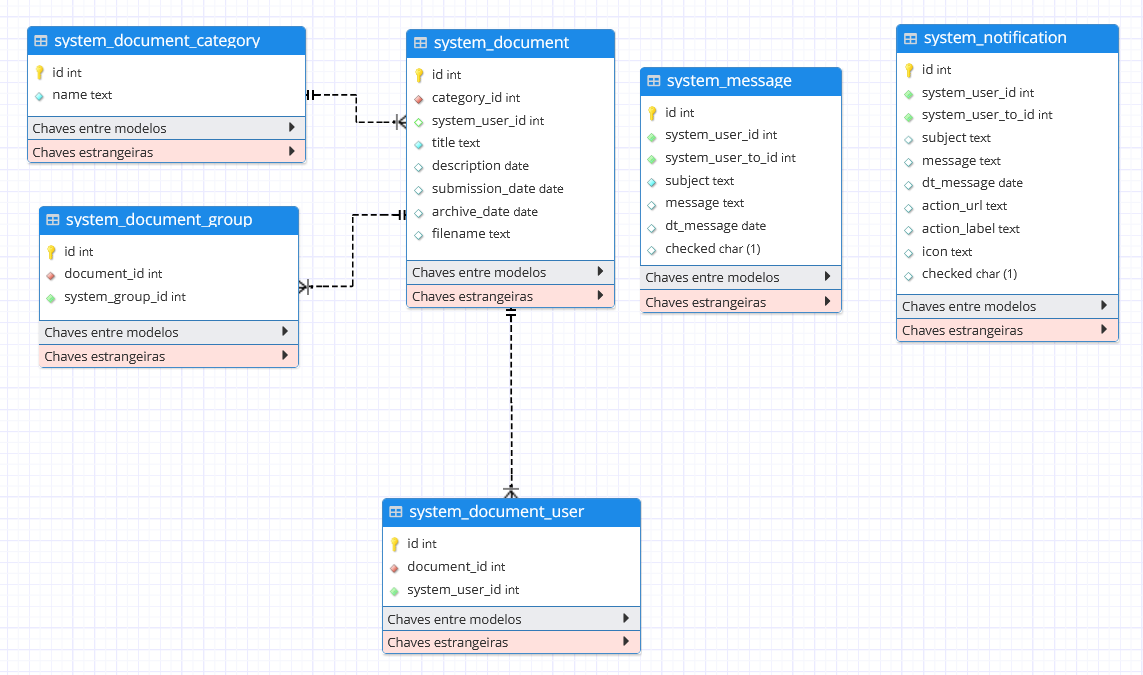
Fonte: elaborado pelo autor.

Figura 7 – Modelo Entidade Relacionamento (permission)



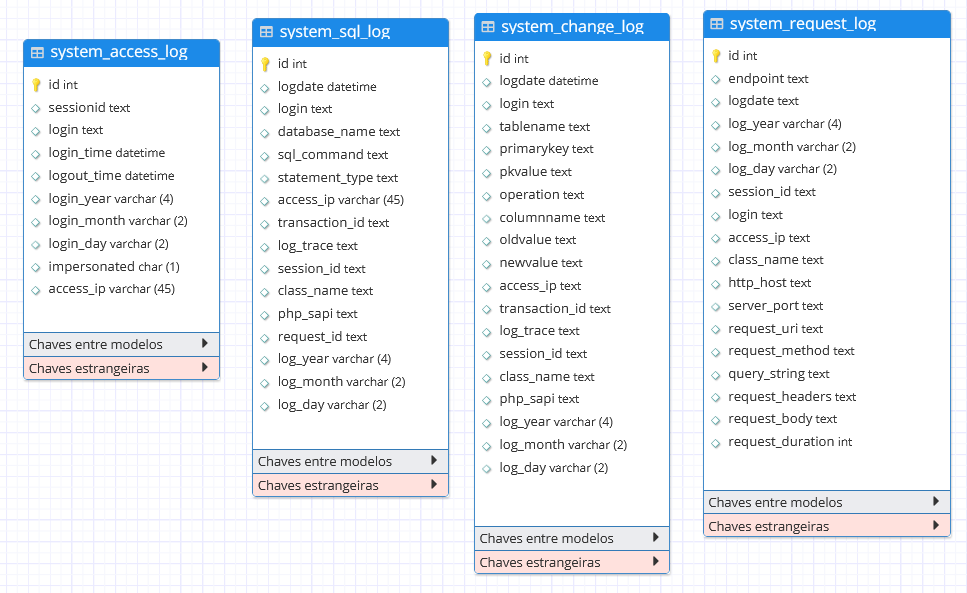
Fonte: AdiantiBuilder (2020).

Figura 8 – Modelo Entidade Relacionamento (communication)



Fonte: AdiantiBuilder (2020).

Figura 9 – Modelo Entidade Relacionamento (log)



Fonte: AdiantiBuilder(2020).

### Diagrama de atividades

O detalhamento da ferramenta desenvolvida pode ser melhor compreendido pelo diagrama de atividades (Figura 11). Na raia referente ao Colaborador, é possível notar as suas funcionalidades atribuídas, sendo elas: Cadastrar usuário, solicitar permissão de gerente caso seja um, cadastrar relatórios e, então, Consultaranálise de assinaturas. A próxima ação pode ser observada na raia do Coordenador, que realizará a análise de assinaturas, no qual o sistema realiza a comparação entre as assinaturas para descobrir se o usuário está pagando caro. O sistema também comunicará o resultado da análise de assinaturas, por meio de um feedback na interface, conforme pode ser observado na raia do Comunicador. Por fim, o Cooperador pode visualizar o resultado da análise.

### Diagrama de componentes ↓

### Diagrama de implantação

### Esquema de tecnologias

## IMPLEMENTAÇÃO

Nesta seção são descritas as técnicas e ferramentas utilizadas para o desenvolvimento do trabalho, além de ser detalhado o processo de implementação da ferramenta. A subseção 3.3.1 mostra as técnicas e ferramentas utilizadas no processo de criação do trabalho, e a subseção 3.3.2 apresenta a operacionalidade da implementação, com os diagramas de atividades e o funcionamento da ferramenta com suas respectivas telas.

### Técnicas e ferramentas utilizadas

[Considerações sobre as técnicas e ferramentas utilizadas para fazer a implementação a partir da especificação – exemplificar mostrando o código implementando.]

### Operacionalidade da implementação

[Apresentação do funcionamento da implementação (em nível de usuário) através de um estudo de caso.]

## RESULTADOS E DISCUSSões

Esta seção está organizada da seguinte forma: na subseção 3.4.1 é realizada a comparação entre os trabalhos correlatos e a ferramenta desenvolvida e na subseção 3.4.2 é apresentada a avaliação de usabilidade e comunicabilidade por meio do método RURUCAg, bem como os resultados obtidos na avaliação.

### Comparação entre os trabalhos correlatos e a ferramenta desenvolvida

### Avaliação de usabilidade e comunicabilidade pelo método RURUCAg

# CONCLUSÕES

[As conclusões devem refletir os principais resultados alcançados, realizando uma avaliação em relação aos objetivos previamente formulados. Deve-se deixar claro se os objetivos foram atendidos, se as ferramentas utilizadas foram adequadas e quais as principais contribuições do trabalho para o seu grupo de usuários ou para o desenvolvimento científico/tecnológico.]

[Deve-se também incluir aqui as principais vantagens do seu trabalho e limitações.]

## EXTENSÕES

[Sugestões para trabalhos futuros.]

Referências

CORREA, Rodolfo O. **Os desafios na adoção de devops em empresa brasileira de tecnologia da informação no setor bancário**. 2017. 102 f. Trabalho de conclusão de curso - Curso de Especialização em Gestão de Tecnologia da Informação, Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, São Paulo.

RIBEIRO, Gabriel D. **Automação de testes aplicados ao DevOps**. 2019. 61 f. Trabalho de conclusão de curso - Bacharel em Sistemas de Informação, Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais.

SILVA, Rachel R. da. **O crescimento de devops no mercado de tecnologia da informação**. 2016. 47 f. Trabalho de Conclusão de Curso (Pós-Graduação em Gestão da Tecnologia da Informação e da Comunicação) - Curso de em Gestão da Tecnologia da Informação e da Comunicação, Universidade Candido Mendes, Rio de Janeiro.

MEDRADO, Adonai Estrela. **DevOps e agregação de valor ao negócio**: Síntese e análise de práticas das organizações com atuação em território nacional. 2015. 57 f. Trabalho de conclusão de curso – Pós-Graduação em Arquitetura e Gestão de Infraestrutura em TI, AVM Faculdade Integrada, Rio de Janeiro.

DEBOIS, Patrick. Agile infrastructure and operations: how infra-gile are you?. In: Agile 2008 Conference, 2008, Toronto. **Anais […]** Toronto: IEEE, 2008. p. 202-207.

[As referências deverão ser apresentadas em ordem alfabética. Só podem ser inseridas nas referências os documentos citados ao longo da monografia. Todos os documentos citados obrigatoriamente têm que estar inseridos nas referências.]

[No formato do nome do autor, após a chamada (sobrenome com todas as letras em caixa alta), o primeiro nome deverá ser apresentado por extenso com a primeira letra em maiúscula e demais em minúscula e os outros nomes abreviados (letra em maiúscula seguida de ponto).]

[Abaixo são mostrados alguns exemplos de referências bibliográficas.]

[livro em meio eletrônico:]

ALVES, Castro. **Navio negreiro**. [S.l.]: Virtual Books, 2000. Disponível em: http://www.terra.com.br/vistualbooks/freebook/port/Lport2/navionegreiro.htm. Acesso em: 10 jan. 2002.

[parte de um documento:]

AMADO, Gilles. Coesão organizacional e ilusão coletiva. In: MOTTA, Fernando C. P.; FREITAS, Maria E. (Org.). **Vida psíquica e organização**. Rio de Janeiro: FGV, 2000. p. 103-115.

[trabalho acadêmico ou monografia (TCC/Estágio, especialização, dissertação, tese):]

AMBONI, Narcisa F. **Estratégias organizacionais**: um estudo de multicasos em sistemas universitários federais das capitais da região sul do país. 1995. 143 f. Dissertação (Mestrado em Administração) - Curso de Pós-Graduação em Administração, Universidade Federal de Santa Catarina, Florianópolis.

[norma técnica:]

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6023**: informação e documentação: referências - elaboração. Rio de Janeiro, 2002a. 24 p.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6024**: informação e documentação: numeração progressiva das seções de um documento escrito - apresentação. Rio de Janeiro, 2012. 4 p.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6027**: informação e documentação: sumário - apresentação. Rio de Janeiro, 2013. 2 p.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6028**: resumos. Rio de Janeiro, 2003. 2 p.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 10520**: informação e documentação: citações em documentos: apresentação. Rio de Janeiro, 2002b. 7 p.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 14724**: informação e documentação: trabalhos acadêmicos: apresentação. Rio de Janeiro, 2011. 11 p.

[livro:]

BARRASS, Robert. **Os cientistas precisam escrever**: guia de redação para cientistas, engenheiros e estudantes. São Paulo: Ed. da Universidade de São Paulo, 1979.

BASTOS, Lília R.; PAIXÃO, Lyra; FERNANDES, Lúcia M. **Manual para a elaboração de projetos e relatórios de pesquisa, teses e dissertações**. Rio de Janeiro: Zahar, 1979.

[guias do usuário:]

BORLAND INTERNATIONAL INC. **Delphi user’s guide**. Scotts Valley: Borland, 1995.

[help:]

BORLAND SOFTWARE CORPORATION. **Delphi enterprise**: help. Version 3.0. [S.l.], 1997. Documento eletrônico disponibilizado com o Ambiente Delphi 3.0.

[trabalho acadêmico ou monografia (TCC/Estágio, especialização, dissertação, tese):]

BRUXEL, Jorge L. **Definição de um interpretador para a linguagem Portugol, utilizando gramática de atributos**. 1996. 77 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

[verbete de enciclopédia em meio eletrônico:]

EDITORES gráficos. In: WIKIPEDIA, a enciclopédia livre. [S.l.]: Wikimedia Foundation, 2006. Disponível em: http://pt.wikipedia.org/wiki/Editores\_graficos. Acesso em: 13 maio 2006.

[artigo em evento:]

FRALEIGH, Arnold. The Algerian of independence. In: ANNUAL MEETING OF THE AMERICAN SOCIETY OF INTERNATIONAL LAW, 61, 1967, Washington. **Proceedings…** Washington: Society of International Law, 1967. p. 6-12.

[artigo em evento em meio eletrônico:]

GUNCHO, Mário R. A educação à distância e a biblioteca universitária. In: SEMINÁRIO DE BIBLIOTECAS UNIVERSITÁRIAS, 10, 1998, Fortaleza. **Anais...** Fortaleza: Tec Treina, 1998. 1 CD-ROM.

[norma técnica:]

IBGE. **Normas para apresentação tabular**. 3. ed. Rio de Janeiro, 1993. 61 p. Disponível em: http://biblioteca.ibge.gov.br/visualizacao/monografias/GEBIS%20-%20RJ/normastabular.pdf. Acesso em: 27 ago. 2013.

[artigo de periódico:]

KNUTH, Donald E. Semantic of context-free languages. **Mathematical Systems Theory**, New York, v. 2, n. 2, p. 33-50, Jan./Mar. 1968.

[parte de um documento:]

LAKATOS, Eva M. Cultura e poder organizacional e novas formas de gestão empresarial. In: LAKATOS, Eva M.. **Sociologia da administração**. São Paulo: Atlas, 1997. cap. 5, p. 122-143.

[artigo em periódico em meio eletrônico:]

MALOFF, Joel. A internet e o valor da "internetização". **Ciência da Informação**, Brasília, v. 26, n. 3, 1997. Disponível em: http://www.ibict.br/cionline/. Acesso em: 18 maio 1998.

[trabalho acadêmico ou monografia (TCC/Estágio, especialização, dissertação, tese):]

SCHIMT, Héldio. **Implementação de produto cartesiano e métodos de passagem de parâmetros no ambiente FURBOL**. 1999. 86 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

SCHUBERT, Lucas A. **Aplicativo para controle de ferrovia utilizando processamento em tempo real e redes de Petri**. 2003. 76 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

[página da internet: se a página não for livro, artigo ou parte de documento em meio eletrônico, deve-se fazer a referência conforme o exemplo abaixo. (O ano da página abaixo descrita não existe explicitamente descrito. Ele foi obtido a partir de informações fornecidas pelo *browse* Mozilla, através da opção “*Page Info*” alcançado através da opção do menu “*View*”. Foi pego a data da última alteração (*modified*). Quando a data for indefinida, colocar uma provável, sendo que neste caso vai entre colchetes e logo após o ano existe o símbolo de interrogação “?” (ex.: ..., [2003?] . Disponível em: ...). Quando a data estiver explicita na página, colocar esta sem colchetes. Se o mês também estiver explicito, colocá-lo (ex.: ..., out. 2003. Disponível em: ...)):]

SCHULER, João P. S. **Tutorial de Delphi**. Porto Alegre, [2002]. Disponível em: http://www.schulers.com/jpss/pascal/dtut/. Acesso em: 27 ago. 2013.

[artigo em evento:]

SILVA, José R. V. et al. Execução controlada de programas. In: SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, 1., 1987, Petrópolis. **Anais**... Petrópolis: UFRJ, 1987. p. 12-19.

[artigo em evento em meio eletrônico:]

SILVA, Roseane N.; OLIVEIRA, Ramon. Os limites pedagógicos do paradigma da qualidade total em educação. In: CONGRESSO DE INICIAÇÃO CIENTÍFICA DA UFPe, 4, 1996, Recife. **Anais eletrônicos...** Recife: UFPe, 1996. Disponível em: http://www.propesq.ufpe.br/anais/anais/educ/ce04..htm. Acesso em: 21 jan. 1997.

[livro:]

SEBESTA, Robert W. **Conceitos de linguagens de programação**. 4. ed. Porto Alegre: Bookman, 2000.

[parte de um documento em meio eletrônico:]

TEODOROWITSCH, Roland. **Manual de ética, estilo e português para a elaboração de trabalhos acadêmicos**. [Gravataí], 2003. Disponível em: http://www.ulbra.tche.br/~roland/pub/etica-est-port-2003-2.pdf. Acesso em: 28 mar. 2006.

[relatório de pesquisa:]

VARGAS, Douglas N. **Editor dirigido por sintaxe**. 1992. Relatório de pesquisa n. 240 arquivado na Pró-Reitoria de Pesquisa, Universidade Regional de Blumenau, Blumenau.

[artigo em periódico em meio eletrônico:]

VIEIRA, Cassio L.; LOPES, Marcelo. A queda do cometa. **Neo Interativa**, Rio de Janeiro, n. 2, inverno 1994. 1 CD-ROM.

WINDOWS 98: o melhor caminho para atualização. **PC World**, São Paulo, n. 75, set. 1998. Disponível em: http://www.idg.com.br/abre.html. Acesso em: 10 set. 1998.

APÊNDICE A – Dicionário de Dados

Este Apêndice apresenta o dicionário de dados com a descrição das tabelas e atributos da ferramenta BeDevOps, apresentados na subseção 3.2.3. Os tipos de dados utilizados foram int e double para o armazenamento de números, varchar para o armazenamento de textos, e date para armazenar variáveis de data. O Quadro 9 apresenta o dicionário de dados da tabela cachaca\_barrica.

APÊNDICE B – Dicionário de Dados

Este Apêndice apresenta a descrição.....

ANEXO A – Exemplo

[Elemento opcional. Anexos são documentos não elaborados pelo autor, que servem de fundamentação, comprovação ou ilustração, como mapas, leis, estatutos, entre outros. Os anexos são identificados por letras maiúsculas consecutivas, travessão e pelos respectivos títulos. Sempre referenciá-las antes.]

1. Disponível em: http://getbootstrap.com. [↑](#footnote-ref-1)
2. Disponível em: https://puppet.com. [↑](#footnote-ref-2)
3. Disponível em: https://www.vagrantup.com/. [↑](#footnote-ref-3)
4. Disponível em: https://www.jenkins.io/. [↑](#footnote-ref-4)