**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO TRIÂNGULO MINEIRO – *CAMPUS* PARACATU**

**TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**CRISTIAN ALVES SILVA**

**ASSISTENTE DE IMPLANTAÇÃO DE SERVIDORES EM DOCKER**

**PARACATU - MG 2024**

**CRISTIAN ALVES SILVA**

**ASSISTENTE DE IMPLANTAÇÃO DE SERVIDORES EM DOCKER**

Trabalho de Conclusão de Curso apresentado ao curso Tecnologia em Análise e Desenvolvimento de Sistemas, do Instituto Federal de Educação, Ciência e Tecnologia do Triângulo Mineiro, Campus Paracatu, como parte dos requisitos para obtenção do título de Analista de Sistemas.

Orientador: Prof. Claiton Luiz Soares

**PARACATU - MG 2024**

Ficha Catalográfica elaborada pelo Setor de Referência do IFTM – Campus Paracatu



Bibliotecário:

**TERMO DE APROVAÇÃO**

**CRISTIAN ALVES SILVA**

**ASSISTENTE DE IMPLANTAÇÃO DE SERVIDORES EM DOCKER**

Trabalho de conclusão de curso apresentado ao Instituto Federal de Educação, Ciência e Tecnologia do Triângulo Mineiro, Campus Paracatu, como exigência para obtenção do diploma de Tecnologia em Análise e Desenvolvimento de Sistemas, sob a orientação do Prof. Claiton Luiz Soares.

Aprovado em // de // de 2024.



IFTM (Prof. Orientador)



IFTM (Prof. Convidado)



(Prof. Convidado)

PARACATU - MG

2024

**DEDICATÓRIA**

Este Trabalho de Conclusão de Curso é dedicado de forma especial aos meus pais, Edilânio Batista Alves e Janaína Silva. Desde o início dos meus estudos até a conclusão desta graduação, vocês têm sido uma presença constante, me oferecendo um apoio inestimável e incansável.

**AGRADECIMENTOS**

Primeiramente, expresso minha profunda gratidão à minha família. Sem o apoio, paciência e incentivo de vocês, esta conquista não teria sido possível. Dedico este trabalho a vocês com todo meu carinho e gratidão.

À minha noiva, agradeço imensamente pela compreensão e apoio incondicional. Sua paciência durante os momentos de ausência e seu incentivo constante foram fundamentais para a realização deste trabalho.

Agradeço ao meu orientador, Prof. Claiton Luiz Soares, pela orientação dedicada e pela sabedoria compartilhada ao longo deste processo. Sua orientação foi essencial para o desenvolvimento e conclusão deste trabalho.

Manifesto meu reconhecimento a todos os professores que contribuíram para minha formação acadêmica. O conhecimento e a inspiração transmitidos por cada um de vocês foram cruciais para a busca pela excelência. Este trabalho reflete a influência e os ensinamentos recebidos ao longo da minha jornada acadêmica.

Por fim, agradeço a Deus pela orientação e fortalecimento que me permitiram superar os desafios enfrentados durante esta jornada acadêmica. Sou grato pelas bênçãos e pelo amparo constante que me possibilitaram alcançar esta conquista.

**EPÍGRAFE**

*"Não é a mais forte das espécies que sobrevive, nem a mais inteligente, mas a que melhor se adapta às mudanças."*

* *Charles Darwin*

**RESUMO**

A automação da implantação de servidores para desenvolvedores é o foco deste trabalho, que busca criar um assistente através de um Shell Script. Esse assistente automatiza o processo de instalação do Docker, bem como a criação e configuração de containers para diversos tipos de servidores, como MySQL, PostgreSQL, MariaDB, Apache, Nginx, VSFTPD e OpenSSH.

Com o intuito de simplificar a configuração e o gerenciamento de ambientes de desenvolvimento, essa proposta oferece uma solução eficiente para os desenvolvedores.

**Palavras-chave:** Docker; Containers; Servidores; Automação; Desenvolvimento.

**ABSTRACT**

This work focuses on automating server deployment for developers by creating an assistant using a Shell Script. The assistant automates the Docker installation process, as well as the creation and configuration of containers for various server types such as MySQL, PostgreSQL, MariaDB, Apache, Nginx, VSFTPD, and OpenSSH.

With the goal of simplifying the setup management of development environments, this proposal provides a pratical and efficient solution for developers.

***Keywords:*** Docker; Containers; Servers; Automation; Development.

**LISTA DE FIGURAS**

[**Figura 1** – Diagrama de planejamento do projeto.](#_1ci93xb) 25

[**Figura 2** – Diagrama de arquitetura do sistema](#_qsh70q). 26

**LISTA DE QUADROS**

[**Quadro 1** – Recursos planejados para implementar o trabalho.](#_4i7ojhp) 24

[**Quadro 2** – Tabela de requisitos.](#_3whwml4) 25

**Quadro 3** – [Testes Unitários](#_1pxezwc). 27

[**Quadro 4** – Testes integrados.](#_2p2csry) 28

**LISTA DE ABREVIATURAS E SIGLAS**

Apache – Servidor Web Apache

IFTM – Instituto Federal do Triângulo Mineiro.

MariaDB – Sistema de Gerenciamento de Banco de Dados derivado do MySQL.  
MySQL – Sistema de Gerenciamento de Banco de Dados SQL.

Nginx – Servidor Web Nginx.  
OpenSSH – *Open Secure Shell* (Shell Seguro Aberto).  
PostgreSQL – Sistema de Gerenciamento de Banco de Dados Relaciona Objeto.

SFTP – *SSH File Transfer Protocol* (Protocolo de Transferência de Arquivos SSH).

SQL – *Structured Query Language*(Linguagem de Consulta Estruturada).  
VSFTPD – *Very Secure FTP Daemon* (Servidor de FTP Muito Seguro).

**SUMÁRIO**

[**1 INTRODUÇÃO 14**](#_1fob9te)

[1.1 Problema 14](#_3znysh7)

[1.2 Tema 15](#_a3wempg694hz)

[1.3 Objetivos 15](#_71odpbvhzspx)

[1.3.1 Objetivo geral 15](#_u0zjlk0lrfs)

[1.3.2 Objetivos específicos 15](#_yo5wwsy3d5xa)

[1.4 Hipótese 16](#_tyjcwt)

[1.5 Justificativa 16](#_3dy6vkm)

[1.6 Delimitação 16](#_1t3h5sf)

[**2 REFERENCIAL TEÓRICO 18**](#_2s8eyo1)

[2.1 Docker 18](#_17dp8vu)

[2.2 Banco de Dados - PostgreSQL 18](#_3rdcrjn)

[2.3 Banco de Dados - MariaDB 19](#_26in1rg)

[2.4 Banco de Dados - MySQL 19](#_lnxbz9)

[2.5 Vsftpd 20](#_35nkun2)

[2.6 OpenSSH 20](#_1ksv4uv)

[2.7 Apache 20](#_44sinio)

[2.8 Nginx 21](#_2jxsxqh)

[**3 MATERIAIS E MÉTODOS 22**](#_z337ya)

[3.1 Recursos 23](#_4i7ojhp)

[3.2 Desenvolvimento do artefato 24](#_2xcytpi)

[3.2.1 Planejamento 24](#_1ci93xb)

[3.2.2 Levantamento de Requisitos 25](#_3whwml4)

[3.2.3 Design do Sistema 26](#_qsh70q)

[3.2.4 Implementação 27](#_3as4poj)

[3.2.5 Testes Unitários 28](#_1pxezwc)

[3.2.6 Testes integrados 29](#_2p2csry)

[**4 RESULTADOS E DISCUSSÕES 30**](#_3o7alnk)

[4.1 Eficácia da automação 30](#_2qx5srepsv90)

[4.2 Facilidade de uso 30](#_tbgkvraygy1h)

[4.3 Robustez e confiabilidade 30](#_55r7tnnjuop7)

[**5 CONCLUSÃO 31**](#_1hmsyys)

[5.1 Contribuições do trabalho 31](#_mjpoyz15g3rr)

[5.2 Limitações e trabalhos futuros 31](#_qwwqlef4yxu)

[5.3 Considerações Finais 32](#_nf2kslaq2u06)

[**REFERÊNCIAS 33**](#_41mghml)

# INTRODUÇÃO

A transformação digital e a crescente complexidade dos ambientes de desenvolvimento têm desafiado os desenvolvedores a buscar soluções mais eficientes e seguras para a implantação de servidores. A configuração manual de diversos serviços pode ser uma tarefa tediosa e suscetível a erros, frequentemente resultando em atrasos e frustrações. Nesse contexto, a criação de ambientes de desenvolvimento consistentes e replicáveis é essencial.

Inspirado pela filosofia de simplicidade de Steve Jobs, que afirmou: “A simplicidade é a máxima sofisticação”, este trabalho propõe o desenvolvimento de um assistente de implantação de servidores para desenvolvedores. Utilizando Shell Script, o assistente automatiza a instalação do Docker e a configuração de containers para uma variedade de servidores, incluindo bancos de dados como MySQL, PostgreSQL e MariaDB, servidores *web* como Apache e Nginx, e servidores SFTP como VSFTPD e OpenSSH.

A escolha do Docker como plataforma principal deve-se à sua capacidade de isolamento de recursos e portabilidade, garantindo ambientes de desenvolvimento consistentes em diferentes sistemas operacionais. A metodologia adotada envolve a criação de scripts Shell que automatizam tarefas críticas de instalação e configuração, reduzindo erros humanos e o tempo dedicado a operações repetitivas. Além da automação básica, o assistente oferece flexibilidade através da personalização dos parâmetros de configuração dos containers, permitindo adaptações conforme as necessidades específicas de cada projeto.

## Problema

A configuração manual de servidores e serviços em ambientes de desenvolvimento é uma tarefa que demanda tempo e está sujeita a erros humanos. Esse processo não apenas atrasa o ciclo de desenvolvimento, mas também introduz inconsistências e vulnerabilidades que podem comprometer a qualidade e a segurança do software. Desenvolvedores frequentemente enfrentam desafios ao tentar replicar ambientes de desenvolvimento em diferentes máquinas e sistemas operacionais, o que dificulta a colaboração e a continuidade dos projetos. A falta de padronização e a complexidade crescente dos sistemas modernos tornam a implantação manual impraticável e ineficaz.

## Tema

Automação da implantação de servidores e serviços utilizando Docker e Shell Script. Através da criação de scripts automatizados, o projeto busca fornecer uma solução prática e eficiente para desenvolvedores, permitindo a configuração rápida e consistente de ambientes de desenvolvimento, independentemente da plataforma utilizada.

## Objetivos

## Objetivo geral

Desenvolver um assistente de implantação automatizada de servidores, utilizando Shell Script para instalar e configurar Docker, facilitando a criação de containers para diferentes serviços como bancos de dados, servidores web e servidores SFTP. Este assistente deve simplificar e padronizar o processo de configuração, reduzindo o tempo e os erros associados à implantação manual.

## Objetivos específicos

* + - 1. Automatizar a instalação do Docker, garantindo que o ambiente esteja pronto para a configuração de containers.
      2. Desenvolver scripts que configurem containers para bancos de dados como MySQL, PostgreSQL e MariaDB, incluindo a personalização de parâmetros conforme necessário.
      3. Automatizar a configuração de containers para servidores web para Proxy Reverso como Nginx e servidores web Apache para páginas estáticas, assegurando a consistência e a segurança das configurações.
      4. Implementar scripts para a configuração de servidores SFTP, incluindo VSFTPD e OpenSSH, com foco na segurança e facilidade de uso.

## Hipótese

Considerando a necessidade de facilitar a implantação de servidores Docker, parte-se da hipótese de que é viável desenvolver um assistente de implantação que utilize uma interface intuitiva baseada em texto para guiar os usuários por meio do processo de configuração e implantação de containers Docker. Essa abordagem tem o potencial de reduzir a complexidade operacional e aumentar a acessibilidade para os usuários menos experientes em administração de servidores.

## Justificativa

Diante do contexto atual de ampla utilização de containers Docker para o gerenciamento de ambientes de aplicativos, bem como da crescente demanda por soluções de implantação simplificadas e acessíveis, a criação de um assistente de implantação de servidores em Docker se mostra altamente relevante.

A necessidade de uma justificativa para esse projeto se baseia na complexidade associada à configuração e à implantação de servidores Docker, bem como na falta de familiaridade de muitos usuários com tais procedimentos. Muitas vezes, a abordagem convencional demanda conhecimentos técnicos avançados e a execução de uma série de comandos complexos, o que pode representar uma barreira significativa para diversos usuários, em particular para aqueles que não possuem familiaridades com as operações de linha de comando.

Nesse sentido, a proposta de desenvolver um assistente de implantação de servidores em Docker visa aprimorar a experiência do usuário ao simplificar e guiar de maneira intuitiva todo o processo de implantação e configuração de containers. Tal abordagem tem o potencial de democratizar o uso da tecnologia Docker, permitindo que um espectro mais amplo de usuários usufrua dos benefícios dos containers e das aplicações neles executadas.

## Delimitação

A pesquisa proposta pertence à área de Análise e Desenvolvimento de Sistemas e tem como foco a automação da implantação de servidores em ambiente de desenvolvimento, utilizando Docker e Shell Script. O assistente desenvolvido foi testado em ambientes de desenvolvimentos locais, abrangendo a configuração de serviços como bancos de dados (MySQL, PostgreSQL, MariaDB), servidores web (Apache, Nginx) e servidores de transferência de arquivos (VSFTPD, OpenSSH).

Conforme o cronograma detalhado no presente documento, a pesquisa foi conduzida entre os meses de março e julho de 2024. Os testes e validações foram realizados em sistemas operacionais baseados em Unix, com o objetivo de garantir a portabilidade e a replicabilidade em ambientes de desenvolvimento automatizados.

# REFERENCIAL TEÓRICO

Nesta seção, são expostas as teorias que servirão de base para a pesquisa em questão, além de serem discutidos os temas principais que constituem o trabalho.

## Docker

O Docker é uma plataforma aberta para desenvolvedores e administradores de sistemas, que facilita a construção, envio e execução de aplicações distribuídas em diferentes ambientes, como *laptops*, VMs, ou na nuvem. Ele permite a virtualização a nível de sistema operacional, criando containers leves e portáteis, executáveis em qualquer máquina com Docker instalado.

Lançado em 2013 por Solomon Hykes, o Docker automatiza a implantação de aplicações dentro de containers, adicionando uma camada de abstração e automação para a virtualização de aplicações. (HYKES, Solomon. "Introducing Docker." Docker Blog, 2013).

Os principais recursos do Docker incluem:

* Empacotamento facilitado de aplicações;
* Isolamento de processos;
* Portabilidade;
* Manutenção e atualização simplificadas.

## Banco de Dados - PostgreSQL

O PostgreSQL é um sistema de gerenciamento de banco de dados relacionais (RDBMS) de código aberto, reconhecido por sua robustez, extensibilidade e conformidade com padrões SQL. Ele suporta uma ampla variedade de tipos de dados e funções avançadas, incluindo consultas complexas, chaves estrangeiras, gatilhos e transações ACID.

Inicialmente desenvolvido no Departamento de Ciência da Computação da Universidade da Califórnia em Berkeley, o PostgreSQL se tornou um dos sistemas de bancos de dados mais avançados e amplamente utilizados. Empresas como Apple, Fujitsu e Red Hat utilizam PostgreSQL em suas operações. (The PostgreSQL Global Development Group. "PostgreSQL: The World's Most Advanced Open Source Relational Database." PostgreSQL, 2024.).

Dentre seus principais recursos estão:

* Extensibilidade;
* Suporte a SQL padrão;
* Tipos de dados avançados;
* Uma comunidade ativa.

## Banco de Dados - MariaDB

O MariaDB é um sistema de gerenciamento de banco de dados relacionais (RDBMS) de código aberto, desenvolvido como uma bifurcação do MySQL após sua aquisição pela Oracle Corporation. Foi criado por Michael “Monty” Widenius, um dos fundadores originais do MySQL, com o objetivo de manter o projeto livre e aberto. (MARIADB FOUNDATION, 2024).

O MariaDB é conhecido por sua alta compatibilidade com MySQL, permitindo uma migração fácil para aqueles que já utilizam MySQL. Além disso, o MariaDB oferece melhorias significativas em desempenho e novos recursos. É amplamente utilizado em várias aplicações, desde pequenas start-ups até grandes corporações, como a Wikipédia e o Google.

Os principais recursos do MariaDB incluem:

* Melhor desempenho e escalabilidade;
* Alta compatibilidade com MySQL;
* Melhorias de segurança;
* Suporte ativo e contínuo pela comunidade.

## Banco de Dados - MySQL

O MySQL é um sistema de gerenciamento de banco de dados relacional (SGBDR) de código aberto, reconhecido pela sua confiabilidade, desempenho e facilidade de uso. Compatível com diversos sistemas operacionais, como Windows, Linux e macOS, o MySQL é amplamente adotado em várias aplicações, desde pequenos *sites* até grandes empresas como Facebook, Twitter e Youtube.

Inicialmente desenvolvido pela empresa sueca MySQL AB e posteriormente adquirido pela Sun Microsystems. (Oracle Corporation. "MySQL :: MySQL Documentation." MySQL, 2024).

O MySQL oferece:

* Alta performance;
* Suporte multiplataformas;
* Segurança robusta;
* Uma grande comunidade de suporte.

## Vsftpd

O VSFTPD (*Very Secure FTP Daemon*) é um servidor FTP (*File Transfer Protocol*)conhecido por sua alta segurança, desempenho e simplicidade. Ele é projetado para ser rápido e leve, oferecendo suporte a funcionalidade como transferência de arquivos em modo seguro através de SSL/TLS, controle de acesso por IP, limitação de banda e isolamento de usuários em chroot. O VSFTPD é frequentemente utilizado em ambientes que necessitam de transferência de arquivos eficientes e seguras (VSFTPD, 2024).

O VSFTPD se destaca por seu foco em segurança, com suporte nativo a SSL/TLS para FTP seguro, isolamento de usuários e diversas opções de configuração de segurança. Ele é amplamente utilizado em ambientes de hospedagem compartilhada e servidores corporativos devido à sua confiabilidade e performance.

## OpenSSH

O *OpenSSH* é um conjunto de ferramentas de rede que permite a comunicação segura entre computadores através do protocolo SSH. Ele inclui um servidor SFTP integrado, que proporciona transferências de arquivos seguras. O *OpenSSH* é amplamente utilizado por sua robustez, segurança e por ser uma solução de código aberto confiável. Além da transferência segura de arquivos, o *OpenSSH* oferece funcionalidades como login remoto seguro, tunelamento seguro de portas e execução remota de comandos (OPENSSH, 2024).

O *OpenSSH* é valorizado por sua segurança robusta, incluindo suporte a autenticação por chave pública, criptografia forte e tunelamento de portas. Essas funcionalidades garantem que as comunicações entre clientes e servidores sejam protegidas contra interceptações e ataques.

## 2.7 Apache

O Apache HTTPS Server, geralmente conhecido como Apache, é um dos servidores *web* mais utilizados no mundo. Desenvolvido pela Apache Software Foundation, o Apache é conhecido por sua robustez, flexibilidade e vasto conjunto de funcionalidades. Ele suporta módulos que podem ser carregados como suporte a linguagens de script, autenticação e SSL/TLS para conexões seguras. O Apache é amplamente utilizado em ambientes de hospedagem compartilhada e em grandes corporações devido à sua capacidade de personalização e escalabilidade (APACHE SOFTWARE FOUNDATION, 2024).

O Apache oferece uma ampla gama de funcionalidades, incluindo suporte a várias linguagens de script como PHP, Perl e Python, autenticação baseada em módulos, e um sistema de configuração extensível baseado em arquivos texto. Ele também suporta balanceamento de carga e configuração de múltiplos *hosts* virtuais, permitindo a hospedagem de várias aplicações em um único servidor.

## 2.8 Nginx

O NGINX é um servidor *web* de alto desempenho, conhecido por sua arquitetura eficiente e escalabilidade. Originalmente desenvolvido como uma solução para o problema de C10k, ou seja, a capacidade de lidar com 10.000 conexões simultâneas, o NGINX é amplamente utilizado não só como servidor *web*, mas também como *proxy* reverso, *load balancer* e cache HTTP. Ele é popular em ambientes de alta demanda, onde a performance e a capacidade de gestão de grandes volumes de tráfego são cruciais (NGINX, 2024).

O NGINX se destaca por sua arquitetura assíncrona e baseada em eventos, que permite um melhor desempenho sob cargas elevadas. Ele suporta HTTP/2, SSL/TLS, e tem funcionalidades de *proxy* reverso e balanceamento de cargas integradas. Essas características o tornam uma escolha ideal para sites de alto tráfego e aplicações *web* em grande escala.

# MATERIAIS E MÉTODOS

Neste trabalho, utilizamos uma série de materiais e ferramentas tecnológicas para desenvolver e testar o assistente de implantação de servidores. Abaixo, listamos os principais componentes utilizados:

1. *Hardware*
   1. Notebook para configuração e programação do script;
2. *Software*
   1. Docker 27.0.2: Utilizado para a criação e gerenciamento de containers;
   2. Shell Bash: Linguagem de script utilizada para automatizar a instalação e configuração dos servidores.
   3. Servidores de Bancos de Dados:
      * MySQL 8.4.0
      * PostgreSQL 16.3
      * MariaDB 11.4.2
   4. Servidores *Web*:
      * Apache 2.4.59
      * NGINX 1.27.0
   5. Servidores SFTP:
      * VSFTPD 3.0.5
      * OpenSSH 9.2

Para o desenvolvimento do assistente de implantação de servidores, seguimos um conjunto estruturado de etapas que abrangem desde a criação do script até a validação e testes finais.

1. Análise de requisitos:
   1. Identificação das necessidades dos desenvolvedores em termos de ambientes de desenvolvimento;
   2. Seleção das tecnologias e serviços essenciais para inclusão no script (Docker, bancos de dados, servidores *web* e SFTP);
2. Desenvolvimento do Script Shell:
   1. Escrita de um script Shell para automatizar a instalação do Docker e a configuração de containers para diferentes tipos de servidores;
   2. Estruturação do script para incluir opções de instalação e configuração para:
      * Bancos de dados (MySQL, PostgreSQL, MariaDB);
      * Servidores *web* (Apache, NGINX);
      * Servidores SFTP (VSFTPD, OpenSSH).
3. Implementação e Testes:
   1. Testes iniciais em ambiente de desenvolvimento local para verificar a correta instalação e funcionamento dos containers;
   2. Ajustes no script baseados no *feedback* obtido durante os testes;
   3. Testes finais para garantir que o script funcione em diferentes configurações de sistema operacional em ambientes de desenvolvimento.
4. Validação
   1. Validação do script por meio de execução de diversos cenários para assegurar a robustez e a flexibilidade da solução.
5. Avaliação dos resultados:
   1. Análise dos benefícios proporcionados pelo uso do assistente de implantação de servidores em comparação com métodos tradicionais de configuração manual.

## Recursos

Para a implementação da pesquisa e do protótipo, o levantamento dos recursos a serem utilizados juntamente com os custos é apresentado no Quadro 1.

Quadro 1 – Recursos planejados para implementar o trabalho.

| **Descrição** **Unidade** **Quantidade Valor Total**  **Unitário** | | | | |
| --- | --- | --- | --- | --- |
| Computador com processador Intel Core i3 | Peça | 1 | R$ 1500,00 | R$ 1500.00 |
| Memória RAM 8 GB | Peça | 1 | R$ 300,00 | R$ 300,00 |
| Armazenamento SSD 256 GB | Peça | 1 | R$ 250,00 | R$ 250,00 |
| Sistema Operacional Debian 12.5 | Licença | 1 | Gratuito | Gratuito |
| Docker 27.0.2 | Licença | 1 | Gratuito | Gratuito |
| Shell Bash | Licença | 1 | Gratuito | Gratuito |
| MySQL 8.4.0 | Licença | 1 | Gratuito | Gratuito |
| PostgreSQL 16.3 | Licença | 1 | Gratuito | Gratuito |
| MariaDB 11.4.2 | Licença | 1 | Gratuito | Gratuito |
| Apache 2.4.59 | Licença | 1 | Gratuito | Gratuito |
| NGINX 1.27.0 | Licença | 1 | Gratuito | Gratuito |
| VSFTPD 3.0.5 | Licença | 1 | Gratuito | Gratuito |
| Visual Studio Code | Licença | 1 | Gratuito | Gratuito |
| Nano | Licença | 1 | Gratuito | Gratuito |
| Git | Licença | 1 | Gratuito | Gratuito |
| Docker Hub | Serviço | 1 | Gratuito | Gratuito |
| **TOTAL** |  |  |  | **R$ 2050,00** |

Fonte – elaborado pelo autor.

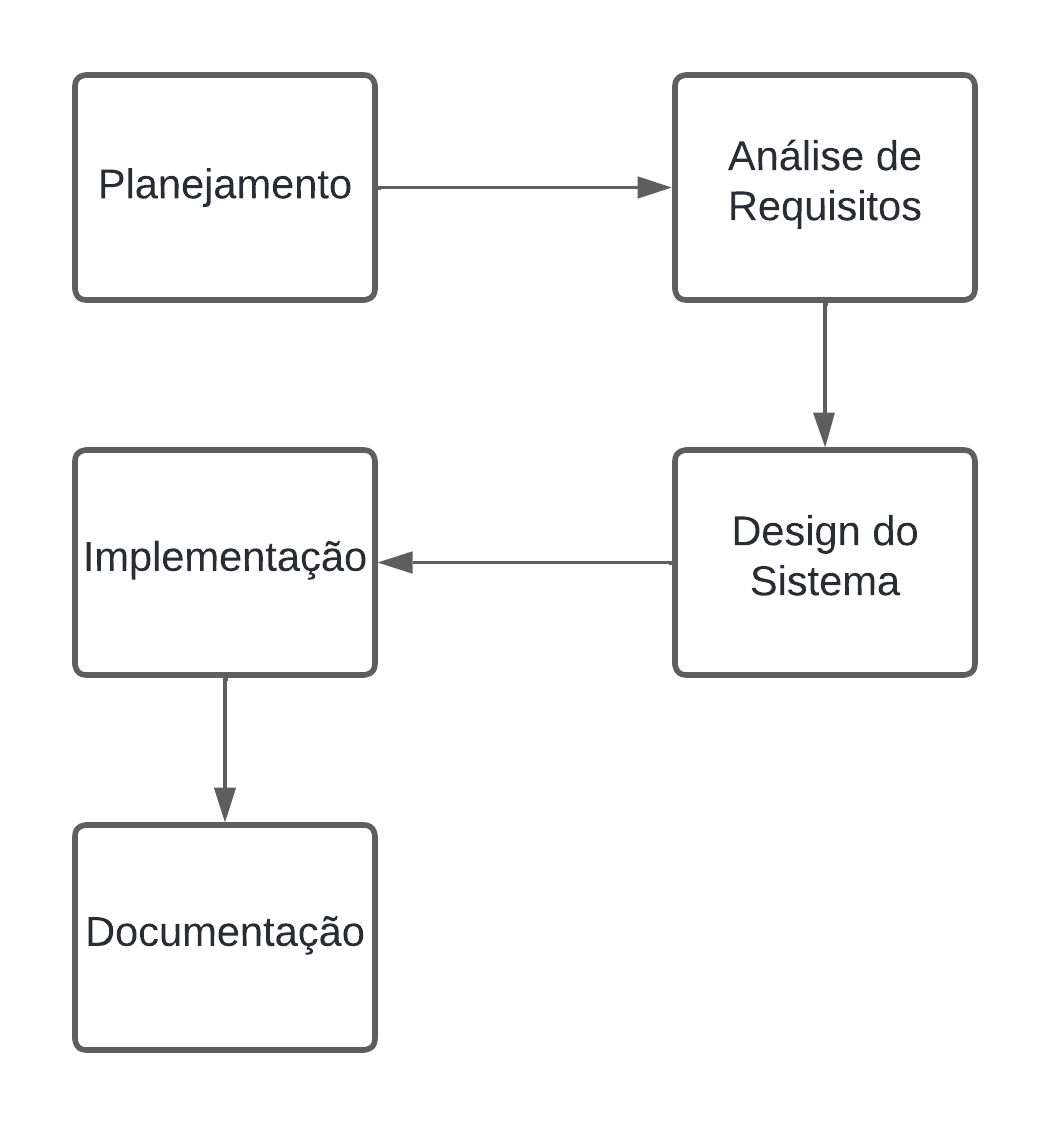
## Desenvolvimento do artefato

Para o desenvolvimento deste trabalho, foram utilizados *design patterns* para definir a estrutura do projeto, a seleção das tecnologias a serem utilizadas e a identificação dos serviços a serem automatizados. As principais atividades realizadas durante o planejamento foram:

## Planejamento

* Identificação das necessidades dos desenvolvedores: Entrevistas e questionários foram aplicados para identificar as principais dificuldades enfrentadas pelos desenvolvedores na configuração de ambientes de desenvolvimento;
* Seleção das tecnologias: Foi decido utilizar Shell Script e Docker devido a sua popularidade e eficiência na automação de tarefas;
* Definição dos requisitos: Foram estabelecidos requisitos funcionais e não funcionais para o assistente, garantindo que ele atendesse às expectativas dos usuários.

Figura 1 – Diagrama de planejamento do projeto.



Fonte – Autor.

## Levantamento de Requisitos

A análise de requisitos envolver a coleta das necessidades dos usuários, bem como a definição das funcionalidades essenciais do assistente. As atividades principais desta fase incluíram:

* Levantamento de requisitos: Foram realizadas entrevistas com desenvolvedores e revisão de documentação técnica para identificar as necessidades e expectativas;
* Definição de funcionalidades: As funcionalidades principais foram definidas, como instalação e configuração automatizada de containers Docker para bancos de dados, servidores *web* e servidores SFTP;
* Especificação dos requisitos: Os requisitos foram documentados de forma detalhada, incluindo requisitos funcionais e não funcionais.

Quadro 2 – Tabela de requisitos.

| **Tipo de Requisito Descrição** | |
| --- | --- |
| Funcional | Instalador de Docker  Configuração de containers MySQL  Configuração de containers PostgreSQL  Configuração de containers MariaDB  Configuração de containers Apache  Configuração de containers Nginx  Configuração de containers VSFTPD  Configuração de containers OpenSSH |
| Não funcional | Segurança  Usabilidade  Desempenho |

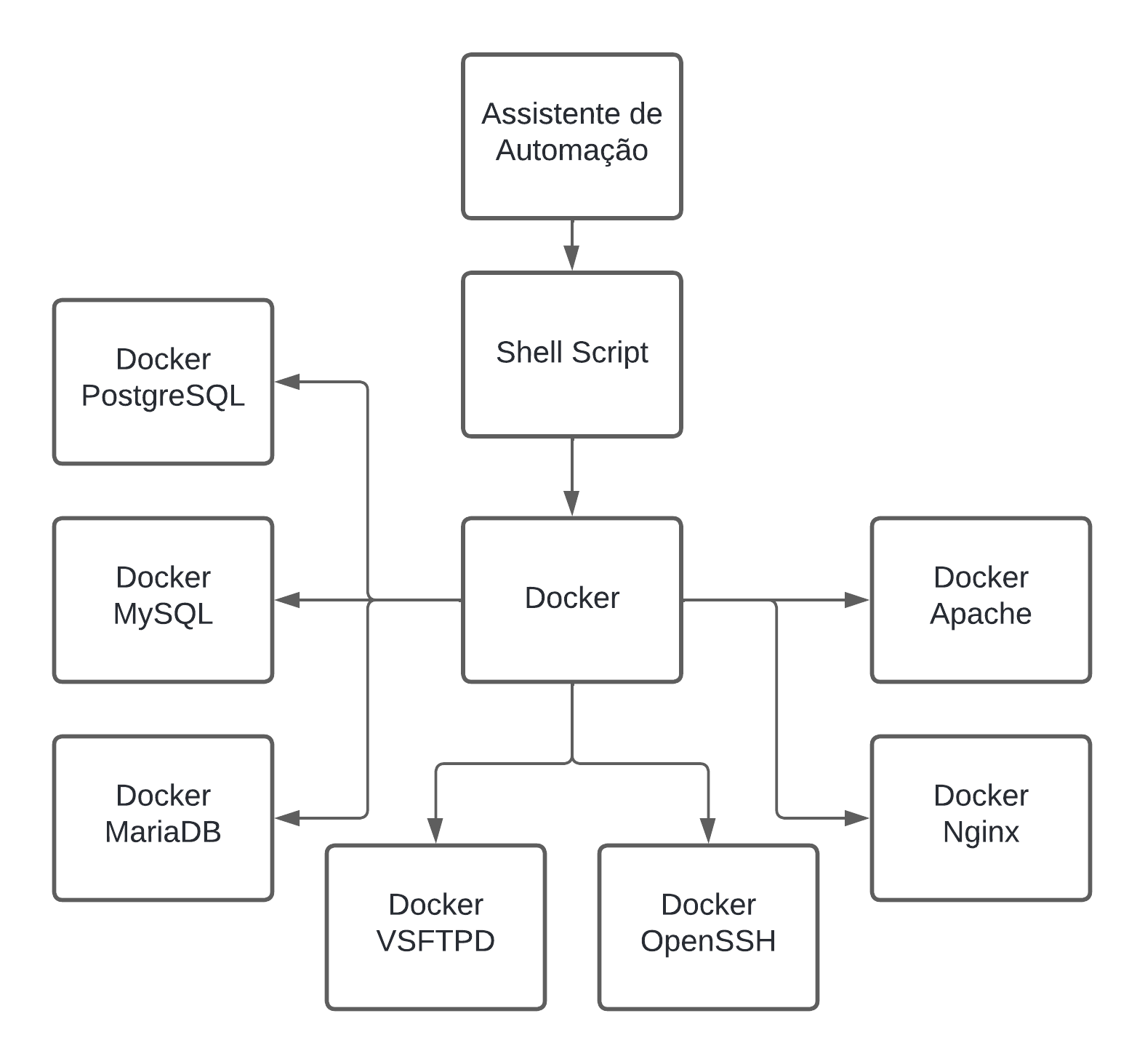
Fonte – elaborado pelo autor.

## *Design* do Sistema

O design do sistema envolver a criação de diagramas e esquemas para definir a arquitetura do assistente. Esta fase incluiu:

* Arquitetura do sistema: Foi definida a estrutura geral do assistente, incluindo os módulos e componentes principais;
* Interface de usuário: Planejamento da interface baseada em texto, garantindo que fosse intuitiva e fácil de usar.

Figura 2 – Diagrama de arquitetura do sistema.

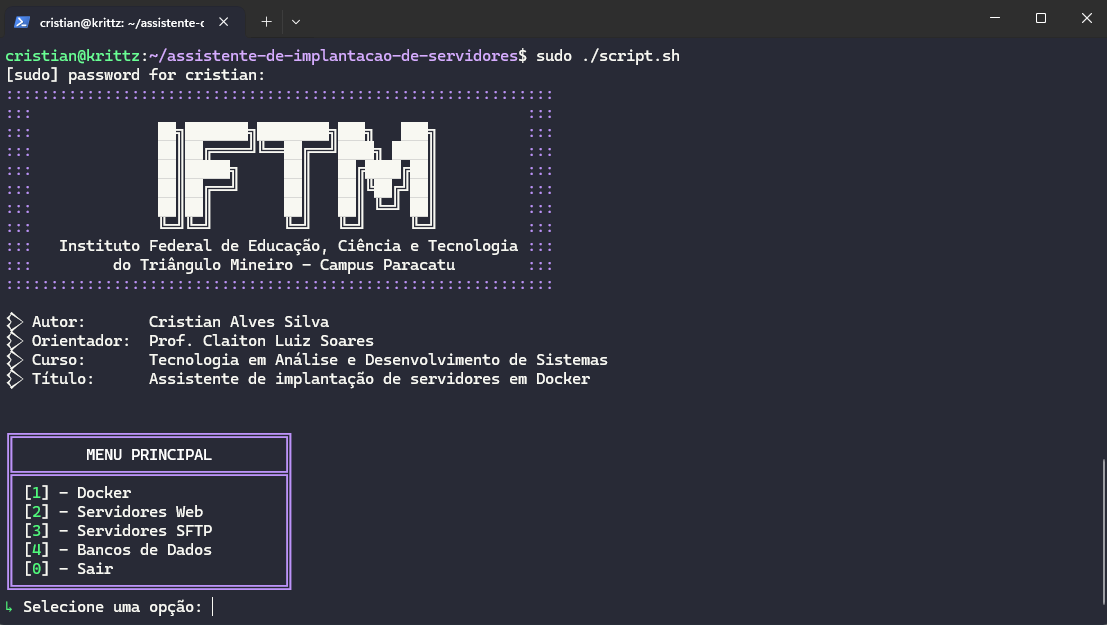


Fonte – Autor.

## Implementação

A implementação foi conduzida utilizando um Shell Script para automatizar a instalação e a configuração dos serviços listados anteriormente. A interface de usuários do script é simples e intuitiva, permitindo ao usuário escolher qual serviço instalar através de um menu. Abaixo está o *layout* da interface de usuário:

Figura 3 - Interface de usuário.



Fonte – Autor.

## Testes Unitários

Os testes unitários foram realizados para cada componente do sistema, garantindo que cada serviço fosse instalado e configurado corretamente. A tabela ilustrada no quadro 3 resume os resultados dos testes unitários.

Quadro 3 – Testes unitários

| **Componentes Testes Resultados** | | |
| --- | --- | --- |
| Instalação Docker | Passados  Falhados | 10  0 |
| Instalação MySQL | Passados  Falhados | 10  0 |
| Instalação PostgreSQL | Passados  Falhados | 10  0 |
| Instalação MariaDB | Passados  Falhados | 10  0 |
| Instalação Apache | Passados  Falhados | 9  1 |
| Instalação Nginx | Passados  Falhados | 8  2 |
| Instalação VSFTPD | Passados  Falhados | 9  1 |
| Instalação OpenSSH | Passados  Falhados | 10  0 |

Fonte – elaborado pelo autor.

## Testes integrados

Os testes integrados foram conduzidos para assegurar que o sistema funcionasse de forma coesa, com todos os componentes instalados e configurados corretamente. A tabela ilustrada no quadro 4 resume os resultados dos testes unitários:

Quadro 4 – Testes integrados.

| **Cenário de Teste Testes Resultados** | | |
| --- | --- | --- |
| Instalação completa (MySQL) | Passados  Falhados | 10  0 |
| Instalação completa (PostgreSQL) | Passados  Falhados | 10  0 |
| Instalação completa (MariaDB) | Passados  Falhados | 10  0 |
| Instalação completa (Apache) | Passados  Falhados | 10  0 |
| Instalação completa (Nginx) | Passados  Falhados | 10  0 |
| Instalação completa (VSFTPD) | Passados  Falhados | 10  0 |
| Instalação completa (OpenSSH) | Passados  Falhados | 10  0 |

Fonte – elaborado pelo autor.

# RESULTADOS E DISCUSSÕES

Nesta seção, serão apresentados os resultados do trabalho, obtidos durante o estudo. Os resultados obtidos demonstram a eficácia do assistente de automação em realizar a instalação e a configuração dos serviços de maneira eficiente e sem erros.

## Eficácia da automação

A automação proporcionada pelo Shell Script mostrou-se altamente eficaz, reduzindo significativamente o tempo necessários para a instalação e configuração dos serviços.

Cada serviço foi instalado corretamente na primeira execução do script, sem a necessidade de intervenções manuais adicionais.

## Facilidade de uso

A interface de usuário do script foi avaliada como intuitiva e fácil de usar. Os usuários podem selecionar os serviços desejados e acompanhar o progresso de instalação através de mensagens claras exibidas pelo script. Esse aspecto é crucial para garantir que mesmo usuários com menos experiência de administração de sistemas possam utilizar a ferramenta com facilidade.

## Robustez e confiabilidade

A robustez do script foi evidenciada pelos resultados dos testes, que não apresentaram falhas durante a instalação e configuração dos serviços. A verificação preliminar da existência do Docker no sistema e a instalação automatizada em caso de ausência contribuem para a confiabilidade do assistente de automação.

# CONCLUSÃO

O desenvolvimento do assistente de implantação de servidores em Docker, descrito ao longo deste trabalho, demonstrou ser uma solução eficaz para a automatização e padronização da configuração de ambientes de desenvolvimento. Através do uso de Shell Script, foi possível criar um processo que não só agiliza a instalação e configuração de containers Docker, mas também minimiza os erros humanos e garante maior consistência entre os ambientes.

## Contribuições do trabalho

O assistente desenvolvido oferece uma série de benefícios práticos para desenvolvedores e administradores de sistemas, entra quais destacam-se:

* Automação de tarefas repetitivas, a implementação de scripts que automatizam a instalação e configuração de diversos tipos de servidores (como MySQL, PostgreSQL, MariaDB, Apache, Nginx, VSFTPD e OpenSSH) facilita a criação de ambientes de desenvolvimento replicáveis e consistentes;
* Redução de erros humanos, ao automatizar tarefas complexas e suscetíveis a erros, o assistente contribui para a redução de falhas que podem comprometer a qualidade e a segurança dos sistemas;
* Aumento da produtividade, desenvolvedores podem focar em atividades mais estratégicas e críticas, deixando as tarefas operacionais para o assistente, o que resulta em um aumento geral da produtividade.

## Limitações e trabalhos futuros

Embora o assistente tenha alcançado seus objetivos principais, algumas limitações foram identificadas durante o desenvolvimento:

* Dependência de ambientes Unix-like, o assistente foi testado majoritariamente em sistemas Unix-like, o que pode limitar sua aplicabilidade em outros sistemas operacionais sem adaptações adicionais;
* Escalabilidade e personalização, embora o assistente permite a personalização dos containers, a escalabilidade do script em projetos de maior complexidade ainda pode ser um desafio.

Para trabalhos futuros, sugere-se a ampliação da compatibilidade do assistente com diferentes sistemas operacionais, bem como o desenvolvimento de uma interface gráfica que torne a ferramenta mais acessível para usuários com menor familiaridade com linha de comando.

## Considerações Finais

O trabalho realizado reforça a importância da automação na configuração de ambientes de desenvolvimento e destaca o potencial do Docker e do Shell Script como ferramentas-chave nesse processo. A adoção de soluções automatizadas não só melhora a eficiência operacional, mas também contribui para a criação de sistemas mais seguros e padronizados.

# REFERÊNCIAS

DOCKER INC. Docker Documentation. Disponível em: <https://docs.docker.com/>. Acesso em: 20 de abril de 2024.

DOCKER INC. Docker Hub. Disponível em: <https://hub.docker.com/>. Acesso em: 20 de abril de 2024.

THE POSTGRESQL GLOBAL DEVELOPMENT GROUP. PostgreSQL: The World’s Most Advanced Open-Source Relational Database. PostgreSQL, 2040. Disponível em: https://postgresql.org/. Acesso em: 23 de abril de 2024.

MARIADB FOUNDATION. MariaDB. MariaDB Foundation, 2024. Disponível em: <https://mariadb.org/>. Acesso em: 26 de abril de 2024.

ORACLE CORPORATION. MySQL :: MySQL Documentation. MySQL, 2024. Disponível em: <https://dev.mysql.com/doc/>. Acesso em: 27 de abril de 2024.

VSFTPD. Very Secure FTP Daemon. VSFTPD, 2024. Disponível em: <https://security.appspot.com/vsftpd.html>. Acesso em: 30 de abril de 2024.

OPENSSH. OpenSSH. OpenSSH, 2024. Disponível em: [https://www.openssh.com/.](https://www.openssh.com/) Acesso em: 04 de maio de 2024.

APACHE SOFTWARE FOUNDATION. Apache HTTP Server. Apache Software Foundation, 2024. Disponível em: <https://httpd.apache.org/>. Acesso em: 7 de maio de 2024.

NGINX. NGINX. NGINX, 2024. Disponível em: <https://nginx.com/>. Acesso em 7 de maio de 2024.

ALVES, Atos Ramos. Administração de Servidores Linux. Rio de Janeiro, RJ: Ciência Moderna, 2013

JARGAS, Aurelio Marinho. Shell Script Profissional. São Paulo, SP: Novatec, 2008.