



**Fundação Educacional do Município de Assis  
Instituto Municipal de Ensino Superior de Assis  
Campus "José Santilli Sobrinho"**

**CLAUDEMIR ROSA**

**SOFTWARE PARA TRANSFERÊNCIA DE DADOS**

**Assis  
2021**



**Fundação Educacional do Município de Assis  
Instituto Municipal de Ensino Superior de Assis  
Campus "José Santilli Sobrinho"**

**CLAUDEMIR ROSA**

## **SOFTWARE PARA TRANSFERÊNCIA DE DADOS**

Trabalho de conclusão de curso apresentado ao curso de Análise e Desenvolvimento de Sistemas do Instituto Municipal de Ensino Superior de Assis – IMESA e a Fundação Educacional do Município de Assis – FEMA, como requisito parcial à obtenção do Certificado de Conclusão.

**Orientando(a): Claudemir Rosa**

**Orientador(a): Prof. Douglas Sanches da Cunha**

**Assis/SP  
2021**

FICHA CATALOGRÁFICA

R788s ROSA, Claudemir  
Software para transferência de dados / Claudemir Rosa. –  
Assis, 2021.

50p.

Trabalho de conclusão do curso (Análise e Desenvolvimento  
de Sistemas). – Fundação Educacional do Município de Assis -  
FEMA

Orientador: Ms. Douglas Sanches da Cunha

1.FTP 2.Nuvem

CDD004.65

# SOFTWARE PARA TRANSFERÊNCIA DE DADOS

**CLAUDEMIR ROSA**

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, avaliado pela seguinte comissão examinadora:

**Orientador:**

---

Prof. Douglas Sanches da Cunha

**Examinador:**

---

Ms. Diomara Martins Reigato Barros

**Assis/SP  
2021**

## DEDICATÓRIA

Dedico este trabalho primeiramente a Deus que está sempre ao meu lado dando forças para continuar apesar de todas as dificuldades, a minha esposa Tanieli Dalsoto da Fonseca ao meu filho mais velho Pablo Vinicius que sempre me apoiaram, para que pudéssemos ter uma vida melhor. Da mesma forma dedico-o para meu amigo, professor e orientador Douglas por confiar no meu potencial, e por colocar novas ideias e agregar o meu conhecimento.

## **AGRADECIMENTOS**

Agradeço a Deus, por sempre estar comigo nessa jornada da vida, pela minha saúde concedida, tanto física e psicológica e poder trabalhar e estudar. Por toda as experiencias vividas sejam ela boas ou ruins, pois fazem parte da vida humana.

Agradeço a minha família em especial minha esposa Tanieli Dalsoto da Fonseca, por cuidar de nós, principalmente do meu filho mais novo, ao meu filho Pablo Vinicius, que ficou triste quando tranquei a faculdade, e me disse que ainda voltaria para finalizar, pois aqui estou, obrigado meu filho por acreditar em seu Pai.

Agradeço aos meus amigos, não somente do trabalho ou faculdade, mas todos que de alguma forma contribuíram pela minha formação.

Agradeço ao meu amigo e orientador Prof. Douglas, por me aceitar como orientando e acreditar em meu potencial, por sempre colocar novas ideias e agregar meu conhecimento e curiosidades nesse mundo de tecnologia.

Agradeço a você que aqui está, lendo meu trabalho.

“A melhor forma de prever o futuro é cria-lo.”

Alan Kay

## RESUMO

Com o grande crescimento de volume de dados na rede de 'internet' e empresas corporativas, de alguma forma precisam transferir e integrar dados importantes entre matrizes, filiais e clientes. Tornou-se, um desafio desenvolver seu próprio 'software' para transferência de dados, devido às tecnologias disponíveis no mercado não atenderem de forma adequada e os custos serem elevados.

Neste trabalho é apresentado, um sistema de controle para integração e transferência de dados, possibilitando a automatização de serviços oferecidos pelo Linux, para atender essas demandas corporativas.

**Palavras-chave:** FTP; HTTP; Escalabilidade; Infraestrutura; Storage; Nuvem; TLS.



## **ABSTRACT**

With the great growth of data volume on the internet network and corporate companies, which somehow need to transfer important data between headquarters, branches and customers. It has become a challenge to develop its own software for data transfer, due to the technologies available on the market not serving adequately and the costs being high.

In this graduation work, a control system for data transfer was presented, enabling the automation of services offered by Linux, to supply these corporate demands.

**Keywords:** FTP; HTTPS; Escalabilidade; Infraestrutura; Storage; Nuvem; SSL.

## LISTA DE FIGURAS E ILUSTRAÇÕES

<b>Figura 1</b> - Mapa Mental.....	18
<b>Figura 2</b> - Fluxo FTP .....	20
<b>Figura 3</b> - Fluxo FTP (Ativo).....	21
<b>Figura 4</b> - Fluxo FTP (Passivo).....	22
<b>Figura 5</b> - Rack Cabeamento estruturado .....	23
<b>Figura 6</b> - Storage de Armazenamento de Dados.....	23
<b>Figura 7</b> - Serviço de Nuvem .....	24
<b>Figura 8</b> - Caso de Uso Geral.....	27
<b>Figura 9</b> - Diagrama de Caso de Uso - Fazer Login .....	28
<b>Figura 10</b> - Diagrama de Caso de Uso - Cadastrar Usuário FTP.....	29
<b>Figura 11</b> - Diagrama de Caso de uso - Enviar e-mail aviso de remoção.....	30
<b>Figura 12</b> - Diagrama de Classes .....	31
<b>Figura 13</b> - Diagrama de Atividades .....	33
<b>Figura 14</b> - Arquitetura do Projeto.....	36
<b>Figura 15</b> - SSH_CONNECT (Criar usuários).....	37
<b>Figura 16</b> - SSH_CONNECT (Excluir usuário) .....	38
<b>Figura 17</b> - SSH2_CONNECT (Alterar Permissão).....	39
<b>Figura 18</b> - Cadastro de Usuário ao Banco de Dados.....	40
<b>Figura 19</b> - Gerador de senhas JavaScript.....	41
<b>Figura 20</b> - Tela de Login .....	42
<b>Figura 21</b> - Tela Principal .....	43
<b>Figura 22</b> - Cadastro Usuário FTP.....	44
<b>Figura 23</b> - Gerenciar Usuários FTP .....	45
<b>Figura 24</b> - Download de arquivos.....	46
<b>Figura 25</b> - Regras Firewall FTP .....	47

## LISTA DE SIGLAS

CLI	Command Line Interface
FTP	File Transfer Protocol
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
IDE	Integrated Development Environment
ISCSI	Internet Small Computer System Interface
PDC	Primary Domain Controller
PDO	PHP Data Object
PHP	Personal Home Page
SFP	Small Form Pluggable
TLS	Transport Layer Security
VPN	Virtual Private Network

# SUMÁRIO

<b>1. INTRODUÇÃO .....</b>	<b>14</b>
1.1 OBJETIVOS .....	15
1.2 JUSTIFICATIVA .....	16
1.3 MOTIVAÇÃO .....	16
1.4 PERSPECTIVAS DE CONTRIBUIÇÃO .....	17
<b>2. METODOLOGIA DO DESENVOLVIMENTO .....</b>	<b>18</b>
2.1 CONCEITO PHP .....	19
2.2 CONCEITO UML .....	19
2.3 CONCEITO APACHE .....	19
2.4 CONCEITO FTP .....	20
2.2.1 SERVIDOR ATIVO X PASSIVO .....	21
2.2.3 SERVIDOR PASSIVO X ATIVO .....	21
2.2.4 CONCEITO SHELL SCRIPT .....	22
<b>3. LEVANTAMENTO E ANÁLISE DE REQUISITOS .....</b>	<b>23</b>
3.1 DETALHE DO PROBLEMA A SER RESOLVIDO .....	24
3.2 RESULTADOS ESPERADOS PÓS IMPLEMENTAÇÃO DO SOFTWARE .....	25
3.3 LEVANTAMENTOS DOS REQUISITOS .....	25
3.4 FUNÇÕES SOLICITADAS .....	25
<b>4. PROPOSTA DE MODELAGEM DO ESTUDO DE CASO .....</b>	<b>27</b>
4.1 DIAGRAMA DE CASO DE USO GERAL .....	27
4.2 DIAGRAMA DE CLASSES .....	31
4.3 DIAGRAMA DE ATIVIDADES .....	33
<b>5. VALIDAÇÃO .....</b>	<b>34</b>
5.1 RECURSOS NECESSÁRIOS PARA DESENVOLVER O SISTEMA .....	34
<b>6. DESENVOLVIMENTO .....</b>	<b>36</b>
6.1 DESENVOLVIMENTO BACK-END .....	37
6.2 CADASTRAR USUÁRIO FTP .....	37
6.3 EXCLUIR USUÁRIO FTP .....	38
6.4 ALTERAR PERMISSÃO PARA USUÁRIO FTP .....	39
6.5 CADASTRO DE USUÁRIO AO BANCO DE DADOS .....	40
6.6 DESENVOLVIMENTO FRONT-END .....	40
<b>7. INTERFACE GRÁFICA .....</b>	<b>42</b>
7.1 TELA DE LOGIN .....	42
7.2 TELA PRINCIPAL .....	43
7.3 TELA CADASTRO DE USUÁRIO FTP .....	44

7.4 TELA GERENCIAR USUÁRIOS FTP .....	45
7.5 TELA DE DOWNLOAD POR HTTP .....	46
<b>8. CONSIDERAÇÕES FINAIS .....</b>	<b>47</b>
8.1 CONFIGURAÇÕES DO FIREWALL.....	47
<b>9. TRABALHOS FUTUROS.....</b>	<b>49</b>
<b>REFERÊNCIAS.....</b>	<b>50</b>

## 1. INTRODUÇÃO

Atualmente, os volumes de dados transferidos na rede de 'internet', têm se tornado cada vez maior, esses volumes cresceram muito em uso pessoais e corporativos com serviços de aplicativos como: *WhatsApp, YouTube, Instagram, Tiktok, Facebook, etc.*

Com essa grande demanda de transferência de dados, também surgiram outros serviços em nuvem, para esses fins, não pensando somente em transferir dados, mas também em vender serviços de backup para usuários comuns e corporativos, serviços em nuvem oferecidos por empresas de tecnologia como Microsoft (*Onedrive*), Google (*Google Drive*), *DropBox, WeTransfer* entre outros.

Apesar de todas essas tecnologias que surgiram, esses serviços não atendem certas empresas com volume de dados muito grande, empresas que precisam transferir dados acima de Terabytes de forma segura e rápida.

Uma rede corporativa, nesse nível de tráfego de dados, na grande maioria, possui cabeamentos estruturados *Category 6 Cable (CAT6)*, com Data Center, equipados com *Storages, Racks* para organizar cabeamento, *Switch* dedicados, ligações 1Gbs entre hosts e servidores de dados, e 10Gbs entre servidores e Storages.

Com toda essa estrutura, ainda é um desafio quando se fala em transferência de dados em empresas desse porte, já que as transferências de dados vão além do suporte físico, envolvem outras tecnologias baseada em software, como os mencionados acima.

Todos esses serviços atuais oferecidos por grandes empresas nesse cenário até aqui mencionado ainda é um problema, os dados de uma empresa nesse nível em uma situação prática teriam que contratar um plano com uma quantidade muito grande de espaço na nuvem, além de existir um tempo de enviar os dados para nuvem e só depois ficar disponível ao cliente.

Como os volumes de dados são grandes, isso é inviável, pois algumas empresas participam de processos de licitações, e no edital se exige que tenha local adequado para armazenar os dados, no caso *Storages*, e que sua transferência seja feita de forma segura, e em alguns casos não permitem enviar os dados para serviços em nuvem, já que não se sabe em quantos servidores na nuvem esses dados podem ficar armazenados e não se conhece

quem tem acesso a essa informação, alguns usuários de serviços em nuvem tiveram seus dados vazados pela internet.

De acordo com o site CNNBrasil (2021), o mundo já registra 4,6 bilhões de dados vazados, os números relacionados a dados vazados na internet, mostra o tamanho da dimensão em lidar com a segurança da informação.

Analisando todos os fatos acima, surgiu a necessidade do desenvolvimento de um software para transferência de dados, um software exclusivo para a empresa, em um cenário onde os dados ficam hospedados na própria empresa, um software, web para cadastro de clientes, controle de acesso à (downloads) e uploads.

Com os dados no próprio servidor da empresa, tornaria os processos mais prático, evitaria o tempo com upload para nuvem, os dados já ficam disponíveis para download mediante o serviço FTP e HTTP. Cada cliente cadastrado no sistema acessa somente as informações com permissão, e com tempo determinado para o acesso. Com essa estrutura montada a sua escalabilidade seria prática e viável e com custo menor em relação aos serviços de nuvem oferecidos por empresas terceiras.

O protocolo de transferência de dados *File Transfer Protocol* (FTP), talvez seja a melhor opção, por ser prático, rápido e de fácil configuração. Para que o acesso aos dados seja feito de forma segura é possível adicionar *Transport Layer Security* (TLS) uma camada de segurança ao protocolo FTP com encriptação, (Morimoto, 2008, p.425).

## 1.1 OBJETIVOS

Este projeto tem por objetivo geral desenvolver um software com ferramentas livres, para atender uma demanda de transferência de dados, entre matriz, filiais e clientes. Um software que possa ser gerenciado por uma interface Web, onde o usuário do sistema possa ter um total controle de usuários que possam acessar e transferir dados.

São objetivos específicos, automatizar através de tecnologia de linguagem de programação, um serviço de transferência de dados no Linux, o protocolo de transferência de arquivos (FTP) e (HTTP). Implementar os módulos, cadastrar clientes, cadastrar

gerentes, cadastrar arquivos, gerenciar espaços em disco, gerar relatórios, gerar tempo de retenção dos dados.

## 1.2 JUSTIFICATIVA

Espera-se que este 'software', possa contribuir com a empresa, facilitando o controle de usuários (FTP), tirando a carga do administrador de redes, que exerce essa função através de linha de comando.

Os serviços oferecidos pela nuvem, mostraram um custo elevado e com algumas deficiências: compactar dados antes do envio, enviar para depois ficar disponível.

O desenvolvimento do 'software' para transferência de dados, seria uma solução para empresa e serviria como base para novos projetos internos, e colocaria como base (fonte de estudos) para novos projetos.

## 1.3 MOTIVAÇÃO

A motivação para produção desse trabalho surgiu da necessidade de automatizar serviço voltados a infraestrutura, o quanto poderia aprender no seu desenvolvimento, quais seriam as etapas a tomar, o que fazer primeiro, como aplicar certas situações teóricas e práticas.

Mas além de tudo conseguir estudar de forma visual outras plataformas que ofereciam os serviços de hospedagem em nuvem e ver o que poderia ser implementado.

A principal motivação, era criar um usuário no Linux, um usuário do sistema, pois continuamente era feito de forma CLI via terminal (Shell), através de script que, até então era uma função manual e, claro como seria feito esse sincronismo de integridades dos dados hospedados. A necessidade surgiu e, a emoção de aprender uma linguagem 'web' e colocar em prática todos os recursos até então feitos em scripts via terminal, se tornou muito motivante, pois era algo novo, algo que ajudaria em muito sanar dúvidas em certos comportamentos e eventos, e de que forma eles eram obedecidos por uma linguagem de



alto nível, exemplo: como subir de forma gráfica os dados para servidor e de que forma o usuário poderia estar fazendo as transferências desses dados, enfim, eram muitas as motivações.

#### 1.4 PERSPECTIVAS DE CONTRIBUIÇÃO

Acredito que a principal contribuição do estudo aqui proposto seja o fato de poder manipular serviços rodando na plataforma Linux, de forma segura e com uma linguagem de programação de alto nível, pois essa foi a grande dificuldade de início no projeto, em saber como manipular o serviço FTP via 'web'.

Com esse estudo aqui mencionado nesse projeto de desenvolvimento de um 'software' para transferência de dados, e mais as experiências já adquiridas pela longa jornada como administrador de redes, principalmente com os conhecimentos dos serviços oferecidos pelo Linux, acredito que esse projeto pode servir de apoio a novos projetos de infraestrutura voltados não somente na transferência de dados, mas também em diversos projetos que envolva outras situações como Firewalls, controle administrativos para serviços como o Samba(Serviço de Compartilhamento de arquivos), VPNs, PDCs etc.

Espera-se que de alguma forma possa ajudar outros alunos ou curiosos, a entender como se inicia uma manipulação via 'web' com servidores (Linux), seguindo o sentimento de que a programação vai além de uma 'interface' gráfica e um banco de dados, compreender que pode manipular um sistema operacional, criando usuários reais, definir senhas, permissões de acesso a dados, entre outras coisas que vão surgindo no mundo da criatividade.

## 2. METODOLOGIA DO DESENVOLVIMENTO

Nesse capítulo, serão apresentados alguns conceitos de tecnologias, que serão utilizados no projeto. Para atender aos objetivos apresentados no presente trabalho, serão conduzidos estudos teóricos para dar suporte ao projeto.

Um mapa mental foi elaborado para melhor entendimento dos requisitos do ‘software’, como mostra a figura abaixo.

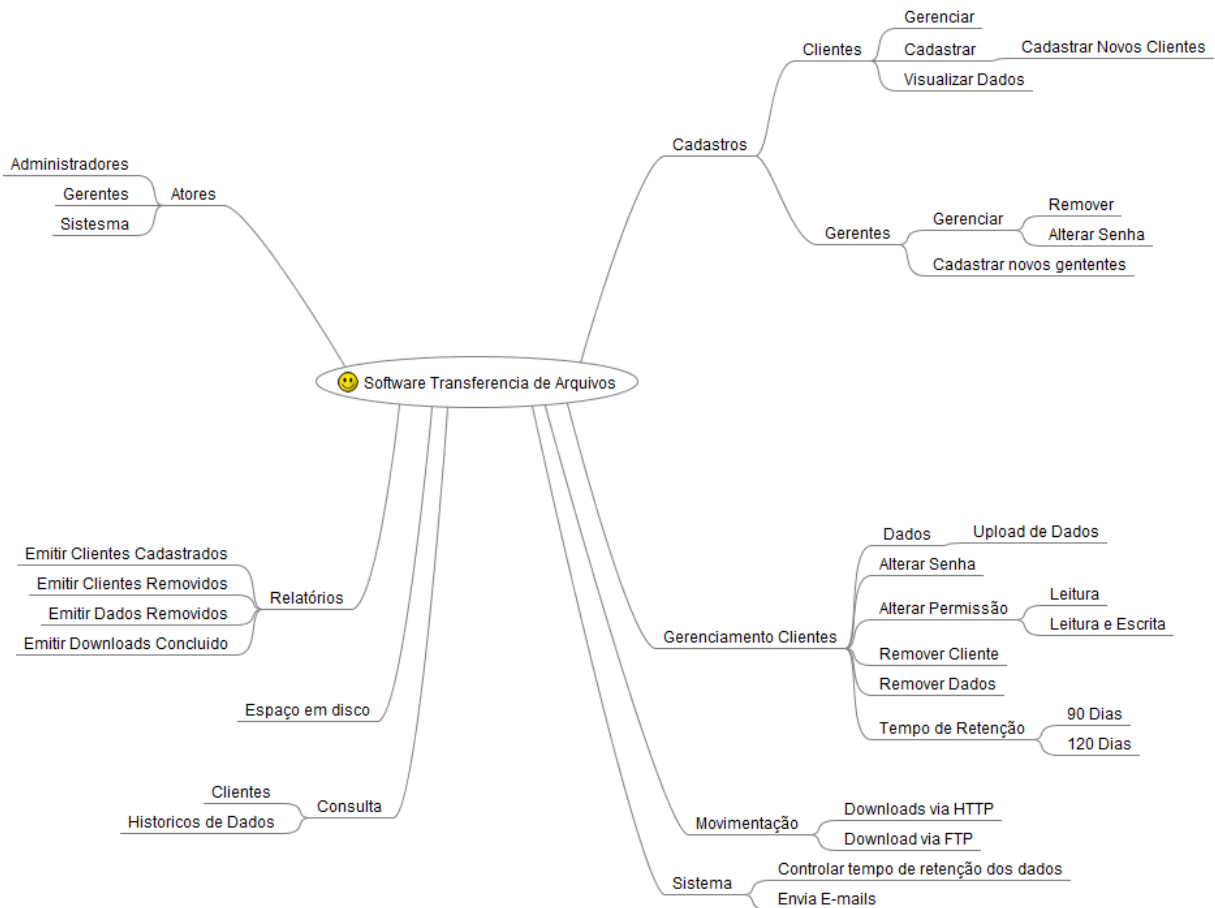


Figura 1 - Mapa Mental

## 2.1 CONCEITO PHP

PHP uma linguagem de programação, voltada para desenvolvimento, 'web', uma linguagem com baixa curva de aprendizado. O próprio Facebook, teve suas primeiras páginas desenvolvidas em PHP.

De acordo com o site [https://www.php.net/manual/pt\\_BR/intro-whatism.php](https://www.php.net/manual/pt_BR/intro-whatism.php) (2020)

O PHP (um acrônimo recursivo para PHP: Hypertext Preprocessor) é uma linguagem de script open source de uso geral, muito utilizada, e especialmente adequada para o desenvolvimento web e que pode ser embutida dentro do HTML.

A melhor coisa em usar o PHP é que ele é extremamente simples para um iniciante, mas oferece muitos recursos avançados para um programador profissional. Não tenha medo de ler a longa lista de recursos do PHP. Pode entrar com tudo, o mais rápido que puder, e começar a escrever scripts simples em poucas horas.

## 2.2 CONCEITO UML

A UML, de acordo com Guedes (2009), é denominada de Linguagem de Modelagem Unificada. É uma linguagem visual utilizada para modelar 'software' baseados no paradigma de orientação a objetos. Ela é uma linguagem de modelagem de propósito geral que pode ser aplicada a todos os domínios de aplicações. Deve ficar bem claro, porém, que a UML não é uma linguagem de programação e sim uma linguagem de comunicação, cujo objetivo é auxiliar os engenheiros de 'software' durante a construção do projeto do sistema. Os diagramas da linguagem estão divididos nos vários tipos: Use Case, Classes, Sequência, Atividades, Objeto, Componentes, Implantação, Pacote, Estrutura e Interação.

## 2.3 CONCEITO APACHE

O serviço Apache ou servidor Apache, é responsável por disponibilizar todo o acesso feito pelos internautas, para ser mais exato, o Apache processa todo o código PHP e devolve em forma de HTML ao usuário, apesar de se dizer "Servidor Apache", que fique claro que não é um servidor físico e sim um Software, de acordo com o site [apache.org](http://apache.org) (2021), o

Apache HTTP Server Project é um esforço para desenvolver e manter um servidor HTTP de código aberto para sistemas operacionais modernos, incluindo UNIX e Windows. O objetivo deste projeto é fornecer um servidor seguro, eficiente e extensível que forneça serviços HTTP em sincronia com os padrões HTTP atuais.

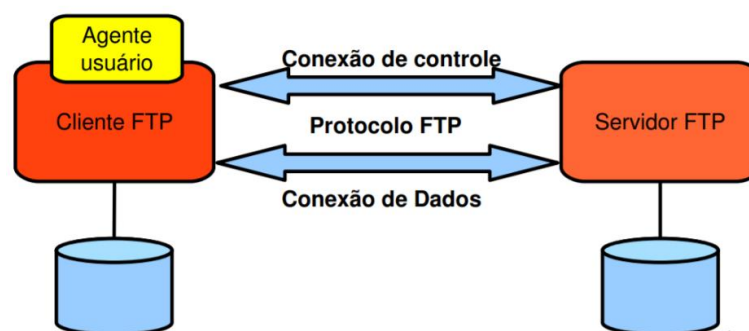
O Apache HTTP Server ("httpd") foi lançado em 1995 e tem sido o servidor web mais popular na Internet desde abril de 1996. Ele comemorou seu 25º aniversário como um projeto em fevereiro de 2020.

## 2.4 CONCEITO FTP

O FTP protocolo de transferência de arquivos (File Transfer Protocol) funciona da seguinte forma, o servidor é o responsável por hospedar os arquivos, os clientes serão os usuários cadastrados no Linux que farão o acesso. A conexão é acessada através de um IP público ou interno, com solicitação do nome do usuário e senha, pois cada cliente tem seu próprio local no servidor, e nenhum cliente enxerga os dados de outros clientes, ficando somente o seu local de acesso disponível. (Morimoto, 2008, p.425).

O serviço não tem limite para transferência de arquivos ou tipo de extensão, sem necessidade de compactar dados para envio. Os dados podem, serem enviados em formatos brutos, onde facilita ainda mais sua usabilidade, o FTP é um dos serviços mais utilizados em provedores de site na 'internet'.

Na figura 2, observa-se os fluxos de trabalho do serviço FTP, o cliente externo faz um acesso ao servidor FTP, através desse protocolo os dados são transferidos.



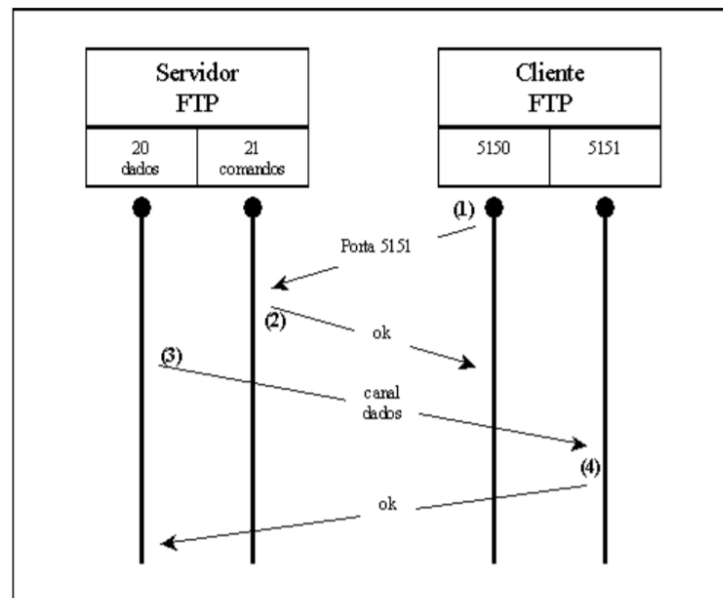
**Figura 2** - Fluxo FTP

Fonte: <http://diatinf.ifrn.edu.br/>

## 2.2.1 SERVIDOR ATIVO X PASSIVO

Ativo, quando o cliente acessa o serviço de FTP na (porta 21) de um servidor, ele indica para qual porta o servidor deverá lhe enviar os dados. Deste modo, quando o download/upload começar, o servidor enviará os dados (partindo da porta 20) para essa porta especificada pelo cliente. (Prof. Pontes, 2009).

### Ativo



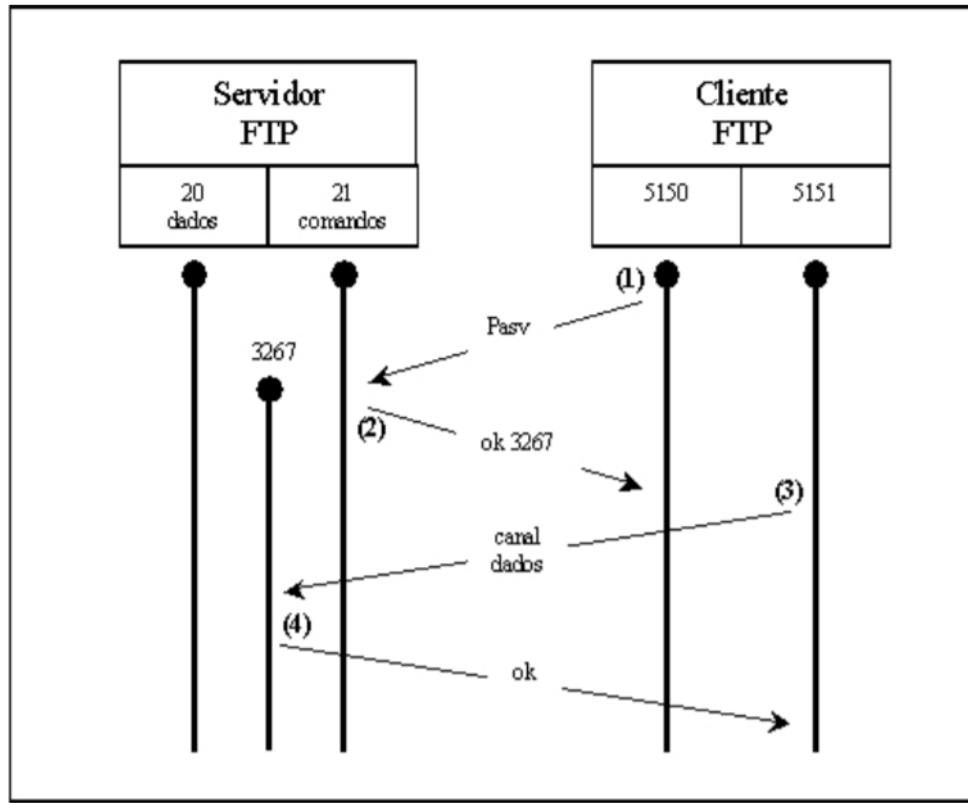
**Figura 3** - Fluxo FTP (Ativo)

Fonte: <http://diatinf.ifrn.edu.br/>

## 2.2.3 SERVIDOR PASSIVO X ATIVO

Passivo, o servidor no início da conexão informa ao cliente por qual de suas portas TCP os dados serão enviados. Quando o cliente acessa o serviço de FTP (porta 21) de um servidor, a conexão já é estabelecida como "PASV" (passiva). (Prof. Pontes, 2009).

# Passivo



**Figura 4** - Fluxo FTP (Passivo)

Fonte: <http://diatinf.ifrn.edu.br/>

## 2.2.4 CONCEITO SHELL SCRIPT

De acordo com Jargas (2008, p.24):

A palavra script também é usada em português para referenciar o roteiro de uma peça de teatro ou produção televisiva. Esse script é o caminho que os atores devem seguir, a descrição de todo o show do início até o fim.

Em uma programação Shell Script não é muito diferente disso. É uma lista de comandos para serem executados em sequência (um comando após o outro). Um roteiro predefinido de comandos e parâmetros. Geralmente o que está no script é aquilo que o usuário digitaria na linha de comando. Realmente, apenas colocando-se todo o histórico da linha de comando em um arquivo, tem-se um script.

### 3. LEVANTAMENTO E ANÁLISE DE REQUISITOS

A empresa possui em seu ambiente de trabalho uma rede estruturada de computadores, possibilitando uma qualidade de tráfego em GigaEthernet e armazenamento de dados volumosos em Storages Dell.



**Figura 5** - Rack Cabeamento estruturado

Fonte: Autoria própria

Storages de Armazenamento Dell.



**Figura 6** - Storage de Armazenamento de Dados

Fonte: Autoria própria

Após a análise do ambiente de trabalho e a segmentação da rede corporativa necessários para partes do projeto. Nessa análise inicial de requisitos, foram coletadas informações importantes, como uma visão sistêmica sobre o ambiente. Saber de que forma a empresa usava seus recursos de TI para transferir dados, e quais equipamentos estavam disponíveis, para integrar ao novo projeto.

### 3.1 DETALHE DO PROBLEMA A SER RESOLVIDO

Com a alta demanda na transferência de dados em ambientes corporativos, ferramentas até então disponíveis e avaliadas no mercado entre os anos de 2016-2020, dependendo da necessidade da empresa se torna um custo financeiro elevado no quesito transferir dados em massa.



**Figura 7** - Serviço de Nuvem

Considerando 3 valores financeiros nesse cenário, que precisam ser avaliados:

- ✓ Valor para o tamanho do armazenamento.
- ✓ Valor da velocidade em relação a Infraestrutura.
- ✓ Valor do profissional (Certificado) ou empresa credenciada parceira das plataformas de Nuvem, que serão necessários para configuração e suporte do ambiente em nuvem.

Serviço como FTP, atende a demanda até certo ponto, pois exige uma mão de obra especializada para controle do sistema FTP, sendo necessário um administrador de redes Linux.



O serviço FTP, fica vinculado, apenas ao controle do administrador de redes, cada solicitação feita através dos gerentes, em fazer novo cadastro de clientes, ou filiais, onde os mesmos acessam informações nos servidores, FTP da empresa.

A distância também é um problema, caso um dos responsáveis pelo projeto esteja viajando a trabalho pela empresa, e em umas dessas viagens, surge a necessidade de fazer um novo cadastro, ou apenas uma simples mudança de permissão de acesso aos dados, precisaria do administrador de redes para cumprir com essa tarefa.

### 3.2 RESULTADOS ESPERADOS PÓS IMPLEMENTAÇÃO DO SOFTWARE

Espera-se de início atender os requisitos básicos e as principais funcionalidades do Software. Com um maior controle de contas cadastradas, sendo versátil em usar o sistema, ficando acessível aos gerentes responsáveis pelos projetos, onde os mesmos possam fazer através de uma 'interface' gráfica, o que um administrador de rede fazia via terminal de comando ainda com mais recursos auxiliares para maior poder de consultas via relatórios, dentre as funções.

### 3.3 LEVANTAMENTOS DOS REQUISITOS

Nessa etapa, uma análise qualitativa foi aplicada nos colaboradores que iriam utilizar o software, essa análise tinha como objetivo levantar os seguintes resultados.

### 3.4 FUNCÕES SOLICITADAS

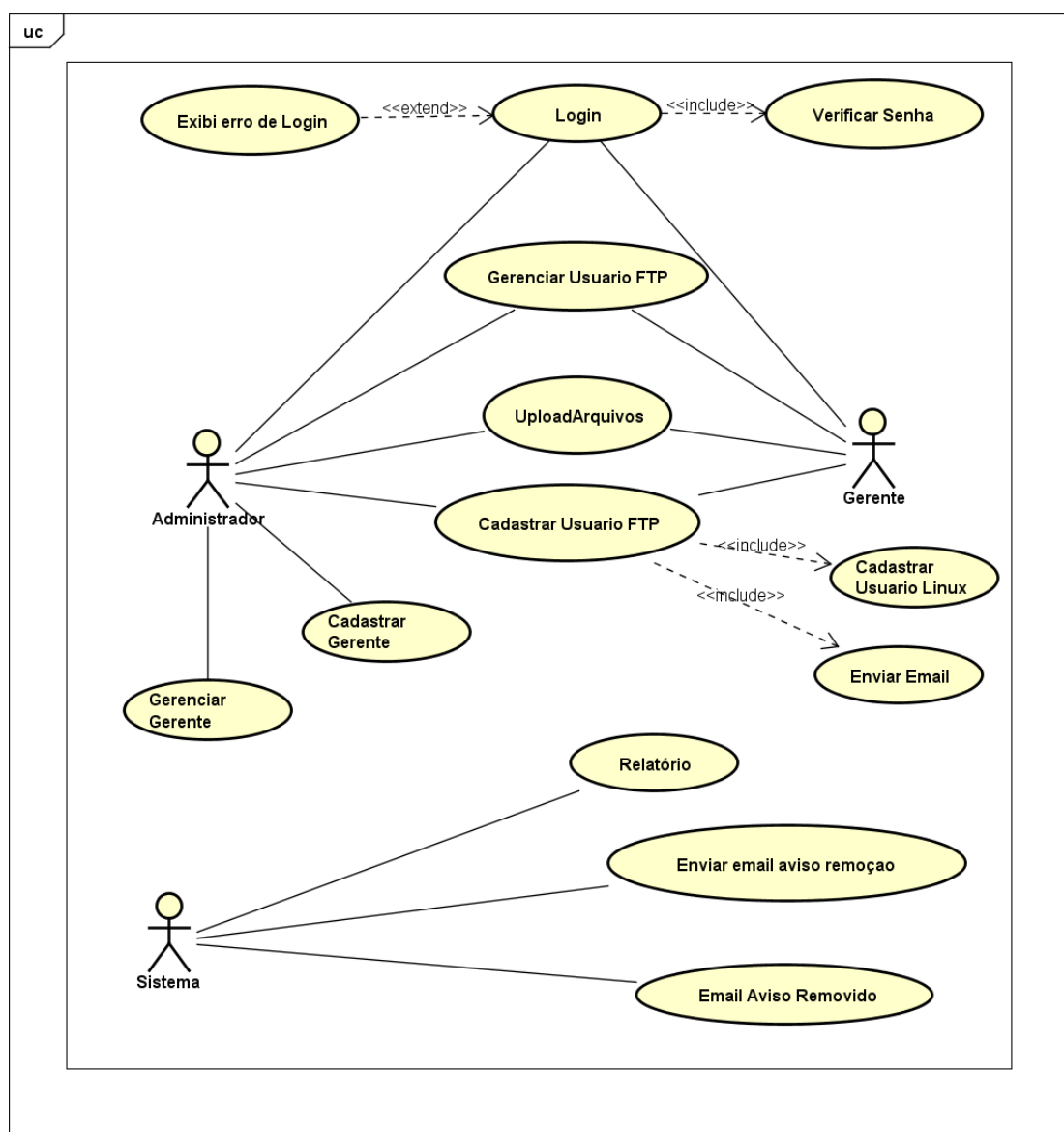
- Tela de Login
- Cadastro de usuários FTP
- Cadastro de dados para download HTTP
- Cadastro Gerentes
- Cadastro de Administradores
- Espaço em disco no servidor

- Relatórios de usuários cadastrados, removidos, dados removidos
- Gerenciamento dos usuários, alterar permissão, alterar senha, remover dados, excluir usuários, tempo de retenção dos dados.
- Gerenciamento de gerentes, cadastro de gerentes, alterar senha, excluir gerentes.
- Gerentes cria usuários, administrador cria gerentes
- Upload e Download de dados, via FTP e HTTP
- Envio de e-mails para notificação de cadastro de usuário FTP
- Envio de e-mails para notificação de cadastro de dados HTTP
- Envio de e-mail para notificação que usuário será removido em breve e os dados serão apagados
- Envio de e-mail para notificação de remoção de usuário FTP

## 4. PROPOSTA DE MODELAGEM DO ESTUDO DE CASO

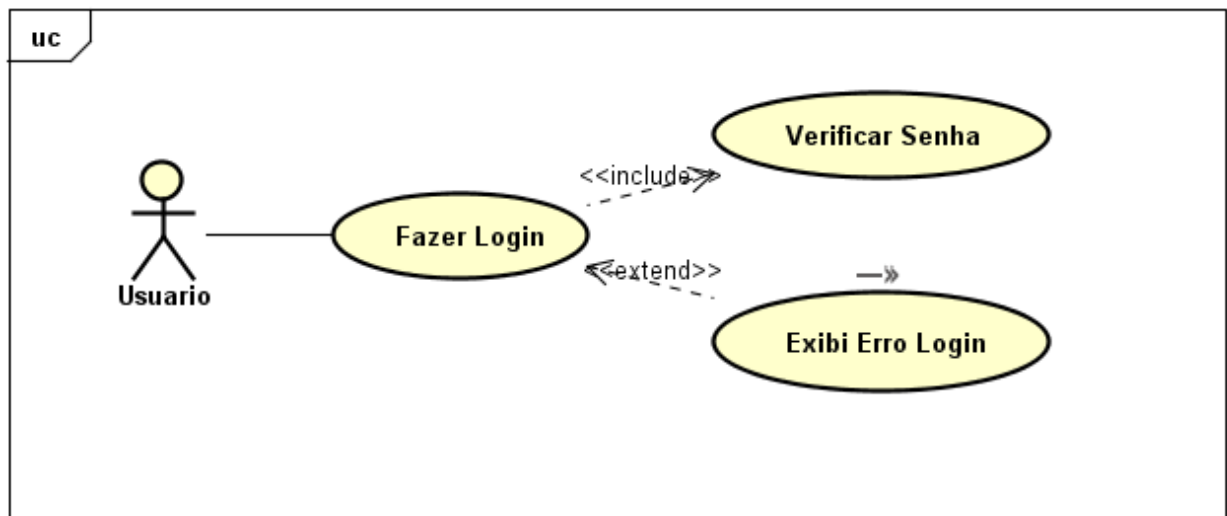
O diagrama de caso de uso é construído logo no início do projeto, pois pode sofrer alterações no decorrer do projeto. Com o diagrama de caso de uso, fica mais claro as funcionalidades do sistema, isso é compreensível até por pessoas leigas no assunto, apenas observando, como ilustra a figura 8.

### 4.1 DIAGRAMA DE CASO DE USO GERAL



powered by Astah

Figura 8 - Caso de Uso Geral

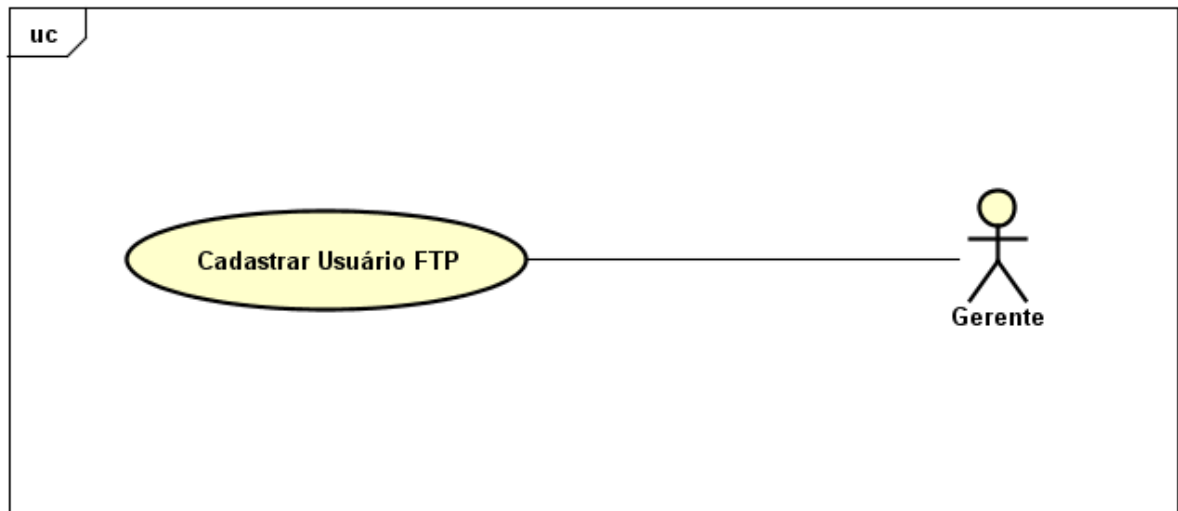


**Figura 9** - Diagrama de Caso de Uso - Fazer Login

- 1 **Finalidade / Objetivo:** o usuário realiza acesso ao sistema
- 2 **Ator:** Usuário
- 3 **Evento Inicial:** O ator insere as informações de acesso para entrar no sistema
- 4 **Fluxo Principal:**
  - a. O sistema oferece uma interface gráfica para o usuário inserir as informações de acesso;
  - b. O usuário informa seus dados para acessar;
  - c. O usuário confirma suas informações e seleciona a opção “Confirmar Acesso” (A1).
- 5 **Fluxo alternativo:**

**A1 – O ator Usuário não tem cadastro:**

  - a. O sistema exibe que não existe cadastro para o usuário inserido;
  - b. O sistema exibe a interface gráfica de acesso.



**Figura 10** - Diagrama de Caso de Uso - Cadastrar Usuário FTP

- 1 **Finalidade / Objetivo:** O gerente cadastra usuário FTP
- 2 **Ator:** Gerente.
- 3 **Evento Inicial:** O gerente inicia a funcionalidade pressionando o botão cadastrar usuário FTP;
- 4 **Fluxo Principal:**
  - a. O sistema oferece uma interface gráfica para iniciar a inserção (A1);
  - b. O gerente insere nome do usuário FTP;
  - c. O gerente gera a senha de acesso;
  - d. O gerente seleciona as permissões de acesso;
  - e. O gerente seleciona o tempo para retenção dos dados;
  - f. O gerente seleciona o disco que os dados ficarão hospedados;
  - g. O gerente preenche e-mail de envio com dados do usuário FTP (opcional);
  - h. O gerente pressiona no botão confirmar cadastro;
  - i. O sistema valida os dados (A2);
  - j. O sistema exibe a interface gráfica de cadastro de usuário FTP.

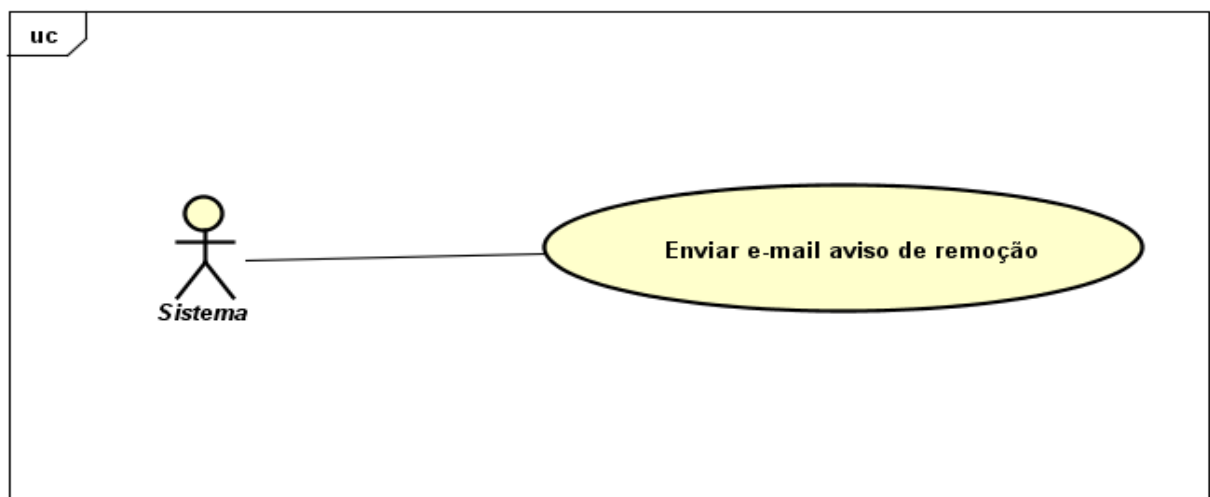
## 5 Fluxo Alternativo:

### A1 – O gerente cancela a operação:

- a. O gerente cancela a inserção;
- b. O sistema exibe a tela inicial de cadastro de usuário FTP.

### A2 – O gerente digita informações inválidas:

- a. O sistema informa ao gerente por meio de uma interface gráfica as informações inválidas;
- b. O sistema exibe a tela inicial de cadastro para ajustar as informações inválidas.



**Figura 11** - Diagrama de Caso de uso - Enviar e-mail aviso de remoção

- 1 **Finalidade / Objetivo:** O sistema e-mail notificando o gerente sobre breve remoção de usuário FTP
- 2 **Ator:** Gerente.
- 3 **Evento inicial:** O sistema faz uma varredura diária no banco de dados a procura de conta de usuários FTP prestes a vencer.
- 4 **Fluxo Principal:**
  - a. O sistema cole informações de uma ou mais conta de usuários FTP;
  - b. O sistema envia e-mail aos gerentes responsáveis, notificando em breve usuário FTP será removido.

## 4.2 DIAGRAMA DE CLASSES

O diagrama de classe tem como objetivo mapear de forma clara a estrutura de um determinado sistema ao modelar suas classes, seus atributos e relações entre objetos, pois de acordo com Tybel (2016), em programação, um diagrama de classes é uma representação da estrutura e relações das classes que servem de modelo para objetos. Podemos afirmar de maneira mais simples que seria um conjunto de objetos com as mesmas características, assim saberemos identificar objetos e agrupá-los, de forma a encontrar suas respectivas classes.

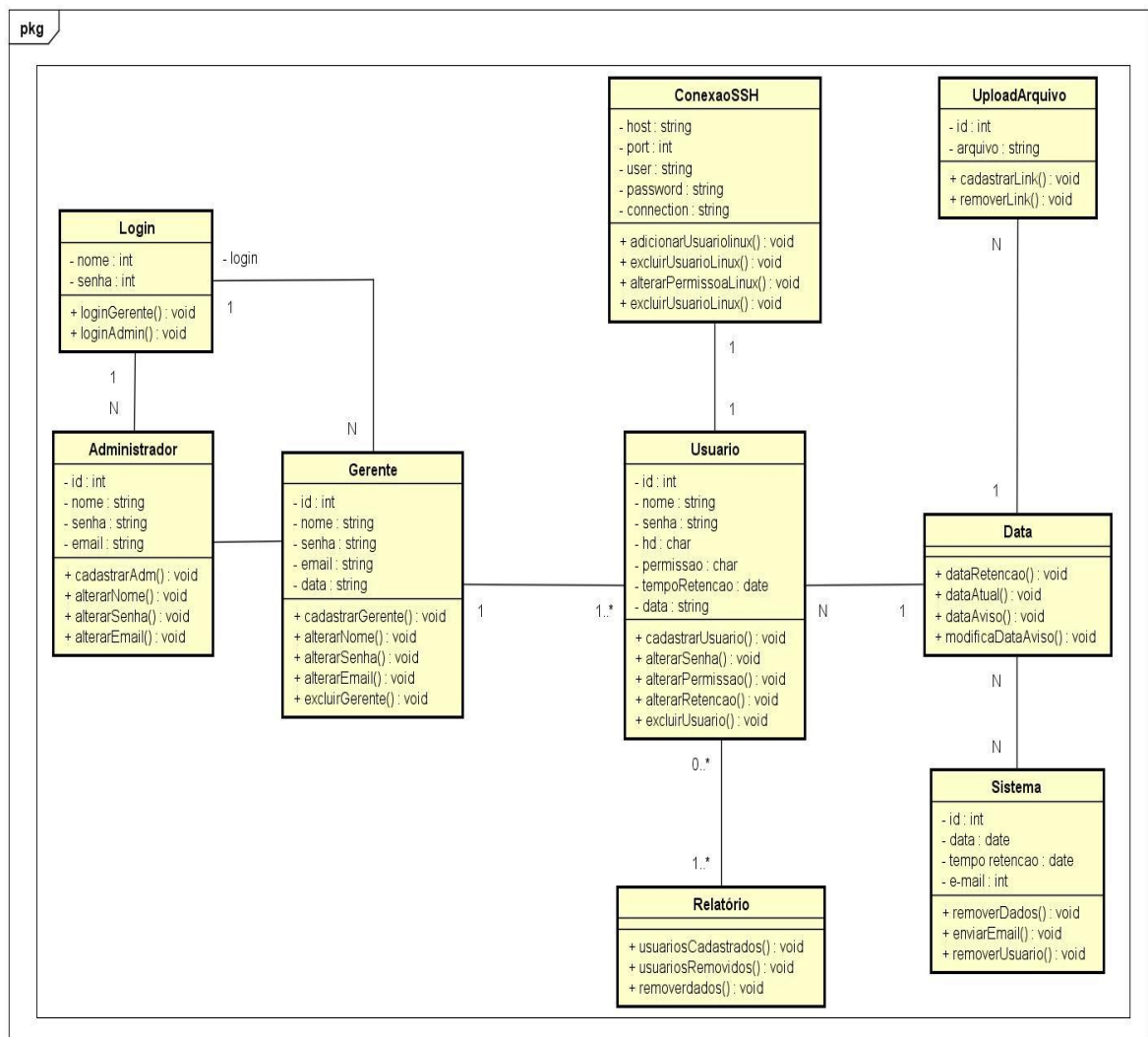


Figura 12 - Diagrama de Classes

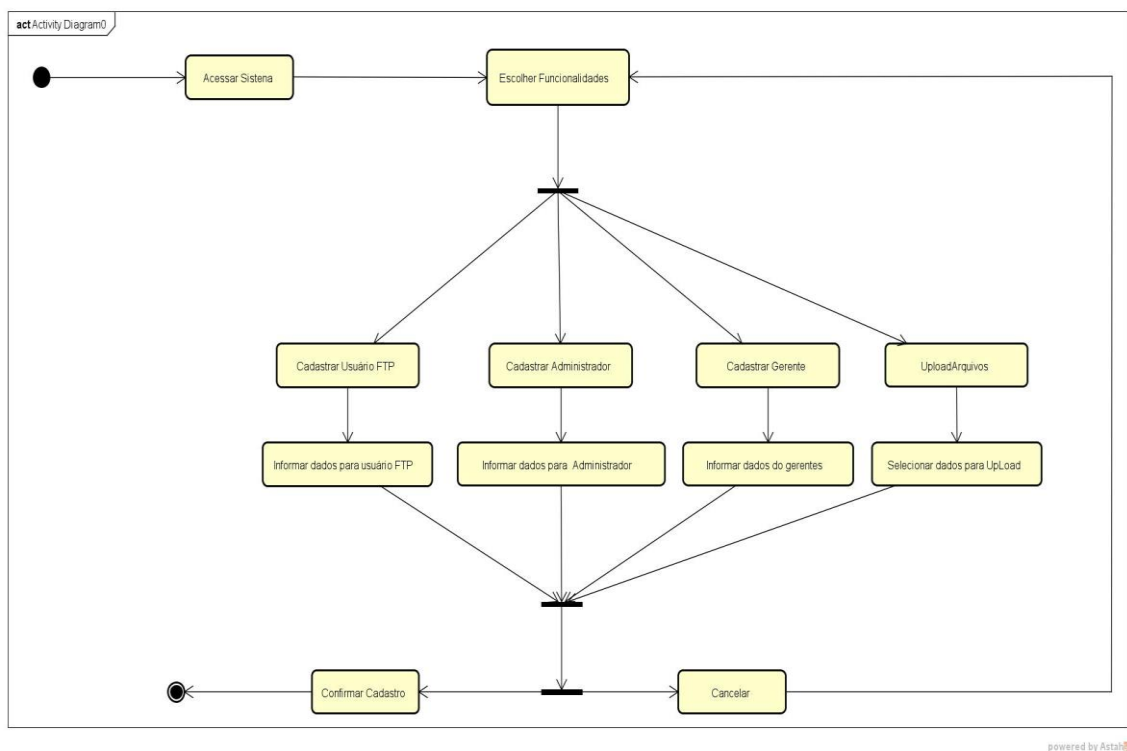
A LEITURA DA FIGURA ILUSTRA ANTERIOR É DA SEGUINTE FORMA:

- O sistema pode haver um administrador ou vários;
- O sistema pode haver um gerente ou vários;
- O sistema pode haver um usuário FTP ou vários;
- Para cada usuário FTP cadastrado no sistema pode haver apenas um gerente responsável pela conta;
- Cada gerente pode cadastrar vários usuários FTP;
- Cada usuário FTP, pode haver apenas uma data de cadastro;
- Cada usuário FTP, pode haver apenas uma data de aviso de retenção dos dados;
- Cada usuário FTP pode haver apenas um gerente;
- Cada upload de arquivos pode haver uma data de cadastro;
- Cada usuário FTP pode haver apenas uma conexão ao SSH;
- O sistema pode enviar vários e-mails;
- O sistema pode haver vários relatórios;



### 4.3 DIAGRAMA DE ATIVIDADES

Para apresentar de forma mais clara as etapas de execução e processos que são percorridos pelo sistema é apresentado um diagrama de atividades, segundo Lucidchart (2018), possui como finalidade principal unir desenvolvedores e pessoas da área de negócio a entender um determinado processo, ou seja, uma atividade.



**Figura 13 - Diagrama de Atividades**

## 5. VALIDAÇÃO

Nessa etapa, após as coletas dos requisitos para o desenvolvimento do sistema, uma análise dos recursos necessários foi validada, observa-se de forma mais clara os primeiros passos para o desenvolvimento.

### 5.1 RECURSOS NECESSÁRIOS PARA DESENVOLVER O SISTEMA

Nessa etapa, foram levantados os recursos necessários para desenvolver o sistema.

- **HARDWARE**

As configurações do servidor eram 8Gb de memória, processador Xeon de 8 núcleos com frequências 2.8Ghz, interface rede 1Gbs, de início com 4TB de armazenamento, esse foi as configurações do host para hospedagem dos dados e a aplicação.

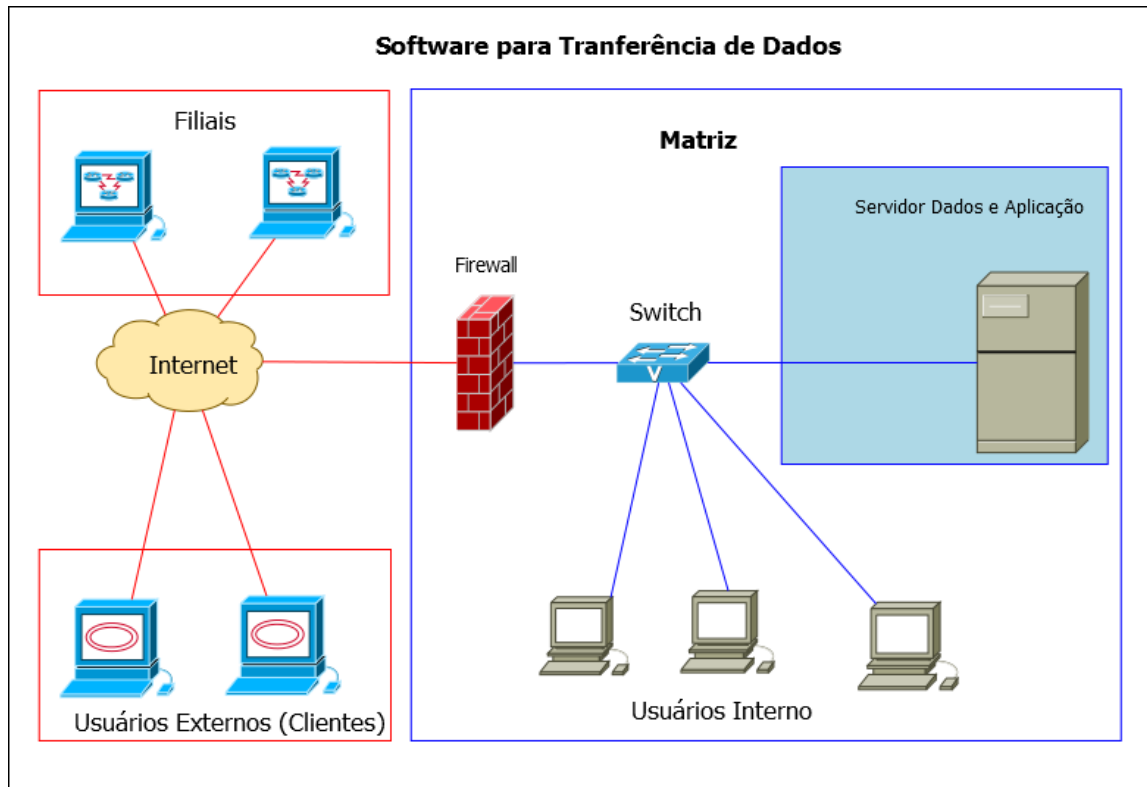
- **TÉCNOLOGIAS**

- ✓ PHP – Linguagem de programação que possibilita o desenvolvimento do software.
- ✓ HTML – BooStrap para manipulação do HTML e CSS, necessário para a criação de do FRON-END.
- ✓ Visual Code – IDE de programação, para manipulação e do código fonte
- ✓ MySQL – Bando de dados para o projeto proposto.
- ✓ Shell Script – Linguagem que possibilita a manipulação de comandos e códigos de programação no Bash do Linux
- ✓ PHPMyAdmin – Interface (GUI), para manipulação do Banco de Dados.
- ✓ JavaScript – Linguagem de programação que possibilita a manipulação de eventos no HTML.
- ✓ Apache – Servidor Web, o qual processará os scripts PHP, e hospedará o sistema.
- ✓ ProFTP – Servidor de transferência de Arquivos (Serviço que será automatizado pela interface Web, proposta no projeto.
- ✓ Debian10 – Sistema operacional do Servidor, o qual armazenará os dados e o sistema.

- ✓ Samba – Servidor de compartilhamento de arquivos, necessários para compartilhar as pastas do projeto via rede.
- ✓ BootStrap – FrameWork (HTML e CSS) que possibilita a integração de componentes visual prontos, acelerando o processo de desenvolvimento.
- ✓ Astah Community – possibilita a criação de todos os diagramas necessário para desenvolvimento e documentação do sistema.

## 6. DESENVOLVIMENTO

Nessa seção demonstra a arquitetura do projeto de sistema, o desenvolvimento do trabalho. Para um melhor entendimento do projeto, na figura 14 ilustra a arquitetura física do Software para Transferência de Dados.



**Figura 14** - Arquitetura do Projeto

Na figura 14, exibem-se a arquitetura do projeto, tendo em foco o servidor de Dados e Aplicação. A ilustração define o funcionamento físico do projeto, como pode ser observado, os usuários externos acessam os dados na Matriz, onde poderão fazer downloads (Baixar dados) e upload (Enviar dados).

Os usuários externos acessam os dados por protocolos FTP, usando como clientes desktop Filezilla, o protocolo FTP é usado quando a necessidades de transferência é volumosa e com dados brutos, sem necessidade de compactação. Para transferência pequenas, onde usuários internos hospeda dados apenas para download, os usuários externos acessam por serviço HTTP, direto via navegador sem a necessidade de um aplicativo cliente FTP.

## 6.1 DESENVOLVIMENTO BACK-END

O desenvolvimento do back-end foi feito em PHP, a escolha da linguagem de programação PHP facilitou a comunicação com o sistema operacional Linux, pelo uso de bibliotecas disponíveis para execução de comandos Shell.

Como serão mostradas nas figuras abaixo, alguns trechos do código do projeto, exibe o uso da biblioteca SSH2\_CONNECT, com essa biblioteca o PHP é capaz de acessar o sistema operacional Linux de forma segura e manipular comandos, nessa figura, mostra a criação de um usuário Linux pelo PHP utilizando o SSH2\_CONNECT.

## 6.2 CADASTRAR USUÁRIO FTP

```
public function adicionarUsuarioLinux ($usuario, $senha, $permissao, $hd){
    // Define local de montagem dos discos /home HD1, /FTP HD2
    $hd1 = '/home';
    $hd2 = '/FTP/FTP';

    if ($hd === 'hd1'){
        ssh2_exec($this->connection, "/usr/bin/sudo /usr/sbin/useradd -m -d $hd1/$usuario $usuario");
        ssh2_exec($this->connection, "echo $usuario:$senha | /usr/bin/sudo /usr/sbin/chpasswd");
        ssh2_exec($this->connection, "/usr/bin/sudo /bin/chmod -R 555 $hd1/$usuario");
        ssh2_exec($this->connection, "/usr/bin/sudo /usr/sbin/usermod -G ftp $usuario");
        ssh2_exec($this->connection, "/usr/bin/sudo /usr/bin/mkdir $hd1/$usuario/$usuario");
        ssh2_exec($this->connection, "/usr/bin/sudo /bin/chown -R $usuario.ftp $hd1/$usuario/$usuario");

        if($permissao === 'leitura'){
            ssh2_exec($this->connection, "/usr/bin/sudo /bin/chmod -R 575 $hd1/$usuario/$usuario");
        }
        elseif($permissao === 'escrita') {
            ssh2_exec($this->connection, "/usr/bin/sudo /bin/chmod -R 777 $hd1/$usuario/$usuario");
        }
    }
}
```

**Figura 15** - SSH\_CONNECT (Criar usuários)

Na figura 15, um trecho do código cadastrar usuário FTP é exibido, onde as informações chegam ao método adicionarUsuarioLinux, as variáveis recebidas por parâmetros são utilizadas de acordo com o IF, um teste é feito para saber em qual diretório o usuário será criado, após validação, a função ssh2\_exec(Pertencente a biblioteca SSH2\_CONNECT) é executada, e comandos shell são repassados ao Linux, então, um usuário é criado com suas devidas permissão de acesso.

### 6.3 EXCLUIR USUÁRIO FTP

```
public function excluirUsuarioLinux ($usuario, $hd){  
  
    $hd1 = '/home';  
    $hd2 = '/FTP/FTP';  
  
    if ($hd === 'hd1'){  
        ssh2_exec($this->connection, "/usr/bin/sudo /usr/sbin/userdel $usuario");  
        ssh2_exec($this->connection, "/usr/bin/sudo /bin/rm -R $hd1/$usuario");  
    }elseif ($hd === 'hd2'){  
        ssh2_exec($this->connection, "/usr/bin/sudo /usr/sbin/userdel $usuario");  
        ssh2_exec($this->connection, "/usr/bin/sudo /bin/rm -R $hd2/$usuario");  
    }  
  
}
```

Figura 16 - SSH\_CONNECT (Excluir usuário)

Na figura 16, uma remoção de usuário FTP é apresentada, o método `excluirUsuarioLinux` é chamado, a função `ssh2_exec` da biblioteca `SSH2_CONNECT` recebe novos comandos bash em seus parâmetros.

## 6.4 ALTERAR PERMISSÃO PARA USUÁRIO FTP

```
public function alterarPermissaoLinux($permissao, $usuario, $hd){  
  
    $hd1 = '/home';  
    $hd2 = '/FTP/FTP';  
  
    if($hd === 'hd1'){  
        if($permissao === 'leitura'){  
            ssh2_exec($this->connection, "/usr/bin/sudo /bin/chmod -R 575 $hd1/$usuario/$usuario");  
        }  
        elseif ($permissao === 'escrita') {  
            ssh2_exec($this->connection, "/usr/bin/sudo /bin/chmod -R 777 $hd1/$usuario/$usuario");  
        }  
    }elseif ($hd === 'hd2'){  
        if($permissao === 'leitura'){  
            ssh2_exec($this->connection, "/usr/bin/sudo /bin/chmod -R 575 $hd2/$usuario/$usuario");  
        }  
        elseif ($permissao === 'escrita') {  
            ssh2_exec($this->connection, "/usr/bin/sudo /bin/chmod -R 777 $hd2/$usuario/$usuario");  
        }  
    }  
}
```

**Figura 17** - SSH2\_CONNECT (Alterar Permissão)

Na figura 17, o processo feito pelo back-end em alterar permissão do usuário, da mesma forma dos anteriores, as informações chegam ao método `alterarPermissaoLinux`, os comandos bash são executados.

## 6.5 CADASTRO DE USUÁRIO AO BANCO DE DADOS.

Para cadastro do usuário ao banco de dados, foi utilizado a biblioteca PDO, o PDO faz todo o tratamento de segurança nos envios de dados ao banco.

```
// Método cadastro de Usuário
public function adicionarUsuario($usuario)
{
    if ($this->existeUsuario($usuario) == false) {
        $sql = "INSERT INTO tb_cliente (nome, senha, permissao, hd, tempoRetencao, data, retencaoAviso, email, status, copiaEmail)
VALUES (:nome, :senha, :permissao, :hd, :retencao, :data, :retencaoAviso, :email, :status, :emailOpcional)";
        $sql = $this->pdo->prepare($sql);
        $sql->bindValue(':nome', $usuario->getNome());
        $sql->bindValue(':senha', $usuario->getSenha());
        $sql->bindValue(':permissao', $usuario->getPermissao());
        $sql->bindValue(':hd', $usuario->getHd());
        $sql->bindValue(':retencao', $usuario->getRetencao());
        $sql->bindValue(':data', $usuario->getDataCadastro());
        $sql->bindValue(':retencaoAviso', $usuario->getRetencaoAviso());
        $sql->bindValue('email', $this->email);
        $sql->bindValue('status', $usuario->getStatus());
        $sql->bindValue('emailOpcional', $this->emailOpcional);
        $sql->execute();
        return true;
    } else {
        return false;
    }
}
```

**Figura 18** - Cadastro de Usuário ao Banco de Dados

## 6.6 DESENVOLVIMENTO FRONT-END

Para desenvolvimento do front-end, foi utilizado o Framework Bootstrap, para trabalhar o efeito visual da aplicação Web, junto com o JavaScript onde manipula alguns eventos de botões.

Nessa etapa do projeto, o front-end, fica responsável em coletar as informações inseridas pelo usuário através de uma interface gráfica, as informações vêm de formulários em HTML, e enviadas ao back-end. Alguns recursos são feitos em JavaScript como o gerador de senhas, um trecho do código é ilustrado na figura abaixo.



```
<script>
  function gerarSenha(event) {
    event.preventDefault();
    $.get('partials/geraSenhas.php', function(data) {
      $('input[name="senha"]').val(data);
    });
  }
</script>
```

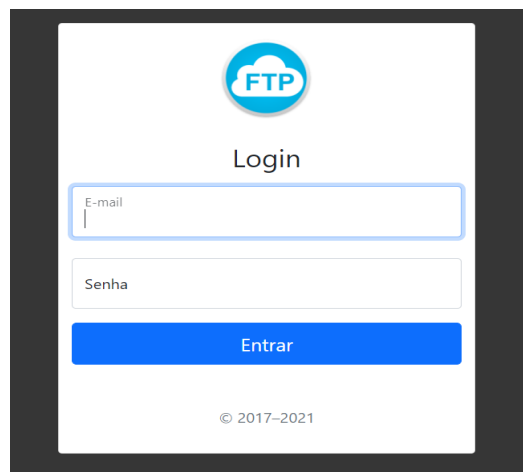
**Figura 19** - Gerador de senhas JavaScript

Na figura 19, ao clicar em um botão gerar senhas, um evento é enviado a uma função PHP, onde a senha é gerada, com as normas padrões de segurança, e uma senha é retornada ao INPUT do HTML. Muitos outros eventos no front-end, foram utilizados o JavaScript.

## 7. INTERFACE GRÁFICA

Com o intuito de melhor ilustrar como ficou a aplicação Web finalizada, serão apresentadas nos próximos parágrafos, diversas imagens da interface gráfica, sendo executada a partir de um servidor dedicado dentro do data center da empresa.

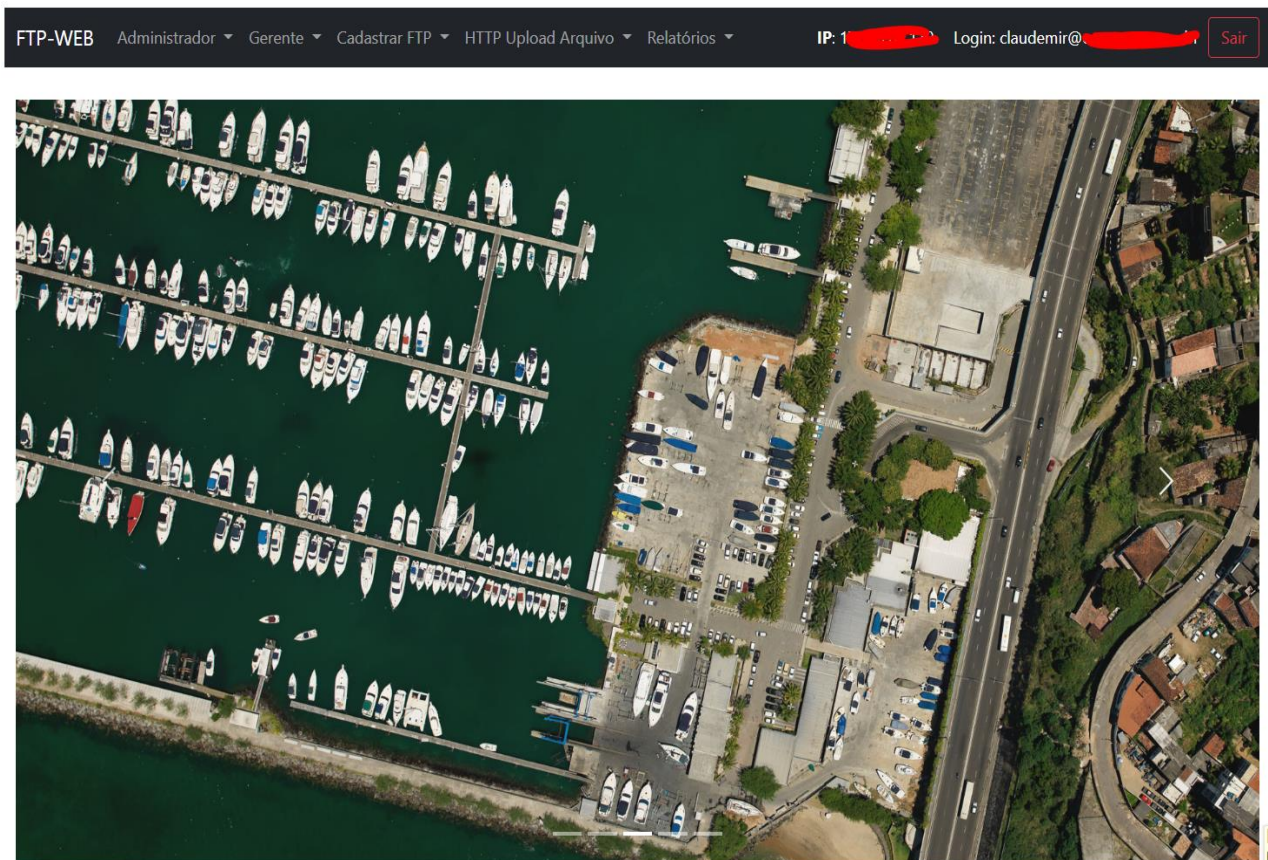
### 7.1 TELA DE LOGIN



**Figura 20** - Tela de Login

Para efetuar login, é necessário e-mail de cadastro e senha, essas informações são fornecidas pelos Administradores do Sistema.

## 7.2 TELA PRINCIPAL



**Figura 21** - Tela Principal

Na figura 21 é apresentada a tela principal do sistema, pode observar nessa figura, todas as opções do menu, cadastro de administrador (voltado ao departamento de TI), profissional de TI, responsável por cadastrar gerentes, e ter total gerencia do sistema.

Os gerentes são os maiores utilizadores do sistema, podendo cadastrar usuários FTP (Para serviços FTP), cadastrar envio de dados via HTTP (Para serviço HTTP Wetransfer). Quando o usuário gerente faz logon no sistema o mesmo não tem acesso ao menu cadastrar administrador.

### 7.3 TELA CADASTRO DE USUÁRIO FTP

**CADASTRO DE USUÁRIO FTP**

**NOME:**

**SENHA:**

**SELECIONE A PERMISSÃO DE ACESSO**

Somente Leitura    Leitura e Escrita

**SELECIONE O TEMPO DE RETENÇÃO**

90 Dias  
 120 Dias

**SELECIONE O HD**

HD-1 Espaço Livre : 240.58 GB  
 HD-2 Espaço Livre : 240.58 GB

E-MAIL OPCIONAL, CASO QUEIRA COPIAR ALGUÉM NO E\_MAIL:

**Grupo Engemap**

APÓS CONFIRMAR CADASTRO UM E-MAIL SERÁ ENVIADO PARA :  
**claudemir@[redacted]**

**Figura 22** - Cadastro Usuário FTP

Na figura 22, é exibida a tela de cadastro de usuário FTP, todos os campos são coletados pelo o usuário e enviados ao back-end. Informações como: nome do usuário, senha de acesso, permissão de acesso, dias para retenção dos dados, volume (HD) e e-mail para envio de informações. Após confirmação de cadastro, um e-mail é enviado ao usuário logado no sistema, com todas as informações necessária para acessos aos dados, também um anexo PDF contendo um pequeno manual de acesso.

## 7.4 TELA GERENCIAR USUÁRIOS FTP

USUÁRIO	NOME	PERMISSÃO	HD	TEMPO DE RETENÇÃO	DATA DE CADASTRO
<a href="#">Editar</a>	ftp100	escrita	hd1	05/09/2021	01/06/2021 21:20
<a href="#">Alterar Senha</a>	ftp19	leitura	hd1	24/08/2021	26/05/2021 16:34
<a href="#">Alterar Permissão</a>	ftp20	leitura	hd1	24/08/2021	26/05/2021 16:36
<a href="#">Alterar Retenção</a>	ftp21	leitura	hd2	23/08/2021	25/05/2021 11:19
<a href="#">Remover Usuário</a>	ftp22	escrita	hd2	23/09/2021	26/05/2021 17:43
<a href="#">Editar</a>	ftp23	leitura	hd1	23/08/2021	25/05/2021 11:22
<a href="#">Editar</a>	ftp24	leitura	hd2	23/08/2021	25/05/2021 11:22
<a href="#">Editar</a>	ftp25	leitura	hd1	23/08/2021	25/05/2021 11:53
<a href="#">Editar</a>	ftp26	leitura	hd1	24/08/2021	26/05/2021 17:46
<a href="#">Editar</a>	ftp27	leitura	hd1	24/08/2021	26/05/2021 17:47
<a href="#">Editar</a>	ftp28	leitura	hd1	24/08/2021	26/05/2021 17:47
<a href="#">Editar</a>	Ftp30	leitura	hd1	25/08/2021	27/05/2021 09:52

**Figura 23** - Gerenciar Usuários FTP

Em gerenciar usuários FTP (figura 23), o gerente visualiza os dados de cadastro do usuário FTP: como nome, permissão, qual HD que os dados estão armazenados, tempo de retenção dos dados e data de cadastro. No botão Editar, o gerente tem as opções oferecidas pelo sistema, como: alterar senha, alterar permissão, alterar retenção e remover usuário, o nome de um usuário FTP não pode ser alterado, pelo fato de ser um usuário de sistema no Linux e todos os dados são vinculados ao seu HOME.

O acesso aos dados é feito por um aplicativo cliente-FTP, o Filezilla é um cliente-FTP mencionado no anexo em PDF enviado junto ao e-mail, quando um novo usuário FTP é cadastrado no sistema.

## 7.5 TELA DE DOWNLOAD POR HTTP

Quando os dados forem enviados via “HTTP Upload Arquivo” disponível no menu principal, o acesso aos dados é feito por qualquer navegador de internet, no momento do cadastro é enviado um e-mail ao usuário contendo informações e um link para download, parecido o que a plataforma Wetransfer oferece, o usuário externo ao clicar no link, ele é redirecionado para uma página de download como mostra a figura 25.



**Figura 24** - Download de arquivos

## 8. CONSIDERAÇÕES FINAIS

O presente trabalho acadêmico só foi possível, pela experiência já adquirida em rede de computadores junto com todo aprendizado adquirido na fase acadêmica, pois o sistema web Software para Transferência de Dados, manipula um serviço rodando em um servidor Linux, esse serviço FTP, foi todo configurando para ter acesso de forma segura usando criptografia TLS.

Além da configuração do serviço FTP via terminal de comando, regras de IPTABLES, precisou ser aplicada no Firewall, o serviço de FTP é acessado via portas 20, 21 como já explicado nesse trabalho no capítulo 2. Metodologia de Desenvolvimento.

### 8.1 CONFIGURAÇÕES DO FIREWALL

Quando um sistema de FTP fica atrás de um Firewall de borda é necessário criar portas com range de acesso no Firewall, casos essas portas não sejam criadas de forma adequadas os usuários externos não conseguirão ter acesso aos dados. É de prática comum o serviço FTP ficar com a face para internet, com um link de internet hospedado direto no servidor de FTP. Na figura 25, é exibida a regra de firewall aplicada para um servidor de FTP.

```
# Regras para roteamento do FTP atrás do firewall
iptables -t nat -A PREROUTING -p tcp -d 192.168.2.34 --dport 20 -j DNAT --to 192.168.2.34:20
iptables -t nat -A PREROUTING -p tcp -d 192.168.2.34 --dport 21 -j DNAT --to 192.168.2.34:21
iptables -t nat -A PREROUTING -p tcp --dport 60000:60100 -j DNAT --to 192.168.2.34:60000-60100
iptables -A OUTPUT -p tcp --sport 21 -m state --state ESTABLISHED -j ACCEPT
iptables -A OUTPUT -p tcp --sport 20 -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -p tcp --sport 60000:60100 --dport 60000:60100 -m state --state ESTABLISHED -j ACCEPT
```

*Figura 25 - Regras Firewall FTP*

Nesse projeto, todas medidas de segurança foram aplicadas, como autenticação usuário e senha, criptografia dos dados em trânsito, o usuário que faz o acesso aos dados fica enjaulado apenas ao seu HOME de acesso, não podendo listar ou visualizar dados de outros usuários.

Outra dificuldade encontrada nesse projeto, talvez a maior delas, foi encontrar uma biblioteca que se conecta ao Linux de forma segura, muitos artigos achados na internet assim como outros trabalhos acadêmicos semelhantes a esse, mas não igual. Todos mencionavam a mesma coisa funções `shell_exec`, funções que precisa ser habilitado no Linux com permissões de `sudo` para cada comando executado no sistema via Apache. Essa função passou a ser ausente, por motivos de segurança. Após longas horas de pesquisa e teste de outras funções, sempre pensando nas questões de segurança, através de conversas no portal Stack OverFlow, encontrei a biblioteca SSH2, já mencionada no trabalho no capítulo 6.1 Desenvolvimento Back-End.



## 9. TRABALHOS FUTUROS

Com a experiência obtida nesse trabalho, há possibilidade de novas implementações, como integração direto no servidor FTP com volumes Storage, por conexão iSCSI, através de interface SFP 10G. Com isso não seriam necessários mais discos locais no servidor FTP para armazenar dados.

Os dados ficariam em um ambiente de Storage com um fator de segurança maior, além de alto desempenho e disponibilidade.

Uma versão mobile da aplicação poderia ser desenvolvida facilitando ainda mais o acesso, podendo gerenciar os dados de usuário FTP, onde quer que esteja, através do celular.

## REFERÊNCIAS

APACHE. **O que é o projeto Apache**. Disponível em <<http://httpd.apache.org/>> Acesso em: 05 maio. 2021

CNNBRASIL, **Vazamento de dados**, cnnbrasil.com.br, 2021. Disponível em <<https://www.cnnbrasil.com.br/business/o-mundo-ja-registra-4-6-bilhoes-de-dados-vazados-em-2021-diz-psafe/>>. Acesso em: 26 de mar. 2021

GOGONI, Ronaldo. **O que é FTP**, tecnoblog.net, 2020. Disponível em <<https://tecnoblog.net/303314/o-que-e-ftp/>>. Acesso em: 28 de out. de 2020.

GUEDES, Gilleanes T. A. **UML 2: Uma Abordagem Prática**, São Paulo: Novatec Editora, 2009.

JARGAS, Aurelio Marinho. **Shell Script Profissional**, Quinta reimpressão. São APaulo: Novatec Editora, 2008.

LUCIDCHART. **O que é um diagrama de atividades**. Disponível em <<https://www.lucidchart.com/pages/pt/o-que-e-diagrama-de-atividades-uml>>. Acesso em: 02 mar. 2021

MORIMOTO, Carlos Eduardo. **Servidores Linux guia prático**. Porto Alegre: Sul Editores, 2009.

CANALTECH. **O que é servidor apache**, canaltech.com.br, 2020. Disponível em: <<https://canaltech.com.br/internet/o-que-e-servidor-apache/>. Acesso em: 28 de outubro de 2020>. Acesso em: 10 jul. 2021.

PHP, **O que é PHP**, php.net, 2021. Disponível em <[https://www.php.net/manual/pt\\_BR/intro-what-is.php/](https://www.php.net/manual/pt_BR/intro-what-is.php/)>. Acesso em: 15 de mar. 2021

PONTES, Prof. Bruno. **Servidor de FTP – ProFTP**. Disponível em: <[http://diatinf.ifrn.edu.br/prof/lib/exe/fetch.php?media=user:1664038:semestr es:2009.1:asa:aulas:material:servidor\\_ftp.pdf](http://diatinf.ifrn.edu.br/prof/lib/exe/fetch.php?media=user:1664038:semestr es:2009.1:asa:aulas:material:servidor_ftp.pdf)>. Acesso em: 20 fev. 2021

SOBRE O DEBIAN, debian.org, 2020. Disponível em: <<https://www.debian.org/intro/about#what>>. Acesso em: 28 de out. 2020.

TYBEL, Douglas. **Orientações básicas na elaboração de um diagrama de classes**.

Disponível em <<https://www.devmedia.com.br/orientacoes-basicas-na-elaboracao-de-umdiagrama-de-classes/37224>>. Acesso em: 04 jul. 2021.