

CENTRO UNIVERSITÁRIO CENTRAL PAULISTA  
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

ALISSON DANIEL RODRIGUES DOS SANTOS  
GISMAR PEREIRA BARBOSA

**RELATÓRIO DE ATIVIDADES MULTIDISCIPLINARES**  
DESENVOLVIMENTO DE UM SERVIDOR WEB

RIO CLARO  
Junho / 2020

ALISSON DANIEL RODRIGUES DOS SANTOS  
GISMAR PEREIRA BARBOSA

**RELATÓRIO DE ATIVIDADES MULTIDISCIPLINARES**  
**DESENVOLVIMENTO DE UM SERVIDOR WEB**

Relatório apresentado à disciplina de Atividades Multidisciplinares do curso Sistemas de Informação do Centro Universitário Central Paulista, campus Rio Claro, a ser utilizado como ferramenta de avaliação de aprendizagem do primeiro semestre de 2020.

Orientador: Prof. Esp. Cristiano José Cecanho.

RIO CLARO  
Junho / 2020

## SUMÁRIO

LISTA DE FIGURAS.....	4
RESUMO.....	5
1 INTRODUÇÃO.....	6
2 DESENVOLVIMENTO.....	7
2.1 Ambiente operacional (Virtual ou Físico).....	7
2.1.1 Sistema operacional: Ubuntu Server 18.04.4.....	8
2.1.2 Sistema de virtualização: Virtualbox.....	10
2.2 Stack de desenvolvimento e operação web.....	13
2.2.1 Servidor web: Apache.....	14
2.2.2 Sistema gerenciador de banco de dados: MySQL.....	16
2.2.3 Linguagem de programação “server side & back-end”: PHP.....	18
2.3 Estrutura complementar (serviços básicos em servidores).....	20
2.3.1 Acesso ao servidor (acesso remoto): SSH.....	20
2.3.2 Sistema de Transferência de Arquivo: FTP.....	22
2.3.3 Sistema de compartilhamento de arquivos: SAMBA.....	24
2.3.4 Sistema de proteção de acessos ao servidor: Firewall.....	25
3 CONCLUSÃO.....	28
4 REFERÊNCIAS.....	30
5 APÊNDICES.....	34
APÊNDICE A – GITHUB ATM1S2020.....	34

## LISTA DE FIGURAS

Figura 1: Versionamento e "Releases" - Ubuntu.....	9
Figura 2: Modelo de virtualização utilizado pelo VirtualBox.....	12
Figura 3: Abstração da Virtualização.....	12
Figura 4: Como ocorre a comunicação Web.....	13
Figura 5: Workflow de um Servidor Web.....	15
Figura 6: Esquema de conexão Banco de Dados e Servidor Web.....	18
Figura 7: Acrônimo L.A.M.P.....	19
Figura 8: Workflow SSH.....	21
Figura 9: Modelo de funcionamento FTP.....	23
Figura 10: Esquema de funcionamento do Firewall.....	27

## RESUMO

Temos como objetivo principal deste projeto o estudo e implementação de um servidor web (baseados nas boas práticas atuais deste seguimento) completo, contendo serviços básicos: acesso remoto seguro (SSH), FTP, compartilhamento de arquivos (SAMBA), segurança para as transações de comunicação de entrada e saída de dados do servidor, assim como tráfego de pacotes (Firewall); e contendo também o serviços essenciais: SGBD (MySQL), webserver (Apache) e uma linguagem de programação back-end (PHP). Seguimos como metodologia de trabalho, pesquisar testar e implementar (em grande maioria do tempo, usando como plataforma uso de máquinas virtuais). A documentação foi realizada em “fragmentos” (retirando as amostras de cada processo de teste) para depois, serem finalizadas na composição deste documento principal (relatório técnico de todo os processos). Ao final dos testes e implementações, conseguimos obter um servidor web robusto e funcional, porém simples e objetivo, com todos os serviços necessário a tal categoria, assim como a qualquer servidor sobre a plataforma Linux, usando todos os recursos que a plataforma nos disponibiliza. Importante ressaltar, que nossa base de testes principal, foi sempre o uso da virtualização (VirtualBox).

**Palavra-chave:** SAMBA; Apache; SSH; PHP; SGBD; banco de dados; MySQL; FTP; servidor web; projeto; VM; virtualização; VirtualBox; firewall; UFW; hospedagem; HTTP; Linux.

## 1 INTRODUÇÃO

Visando entender, pesquisar e desenvolver um servidor web, durante a construção desta obra, iremos analisar e adentrar inúmeras tecnologias e serviços referentes ao tema.

Falaremos sobre servidores web em geral e sua utilização nos dias atuais, assim como tudo relacionado á eles. Desta forma, estaremos , de forma específica, estudando o universos dos servidores e suas construções, sobretudo o que estão a funcionar sobre o sistema operacional Linux (como distro principal, e artefato deste estudo, o Ubuntu Server).

Iremos implementar serviços básico para todos os servidores Linux: FTP, Firewall, compartilhamento de arquivos (SAMBA), acesso remoto seguro (SSH), dentre outros básicos, como a configuração e utilização de IPs fixos para servidores.

Indo adiante, iremos também implementar o serviços essenciais para um servidor web: um SGBD (MySQL), um servidor Web (Apache) e a linguagem de programação e operação de back-end (PHP), visando a implementação da nosso stack web.

A intenção será entender e compreender o universo dos servidores web, tendo em vista, sobretudo, nos dias atuais, a crescente demanda por sistemas web (a era dos sistemas em nuvem).

Todos os trabalhos serão documentados neste documento, e toda a parte de testes e implementações estarão disponíveis no repositório do Github que foi criado para o projeto (APÊNDICE A – Github ATM1s2020).

A realização do projeto será de grande valia para nossa área de estudo e formação profissional, tendo em vista, agregar não somente de forma acadêmica, mas como também prática, tendo em vista que todos os procedimentos realizados durante este projeto, configuram o dia a dia de um administrador de redes ou profissional que trabalha com redes, servidores e/ou provedores de Internet.

## **2 DESENVOLVIMENTO**

A ideia de construir um servidor Web, vai além de, unicamente, disponibilizar os recursos específicos. Para que tudo funcione perfeitamente, sendo este servidor disponível, seguro e produtivo, é necessário trabalhar sobre a vertente de uma conjunto de ferramentas, programas e/ou utilitários que criaram um ambiente característico e completo de servidores: compartilhamentos de arquivos, serviços de transferência de arquivos, acesso remoto seguro, dentre outros.

Sobretudo, ao escolher um servidor Web, nos dias atuais, temos a possibilidade alocar o mesmo em um ambiente virtual, nuvem, ou máquina física local, pois independente de qual seja o escopo onde o mesmo estará funcionando, ele será construído sobre uma máquina Linux, já que a maioria das ferramentas de desenvolvimento web, assim como base de infraestrutura para o mesmo, são desenvolvidas nativamente para o Linux (assim como nosso alvo principal de estudo para este trabalho).

Seguindo a linha de raciocínio dos parágrafos acima, teremos que modularizar (separar, dividir) a implementação dos processos (instalações e configurações), para que todos os serviços funcionem perfeitamente, em sincronia com o sistema operacional, assim como proposta a solução de acordo com sua demanda principal.

Desta forma, além dos serviços Web, serão necessário a implementação de outros serviços, que serão descritos daqui em diante, assim como o ambiente onde o servidor será instalado (o próprio sistema operacional).

### **2.1 Ambiente operacional (Virtual ou Físico)**

Como sendo a primeira atividade a ser pensada e trabalhada, após a decisão da construção de um Servidor Web, o ambiente onde o mesmo estará disponível deverá estar de acordo e compatível com a solução que o mesmo irá suportar.

A construção será feita em uma máquina Linux, e a distribuição será o Ubuntu Server.

Quanto ao ambiente, o mesmo poderá ser realizado em uma máquina virtual (VM) ou máquina física. Em nossos testes, foram realizados os dois tipos de instalação, configuração e utilização (para dimensionar as possibilidades, assim com semelhanças e compatibilidades entre os dois meios – físico e virtual).

A opção por modelo de estrutura de máquina virtual (VM), foi o VirtualBox (ferramenta de virtualização da Oracle).

### **2.1.1 Sistema operacional: Ubuntu Server 18.04.4**

O S.O. (Sistema Operacional) escolhido para ser utilizado no projeto foi o Ubuntu, uma distro Linux (das várias existentes) que disponibiliza versões LTS em vários formatos (IoT, Desktop, etc.), sendo nosso escopo para utilização da versão Ubuntu Server LTS 18.04.4.

Historicamente, o Ubuntu é uma das primeiras distros Linux:

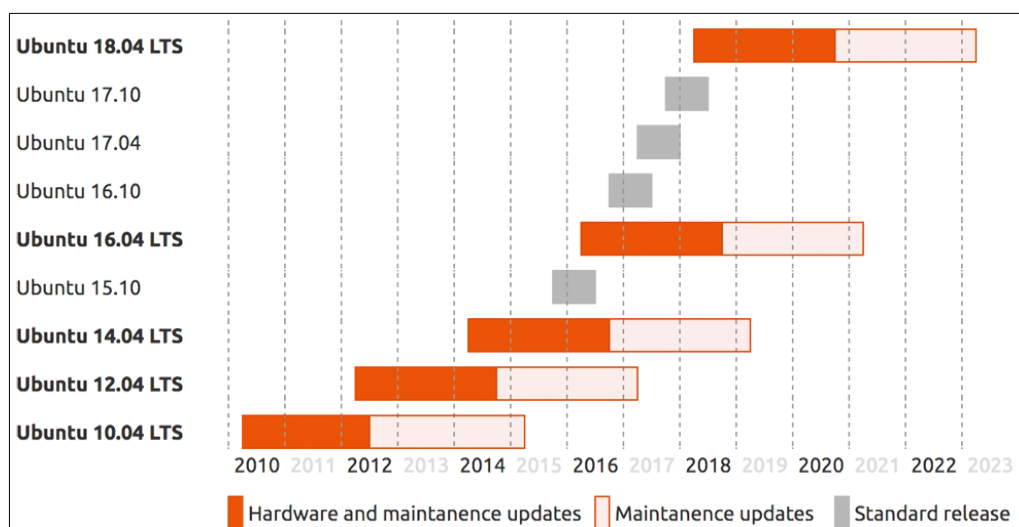
Ubuntu foi concebido em 2004 por Mark Shuttleworth, um empreendedor Sul-Africano de sucesso, e sua companhia Canonical. Shuttleworth reconheceu o poder do Linux e do código-aberto, mas reconhecia as fraquezas que impediam o seu uso disseminado. Shuttleworth pretendia abertamente resolver estes pontos fracos e criar um sistema que fosse fácil de usar, completamente livre ..., e que pudesse competir com outros sistemas operacionais populares. Com o sistema Debian como base, Shuttleworth começou a construir o Ubuntu. Usando seus próprios fundos a princípio, CDs de instalação eram prensados e enviados para todo o mundo sem nenhum custo para o usuário final. O Ubuntu se espalhou rapidamente, o tamanho da comunidade rapidamente cresceu, e logo se tornou a mais popular distribuição Linux baseada em Debian disponível (WIKILIVROS, Manual do Ubuntu / Uma breve história do Ubuntu, 2019).

Segundo (WIKILIVROS, 2019, Versões do Ubuntu) o versionamento do sistema operacional estão diretamente relacionadas com sua data de lançamento, que por sua vez vem estão no formato “ANO.MES” (A.M.). Desta forma, o primeiro dígito será referente ao ano de lançamento, e o segundo o mês. Ainda, em relação ao versionamento, historicamente, as versões LTS (acrônimo em inglês long-term support, em português suporte de longo prazo) são lançadas a cada 4 anos, e



sempre no mês de abril, sendo as versões de teste, lançadas todos os anos, no intervalo entre as LTS, nos meses de abril e outubro, com podemos ver na Figura 1.

*Figura 1: Versionamento e "Releases" - Ubuntu*



Fonte: <https://ubuntu.com/>

Conforme podemos checar no ranking do site DistroWatch (Distrowatch, 2020) podemos checar que as distribuição Linux é a 4ª colocada no ranking de distribuições mais populares. Em relação a versão Server, a mesma, por ser derivada do Debian Linux (Debian, 2020), também é extremamente popular, com o adicional de já “portar” drivers e ferramentas de empresas privadas (ferramentas proprietárias).

Além de tudo já citado, o Ubuntu tem um repositório completo, com todas as aplicações necessárias para este projeto, facilitando instalações, configurações, etc.

E, tendo todos estes itens já citados, podemos definir a escolha do Ubuntu Server 18.04.4 (versão corrente no momento do startup do projeto) sendo como a melhor escolha para o momento (visando não só implementar a solução, mas desenvolver o background necessário para o mesmo).

O procedimento de instalação, na maioria dos passos a serem seguidos, será sempre o mesmo, sendo máquina física ou máquina virtual (com algumas pequenas exceções).

Todo o processo técnico de instalação e configuração poderá ser verificado no repositório do Github (APÊNDICE A – Github ATM1s2020). A instalação é básica e simples, mas deve-se atentar no momento do particionamento de disco, para evitar erros indesejáveis após o sistema estar operando.

Outra situação que deve ser bem entendida, é no que tange a configuração das interfaces de rede do servidor, já que existem “n” formas de realizar determinada configuração, sendo a descrita no roteiro a mais indicada.

Durante os trabalhos (testes), trabalhamos com instalações em máquinas físicas (notebooks), e também em máquinas virtuais. Isso ocorreu, devido a necessidade de testar a metodologia utilizada no projeto, assim como aferir os dados coletados testando o funcionamento da solução, uma vez que o ambiente virtual pode não fornecer, em certo pontos, situações corriqueiras que uma máquina física possui.

Outra situação, é que apesar deste documento estar considerando a versão do Ubuntu Server 18.04.4, como a oficial para uso no projeto, também foram testadas as versões 13.04, 13.10, para verificar as possibilidades de retro utilização dos procedimentos aqui citados. Algo que deve ser levado em consideração quanto a versão que se deve utilizar do sistema operacional (Ubuntu Server), é que a mesma deva ser uma versão LTS (versão com suporte a update de segurança), e a mais atual possível, por questões de segurança. O uso, nos testes com versões ultrapassadas se deu apenas como experimentos (e por questões do hardware onde o mesmo foi testado), porém para um servidor real, devemos prezar pela segurança, e as versões atuais são mais estáveis e seguras (ao menos, por que continuam recebendo atualizações de segurança).

### **2.1.2 Sistema de virtualização: Virtualbox**

Para entendermos e introduzir o VirtualBox, software e/ou suíte de virtualização multi plataforma da Oracle (licenciado para uso gratuito para projetos de estudo de virtualização, opensource), primeiramente temos que entender o que é virtualização:

Para entender perfeitamente o conceito da tecnologia, deve-se traçar um paralelo entre o que é real e o que é virtual. Seguindo essa linha de raciocínio, algo real teria características físicas, concretas; já o virtual está associado àquilo que é simulado, abstrato. Dessa forma a virtualização pode ser definida como a criação de um ambiente virtual que simula um ambiente real, propiciando a utilização de diversos sistemas e aplicativos sem a necessidade de acesso físico à máquina na qual estão hospedados (TECMUNDO, 2009).

A Virtualização (em inglês Virtualization) é uma tecnologia que otimiza o uso do hardware através da criação de "versões virtuais" das aplicações ou recursos de toda infraestrutura de TI como hardware, sistemas operacionais, computadores, servidores e/ou dispositivos de armazenamento e incorporá-los a um ou mais servidores físicos. Em outras palavras, essa tecnologia permite compartilhar recursos de hardware de um computador, por exemplo, e utilizá-los para criar diversos ambientes virtuais isolados dentro do mesmo equipamento (CONTROLE NET, 2020).

Com esta devida introdução, podemos afirmar, de forma resumidamente, que o VirtualBox, é o responsável pela processo (todo o processo) da virtualização. Sendo mais específico:

VirtualBox é um software opensource, aplicativo gratuito, multi-plataforma para criar, gerenciar e executando máquinas virtuais (VMs) – computadores cujos componentes de hardware são emulados pelo computador host, o computador que executa o programa. VirtualBox pode ser executado em Windows, Mac OS X, Linux e Solaris (E-TINET, 2020).

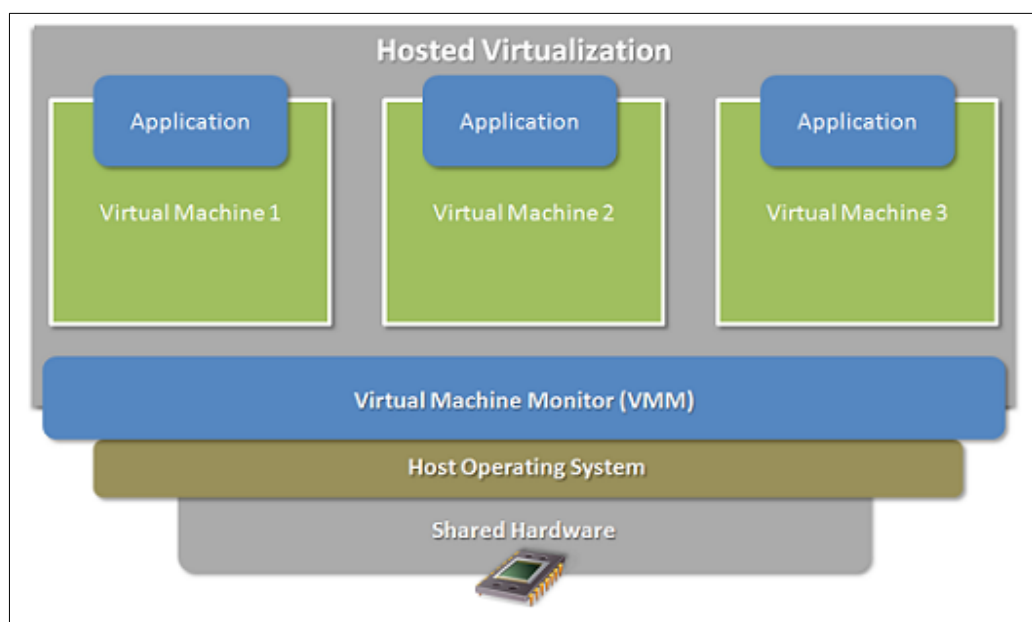
O uso de uma máquina virtual é facultativo (de acordo com o escopo de trabalho). Em nosso caso, por se tratar de uma atividade avaliativa, envolvendo todas as disciplinas já ministradas nos semestres anteriores, e correntes no semestre atual, usamos o recurso da virtualização para homologar o projeto. Outro fator, que também foi favorável para uso da virtualização, foi a falta de um servidor físico para testes, aliado ao fato da escalabilidade da virtualização para os testes.

Quanto a questão da utilização de virtualização, ser facultativa na reprodução deste projeto, mesmo assim é possível encontrar o procedimento de instalação e configuração no Github do projeto (APÊNDICE A – Github ATM1s2020).

De forma simples, visando apenas descrever o uso da prática de virtualização adotada (existem formas variadas para implementação de virtualização), ao instalarmos o VirtualBox, o mesmo criou uma camada de virtualização (especificamente no caso do VirtualBox, uma camada de emulação) do hardware das máquinas onde o mesmo foi instalado. Desta forma, conseguimos

“instalar um sistema operacional dentro de outro”, o que pode ser verificado na Figura 2, demonstrando como funciona o processo de virtualização:

*Figura 2: Modelo de virtualização utilizado pelo VirtualBox.*



*Fonte: <https://tecnologiasemsegredos.wordpress.com>*

Já na Figura 3, temos uma abstração do processo para o usuário:

*Figura 3: Abstração da Virtualização*



*Fonte: <https://www.qnapbrasil.com.br>*

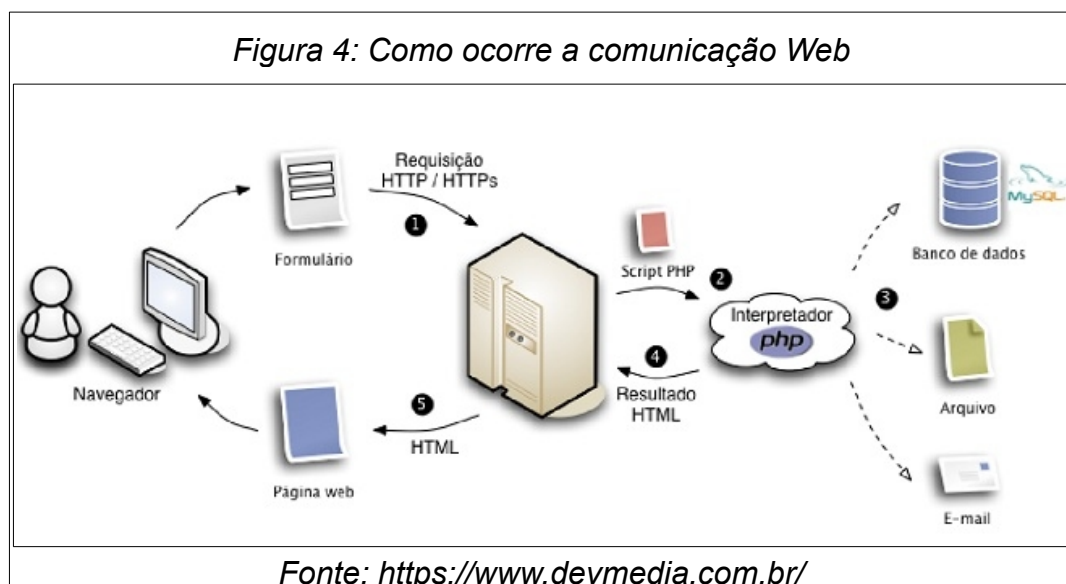
Na prática, isso nos favorece, sobretudo, como já citado, para realizar os testes e implementar uma solução, ainda que na ausência de uma máquina dedicada para o assunto (servidor).

A virtualização, pode ser aplicada também em ambientes de produção, mas isso dependerá do escopo para o qual a mesma for definida.

## 2.2 Stack de desenvolvimento e operação web

Quando estamos a falar de um ambiente Web, seja ele para operação e/ou produção, assim como para desenvolvimento e/ou testes, temos a necessidade de seguir um check-list (ainda que não literal) de serviços que terão que coexistir no mesmo ecossistema: um linguagem de programação de back-end (já que estamos falando de um servidor), um sistema gerenciador de banco de dados e o servidor web. E desta forma, nós temos uma “receita” para construção do “core” do nosso servidor Web.

Em nosso projeto, iremos utilizar a solução batizada como L.A.M.P. (Linux + Apache + MySQL + PHP). Já falamos do Linux, em capítulos anteriores, e agora, o foco é atuar sobre os demais, Apache (servidor web), MySQL Server (SGBD) e o PHP (linguagem de programação back-end). A intenção, em um diagrama gráfico, é algo perto do que é demonstrado na Figura 4.



### 2.2.1 Servidor web: Apache

Ao iniciar esta seção do documento, estamos a falar de um dos principais ativos do projeto, o Servidor Web Apache:

The Apache Software Foundation, Apache Server Project é um projeto 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 (APACHE HTTP SERVER PROJECT, 2020).

Entre todos os servidores web, o Apache é o que mais se destaca, por existir um enorme número de maneira que podemos configurá-lo. O mesmo pode ser configurado para vários domínios (hospedagem virtual), fornecendo comunicações criptografadas (HTTPS) e protegendo parte de ou todo um site usando diferentes tipos de autenticação.

O Apache começou como uma coleção de correções e melhorias do servidor HTTP feitas pelo Centro Nacional de Aplicações de Supercomputação (NCSA), até então, O daemon NCSA HTTP era o mais popular servidor HTTP da época, mas logo depois da saída de seu autor, Rob McCool, da NCSA, o mesmo veio a ficar defasado. Em 1995, um grupo de desenvolvedores formou o Apache Group e começou a fazer modificações importantes na base de código do NCSA HTTPD, tão, logo o Apache, veio a substituí-lo como o servidor web mais popular até nos dias de hoje. O mesmo grupo, mais tarde formou a Apache Software Foundation (ASF), para promover o desenvolvimento do Apache e outros softwares livres.

Para entender um pouco sobre seu funcionamento, falaremos aqui sobre o HTTPD (Hypertext Transfer Protocol Server Daemon), um componente do servidor web Apache. Ele fornece o serviço com o qual os navegadores web clientes se comunicam executando em segundo plano em seu servidor e aguarda solicitações de clientes web. Os navegadores da web fornecem essas conexões ao daemon HTTP e enviam solicitações, que o daemon interpreta enviando os dados apropriados.

Optamos por trabalhar com o Apache (versão atual Apache2), por conta da facilidade de encontrar documentação do mesmo. Outro motivo, é que o mesmo

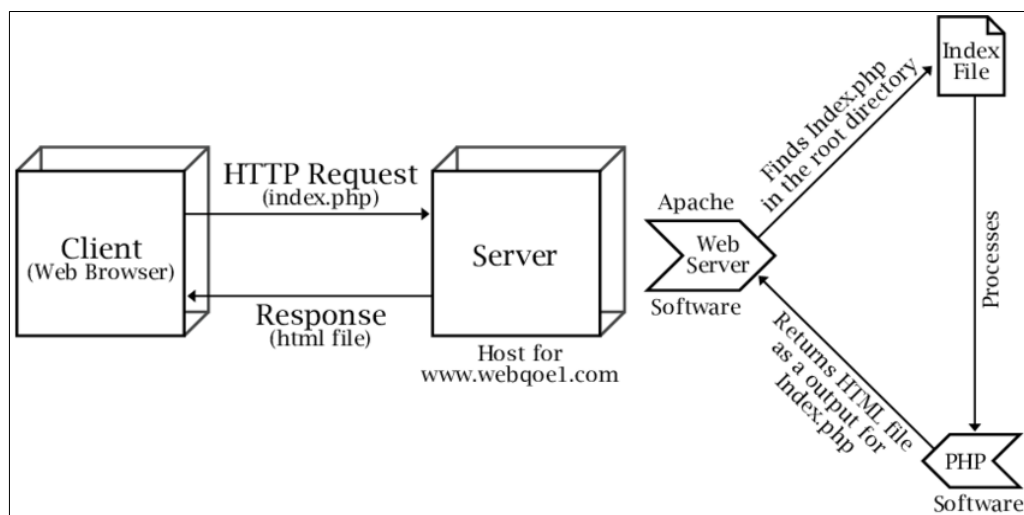
possui “libs” de integração de serviços com o PHP e o MySQL, sendo esse mais um facilitador para todo o processo de implementação.

No que tange a parte da instalação e configuração (básica para que o mesmo funcione) pode ser encontrada no Github do projeto (APÊNDICE A – Github ATM1s2020).

Os processo são simples, e podem variar de implementação para implementação.

Aqui temos o core de nosso servidor web, sendo o Apache, o responsável por criar a interface de acesso e operação dos programas (scripts) criados com o PHP, assim como também, no que tange a operação de todo o sistema web, fará toda as interações (requests / responses) do fluxo de comunicação entre cliente e servidor (workflow de operação do servidor web), como podemos verifica na Figura 5

*Figura 5: Workflow de um Servidor Web*



*Fonte: <https://www.researchgate.net>*

Partindo do pré suposto que o Apache é um servidor web, entendemos que existem variados tipos de implementações, sendo o modelo utilizado em nossa implementação escolhido de acordo com nossos objetivos (demonstração de um servidor web).

### 2.2.2 Sistema gerenciador de banco de dados: MySQL

Para nosso projeto, como já citado anteriormente, assim como em qualquer projeto de servidor web, é essencial o uso de um SGBD. Em nosso caso, escolhemos o MySQL Server:

O MySQL é um sistema de gerenciamento de banco de dados (SGBD), que utiliza a linguagem SQL (Linguagem de Consulta Estruturada, do inglês Structured Query Language) como interface. É atualmente um dos sistemas de gerenciamento de bancos de dados mais populares da Oracle Corporation, com mais de 10 milhões de instalações pelo mundo (Wikipédia, A enciclopédia livre, 2020).

Sendo os mais populares sistemas de gerenciamento de bancos de dados “SQL Open Source”, é desenvolvido pela MySQL AB, empresa ao qual, é fornecedora de serviços relacionados a sistemas de gerenciamento de banco de dados (SGBD). Esta popularidade está ligada pela sua facilidade de uso e capacidade de rodar em diferentes sistemas operacionais e também por ser uma tecnologia “open source”, sendo possível realizar seu desenvolvimento de acordo com as necessidades da organização.

O MySQL foi criado por David Axmark, Allan Larsson e Michael "Monty" Widenius, na Suécia, após a necessidade de utilizar mecanismos que permitissem a conexão de tabelas criadas na linguagem SQL. A princípio, a equipe iria fazer o uso de uma ferramenta inferior, mas logo perceberam que a mesma não iria ter um bom desempenho para atender as necessidades do projeto. Michael Widenius da companhia suíça TcX, desenvolveu um banco de dados chamado UNIREG, sendo reescrito em varias linguagens. Basicamente, entre os entrelaço da história, surgiu a primeira versão do MySQL baseado no banco de dados da companhia TcX, desenvolvida por Michael Widenius.

O SGDB, faz parte da combinação do pacote LAMP (ver em Stack de desenvolvimento e operação web),

Para a utilização do mesmo (de forma profissional, como é realizada em provedores e hospedagens de sites), é necessário ter instalado um servidor e um aplicação cliente (SSH, vem em Acesso ao servidor (acesso remoto): SSH), o



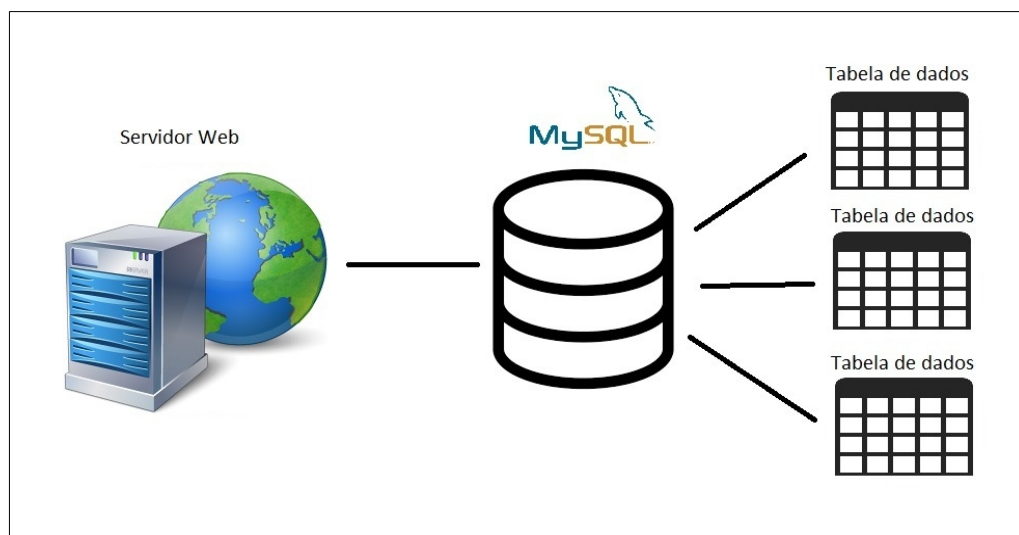
servidor é o responsável por armazenar os dados, responder as requisições, controlar a consistências dos dados entre outras. A comunicação entre cliente e servidor é feita através de strings de conexões e/ou métodos encapsulados (abstrações como DAO ou componentes do MySQL como PDO). Tais conexões, podem ser locais também, como é nosso caso (escopo do nosso projeto).

A implementação do MySQL Server é tão prioritariamente e importante, como a questão da implementação do Apache. Todos os detalhes da implementação realizada em nosso projeto podem ser conferidos no Github do projeto (APÊNDICE A – Github ATM1s2020).

Ainda quanto a sua implementação, devemos salientar que em nosso projeto, todos os ativos do servidor web (inclusive o MySQL) estão instalados na mesma instância virtual (como também poderia ser um instância física). E desta forma, conforme este modelo, seguimos as instalações. Todavia, caso você esteja utilizando outro modelo de arquitetura (como por exemplo, um servidor remoto do MySQL), o mais correto seria dar uma atenção especial quanto a parte de segurança para conexões (sejam para manipulação e/ou para consumir recursos) entre o cliente e o servidor remoto (com o uso de SSH, como citado anteriormente).

Ainda no ponto das configurações técnicas, nosso foco de trabalho, foi único e exclusivo, para disponibilizar o ambiente de banco de dados para os usuários remotos, mas onde o banco de dados só responde a requisições internado do servidor, intermediadas pelo PHP e o Apache (o usuário não cria uma instância ou conexão direta ao SQL Server, ele passa pelo sistema do site), como pode ser conferido na Figura 6. O mesmo ocorre com o programado, que para criar sua base de dados, deve repassar seu script de banco de dados (ou solicitar um usuário local) para o DBA (acrônimo em inglês para Databse Administrator, em português Administrador de Banco de Dados) crie a base de dados e libere a conexão para o sistema (sites ou programas web que estarão executando a partir do servidor Apache).

*Figura 6: Esquema de conexão Banco de Dados e Servidor Web*



*Fonte: <https://www.homehost.com.br/>*

Basicamente, temos um servidor, que só atende requisições locais, sem interações remotas (diretas).

### **2.2.3 Linguagem de programação “server side & back-end”: PHP**

Para completar nosso Stack de desenvolvimento web, assim como sinalizar qual linguagem irá trabalhar com as requisições, acessos ao banco de dados, regras de negócio do site (tudo que é relacionado com back-end de um site e/ou aplicação web), iremos falar sobre o PHP.

De acordo com o portal oficial do PHP:

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 (PHP.NET, O que é PHP?).

Seguindo ainda, quanto a definição do PHP:

Em vez de muitos comandos para mostrar HTML (como acontece com C ou Perl), as páginas PHP contém HTML em código mesclado que faz "alguma coisa" (neste caso, mostra "Olá, eu sou um script PHP!"). O código PHP é delimitado pelas instruções de processamento (tags) de início e fim `<?php` e `?>` que permitem que você entre e saia do "modo PHP" (PHP.NET, O que é PHP?) .

Desta forma, temos uma linguagem que se mescla com as TAGs HTML, assim como com demais linguagens como o Javascript, porém, diferentes destas, o código PHP é interpretado e executado do lado do servidor (server-side e/ou back-end), enquanto o restante é feito no lado cliente (client-side e/ou front-end). Isso traz certa segurança, já que os usuários não conseguem ter acesso ao código fonte do que está executando diretamente no servidor.

O PHP é uma das linguagens de programação mais populares em todo mundo, tendo em vista que sua facilidade de implementação, e as grandes comunidades existentes. Por se tratar de uma linguagem de código fonte aberto, está em plena construção e melhoria, ainda nos dias atuais.

Em nosso projeto, temos a necessidade realizar a instalação do ambiente para utilização da mesma. Desta forma, como pode ser verificado no repositório do Github do projeto (APÊNDICE A – Github ATM1s2020), sua disponibilização para o servidor Apache, assim como utilização para o MySQL é simples e objetiva, já que a linguagem possui plugins e componentes que facilitam todo o processo.

Podemos falar que o PHP, desde sua criação (por Rasmus Lerdorf, em 1995), domina a Internet, sendo a linguagem mais popular e mais utilizada em ambientes web, desde a criação e popularização da Internet.

É uma linguagem extremamente flexível, sendo multi paradigmas (linguagem de scripts, estruturada, orientada a objetos, etc.), os aspectos acima, definindo a utilização do PHP, e fechando a suíte LAMP.

*Figura 7: Acrônimo L.A.M.P.*



*Fonte: <https://appiomatic.com/>*

## **2.3 Estrutura complementar (serviços básicos em servidores)**

Quando falamos em serviços complementares em um servidor Linux, estamos nos referindo aos serviços essenciais, que todo servidor Linux deve prover. Tais serviços se resumem ao compartilhamento e/ou gerenciamento de arquivos entre o servidor e seus clientes, assim como também as formas de acesso remoto e validação para o uso dos referidos compartilhamentos. E, é claro, determinando quais são as regras de entrada e saídas de dados do servidor para com os clientes em rede.

Por este motivo, no projeto de nosso servidor web, também seguimos esta regra e disponibilizamos os seguintes serviços: compartilhamento de Arquivos (SAMBA), transferência de arquivos (VSFTPD – FTP), acesso remoto seguro (SSH) e segurança para as conexões e portas de acesso ao servidor (UFW – Firewall).

### **2.3.1 Acesso ao servidor (acesso remoto): SSH**

Para o acesso ao servidor, visando simular um ambiente real, e como as coisas funcionam neste tipo de ambiente, a intenção foi criar e utilizar as conexões remotas, os acesso via a SSH:

O SSH é um pacote de software que permite a administração segura do sistema e transferências de arquivos por redes inseguras. É usado em quase todos os data centers e em todas as grandes empresas (SSH, Secure Shell, SSH.com, 2020).

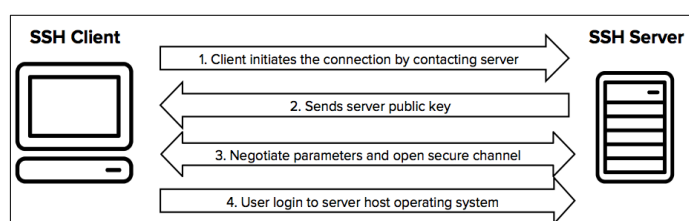
As ferramentas do protocolo Secure Socket Shell são um conjunto de aplicativos cliente-servidor que permite fazer acesso remoto seguro entre computadores clientes e servidores. Nelas, incluem os protocolos ssh, para fazer logins e execuções remotas, como também os protocolos scp e rsync para fazer cópias de arquivos entre os sistemas. Como a troca de informação é totalmente criptografada entre servidor e os clientes, essas ferramentas são mais seguras do que as antigas ferramentas, como por exemplo, o antigo protocolo telnet (comando de login), onde expunham todos os dados para ataques cibernéticos. Hoje este tipo de comando só deve ser usado para testes com portas remotas ou outras operações que não contenha dados privados.

SSH foi desenvolvido em 1995, logo após a um incidente de hacking na rede universitária finlandesa. Um sniffer (análise de pacotes) de senha havia sido instalado em um servidor conectado diretamente á rede principal de tráfego de dados (espinha dorsal), logo quando a falha foi descoberta, ele tinha milhares de nomes e senhas de usuários em seu banco de dados, incluindo vários de sua própria empresa. Com este ocorrido, Ylonen se dedicou totalmente nos estudos sobre criptografia e no desenvolvimento de uma solução para o uso de login remoto pela internet com segurança. E em julho de 1995, Ylonen publicou a primeira versão do código aberto OpenSSH.

Atualmente, este protocolo é usado no gerenciamento de mais na metade dos servidores web do mundo e praticamente todos os S.O UNIX, tanto local com também em nuvem. Especialista na área de segurança de informação e administradores de sistemas, usam o protocolo para configurar a maioria dos firewalls, roteadores, switches e servidores em ambientes de missão crítica do nosso mundo digital.

O protocolo SSH, funciona no modelo cliente-servidor, a conexão é feita pelo cliente SSH conectando-se ao servidor SSH. Quem faz ao processo de configuração do protocolo é o cliente SSH que usa criptografia de chave pública para verificar a identidade do servidor SSH. Após esta fase, o protocolo faz o uso de algoritmo que mapeia dados grandes e de tamanho variável para pequenos dados de tamanho fixo, o que chamamos de algoritmos de hash, para assim, garantir a privacidade e a integridade dos dados que são trocados entre o cliente e o servidor.

*Figura 8: Workflow SSH*



*Fonte: <https://www.ssh.com/>*

Todos os detalhes da implementação técnica do SSH em nosso projeto, podem ser encontrados no Github do projeto (APÊNDICE A – Github ATM1s2020).

Em nosso caso, usamos uma implementação básica do serviço (sem uso de chaves criptográficas). Todavia, caso o projeto fosse aplicado em um ambiente real, teríamos esta demanda, da geração de chaves criptográficas (chaves públicas e privadas), recurso este fornecido, de forma nativa, pelo serviço (ao menos nos sistemas Unix).

### **2.3.2 Sistema de Transferência de Arquivo: FTP**

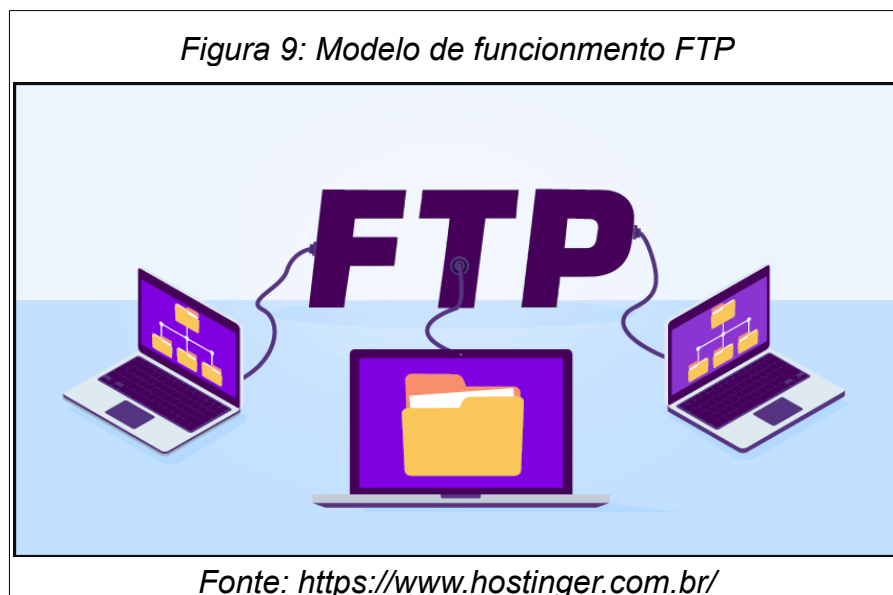
O FTP, serviço essencial em qualquer servidor web, não poderia faltar em nosso projeto. Seguindo a lógica de pensamento, a definição de FTP:

Muito antes dos sistemas de armazenamento em nuvem, nasceu o FTP (File Transfer Protocol). Desde os primórdios da Internet, ele é o responsável por enviar arquivos para a web. Forma prática e versátil de transferência de arquivos, ela serve basicamente para que usuários possam enviar ou receber documentos da Grande Rede por meio de um endereço no navegador ou um software instalado no PC (BARROS, Tecmundo, 2012).

Do ponto de vista histórico:

A origem do FTP data da década de 1970, num momento evolução dos computadores em que as primeiras redes e ligações entre as máquinas existentes estavam ainda engatinhando, mas crescendo gradativamente. A tecnologia foi desenvolvida por Abhay Bhushan, na época, um estudante de engenharia elétrica no MIT (Massachusetts Institute of Technology), um dos centros de ensinos acadêmicos mais conceituados do mundo (HOSTINGER, 2019).

Tão logo, o FTP será uma das principais ferramentas utilizadas em nosso projeto, levando em consideração que o servidor web terá o enfoque para disponibilizar um ambiente, para que outros alunos, hospedem seus projetos web, fazendo isso por intermédio do FTP (VSFTPD), seguindo uma lógica, podemos pensar na em algo, como demonstrado abaixo, na Figura 9, demonstrando como ocorrem as comunicações e transferências de arquivos entre servidor e cliente.



Todos os detalhes da implementação técnica do FTP, poderão ser verificados no Github do projeto (APÊNDICE A – Github ATM1s2020).

Como já citado, nos demais serviços integrantes deste servidor web, existem várias formas de configuração do serviço de FTP, sendo a adotada neste projeto, o a configuração de usuários e senhas, com a acesso a pasta “public”, em suas pastas “home” de usuário (a pasta “/home/public” se torna a raiz, “/” para o FTP). Desta forma, cada usuário poderá ter acesso apenas a sua pasta “public”. Esse controle foi implementado para criar certa camada de segurança.

Outra forma, possível de usar, em nosso projeto, é o SFTP (Secure FTP), que no caso, vai utilizar uma conexão SSH, juntamente com o FTP. Como já dito, maiores detalhes, consultar o Github do projeto (APÊNDICE A – Github ATM1s2020).

No que tange o desenvolvimento de todo o projeto, neste capítulo em específico (sobre o FTP), a implementação técnica obteve sucesso durante quase todos os testes, com exceção dos testes utilizando a versão do Ubuntu Server 13.04. Nesta versão específica, todos os procedimento técnicos empenhados para que a autenticação FTP (utilizando o programa VSFTPD) falharam. Em comparação, na versão 13.10, os mesmos testes funcionaram, com sucesso em 100% (assim como na versão de estudo deste projeto, a 18.04.4). Acreditamos que isso tenha ocorrido

por algum tipo de inconsistência e/ou compatibilidade entre a versão do sistema operacional, e a versão disponível “a época” do programa VSFTPD (FTP).

### **2.3.3 Sistema de compartilhamento de arquivos: SAMBA**

O Samba é um projeto de código aberto dos protocolos usados para compartilhar arquivos e impressoras, como também, para autenticar usuários e restringir hosts, entre sistemas. É um conjunto de programas que permitem que Linux, UNIX e outros sistemas interoperarem com protocolos de compartilhamento de serviços (impressão, arquivos, diretório) do Microsoft Windows. Com ele, podemos usar o padrão TCP/IP para se comunicar com os clientes. O Samba suporta hostnames, TCP/IP, assim como os nomes de máquinas NetBIOS.

As pessoas preferem os servidores Samba em vez dos servidores de arquivo do Windows, por causa da maior segurança que é inerente à execução de serviços do Windows de compartilhamento de arquivos no Linux ou outros sistemas operacionais tipo UNIX. Com isso, o Samba sempre se esforça para tornar seu software seguro e robusto. Além de todas as discussões técnicas, podemos dizer, por base de estudos, que o Samba torna fácil compartilhar seus serviços entre servidores Linux e sistemas desktop Windows.

O Samba foi criado e publicado em 1992, por Andrew Triggell, ele precisava configurar um volume de disco em seu PC, para um servidor UNIX. O S.O (Sistema Operacional) era o DOS que utilizava o sistema de arquivos de formato NFS (Network File System) para o acesso. Mas o que menos esperava, era que um aplicativo de seu computador necessitaria do protocolo NetBios (incompatível com NTS). Triggell então, escreveu um Sniffer (programa para captura de tráfego de dados em rede) para a análise de tráfego de dados gerado pelo protocolo NetBios. Uma vez implementado, ele utilizou de engenharia reversa em cima do protocolo SMB (Server Message Block), e implementou no UNIX. Isso fez com que o servidor Unix aparecesse como um servidor Windows em seu computador baseado no DOS.

Para a implementação da ferramenta no lado do servidor basta instalar o pacote do Samba, definir no “smb.conf” quais os diretórios serão compartilhados,



criar usuários, definir senhas, ajustar as permissões, criar o script para mapear unidades de rede no login. No lado cliente, estações Windows, basta configurar para fazer login sob domínio do NT.

O Samba em nosso projeto tem a missão de fornecer a gerencia e compartilhamento de arquivo, entre os usuários remotos do servidor Web (basicamente, aqui estamos falando de usuários programadores, que usarão o servidor remoto para compartilhar seus arquivos).

Habitualmente, independente da plataforma cliente, o Samba irá funcionar para compartilhamento de arquivos, dentro das especificações que utilizamos em nosso projeto. Você poderá verificar mais detalhes técnicos no Github de nosso projeto (APÊNDICE A – Github ATM1s2020).

Resumidamente, na questão do compartilhamento de arquivo, ele estará presente como uma auxílio ao FTP (que também pode ser usado para compartilhar arquivos).

Como já citado em outros momentos anteriores, a configuração utilizada em nosso projeto, referente ao compartilhamento de arquivos via Samba, é a básica para que o compartilhamento funcione.

#### **2.3.4 Sistema de proteção de acessos ao servidor: Firewall**

Um Firewall é um dispositivo de segurança de rede que monitora e filtra o tráfego de rede de entrada e saída com base nas políticas de segurança previamente estabelecidas por uma organização. No seu mais básico, um firewall é essencialmente a barreira que fica entre uma rede interna privada e a Internet pública. O principal objetivo de um firewall é permitir que o tráfego não ameaçador entre e mantenha o tráfego perigoso fora. (Check Point, Software Technologies LTD)

Por medidas de segurança de uma rede, é aconselhável o uso do dispositivo para bloquear ataques de malware (software malicioso, em tradução livre) à camadas de aplicativos. Em parceria com um sistema integrado de prevenção de instruções (IPS), um Firewalls é capaz de identificar de forma rápida um ataque em toda a rede. Os sistemas de segurança podem agir em políticas previamente

definidas para proteger a rede e tendem a ter avaliações rápidas para detectar atividades invasivas ou suspeitas de fragmentos estranhos, e desligá-las.

Os firewalls existem desde o final de 1980, e praticamente eram filtros de pacotes para examinar pacotes ou bytes transferidos em redes de computadores. Embora estas tecnologias de filtragem de pacotes ainda estejam em uso nos dias de hoje, os firewalls foram mais além do que isso com o avanço da tecnologia. Os firewalls de hoje, suportam uma ampla variedade de funções e recursos incorporados, como por exemplo, prevenção de ameaças de rede, controle baseado em aplicativos e identidades, suporte nuvem híbrida e desempenho escalável.

Os tipos de firewalls são:

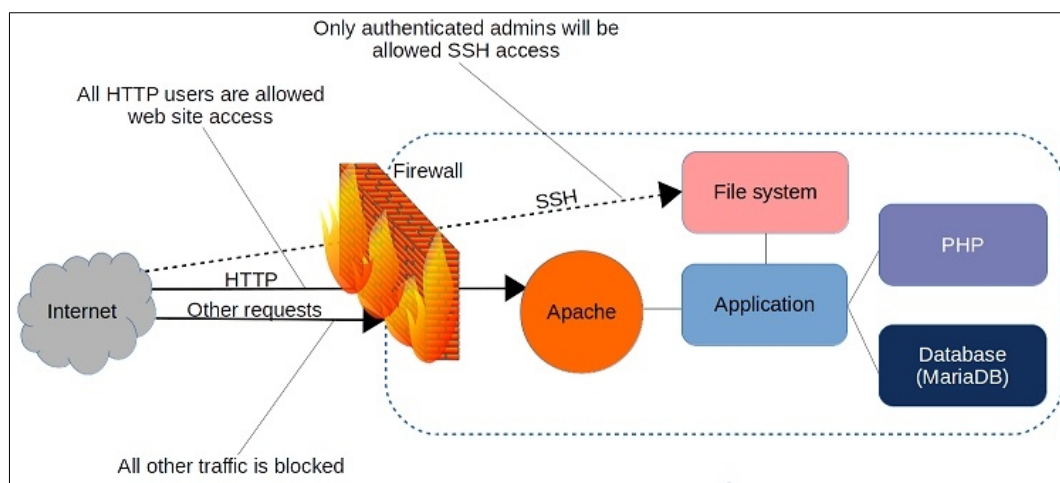
- Filtragem de pacotes: Analisa uma pequena quantidade de dados e é distribuída de acordo com as definições de normas do filtro.
- Serviço de Proxy: Dispositivo de opera na filtragem de mensagens na camada aplicativo (HTTP, HTTPS).
- Inspeção componente: Filtragem de pacotes, monitora conexões ativas para definir quais os pacotes de rede permitir através do Firewall.
- Firewall de próxima geração: Firewall de inspeção de pacotes profundos na camada de aplicativo.

Em comparativos de inspeções de camada de rede vs. camada de aplicativos, os filtros da camada de rede inspecionam pacotes em um nível relativamente baixo da pilha de protocolos TCP/IP, isso faz com que os pacotes não passem pelo firewall, a menos que correspondam as políticas definidas onde, a origem e o destino do conjunto de regras são baseados em endereços e portas IP no Protocolo da Internet. Sendo assim, os dispositivos que fazem as inspeções nas camadas de rede têm um desempenho melhor dos dispositivos similares que inspecionam a camada de aplicativos. A desvantagem é que os softwares maliciosos, podem passar por rotas permitidas, como por exemplo, tráfego de internet de saída por protocolos de internet HTTP e HTTPS.

Resumidamente, em nossa implementação de Firewall, onde utilizamos o programa padrão UFW (que acompanha a maioria das distribuições Linux), apenas atuamos no bloqueio de entrada, liberando a penas as portas específicas para cada serviço instalado no servidor (por exemplo, a porta TCP/22 para o SSH).

Todas as configurações técnicas referentes ao Firewall, podem ser verificadas no Github do projeto (APÊNDICE A – Github ATM1s2020). Elas foram implementadas de acordo com a demanda que nosso projeto necessita.

*Figura 10: Esquema de funcionamento do Firewall*



Fonte: <https://opensource.com/>

Conforme podemos ver na Figura 10, o esquema de funcionamento do Firewall, está em impedir qualquer tipo de comunicação que não esteja dentro das regras de Firewall, previamente cadastradas.

### 3 CONCLUSÃO

A implementação de um Servidor Web, de fato é algo complexo e extremamente trabalhoso, e que vai além da sua definição em si, tendo em vista que, para se ter um servidor seguro, robusto e eficiente, temos uma suíte de serviços a serem implementados para que a solução final possa, de fato, fornecer seu serviços.

Partindo da introdução acima, para a construção deste projeto, tivemos a necessidade buscar (pesquisar) muito sobre a plataforma Linux (S.O.). Tendo o Linux, como nosso sistema operacional, foi necessário entender como as coisas funcionam, para só depois iniciar os testes e implementações. E, seguindo uma linha lógica, tivemos que entender como um servidor web funciona, para que fosse possível propor a solução para o problema do projeto.

Alinhado a tudo que já foi descrito acima, tínhamos a necessidade separar a construção do trabalho e módulos (separação de conceitos), para que cada parte do servidor estivesse bem definida e funcional. E isso, não somente por que era nosso pensamento, mas por que é assim que as coisas funcionam no Linux, sendo que cada parte ou serviço responsável por um demanda neste ambiente.

Seguindo a estratégia, separamos o desenvolvimento do servidor, em serviços e/ou sistemas básicos (FTP, SSH, SAMBA, Firewall), e sistemas essenciais (Apache, MySQL e PHP).

Os serviços básicos foram implementados, devido a necessidade criar robustez ao servidor, assim como praticidade de seu uso. Seguindo a premissa de que, um servidor deveria ter acesso restrito e controlado, assim como tudo que o mesmo disponibiliza, e visando atribuir confidencialidade, integridade e disponibilidade (CID), todos os serviços necessários (citados no paragrafo anterior) forma implementados.

Já em relação aos serviços essenciais, tivemos o cuidado de liberar as funções, estritamente, necessárias, para que o servidor cumprisse seu objetivo de forma objetiva e com determinada agilidade.

Durante o processo de construção, percebemos que algumas situações deveriam ser revistas (como ocorreu com o FTP), mas sobretudo, em grande maioria dos eventos obtivemos êxito.

É claro, que, partindo deste estudo, podemos iniciar inúmeros estudos específicos, como por exemplo, estudos na vertente de segurança de um servidor web, como também, as diferenças existentes entre implementações em um ambiente virtual ou físico, etc; inúmeros outros estudos e pontos de pesquisa poderiam ser gerados deste projeto em específico.

E ao final do projeto, ainda que a construção seguisse em linhas paralelas (ao mesmo tempo desenvolvendo vários serviços), conseguimos obter um servidor eficiente e funcional.

#### 4 REFERÊNCIAS

Apache, Http Server Project. Disponível via URL: <<https://httpd.apache.org/>>. Acesso em 15 de abr. de 2020.

Arquivos.wiki.br, “Como resolver erros relacionados com o arquivo httpd”. Github, 2020. Disponível via URL: <<http://www.arquivo.wiki.br/processo/httpd.exe.html>>. Acesso em 29 de mar. de 2020.

ATM\_1S2020, “Atividades Multidisciplinares 1º Semestre 2020 - (ATM1s2020)”. Github, 2020. Disponível via URL: <[https://github.com/gismarb/atm\\_1s2020](https://github.com/gismarb/atm_1s2020)>. Acesso em 29 de mar. de 2020.

BalenaEther, Download balenaEtcher 1.5.80. Disponível via URL: <<https://www.balena.io/etcher/>>. Acesso em 10 de mar. de 2020;

BARROS, Thiago. “O que é FTP e como usar?”. Techtudo, 2012. Disponível via URL: <<https://www.techtudo.com.br/artigos/noticia/2012/07/o-que-ftp-e-como-usar.html>>. Acessado em 30 de jun. de 2020.

Becode, Principais SGBDS. Disponível via URL: <<https://becode.com.br/principais-sgbds/>>. Acessado em 29 de jun. de 2020.

Bóson Treinamentos em Tecnologia, Como configurar endereço IP estático no Linux Ubuntu 18.04 com netplan. Disponível via URL: <<http://www.bosontreinamentos.com.br/linux/como-configurar-endereco-ip-estatico-no-linux-ubuntu-18-04-com-netplan/>>. Acesso em 10 de mar. de 2020;

CHECK POINT, O que é um firewall. Disponível via URL: <<https://www.checkpoint.com/cyber-hub/network-security/what-is-firewall/>>. Acesso em 30 de JUN de 2020.

CLINTON, David, “Linux firewalls: What you need to know about iptables and firewall”. Opensource, 2018. Disponível via URL: <<https://www.checkpoint.com/cyber-hub/network-security/what-is-firewall/>>. Acesso em 30 de JUN de 2020.

ClubeDelphi 167. “Introdução ao desenvolvimento web no Delphi”. Devmedia, 2020. Disponível via URL: <<https://www.devmedia.com.br/introducao-ao-desenvolvimento-web-no-delphi/34423>>. Acessado em 30 de jun. de 2020.

Controle NET. “O que é Virtualização ou Virtualization?”, 2009. Disponível via URL: <<https://www.tecmundo.com.br/web/1624-o-que-e-virtualizacao-.htm>>. Acessado em 30 de jun. de 2020.

Debian, 2020. Disponível via URL: <<https://www.debian.org/>>. Acessado em 30 de jun. de 2020.

DELFINO, Pedro. “O que é Virtualização ou Virtualization?”. E-TINET, 2020. Disponível via URL: <<https://e-tinet.com/linux/virtualbox-porque-utilizar/>>. Acessado em 30 de jun. de 2020.

E. MORINOTO, Carlos. Servidores Linux, guia Prático. Porto Alegre: SUL, 2011;

FileZilla, Overview. Disponível via URL: <<https://filezilla-project.org/>>. Acesso em 15 de mai. de 2020.

HASSAN, Zunayed. “Installing Apache, MySQL and PHP on Linux”. Developer’s Diary, 2020. Disponível via URL: <<https://appiomatic.com/blog/installing-apache-mysql-and-php-on-linux/>>. Acessado em 30 de jun. de 2020.

Home Host. “O que é MySQL?”, 2020. Disponível via URL: <<https://www.homehost.com.br/blog/tutoriais/mysql/o-que-e-mysql/>>. Acessado em 30 de jun. de 2020.

ISLAM, Nazrul. “Workflow of apache web server”. ResearchGate, 2014 . Disponível via URL: <[https://www.researchgate.net/figure/Workflow-of-apache-web-server\\_fig4\\_277816114](https://www.researchgate.net/figure/Workflow-of-apache-web-server_fig4_277816114)>. Acessado em 30 de jun. de 2020.

L, Andrei. “FTP: o que é, como funciona e qual o melhor tipo para gerenciar arquivos na internet”. Hostinger Tutoriais, 2019. Disponível via URL: <<https://www.hostinger.com.br/tutoriais/ftp-o-que-e-como-funciona>>. Acessado em 30 de jun. de 2020.

MySQL, mysql.com. Disponível via URL: <<https://www.mysql.com/>>. Acesso em 18 de mai. de 2020.

NEGUS, Cristopher. Linux: A Bíblia. Tradução da 8ª edição. Rio de Janeiro: Alta Books, 2014;

OpenSSH, SSH... OpenSSH Keeping your communiqués secret. Disponível via URL: <<https://www.openssh.com/>>. Acesso em 08 de mar. de 2020.

OSTechNix, Install Apache, MySQL, PHP (LAMP) Stack On Ubuntu 18.04 LTS. Disponível via URL: <<https://www.ostechnix.com/install-apache-mysql-php-lamp-stack-on-ubuntu-18-04-lts/>>. Acesso em 10 de mar. de 2020.

PHP, PHP Releases. Disponível via URL: <<https://www.php.net/>>. Acesso em 23 de mai. de 2020.

PKGS, Ubuntu Repositories. Disponível via URL: <<https://ubuntu.pkgs.org/>>. Acesso em 15 de abr. de 2020.

Putty, Download putty. Disponível via URL: <<https://www.putty.org/>>. Acesso em 15 de mar. de 2020.

QNAP. “O que é Virtualização”. QNAP, 2019. Disponível via URL: <<https://www.qnapbrasil.com.br/blog/post/o-que-e-virtualizacao>>. Acessado em 30 de jun. de 2020.

Rackspace, Install MySQL Server on the Ubuntu operating system. Disponível via URL: <<https://support.rackspace.com/how-to/install-mysql-server-on-the-ubuntu-operating-system/>>. Acesso em 10 de mar. de 2020;

Ranking. “DistroWatch Page Hit Ranking” , 2020 . Disponível via URL: <<https://distrowatch.com/dwres.php?resource=popularity>>. Acessado em 30 de jun. de 2020.

SAMBA, About Samba. Disponível via URL: <<https://www.samba.org/>>. Acesso em 05 de mai. de 2020.

Site Antigo, História do MySQL. Disponível via URL: <<https://siteantigo.portaleducacao.com.br/conteudo/artigos/informatica/historia-do-mysql>>. Acessado em 29 de jun. de 2020.

SSH.COM. “SSH Protocol”. SSH.COM, 2020. Disponível via URL: <<https://www.ssh.com/ssh/protocol/#how-does-the-ssh-protocol-work>>. Acesso em 29 de mar. de 2020.

Tecmundo. “O que é Virtualização?”, 2009. Disponível via URL: <<https://www.tecmundo.com.br/web/1624-o-que-e-virtualizacao-.htm>>. Acessado em 30 de jun. de 2020.

Tecnologia Sem Segredos. “Visão geral sobre virtualização”, 2020. Disponível via URL: <<https://tecnologiasemsegredos.wordpress.com/category/virtualizacao/>>. Acessado em 30 de jun. de 2020.

TEO, Configurando SSH Keys para acesso por chave pública. Disponível via URL: <<https://www.todoespacoonline.com/w/2015/08/configurando-ssh-keys-para-acesso-por-chave-publica/>>. Acesso em 10 de mar. de 2020;

Ubuntu Server 14.04.6 LTS, Download ISO amd64. Disponível via URL: <<http://releases.ubuntu.com/14.04/>>. Acesso em 10 de mar. de 2020;

Ubuntu Server 16.04.6 LTS, Download ISO amd64. Disponível via URL: <<http://releases.ubuntu.com/16.04/>>. Acesso em 10 de mar. de 2020;

Ubuntu Server 18.04.4 LTS, Download ISO amd64. Disponível via URL: <<https://ubuntu.com/download/server/thank-you?version=18.04.4&architecture=amd64>>. Acesso em 10 de mar. de 2020;



UBUNTU. “Ubuntu Releases”. Canonical, 2020 . Disponível via URL: <<https://assets.ubuntu.com/v1/ac31b45e-release-chart-desktop-big.png>>. Acessado em 30 de jun. de 2020.

UFW, Ubuntu Wiki. Disponível via URL: <<http://wiki.ubuntu-br.org/UFW>>. Acesso em 15 de abr. de 2020.

Ultradownloads, Download ImgBurn 2.5.8.0. Disponível via URL: <<http://ultradownloads.com.br/download/ImgBurn/>>. Acesso em 10 de mar. de 2020;

VirtualBox, “Welcome to VirtualBox.org!”. Oracle, 2020. Disponível via URL: <<https://www.virtualbox.org/>>. Acesso em 08 de mar. de 2020.

Vsftpd, Probably the most secure and fastest FTP server for UNIX-like systems. Disponível via URL: <<https://security.appspot.com/vsftpd.html>>. Acesso em 05 de mai. de 2020.

VSFTPD.CONF, VSFTPD Beasts. Disponível via URL: <[http://vsftpd.beasts.org/vsftpd\\_conf.html](http://vsftpd.beasts.org/vsftpd_conf.html)>. Acesso em 30 de abr. de 2020.

WIKILIVROS. “Manual do Ubuntu / Uma breve história do Ubuntu”. WIKILIVROS, 219. Disponível via URL: <[https://pt.wikibooks.org/wiki/Manual\\_do\\_Ubuntu/Uma\\_breve\\_hist%C3%B3ria\\_do\\_Ubuntu#:~:text=Ubuntu%20foi%20concebido%20em%202004,impediam%20o%20seu%20uso%20disseminado.&text=Com%20o%20sistema%20Debian%20como,come%C3%A7ou%20a%20construir%20o%20Ubuntu.>](https://pt.wikibooks.org/wiki/Manual_do_Ubuntu/Uma_breve_hist%C3%B3ria_do_Ubuntu#:~:text=Ubuntu%20foi%20concebido%20em%202004,impediam%20o%20seu%20uso%20disseminado.&text=Com%20o%20sistema%20Debian%20como,come%C3%A7ou%20a%20construir%20o%20Ubuntu.>)>. Acessado em 30 de jun. de 2020.

Wikipedia, MySQL. Disponível via URL: <<https://pt.wikipedia.org/wiki/MySQL>>. Acessado em 29 de jun. de 2020.

Wikipédia, Samba (software). Disponível via URL: <[https://pt.wikipedia.org/wiki/Samba\\_\(software\)](https://pt.wikipedia.org/wiki/Samba_(software))>. Acesso em 30 de Jun de 2020.

Wikipédia, Secure Shell. Disponível via URL: <[https://pt.wikipedia.org/wiki/Secure\\_Shell](https://pt.wikipedia.org/wiki/Secure_Shell)>. Acesso em 08 de mar. de 2020.

ZAGO, Samba - Como funciona, como usar, serviços do servidor Samba. Disponível via URL: <<http://www.zago.eti.br/samba/samba.html#toc5>>. Acesso em 30 de jun de 2020.

## 5 APÊNDICES

### APÊNDICE A – Github ATM1s2020

[\[Howto\]](#) Ubuntu Server 18.04: Procedimentos de Instalação e Configuração do Sistema Operacional

[\[Howto\]](#) VirtualBox 6.1: Procedimentos de Instalação e Configuração do Sistema de Virtualização

[\[Howto\]](#) Apache: Procedimentos de Instalação e Configuração do serviço de acesso Web (Servidor Web)

[\[Howto\]](#) MySQL Server : Procedimentos de Instalação e Configuração do Sistema Gerenciador de Banco de Dados (SGBD) Web

[\[Howto\]](#) PHP : Procedimentos de Instalação e Configuração do ambiente de desenvolvimento web (Stack de desenvolvimento PHP)

[\[Howto\]](#) Secure Socket Shell (SSH): Procedimentos de Instalação e Configuração do acesso remoto ao servidor Web

[\[Howto\]](#) VSFTPD: Procedimentos de Instalação e Configuração do serviço de transferência de arquivo (FTP)

[\[Howto\]](#) SAMBA : Procedimentos de Instalação e Configuração do serviço de compartilhamento de arquivos do servidor (SAMBA)

[\[Howto\]](#) UFW : Procedimentos de Instalação e Configuração do serviço de Firewall