|  |
| --- |
| 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

ABSTRACT

LISTA DE Figuras

[Figura 1- Puppet Enterprise Status Dashboard 19](#_Toc56952385)

[Figura 2- Utilização do Puppet pelo software Ovenbird 22](#_Toc56952386)

[Figura 3- Tela de pipeline de implantação 23](#_Toc56952387)

[Figura 4 - Interface gráfica e exemplos de Jobs da ferramenta Jenkins 24](#_Toc56952388)

[Figura 5 – Diagrama de Caso de Uso 27](#_Toc56952389)

[Figura 6 – Modelo Entidade Relacionamento (bedevops) 29](#_Toc56952390)

[Figura 7 – Modelo Entidade Relacionamento (permission) 30](#_Toc56952391)

[Figura 8 – Modelo Entidade Relacionamento (communication) 30](#_Toc56952392)

[Figura 9 – Modelo Entidade Relacionamento (log) 31](#_Toc56952393)

[Figura 10 - Diagrama de atividades 32](#_Toc56952394)

[Figura 11 – Diagrama de componentes 34](#_Toc56952395)

[Figura 12 – Diagrama de tecnologias 35](#_Toc56952396)

[Figura 13 – Configuração da EC2 – Etapa 1 - Amazon Machine Image (AMI) 36](#_Toc56952397)

[Figura 14 - Configuração da EC2 – Etapa 1 - Tipo da instância 37](#_Toc56952398)

[Figura 15 – Configuração da EC2 – Etapa 3 - Configuração detalhada da instância 38](#_Toc56952399)

[Figura 16 – Configuração da EC2 – Etapa 4 - Configuração de *storage* 38](#_Toc56952400)

[Figura 17 – Verificação de pré-requisitos 39](#_Toc56952401)

[Figura 18 - Painel de configuração do domínio 39](#_Toc56952402)

[Figura 19 – Configuração de *Host* 39](#_Toc56952403)

[Figura 20 – Configuração de *Namespaces* 40](#_Toc56952404)

[Figura 21 – Console AWS – Identificação de *Host* 40](#_Toc56952405)

[Figura 22 – Configuração do Route 53 40](#_Toc56952406)

[Figura 23 – Configuração da instância RDS 42](#_Toc56952407)

[Figura 24 – Pagina inicial do Adianti Builder 43](#_Toc56952408)

[Figura 25 – Planos do Adinti Builder 43](#_Toc56952409)

[Figura 26 – Criar projeto no Adianti Builder 44](#_Toc56952410)

[Figura 27 – Fórum da comunidade do Adianti Builder 44](#_Toc56952411)

[Figura 28 – Página de suporte do Adianti Builder 45](#_Toc56952412)

[Figura 29 – Modelagem da base de dados no Adianti Builder 45](#_Toc56952413)

[Figura 30 – Modelos provisionados pelo Adianti Builder 46](#_Toc56952414)

[Figura 31 – Criar telas CRUD com base no modelo de dados 46](#_Toc56952415)

[Figura 32 - Geração automática das CRUDS básicas da aplicação 47](#_Toc56952416)

[Figura 33 – Edição em modo designer 47](#_Toc56952417)

[Figura 34 – Edição em modo código 48](#_Toc56952418)

LISTA DE Quadros

[Quadro 1- Comparativo de resultados em caso de falhas no ambiente produtivo 19](#_Toc56891273)

[Quadro 2 - Requisitos Funcionais 24](#_Toc56891274)

[Quadro 3 - Requisitos Não Funcionais 24](#_Toc56891275)

[Quadro 4 - Matriz de rastreabilidade dos RFs e sua relação com os UC 27](#_Toc56891276)

Lista de tabelas

**Nenhuma entrada de índice de ilustrações foi encontrada.**

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

AWS – Amazon Web Services

SO – Sistema Operacional

AMI – Amazon Machine Image

EC2 – Elastic Compute Cloud

RDS - Relational Database Service

SUMÁRIO

[1 Introdução 14](#_Toc56891411)

[1.1 OBJETIVOS 15](#_Toc56891412)

[1.2 estrutura 15](#_Toc56891413)

[2 FUNDAMENTAÇÃO TEÓRICA 16](#_Toc56891414)

[2.1 DEVOPS 16](#_Toc56891415)

[2.2 FERRAMENTAS DE AUTOMAÇÃO 17](#_Toc56891416)

[2.3 CONTINUOS DELIVERY 17](#_Toc56891417)

[2.4 TRABALHOS CORRELATOS 17](#_Toc56891418)

[2.4.1 ESTUDO DE CASO DE UTILIZAÇÃO DA METODOLOGIA DEVOPS PARA ATENDER AO PROCESSO DE CONTINUIDADE DE SERVIÇOS CONFORME O FRAMEWORK ITIL 17](#_Toc56891419)

[2.4.2 AUTOMATIZAÇÃO DE ENTREGA DE SOFTWARE EM AMBIENTE ÁGIL DE DESENVOLVIMENTO 20](#_Toc56891420)

[2.4.3 UM PROCESSO DE INTRODUÇÃO DE DEVOPS EM SISTEMAS LEGADOS 22](#_Toc56891421)

[3 DESENVOLVIMENTO DA FERRAMENTA 24](#_Toc56891422)

[3.1 LEVANTAMENTO DE REQUISITOS 24](#_Toc56891423)

[3.2 ESPECIFICAÇÃO 25](#_Toc56891424)

[3.2.1 Diagrama de Caso de Uso 25](#_Toc56891425)

[3.2.2 Matriz de rastreabilidade dos RFs e sua relação com os Casos de Uso 26](#_Toc56891426)

[3.2.3 Modelo Entidade Relacionamento 27](#_Toc56891427)

[3.2.4 Diagrama de atividades 30](#_Toc56891428)

[3.2.5 Diagrama de componentes 32](#_Toc56891429)

[3.2.6 Diagrama de tecnologias 33](#_Toc56891430)

[3.3 IMPLEMENTAÇÃO 34](#_Toc56891431)

[3.3.1 Técnicas e ferramentas utilizadas 34](#_Toc56891432)

[3.3.2 Operacionalidade da implementação 36](#_Toc56891433)

[3.4 RESULTADOS E DISCUSSões 37](#_Toc56891434)

[3.4.1 Comparação entre os trabalhos correlatos e a ferramenta desenvolvida 37](#_Toc56891435)

[3.4.2 Avaliação de usabilidade e comunicabilidade pelo método RURUCAg 37](#_Toc56891436)

[4 CONCLUSÕES 38](#_Toc56891437)

[4.1 EXTENSÕES 38](#_Toc56891438)

[Referências 39](#_Toc56891439)

[APÊNDICE A – Dicionário de Dados 43](#_Toc56891440)

[ANEXO A – Exemplo 44](#_Toc56891441)

# 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 AUTOMAÇÃO

## CONTINUOS DELIVERY

## TRABALHOS CORRELATOS

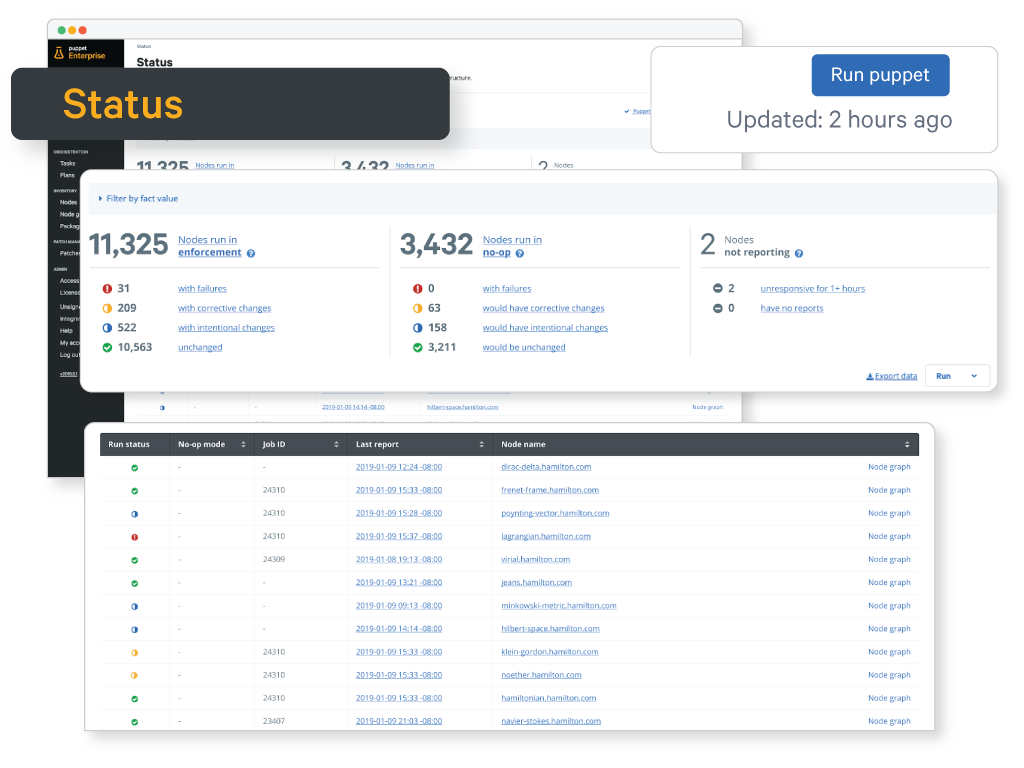
Nesta seção são apresentados trabalhos que apresentam semelhança com os principais objetivos do trabalho proposto. A subseção 2.1 aborda a implantação do DevOps em conformidade com a Biblioteca de Infraestrutura de Tecnologia da Informação, traduzida do inglês Information Technology Infrastructure Library (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 - 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

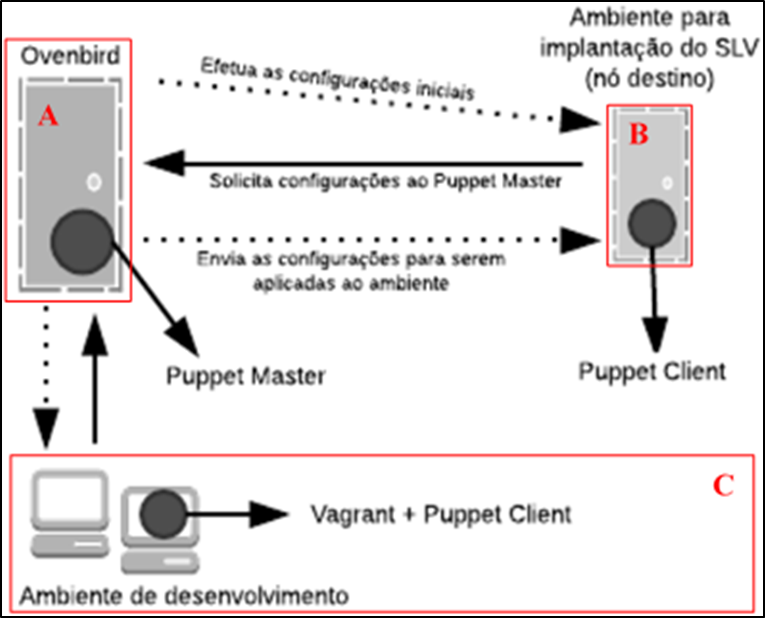
Nunes (2014) visou a construção de um software que pudesse integrar ferramentas, tecnologias e boas práticas para ajudar a realizar padronizações de automatização no desenvolvimento de sistemas web. Em seu desenvolvimento foram utilizadas como ferramentas web, a biblioteca Bootstrap[[1]](#footnote-1), 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).

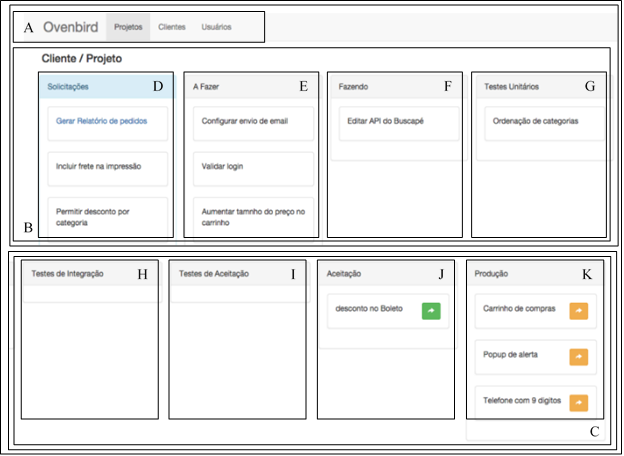
Figura - Utilização do Puppet pelo software Ovenbird



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.

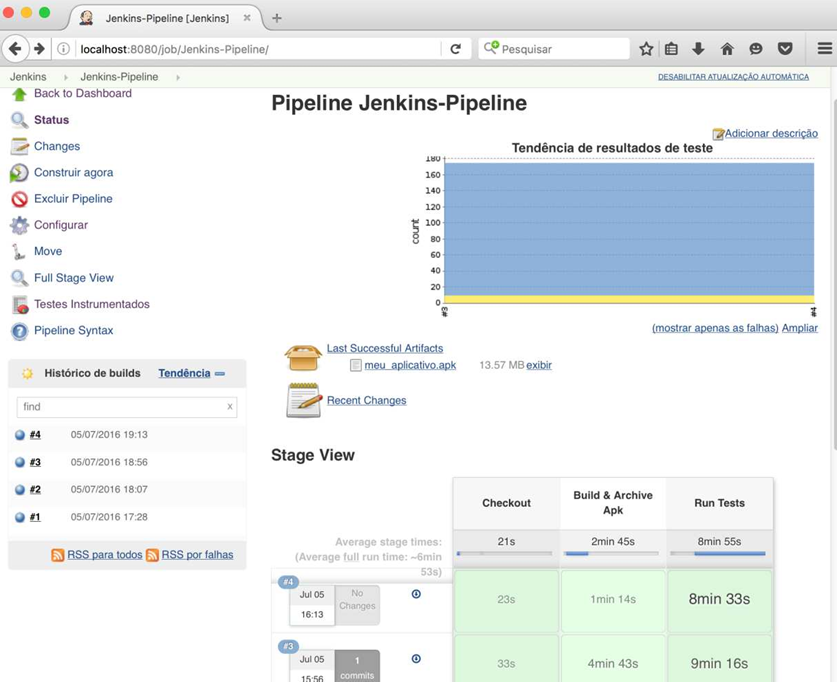
Figura - Tela de pipeline de implantação

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 - 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 - 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 - 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.

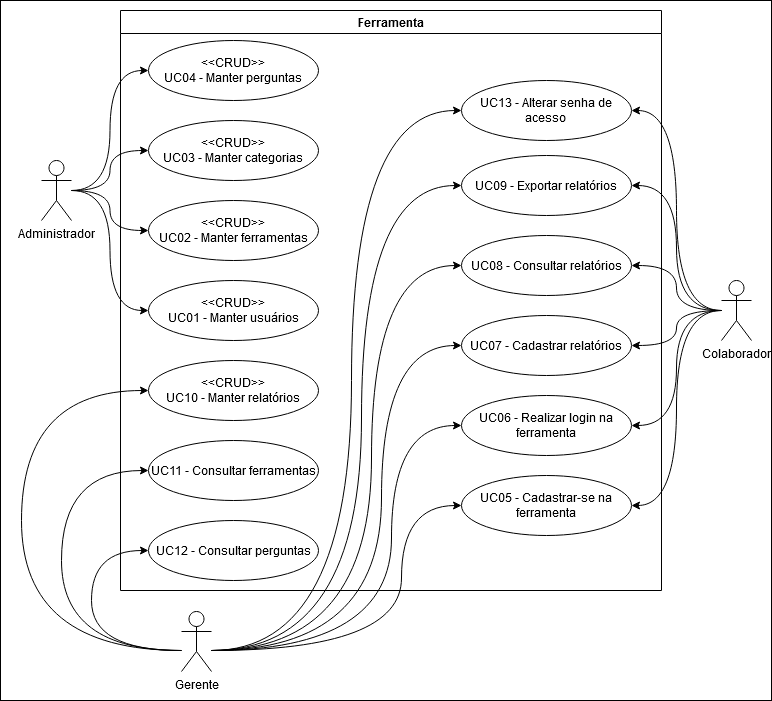
Os usuários (Gerente) e (Colaborador) podem realizar a ação de cadastrar-se, demonstrado por meio do UC05 – Cadastrar-se na ferramenta. Após o cadastro, eles podem realizar o log in, explicado por meio do UC06 – Realizar log in para ter acesso às demais funcionalidades da ferramenta. Um usuário (Gerente) 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 – 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 - Matriz de rastreabilidade dos RFs e sua relação com os UC

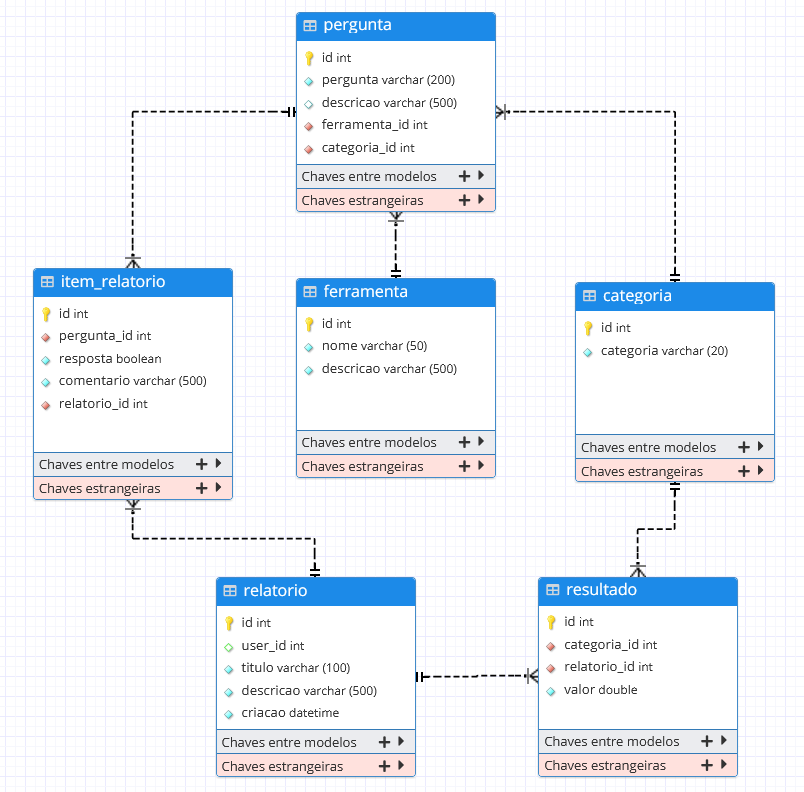
|  |  |
| --- | --- |
| **Requisitos Funcionais** | **UC** |
| RF01: manter o cadastro de usuários (Create, Read, Update and Delete - CRUD). | UC01 |
| RF02: manter o cadastro de perguntas do questionário (CRUD). | UC04 |
| RF03: manter o cadastro de categorias (CRUD). | UC03 |
| RF04: manter o cadastro 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.

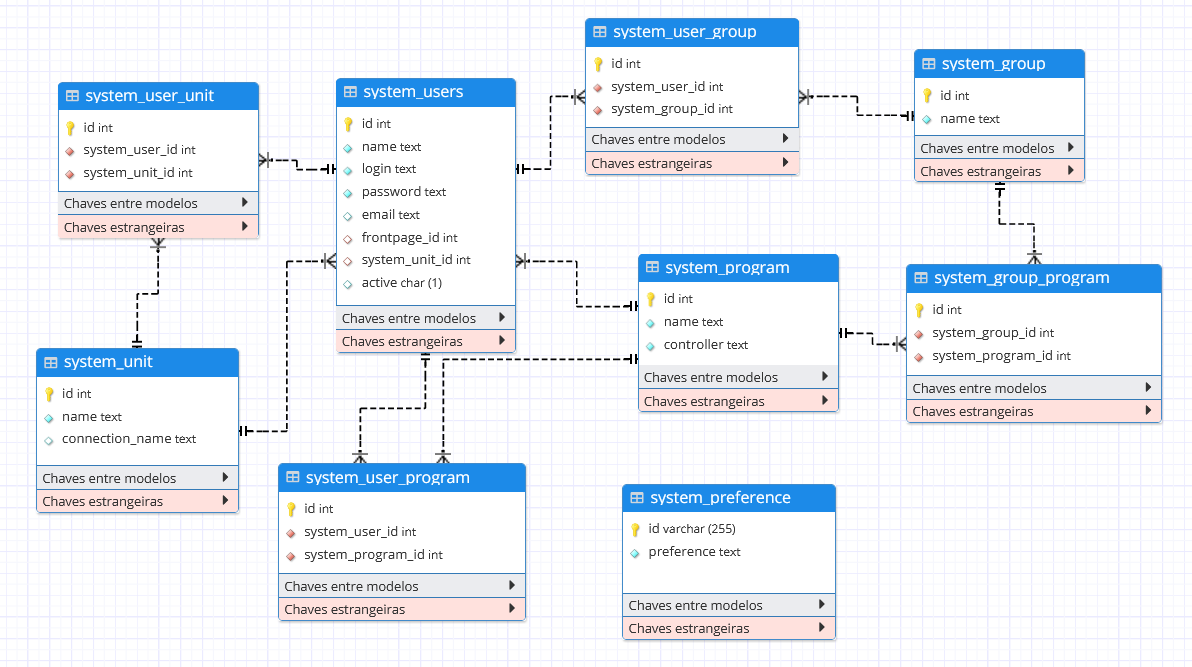
Figura – Modelo Entidade Relacionamento (bedevops)



Fonte: elaborado pelo autor.

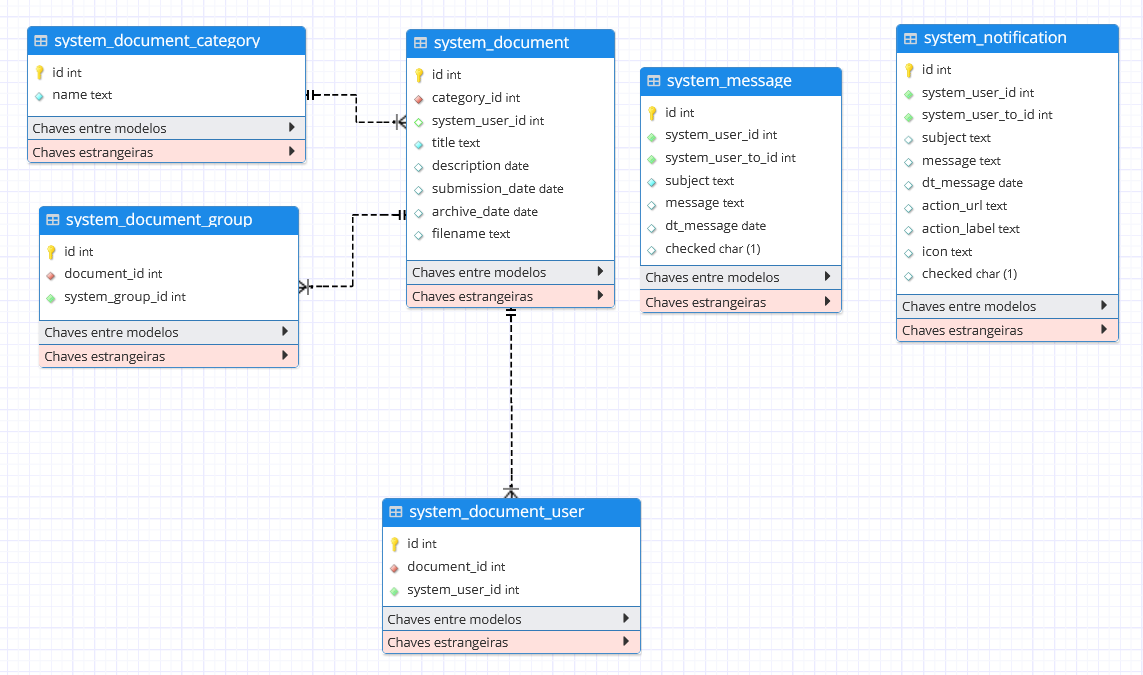
A ferramenta utilizada para realizar o desenvolvimento do trabalho pré-dispõem de modelos de estrutura do banco de dados para o funcionamento base da ferramenta, sendo elas de: permission (Figura 7); communication (Figura 8); e log (Figura 9). Já no Apêndice A pode ser encontrado o dicionário de dados 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 – Modelo Entidade Relacionamento (permission)



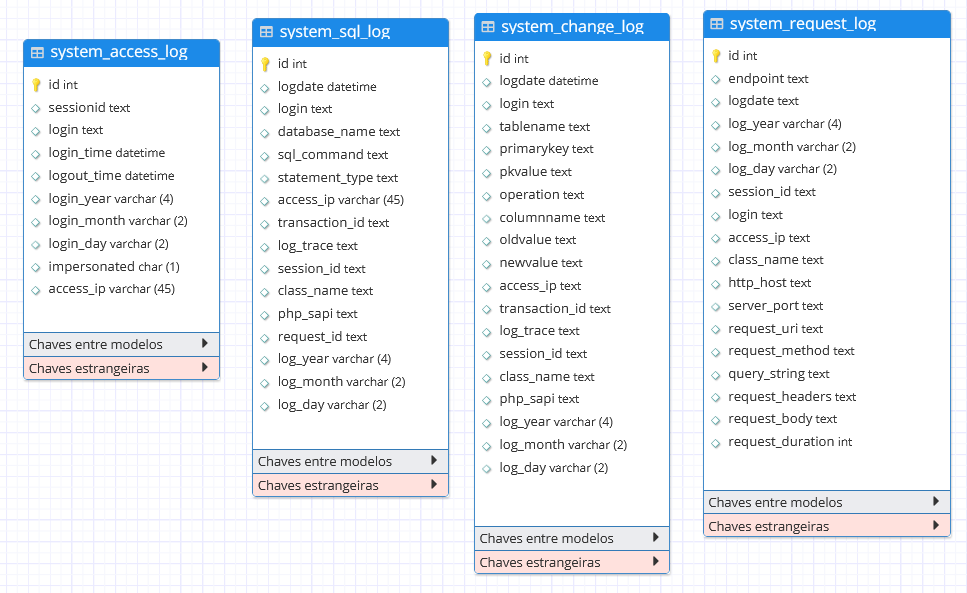
Fonte: AdiantiBuilder (2020).

Figura – Modelo Entidade Relacionamento (communication)



Fonte: AdiantiBuilder (2020).

Figura – Modelo Entidade Relacionamento (log)

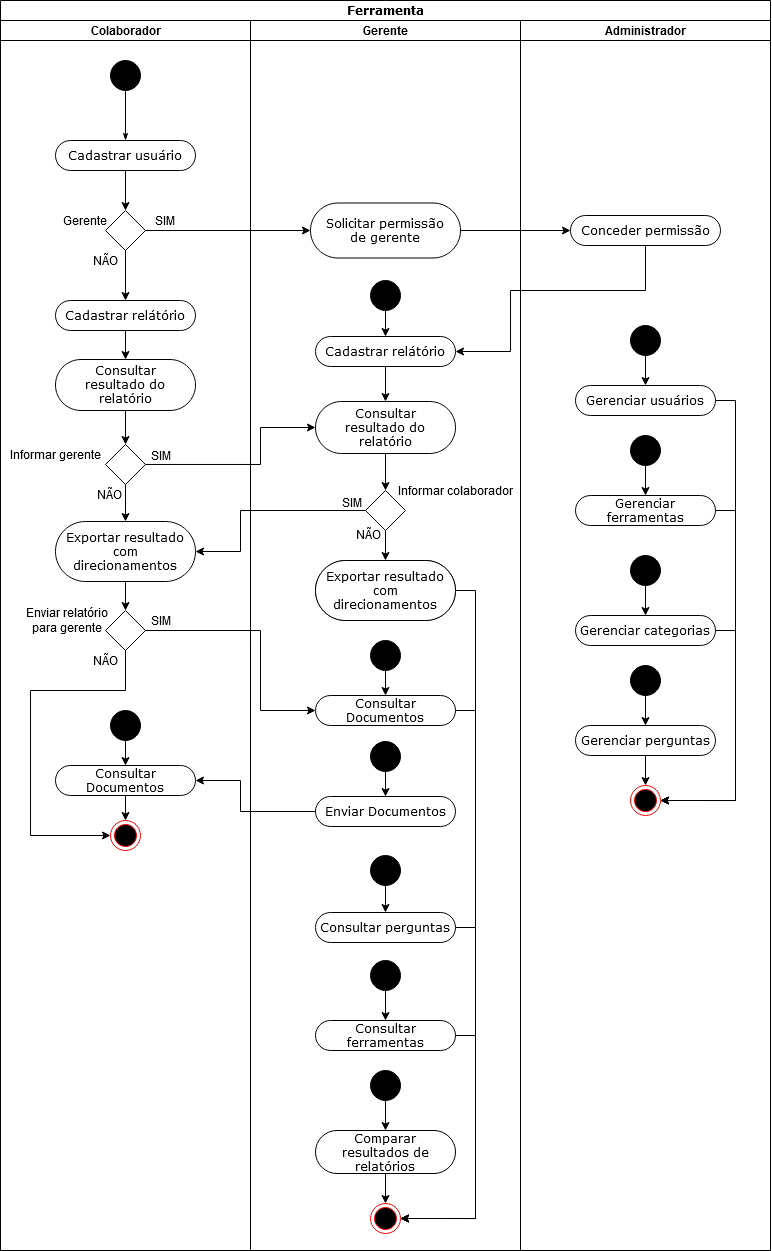


Fonte: AdiantiBuilder(2020).

### Diagrama de atividades

O detalhamento da ferramenta desenvolvida pode ser melhor compreendido pelo diagrama de atividades (Figura 10). Na raia referente ao Colaborador, é possível notar as suas funcionalidades atribuídas, sendo elas: Cadastrar usuário, cadastrar relatório, consultar resultado do relatório, informar gerente, exportar resultado com direcionamentos, e então, enviar resultado para gerente. Além destas ações, o Colaborador também pode consultar documentos enviados a ele pelo gerente. Uma outra ação possível pelo colaborador é a de consultar documentos. Na raia central é possível ver as ações possíveis pelo Gerente, que, assim como o Colaborador, também pode: Cadastrar relatório, consultar resultado do relatório e exportar resultado com direcionamentos. Além destas atribuições o Gerente pode solicitar permissão de gerente, consultar documentos, enviar documentos, consultar perguntas, consultar ferramentas e comparar resultados de relatórios. Por fim, na raia referente ao Administrador, são apresentadas suas funcionalidades, sendo elas: conceder permissão, gerenciar usuários, gerenciar ferramentas, gerenciar categorias e gerenciar perguntas.

Figura - Diagrama de atividades



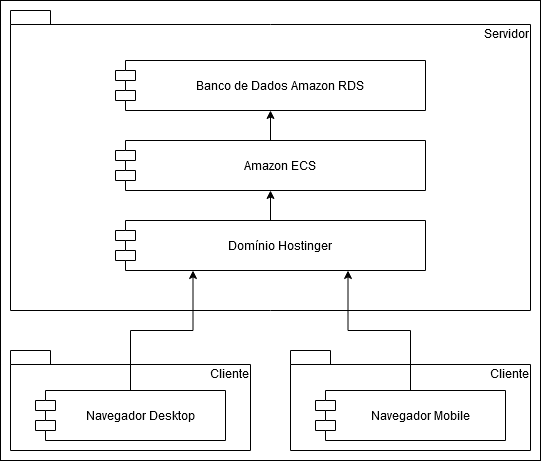
Fonte: elaborada pelo autor.

### Diagrama de componentes

No desenvolvimento da ferramenta BeDevOps foi utilizado a arquitetura cliente-servidor (Figura 11). O cliente acessa a ferramenta por meio do seu navegador, representados por Navegador Desktop e Navegador Mobile, que será o responsável por interpretar os códigos HTML, CSS e JavaScript. Estes códigos são provisionados por meio de uma instância Apache web server, que então interpreta os códigos PHP. Na arquitetura do servidor, utilizou-se o serviço Amazon Elastic Compute Cloud (EC2). O EC2 que é um serviço Web que fornece capacidade segura e redimensionável de computação em nuvem (AMAZON, 2020a).

O serviço Amazon Relational Database Service (RDS) foi utilizado para a hospedagem de dados, onde foi criada uma instância de MySQL. Segundo Amazon (2020) o serviço RDS facilita a configuração, operação e escalabilidade de bancos de dados relacionais na nuvem, sendo um serviço que oferece capacidade econômica e escalável. O serviço RDS automatiza tarefas custosas de gerenciamento, como configuração de hardware, configuração de banco de dados, *patching* e backup. Para tornar da ferramenta acessível globalmente, foi registrado um domínio na provedora de hospedagem na web chamada Hostinger, local onde foi configurada toda a conexão para os serviços disponibilizados na Amazon. O funcionamento dos serviços da Amazon é abordado com mais profundidade na subseção 3.2.7.

Figura – Diagrama de componentes



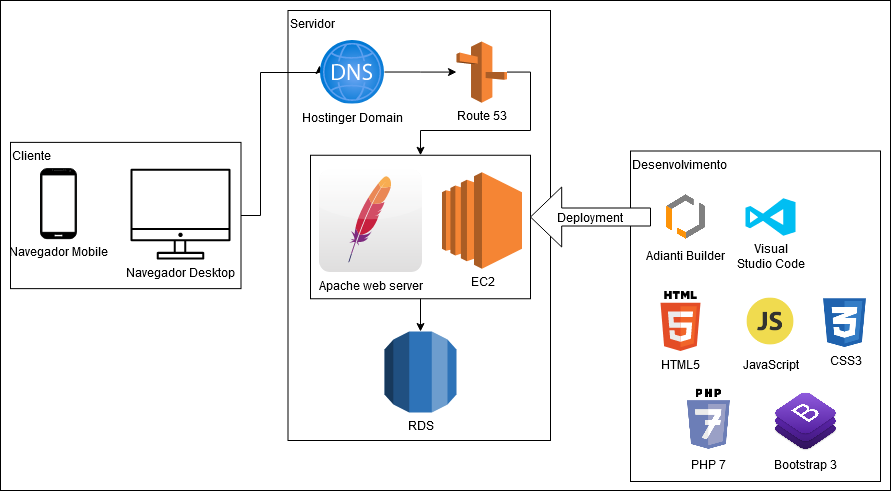
Fonte: elaborada pelo autor.

### Diagrama de tecnologias

Na Figura 12 é apresentada a relação das tecnologias utilizadas em cada etapa do desenvolvimento da ferramenta. O cliente é representado por meio do navegador desktop e navegador mobile, indicando que a ferramenta pode ser acessada tanto por dispositivos celulares ou tablets, quanto por computadores. No servidor, utilizou-se o serviço de registro de domínios da Hostinger. A nível computacional foram utilizados os serviços da Amazon Web Services (AWS) que garantem a disponibilidade, integridade e resiliência da ferramenta, bem como promovem o desenvolvimento por meio da abstração fornecida pela AWS.

O serviço da Hostinger domain tem como objetivo disponibilizar a ferramenta a nível global, provendo o domínio da ferramenta (bedevops.space); o Route 53 que, conforme Amazon (2020e), é um serviço de Domain Name System (DNS), com o principal objetivo de direcionar usuários finais aos sistemas móveis de Internet. Sua função é gerenciar e disponibilizar o trafego entre a Hostinger e os serviços AWS; o Apache web server sendo executado sobre a instancia EC2 da Amazon tem como objetivo provisionar para o cliente por meio do domínio os arquivos estáticos (HTML, CSS, JavaScript) e interpretar os códigos PHP; para a persistência de dados foi utilizado o serviço RDS com a execução de uma instância MySQL; as ferramentas de desenvolvimento utilizadas foram o Adianti Buider, que é abordado com mais profundidade na seção 3.3.1, e para realização de *debug* de problemas foi utilizada como Interface de Desenvolvimento (Integrated Development Environment – IDE) o Visual Studio Code; a interface gráfica foi implementada utilizando os componentes do framework Bootstrap 3.

Figura – Diagrama de tecnologias



Fonte: elaborada pelo autor.

## IMPLEMENTAÇÃO

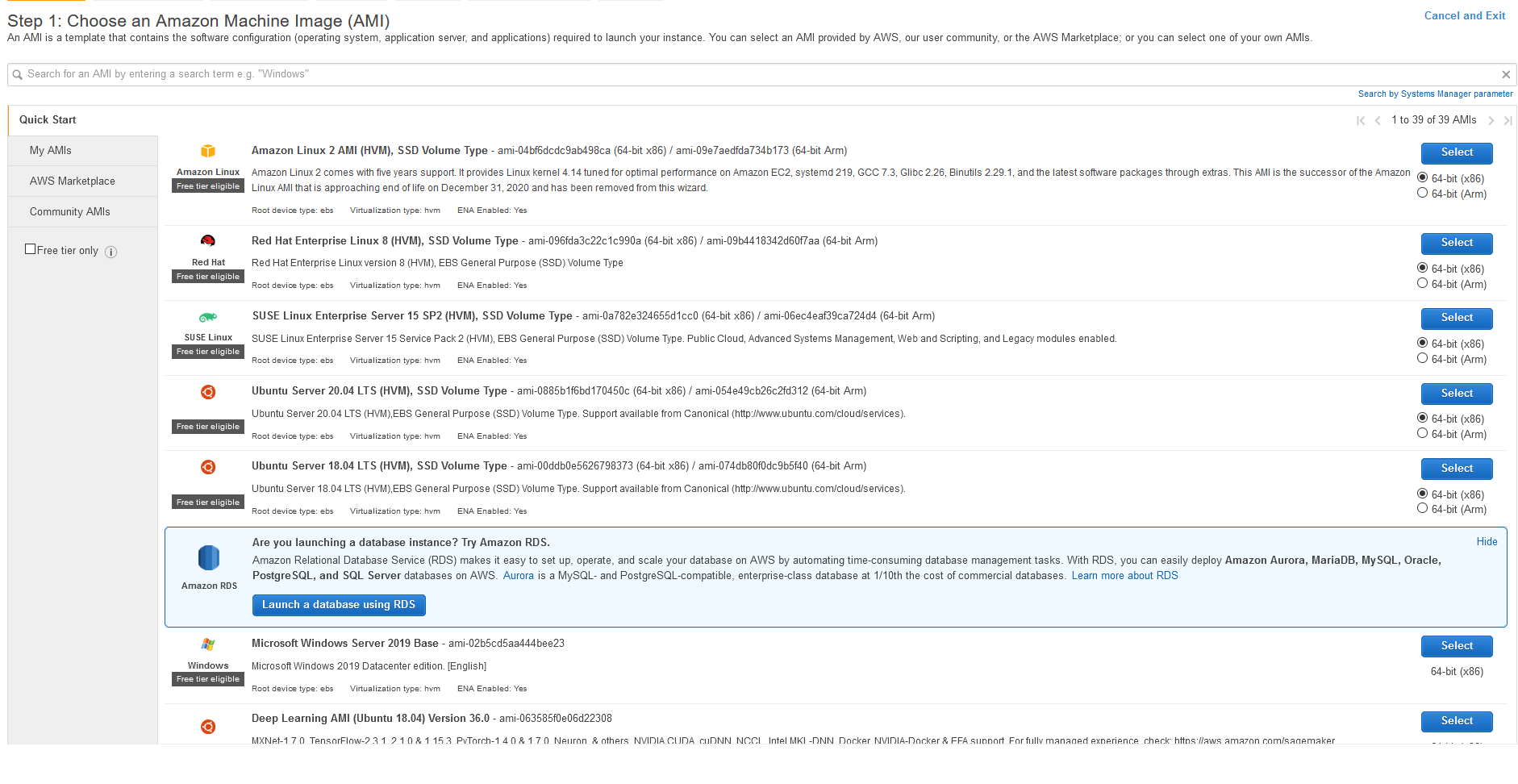
Nesta seção, são descritas as técnicas e as ferramentas utilizadas para o desenvolvimento da ferramenta BeDevOps por meio de figuras do código desenvolvido, bem como de configurações realizadas nos serviços da AWS, estando estruturado em três subseções. A subseção 3.3.1 traz as técnicas e ferramentas utilizadas; a subseção 3.3.2 mostra a codificação da implementação e a subseção 3.3.3 apresenta a operacionalidade da implementação.

### Técnicas e ferramentas utilizadas

Para desenvolver o sistema web responsivo BeDevOps foi utilizado a arquitetura cliente-servidor. A interface gráfica foi implementada utilizando os componentes do framework Bootstrap 3, para garantir a responsividade das telas do visualizados a partir de dispositivos móveis como celulares e tablets. Todos os recursos computacionais foram provisionados por meio de serviços AWS, para a qual foi necessário registrar-se.

A configuração dos serviços AWS pode ser realizada no Console AWS, sistema responsável pela interação com usuário para configuração de seus serviços. No console da AWS, basta selecionar o serviço EC3 e criar uma nova instância, e então selecionar uma Imagem de Máquina (Amazon Machine Image – AMI) (Figura 13). Após isso, será necessário informar as configurações da instância, como: tipo da instância; configuração detalhada da instância; e configuração de *storage*. Conforme é possível observar nas Figura 14, Figura 15 e Figura 16.

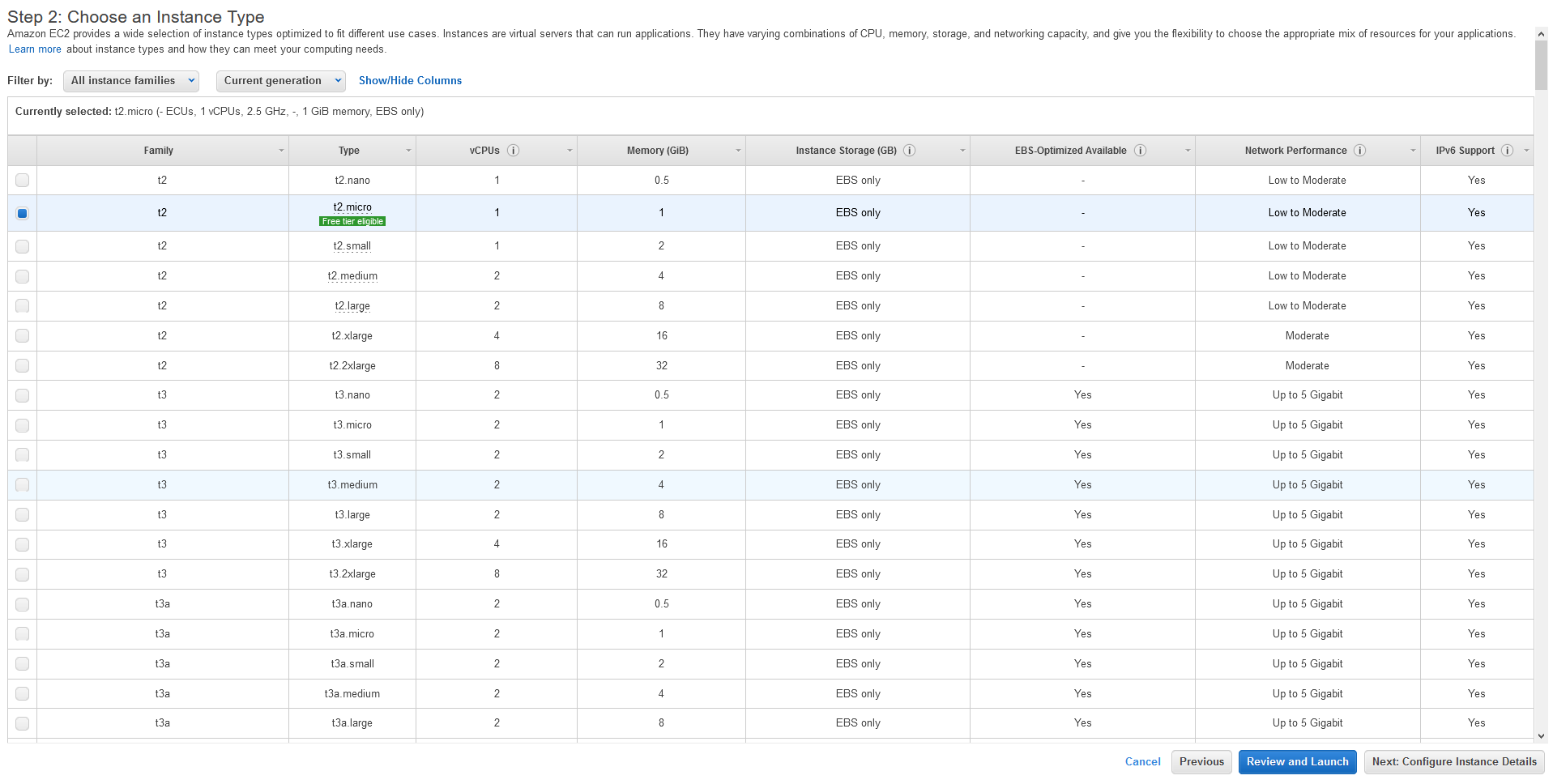
Figura – Configuração da EC2 – Etapa 1 - Amazon Machine Image (AMI)



Fonte: Amazon (2020).

A Figura 13 apresenta a etapa 1 da configuração. No caso da ferramenta BeDevOps, utilizou-se a AMI Ubuntu 20.04 LTS. Na segunda etapa de configuração da instância (Figura 14), é possível selecionar qual tipo de instância se deseja utilizar. A ferramenta desenvolvida neste trabalho não necessita muitos recursos computacionais, motivo pelo qual utilizou-se o tipo t2.micro classificado no nível gratuito (*Free tier)*. A Amazon (2020) fornece um período de uso gratuito, que pode ser utilizado para qualquer coisa que você queira executar na nuvem ou simplesmente para obter experiência prática com a AWS.

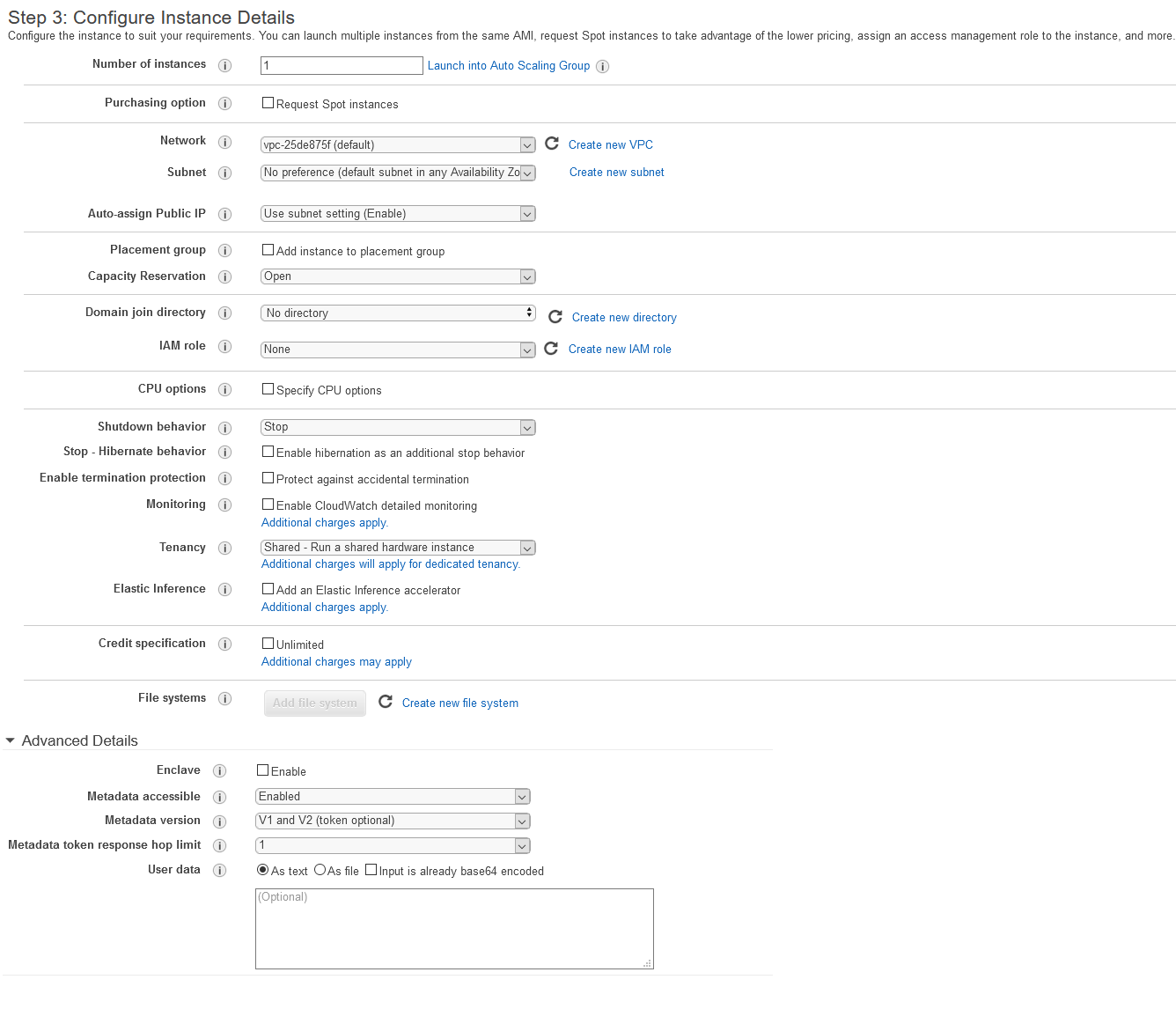
Figura - Configuração da EC2 – Etapa 1 - Tipo da instância



Fonte: Amazon (2020).

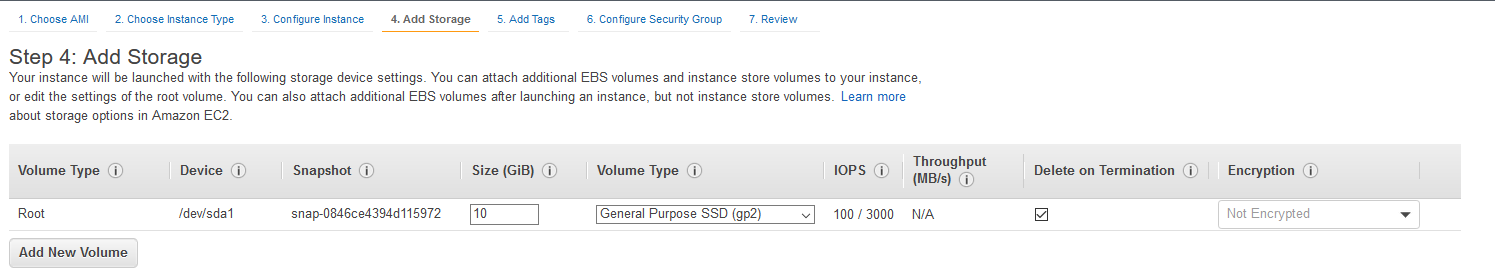
Na Figura 15 é demonstrada a terceira etapa de configuração da instância EC2, na qual aplicou-se as configurações padrões fornecidas pelo serviço. Cabe destacar que foi utilizada uma configuração mínima visando o baixo custos operacional, podendo, de acordo com a demanda da ferramenta ser configurado com maiores recursos computacionais. Na Figura 16 é demonstrado a etapa 4 de configuração da instância com a alocação de espaço em disco. Para a ferramenta BeDevOps foram alocados 10GB à instância.

Figura – Configuração da EC2 – Etapa 3 - Configuração detalhada da instância



Fonte: Amazon (2020).

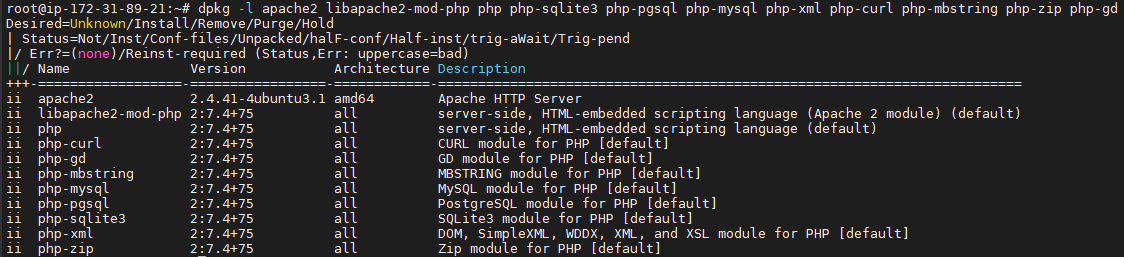
Figura – Configuração da EC2 – Etapa 4 - Configuração de *storage*



Fonte: Amazon (2020).

Após a criação da instância EC2 é necessária a preparação do ambiente para a execução da ferramenta BeDevOps. Os principais pré-requisitos são o Apache e o PHP que podem facilmente serem instalados no Ubuntu 20.04 pelo comando “apt-get install apache2 php”. Além dos pacotes principais, são necessárias instalar as seguintes bibliotecas: libapache2-mod-php, php-sqlite3, php-pgsql, php-mysql, php-xml, php-curl, php-mbstring, php-zip e php-gd. A instalação destas bibliotecas no Ubuntu 20.04 pode ser feita pelo comando “apt-get install libapache2-mod-php php-sqlite3 php-pgsql php-mysql php-xml php-curl php-mbstring php-zip php-gd”. A Figura 17 demonstra como é possível verificar se os pacotes encontram-se instalados, ou, caso tenha realizado a instalação, se os mesmos foram instalados.

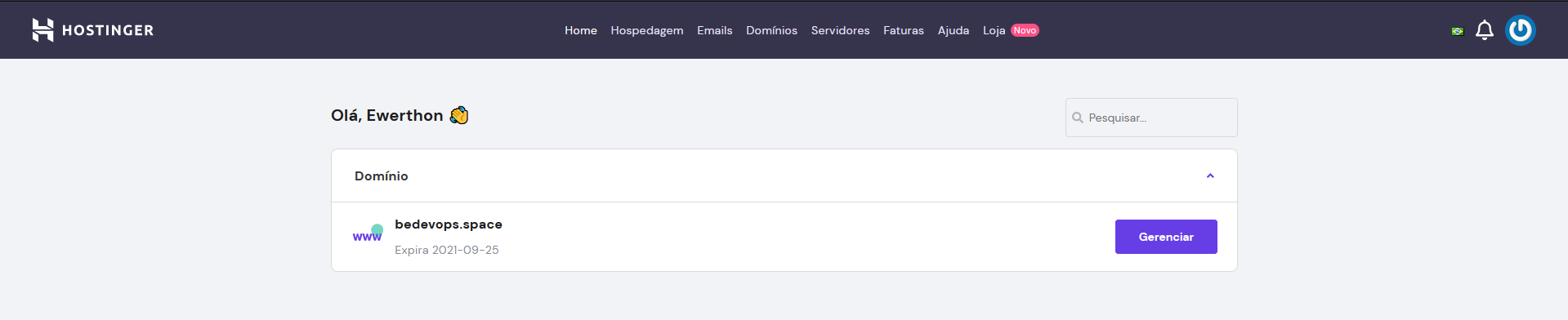
Figura – Verificação de pré-requisitos



Fonte: elaborada pelo autor.

Para disponibilizar o acesso a ferramenta foi hospedado um domínio na Hostinger (bedevops.space) para o qual foi necessário registrar-se. A configuração do domínio é realizada pelo painel de configuração da Hostinger, conforme apresentado pela Figura 18. A Figura 19 e a Figura 20 demonstram quais configurações de Zona DNS foram necessárias para tornar o domínio funcional.

Figura - Painel de configuração do domínio



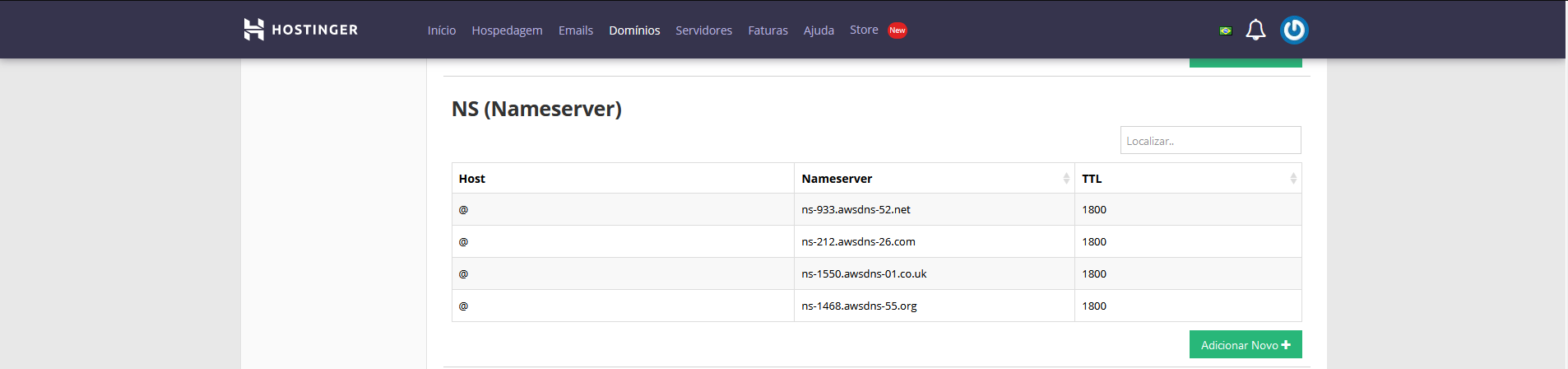
Fonte:Hostinger (2020).

Figura – Configuração de *Host*



Fonte:Hostinger (2020).

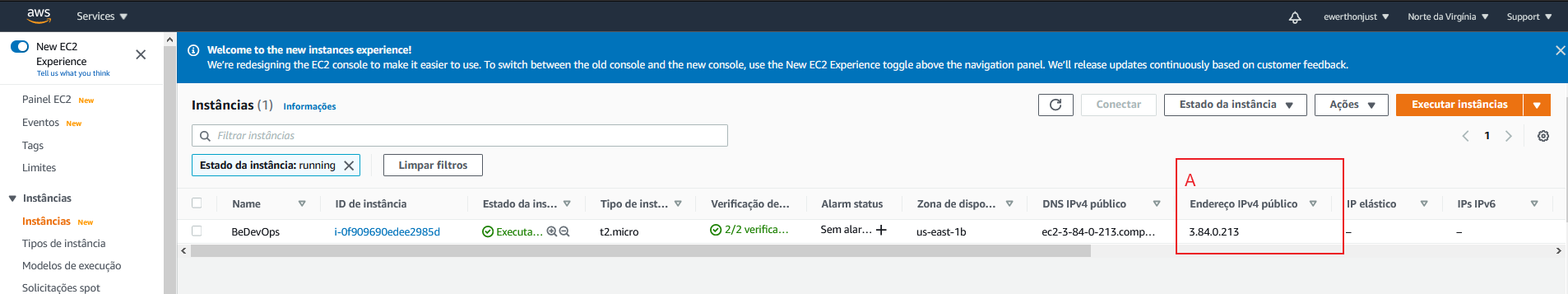
Figura – Configuração de *Namespaces*



Fonte:Hostinger (2020).

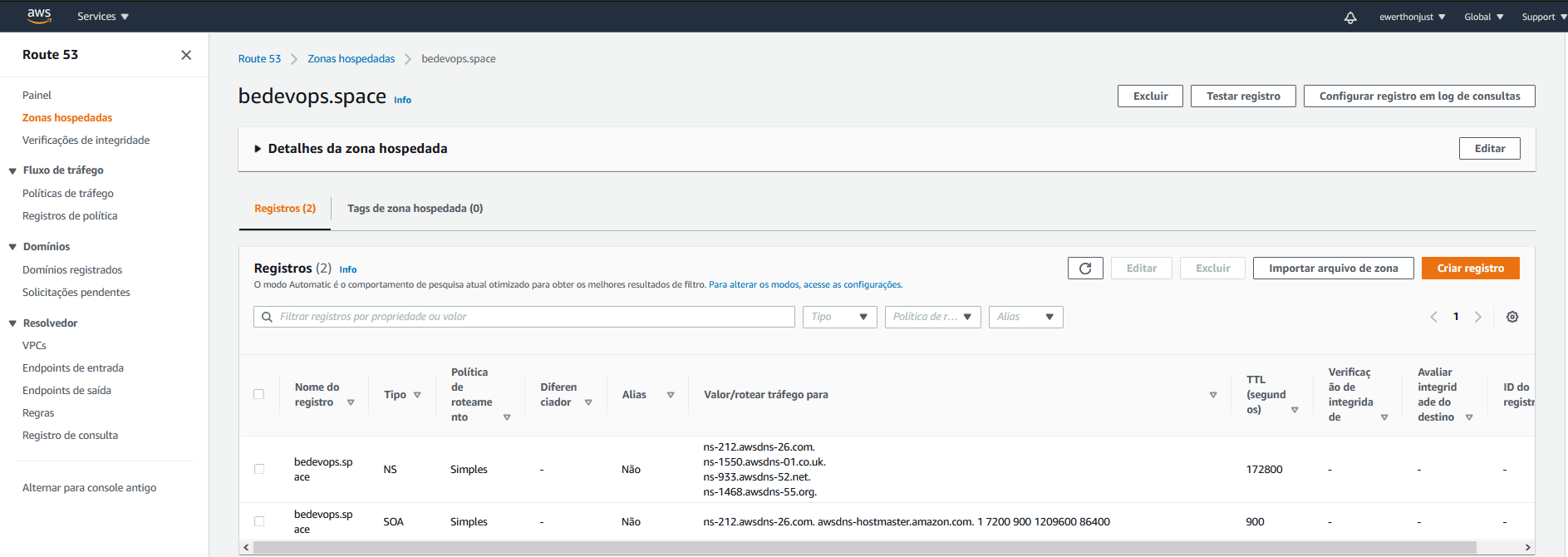
A informação necessária para configurar o *Host*, apresentada na Figura 19, pode ser identificada no console da AWS, conforme demonstrado no retângulo A da Figura 21. O provisionamento da conexão entre a Hostinger e os serviços AWS é feita pelo serviço Route 53 registrando uma zona hospedada (Figura 22) onde são possíveis de serem identificar os *Namespaces* a serem informados na Figura 20.

Figura – Console AWS – Identificação de *Host*



Fonte: adaptado de Amazon (2020).

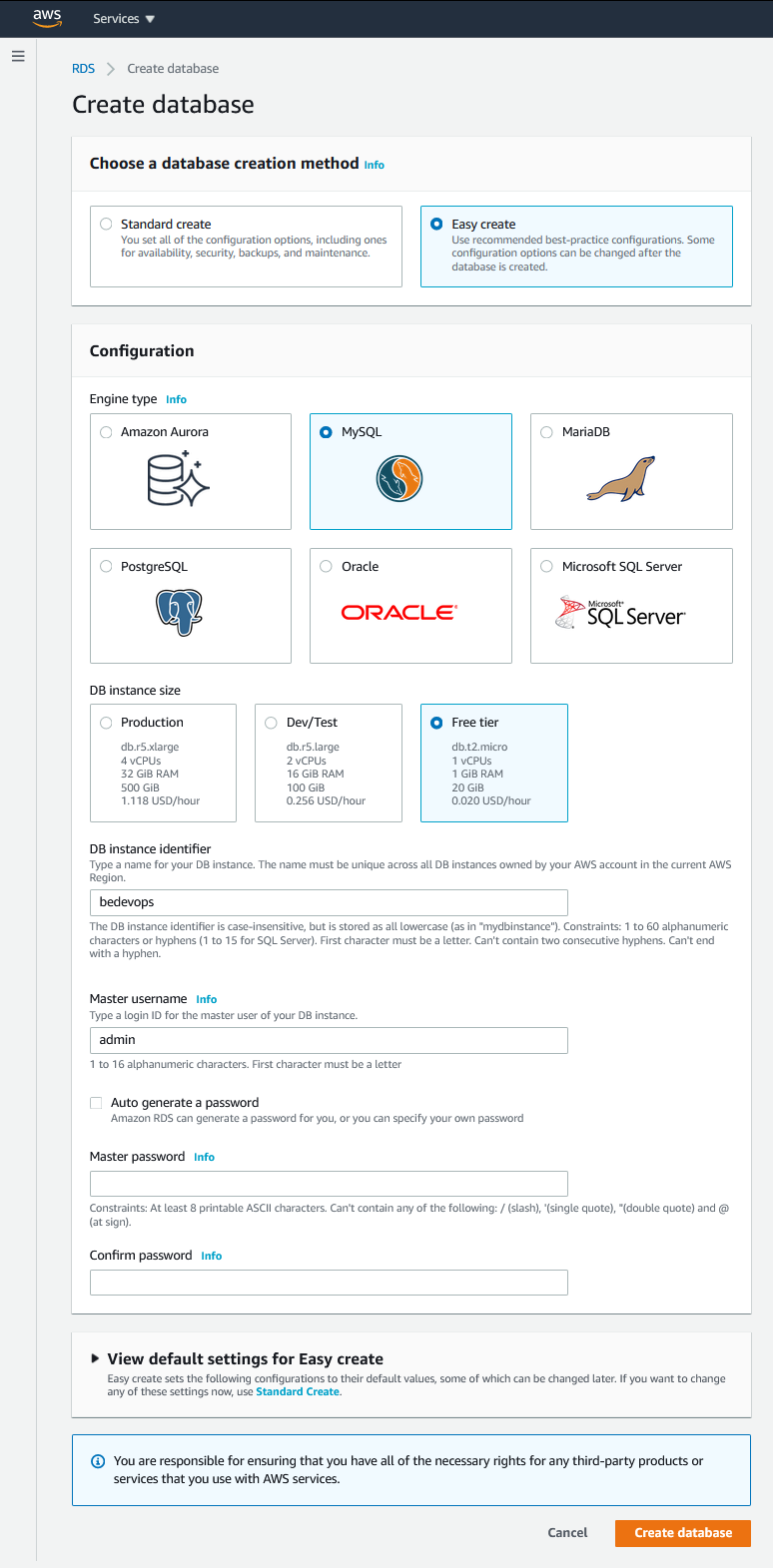
Figura – Configuração do Route 53



Fonte: Amazon (2020).

O banco de dados escolhido para o projeto foi o MySQL provisionado pelo serviço Amazon RDS e o dicionário de dados se encontrado no Apêndice A. Na Figura 23 é demonstrado a configuração da instância RDS utilizando o modelo relacional MySQL. O modelo relacional é uma estrutura de dados onde a parte lógica: tabelas de dados, exibições e índices são armazenadas separadamente da estrutura física (Oracle, 2020). Com essa separação os *Data Base Administrators* (DBA) podem gerenciar o armazenamento de dados físicos sem afetar o acesso a esses dados como uma estrutura lógica.

Figura – Configuração da instância RDS



Fonte: Amazon (2020).

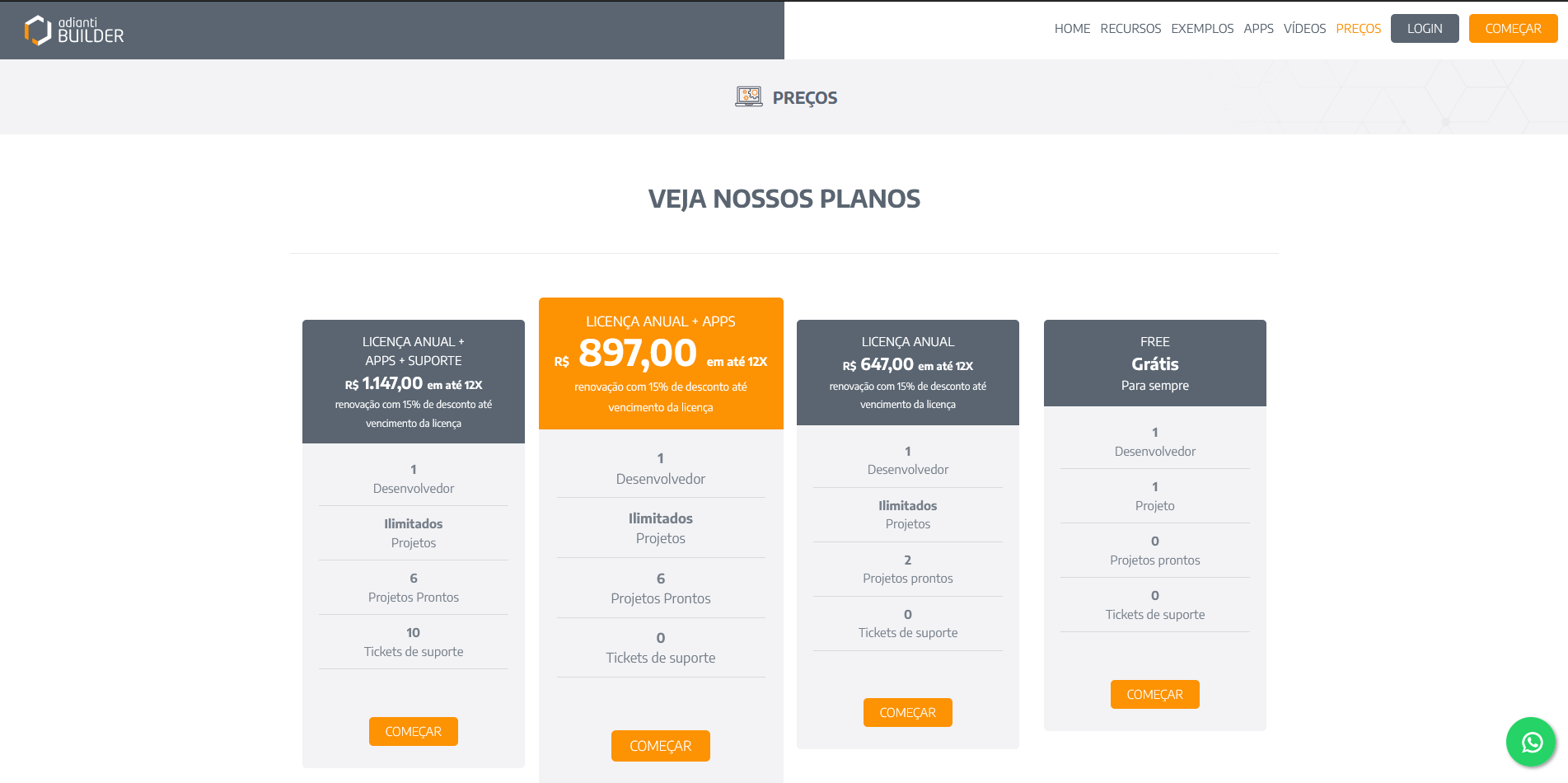
Conforme mencionado na subseção 3.2.6, a ferramenta Adianti Buider foi utilizada no desenvolvimento da ferramenta BeDevOps, que é segundo Adiantibuilder (2020) “uma plataforma Low-Code de criação de sistemas online em PHP. Permite desde a modelagem da aplicação até a criação de suas telas de forma visual utilizando componentes de alto nível.”. A ferramenta é web e para ter acesso e necessário registrar-se (Figura 24), além de que para obter acesso completo as funcionalidades, é necessário comprar uma licença (Figura 25).

Figura – Pagina inicial do Adianti Builder



Fonte: AdiantiBuilder (2020).

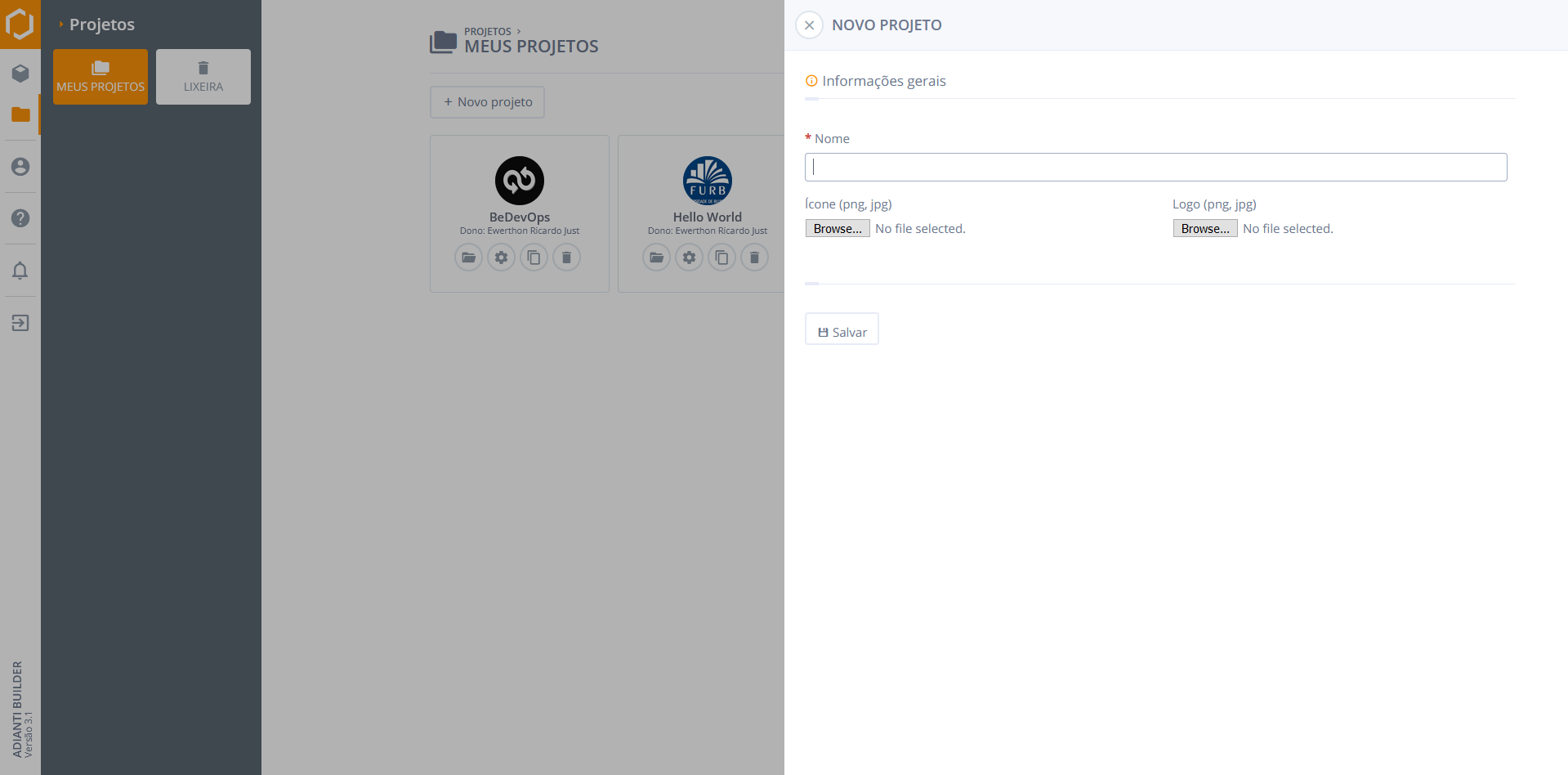
Figura – Planos do Adinti Builder



Fonte: AdiantiBuilder (2020).

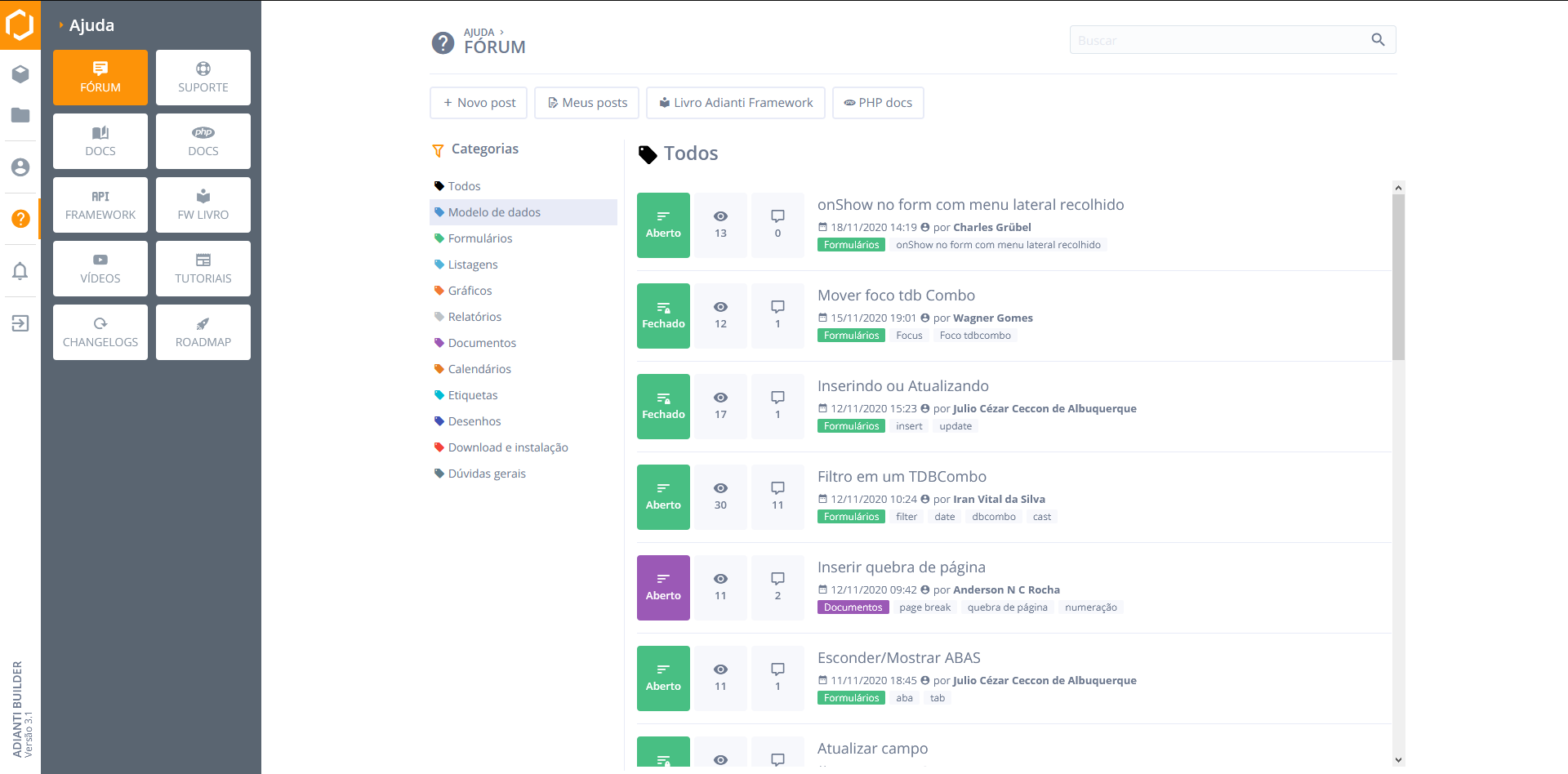
Após ter se cadastrado é possível criar um projeto (Figura 26) e caso você tenha comprado uma licença, dependendo do plano, será possível: abrir chamados para a equipe técnica; criar mais de um projeto; ter acesso a projetos prontos; e ao fórum da comunidade utilizadora (Figura 28). Tendo criado um projeto, será provisionada toda a estrutura base para o desenvolvimento do projeto. Na Figura 27 é demonstrado o Dashboard que apresenta os tipos de páginas criadas no projeto.

Figura – Criar projeto no Adianti Builder



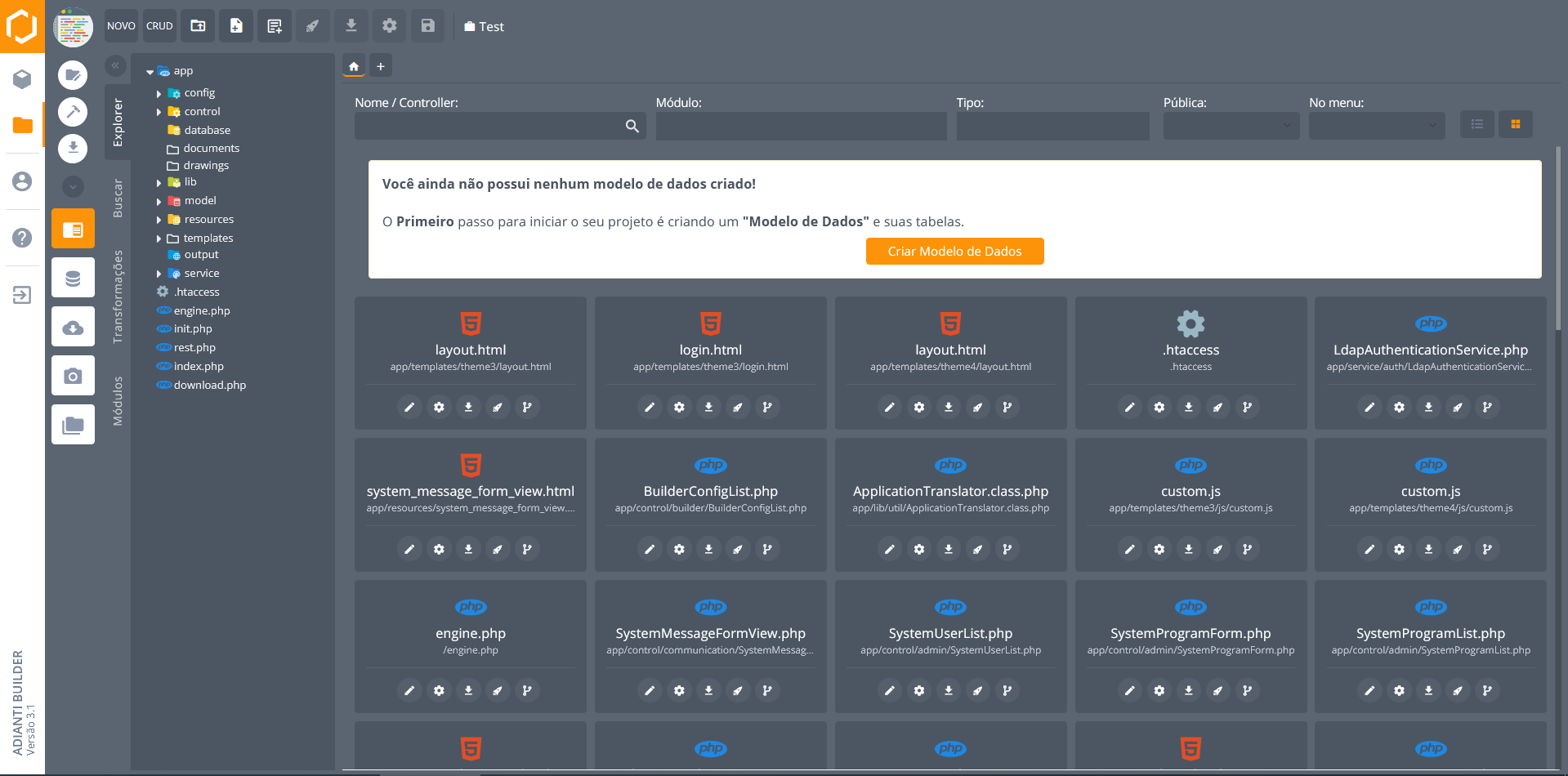
Fonte: AdiantiBuilder (2020).

Figura – Fórum da comunidade do Adianti Builder



Fonte: AdiantiBuilder (2020).

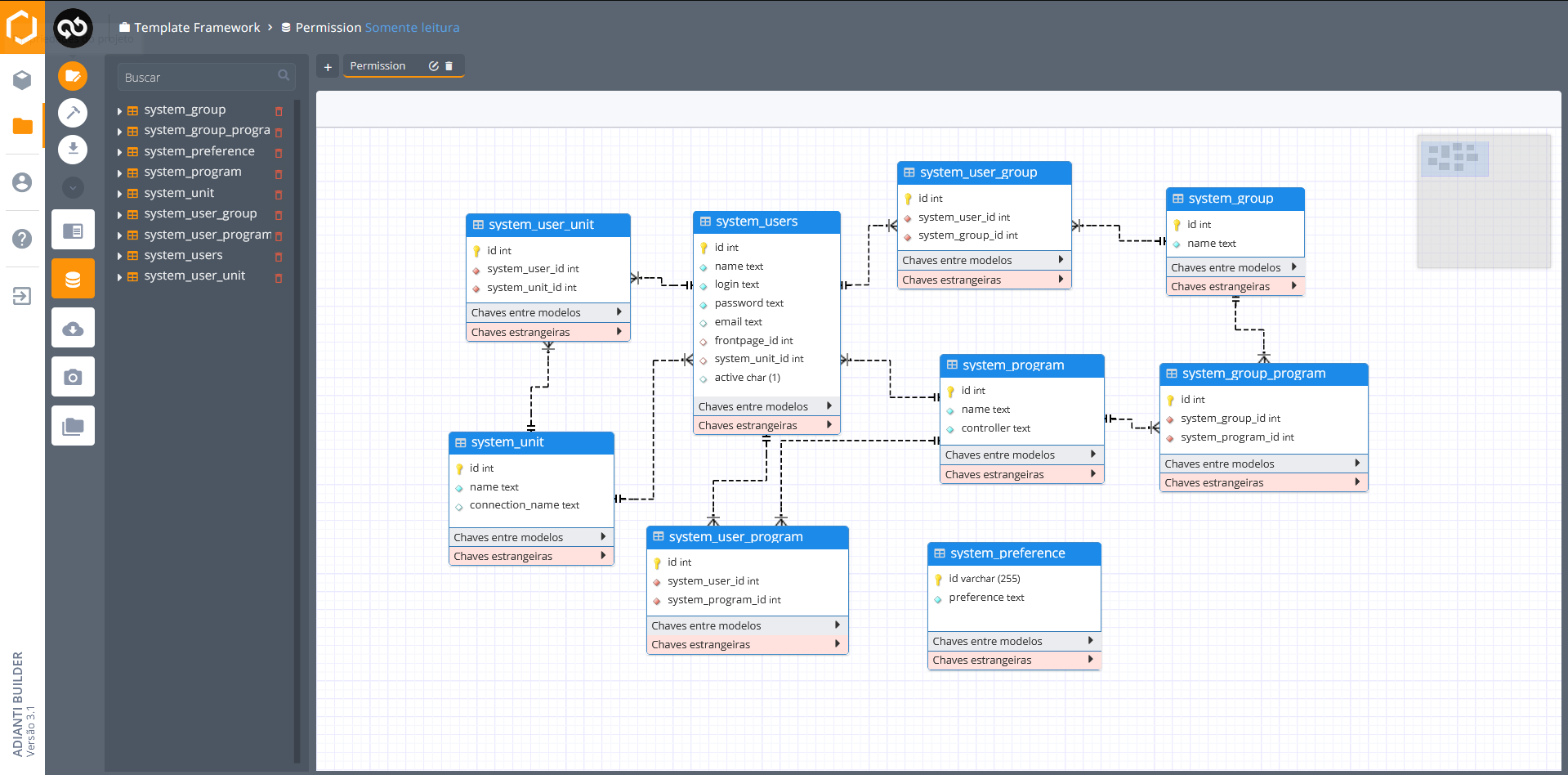
Figura – Página de suporte do Adianti Builder



Fonte: AdiantiBuilder (2020).

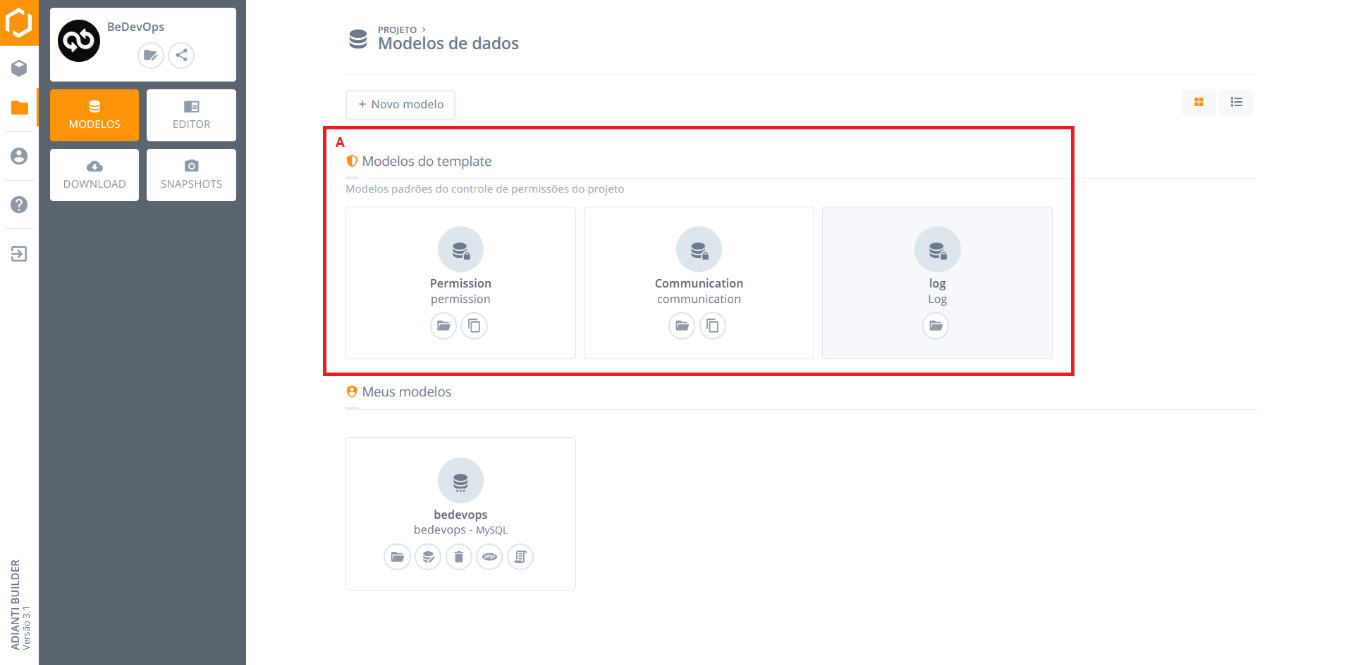
A ferramenta dispõe de um modulo para modelagem de dados no qual é possível estruturar todo o banco de dados por meio de interface gráfica baseado em diagrama MER. Na Figura 29é demonstrada a tela de modelagem da estrutura de dados do Adianti Builder. O Adianti Builder, conforme mencionado na subseção 3.2.3, pré-dispõem de modelos de estrutura do banco de dados para o funcionamento base da ferramenta, conforme retângulo A da Figura 30.

Figura – Modelagem da base de dados no Adianti Builder



Fonte: AdiantiBuilder (2020).

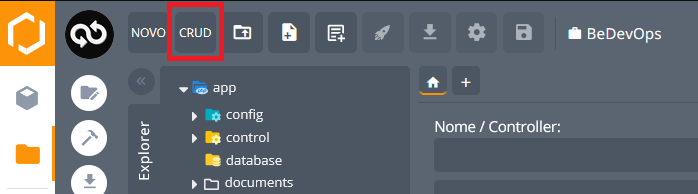
Figura – Modelos provisionados pelo Adianti Builder



Fonte: adaptado de AdiantiBuilder (2020).

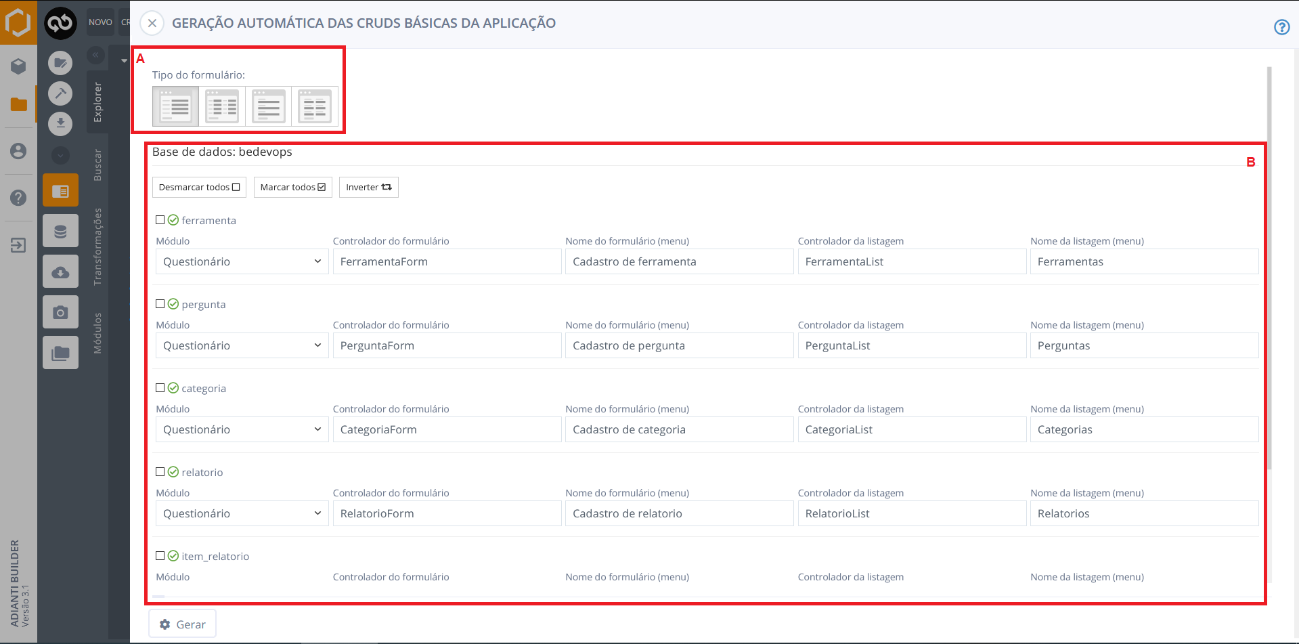
Um modulo gerador de telas de CRUD é disponibilizada pelo Adianti Builder, que utiliza como base o modelo de dados estruturado. Na Figura 31 é demonstrado o botão CRUD responsável por chamar o modulo. Uma tela de Geração automática das CRUDS básicas da aplicação é apresentada. Conforme retângulo A da Figura 32 é possível selecionar o Tipo de formulário e no Retângulo B da Figura 32, a partir de qual tabela do modelo se deseja criar a tela.

Figura – Criar telas CRUD com base no modelo de dados



Fonte: adaptado de AdintiBuilder (2020).

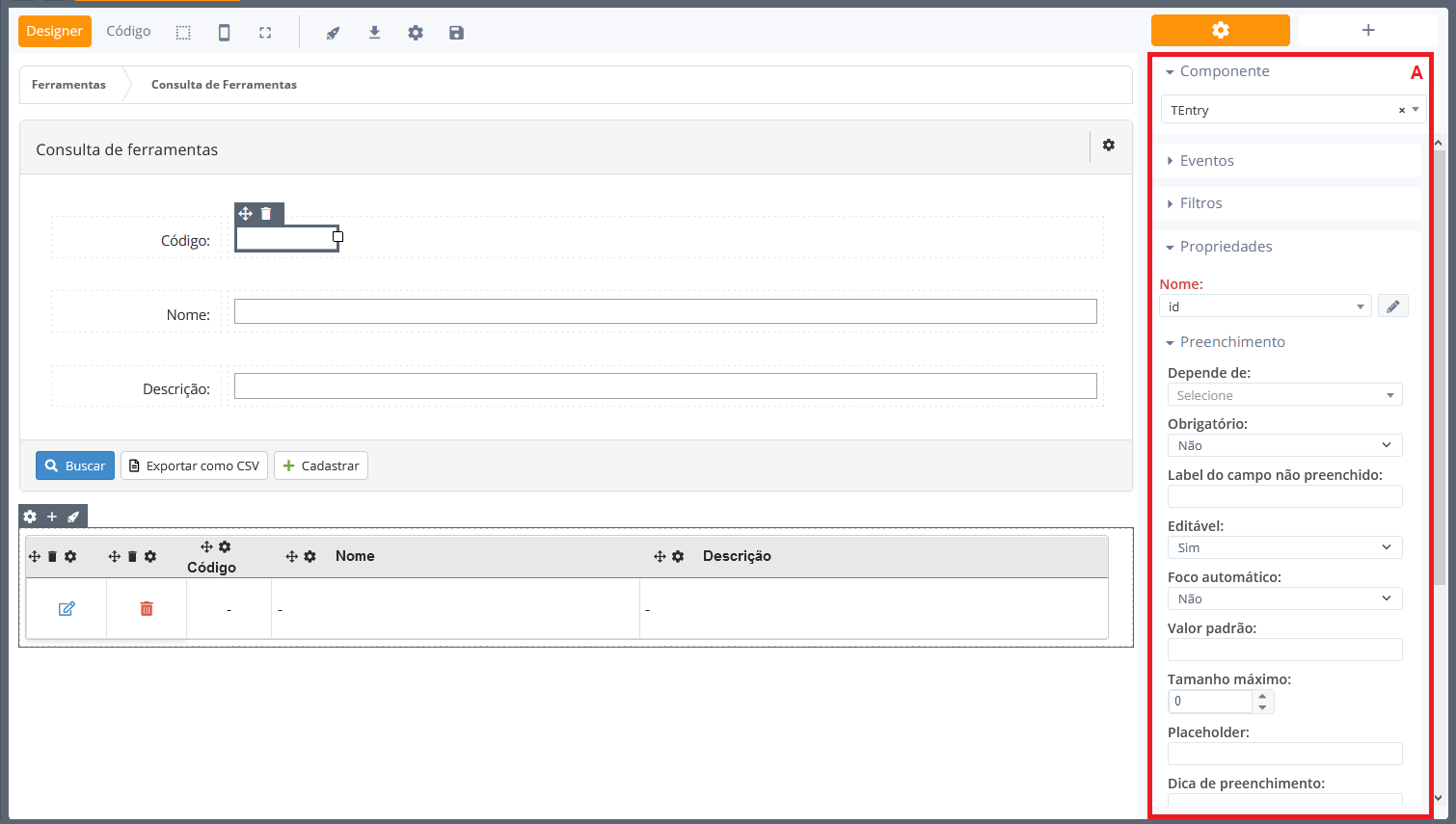
Figura - Geração automática das CRUDS básicas da aplicação



Fonte: adaptado de AdintiBuilder (2020).

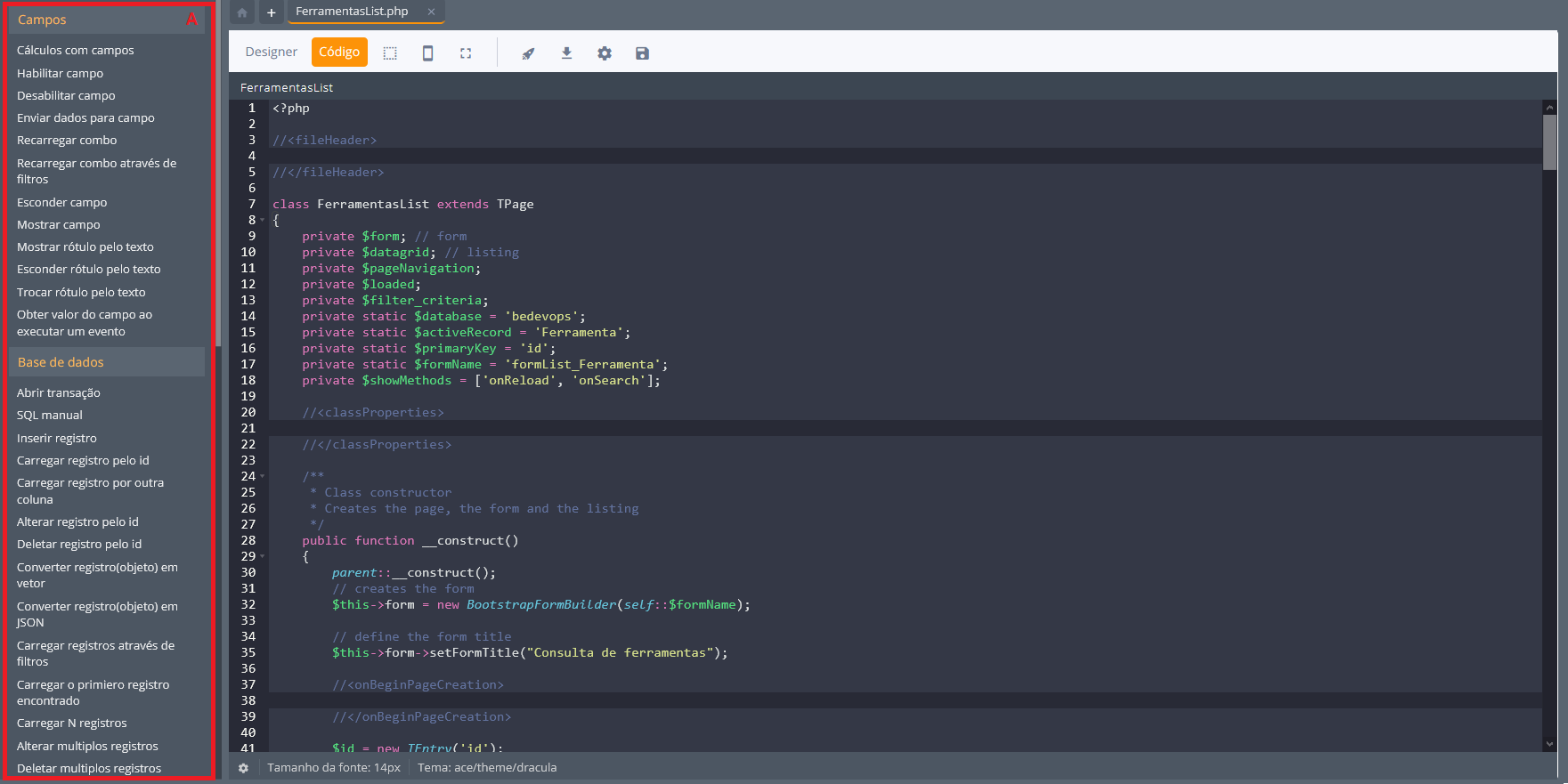
Após a criação de uma tela utilizando o gerador, é apresentado um ambiente como dois modos de edição para o ajuste de campos, formatação de textos e demais propriedades da tela. A Figura 33 refere-se a aba Designer para a edição da tela em modo interface gráfica em que o Retângulo A trata de todas as propriedades de componentes da tela. A Figura 34 apresenta o modo de edição em Código, onde é possível realizar a inserção de funções, rotinas, regras e campos personificados. O Retângulo B da Figura 34 é demonstrada a aba Snippets, sendo ela responsável principalmente por trazer direcionamentos para interação com a base de dados, componentes da tela, chamada de janelas e interações Representational State Transfer (REST).

Figura – Edição em modo designer



Fonte: adaptado de AdintiBuilder (2020).

Figura – Edição em modo código



Fonte: adaptado de AdintiBuilder (2020).

### Operacionalidade da implementação

## 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)