

**servidoras**

**aprendendo  
a colocar  
sites no ar**

**Aviso:** Nós da MariaLab gostamos de gendricar termos técnicos como 'servidora' e 'roteadora' para o feminino, como uma forma de questionar a noção de que o campo de infraestruturas técnicas é um setor de domínio masculino - e não limitar o imaginário coletivo e nossas próprias noções a esses conceitos traduzidos do inglês.

# 1 - Internet, sites, servidoras e a tal nuvem

**8\_ 1.1 - Internet:** O acesso à internet passou a estar disponível a 51% da população mundial, mas e os outros 49%? Aqui iremos entender o que é a internet, a sua estrutura e as políticas envolvidas para garantir conectividade aos usuários.

**9\_** O que é internet

**12\_** Infraestrutura da Internet

**15\_** Camadas da internet

**20\_ 1.2 - Sites:** Todas nós acessamos sites pela internet, mas sabemos o caminho percorrido do nosso dispositivo até ver o conteúdo em nossa tela? Aqui iremos ver como funcionam as páginas que acessamos.

**21\_** O que é site

**24\_** Seu nome na Internet

**27\_** Seu IP na internet

**32\_** Protocolos e Portas

**36\_** Como um site vai pro ar?

**38\_ 1.3 - Servidoras:** Se hoje temos uma sociedade repleta de computadores e smartphones, devemos isso a um homossexual que usou como referência o trabalho de uma mulher. Graças aos trabalhos dele, se criaram tecnologias incríveis acessadas pela rede. Mas onde estão hospedadas estas tecnologias?

**39\_** O que são Servidores/as?

**43\_** Configuração de Hardware da Servidora

[48\\_](#) Sistemas Operacionais para Servidoras

[50\\_](#) Servidora Web

[55\\_](#) Arquitetura Cliente/Servidora

[59\\_](#) **1.4 - Nuvem:** Já ouviu falar que seus dados estão na nuvem? Será que está no céu? Vamos entender essa história...

[60\\_](#) Essa tal “nuvem”

[63\\_](#) DataCenter

[66\\_](#) Modelos de Nuvem

## 2. Assumindo o controle

[69\\_](#) **2.1 - Criando uma Servidora:** Antes de tudo é necessário planejamento. Precisamos pensar qual será o propósito da nossa servidora e o que precisaremos para podermos colocar a mão na massa e criar. Assim como fazer um bolo: primeiro reunir os ingredientes para depois começar o preparo.

[70\\_](#) Linux e softwares de código aberto

[74\\_](#) Aluguel de Servidoras

[82\\_](#) Reunindo as informações para criar nossa Servidora

[84\\_](#) **2.2 - Colocando a mão na massa:** Agora criaremos nossa servidora. Para fins de aprendizado, ela será criada dentro de nosso computador com uma técnica chamada virtualização e faremos todas as configurações necessárias para ter a nossa servidora Web.

[85\\_](#) Criando uma Máquina Virtual

[96\\_](#) Configurando o Ubuntu Server

[105\\_](#) Conhecendo o Linux Server

[112\\_](#) Navegando pelo Terminal

[133\\_](#) Instalação de Programas

[140\\_](#) **2.3 - Olá Mundo!:** Chegou a hora de colocarmos nosso site no ar! Iremos instalar o servidor web Apache e criar a nossa página inicial.

[141\\_](#) Instalando uma Servidora Web

[154\\_ Gerenciando o acesso aos meus arquivos](#)

[159\\_ Transferindo arquivos com o protocolo FTP/SFTP](#)

## 3. Colocando tudo no ar

**164\_ 3.1 - Preparando o ambiente:** Para podermos instalar nossos primeiros serviços, iremos configurar um interpretador da linguagem PHP e um serviço de Banco de Dados. Com esses dois configurados, estaremos com tudo pronto pra instalar nossos serviços.

[165\\_ Ambiente LAMP - Linux + Apache + MySQL + PHP](#)

[166\\_ Instalando e Configurando o Interpretador PHP](#)

[172\\_ Instalando e Configurando o Banco de Dados MySQL](#)

**181\_ 3.2 - Criando nosso blog com o Wordpress:** O WordPress é um projeto de código aberto que você pode usar para criar sites, blogs ou aplicativos. Com ele iremos criar o nosso Blog.

[182\\_ Introdução ao Wordpress](#)

[183\\_ Instalando o Wordpress](#)

[194\\_ Configurando nosso blog](#)

**197\_ 3.3 - Criando nossa rede social com HumHub:** HumHub é um software gratuito para redes sociais e uma plataforma construída para lhe dar as ferramentas para tornar a comunicação e a colaboração fáceis e bem sucedidas. Com ele iremos criar nossa rede social.

[198\\_ Introdução ao Humhub](#)

[199\\_ Instalando o Humhub](#)

**209\_ 3.4 - Criando nossa loja com o PrestaShop:** A PrestaShop é uma solução completa, inovadora e eficiente para criar sua loja virtual.

[210\\_ Introdução ao PrestaShop](#)

[211\\_ Instalando o PrestaShop](#)

# Servidoras Feministas

## Introdução

A evolução da tecnologia digital afetou a realidade social, estando presente em todas as esferas de nossas atividades, gerando novas relações a serem reguladas pelo sistema jurídico. A falta de leis consolidadas com relação a privacidade digital exige que cada indivíduo esteja preparado para enfrentar as novas situações e a invasão de privacidade decorrentes dessa nova visão sobre a privacidade entrelaçada com o surgimento de novas tecnologias de informação.

Para abordarmos o tema, no ano de 2017 iniciamos um projeto na [MariaLab](#) sobre Infraestruturas Feministas, dedicado a disseminar e incentivar a troca de conhecimentos sobre servidoras, redes autônomas, hacking e tecnologia feminista por grupos de mulheres.

Em linhas gerais, o nosso discurso acerca das “redes e tecnologias autônomas”, é de afirmar possibilidades de resistência à [vigilância massiva](#) e ao [controle das comunicações](#) realizadas por Estados e grandes corporações. Aceitarmos que isso ocorra, significa abrir mão da nossa privacidade e ao fazer isso estamos concordando com o fato de que qualquer governo ou corporação que se utilize desses recursos de vigilância, tenha acesso ilimitado às informações coletadas sobre nossas vidas.

Sob o olhar feminista, este conceito também significa resistir a um sistema e a uma infraestrutura de comunicação e informação que se constituiu em um espaço patriarcal, masculinizado, e sob o controle do norte global, estando as mulheres (mais ainda as mulheres do sul global) invisibilizadas, ou até mesmo [excluídas nesse processo](#). Prova disso, é a invisibilidade das criações de mulheres nas áreas de STEAM e da falta de presença de mulheres decidindo como essas estruturas serão incluídas na sociedade.

O direito fundamental à privacidade deve ser interpretado, não só como tutela de um interesse individual, mas como fundamento do Estado Democrático de Direito. Em termos práticos, tudo que está disponível na internet que for coletado por esses órgãos e que não “agrade” ou sirva aos interesses de quem vigia você, hoje ou daqui a alguns anos, pode te tornar um alvo. Pode ser uma opinião sobre um governante ou marca nas redes sociais que te levem a ser processado ou até o valor diferenciado a ser cobrado por produtos e serviços baseados em análises do que você consome ou seu histórico de navegação.

Olhando o cenário por uma perspectiva de gênero, enxergamos uma grande dependência de terceiros (empresas e/ou profissionais técnicos homens cisgêneros) para gestão de dados, equipamentos eletrônicos e serviços hospedados na web (sites, email, repositório de arquivos, etc). Neste contexto verificamos a escassez de mulheres na tecnologia, onde a maioria de nós não encontra conteúdos acessíveis para a produção de um conhecimento de acordo com as nossas realidades.

Com isso, ainda em 2017, iniciamos dois cursos presenciais focando em coletivos de mulheres: “Redês Autônomas” e “Servidoras Feministas”. Este é o resultado ampliado do conteúdo aplicado em sala de aula para que possam ter acesso a este conhecimento e compartilhar com outras mulheres sem que a geografia impeça o aprendizado.

## **Como ler o material**

Esta guia é destinada a quem tem curiosidade de entender como funciona a maior rede de computadores do mundo, conhecida como Internet. E para quem já tem esse conhecimento e quer ter seu endereço(*site*) nessa teia(*web*) e ir além aprendendo a criar e gerir o espaço deste endereço.

Este material foi dividido em 3 seções complementares a fim de facilitar o entendimento dos conceitos e técnicas que serão apresentados.

Na primeira seção **Internet, sites, servidores e a tal nuvem** nós veremos todo o ecossistema por trás dos sites que visitamos na internet, o que são, onde ficam, do que são compostos e as políticas envolvidas para conseguirmos acessá-los.

Na segunda seção **Assumindo o controle** iremos entender mais sobre servidoras (onde os sites são criados), sua estrutura e, também, iremos criar uma servidora com Sistema Operacional Linux, que é o mais recomendado. Caso você não saiba nada sobre Linux ou nunca teve contato, aconselhamos muito que assista esse [vídeo](#) antes, pois assim conseguirá aproveitar melhor.

Na terceira seção **Colocando tudo no ar** iremos instalar os programas necessários em nossa servidora para que nosso site esteja disponível em nossos navegadores de Internet. Caso você já tenha feito algum curso de programação front-end, essa seção será um extra a seu aprendizado.

Dentro das seções, além do conteúdo acima, você irá encontrar dicas de leitura complementar, dicas de segurança e privacidade, indicações de outras guias para aprofundamento e exercícios para que a leitura deste material seja dinâmica. Em relação aos exercícios tudo o que for necessário será disponibilizado em nosso site para download.

Aconselhamos que você crie uma pasta em seu computador desde já, para centralizar o conteúdo e também separar um caderninho de anotações. Juntas estas tecnologias são imbatíveis!

Agora que você já sabe o que vamos fazer e o que vamos precisar, podemos começar. Boa leitura, bons estudos!

# 1. Internet, sites, servidoras e a tal nuvem

## 1.1 INTERNET



# O que é Internet

A Internet é uma rede que interliga computadores do mundo inteiro. O que faz a Internet tão poderosa assim é um processo da informática que atende pelas siglas TCP/IP (Protocolo de Controle de Transferência/Protocolo Internet). Todos os computadores que entendem essa linguagem são capazes de trocar informações entre si. Assim, eles podem se conectar a máquinas e dispositivos de diferentes tipos, sejam eles computadores, smartphones, Smart TVs, etc.

A internet surge para fins bélicos nos Estados Unidos, na década de 1960, na época da Guerra Fria, a partir de uma rede de informações militares que interligava centros de comando e de pesquisa. Para atender à necessidade militar de proteger os sistemas de defesa do país no caso de um ataque nuclear.

A rede não tinha um “centro” que poderia servir de alvo principal ao inimigo.

Nos anos 1970, a internet começa a ser utilizada pela comunidade acadêmica mundial e, em 1975, são feitas as primeiras ligações internacionais. Nesse período, os computadores conectados não passavam de 200.

Em 1997, segundo a Direct Marketing Association (DMA) e a Price Waterhouse, empresas norte-americanas de consultoria em marketing, o número de usuários chegou a 60 milhões em todo o mundo. Um estudo recente da União Internacional de Telecomunicações (UIT), agência das Nações Unidas, apontou que mais da metade da população mundial está conectada à internet. São 3,9 bilhões de pessoas (o equivalente a 51% da população mundial) ligadas à rede.

A Internet tem revolucionado a comunicação mundial ao permitir, por exemplo, a conversa entre usuários a milhares de quilômetros de distância pelo preço de uma ligação local. O grande número de pessoas que a utilizam também é responsável pelo maior problema da rede: o congestionamento e a lentidão no acesso aos serviços. Com a Internet surge a expressão ciberespaço, que significa o espaço virtual e sem fronteiras, no qual circulam as milhares de informações veiculadas na rede.

De acordo com um levantamento de 2019 da [Visual Capitalist](#), em 1 minuto de internet acontece:

- 📊 1 milhão de logins no Facebook
- 📊 41,6 milhões de mensagens enviadas no Facebook Messenger e no WhatsApp
- 📊 3,8 milhões de buscas no Google
- 📊 4,5 milhões de vídeos vistos no YouTube
- 📊 390 mil aplicativos baixados na Google Play e na Apple Store
- 📊 87 mil pessoas tuitando
- 📊 695 mil horas de vídeos assistidas na Netflix
- 📊 347 mil “scrolls” (movimento de baixar a tela) no Instagram


Isso demonstra o volume de informações pessoais que estão sendo coletadas por usuários e processadas em servidores por empresas privadas, onde muitos destes dados são compartilhados com nosso consentimento sem sabermos de fato os seus fins. Só o Facebook gera mais de 500TB de dados a cada 24 horas, onde parte desses dados é usada para direcionar propagandas e outros são cedidos para empresas.

Você já leu algum termo de uso antes de clicar em “Li e aceito os termos de uso”? Nele está descrito, ou deveria estar, tudo que a empresa coleta de dados e o que ela faz com os dados. O objetivo dos termos de uso é selar um contrato entre o usuário e a provedora do serviço, assim ambas as partes têm amparo jurídico para resolver disputas a respeito destes dados gerados pelas plataformas.

Vamos pensar nas seguintes situações:

📊 Se o aplicativo de um banco dissesse que compartilha seus dados de transação para lojas, prestadoras de serviços e governos desconhecidos sem dizer para que eles irão utilizar, você pensaria duas vezes antes de instalar?

📊 Imagine se uma loja te cobrasse mais caro porque sabe que seus hábitos de compra giram em torno de um valor específico, ou porque o dispositivo que está usando é da Apple.

 Vamos supor que ao dar entrada em um novo plano de saúde seu pedido é negado, pois descobriram que você esteve doente através das redes sociais ou histórico de compras em uma farmácia.

Essas situações parecem absurdas e que rompem com seu direito à privacidade, certo? Mas infelizmente as duas últimas realmente estão acontecendo. Por isso, temos que pensar duas vezes antes de compartilhar informações pessoais na internet.

## Notícias relacionadas

MENSTRUAPPS – Como transformar sua menstruação em dinheiro (para os outros)?

Facebook usa inteligência artificial para prever o comportamento de usuário para anunciantes

O que está em jogo quando você dá seu CPF na hora da compra

Entidade diz que planos de saúde avaliam cobrar mais de quem tem mais risco

Preço de produtos em sites muda quando pesquisado em iPhone, Android e computador

Você não aguentaria a verdade sobre os anúncios do Facebook, mostra estudo de Harvard

## Dicas de segurança

Fuzzify.me – extensão de código aberto para navegador que ajuda a impedir anúncios direcionados no Facebook

# Infraestrutura da Internet

A Internet é organizada na forma de uma teia e as informações na Internet passam por vários computadores até chegarem aos seus destinos. Se você pretende acessar um site na Austrália, por exemplo, não é necessário fazer um interurbano internacional como em uma ligação telefônica, basta conectar-se a um computador ligado à Internet em sua cidade. Este computador faz um pedido ao provedor de internet que traça uma rota até chegar ao destino. Vamos entender essa viagem:

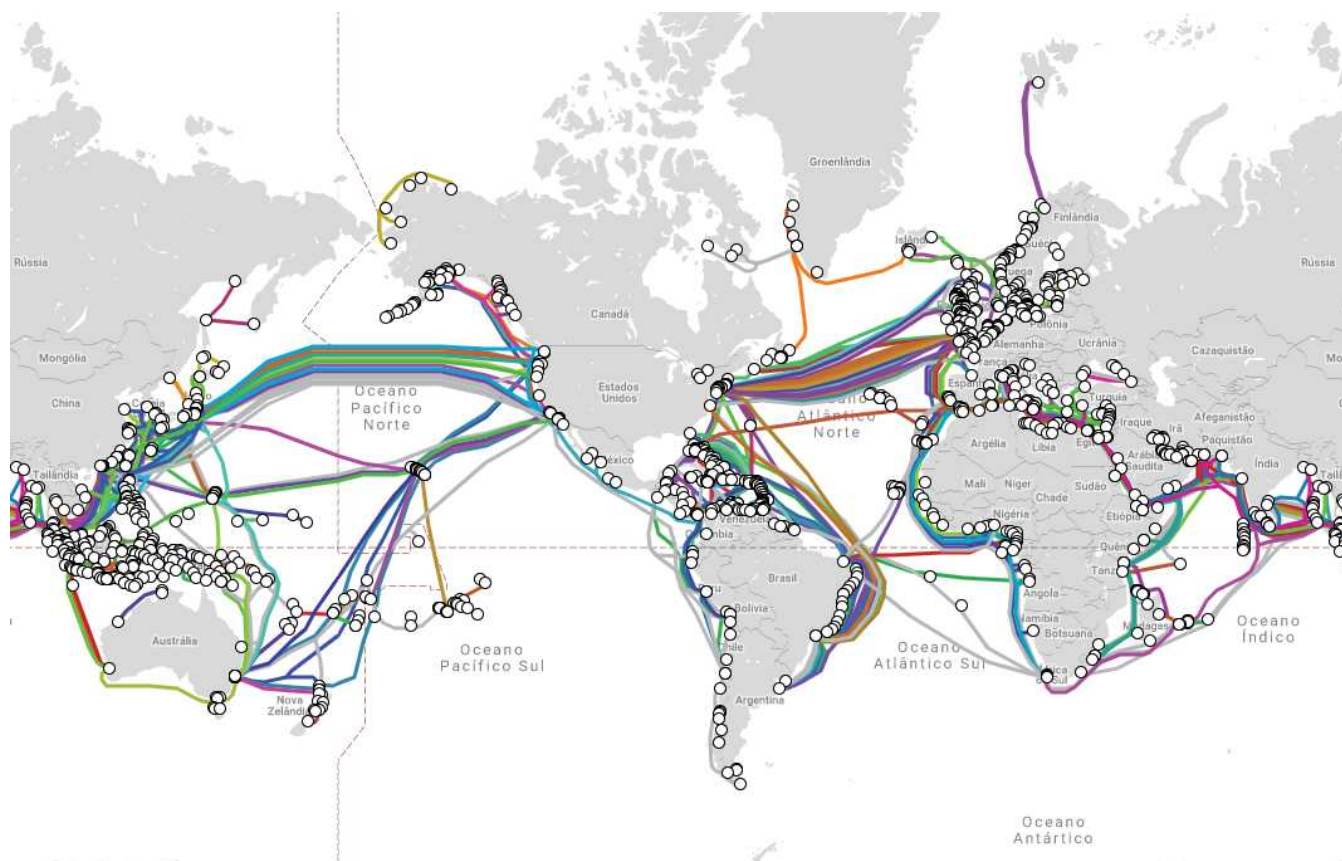
Vídeo: [como funciona a internet por dentro](#)

**1\_** O pedido feito no navegador passa por um cabo de internet que geralmente seguem o mesmo caminho das linhas telefônicas, dividindo praticamente a mesma estrutura subterrânea. O meio mais utilizado para fornecer internet era o cabo de cobre. Porém, esse recurso não oferece para o sinal de internet a proteção necessária contra as interferências de outras redes, fazendo com que a conexão de internet sofra bastante oscilações de sinal e perda de velocidade de conexão. Atualmente, as principais fornecedoras de internet já oferecem o serviço via fibra óptica como opção. Esse recurso é composto por um material dielétrico, ou seja, imune a ondas eletromagnéticas de outras redes que podem causar interferências no seu sinal de internet. Dessa forma, a fibra óptica oferece estabilidade, mais qualidade de conexão e velocidade para a sua internet.

**2\_** O pedido navega por esses cabos até uma central em sua rua, geralmente localizadas em postes ou bueiros. Essas centrais locais estão conectadas com uma central maior, que é de responsabilidade de sua operadora de internet, localizadas em um ponto específico da sua região.

**3\_** Na central maior existem computadores de alta capacidade, com grande poder de processamento e conexões velozes, conhecidas como servidores (para nós, As Servidoras), controladas por universidades, empresas e órgãos do Governo. Essa empresa ou instituição cobra taxa mensal de cada usuário para

cobrir, entre outros, os custos da conexão com a rede. Essa forma de funcionamento garante um custo baixo de conexão (ignore os impostos brasileiros, esta é outra discussão). Essas servidoras encaminham a requisição para as servidoras onde se encontra o conteúdo. No caso de sites internacionais, ela é transmitida por meio de cabos submarinos que partem do Brasil para os Estados Unidos, por exemplo.



Backbones submarinos distribuídos pelo planeta Terra. Disponível em: <https://www.submarinecablemap.com>

**4\_** Na imagem acima você pode ter uma ideia do quão complexo é esse emaranhado. Todas aquelas linhas coloridas em destaque que você vê no mapa mundial representam cabos submarinos que interligam um ponto de terra a outro. Existem centenas de cabos como esses, cumprindo as mais variadas rotas. A longa jornada é transposta em questão de milissegundos.

**5\_** Quando o pedido chega ao destino final, ele é validado e a seguir um pacote com a página solicitada (no caso a página inicial) é gerado como resposta a esse pedido, remetido para você, percorrendo o mesmo caminho de volta, e aberto no seu navegador.

Ou seja, da próxima vez que você clicar em um simples link, pode ser que a sua requisição precise viajar milhares de quilômetros de distância até que ela possa chegar ao destino final e voltar. E tudo isso em poucos segundos.

Agora vamos refletir: Se a transmissão de dados é física e passa por centrais que redirecionam informações a seus destinos, o que impede que essa transmissão seja interceptada ou censurada?

Em julho de 2016, o WhatsApp (da empresa Facebook) foi bloqueado pela 4ª vez no país inteiro a partir de uma ordem judicial da 2ª Vara Criminal de Duque de Caxias, no Rio de Janeiro. O motivo: O WhatsApp se recusa a ceder o conteúdo e acaba bloqueado por desrespeito à decisão judicial, com impacto nos 100 milhões de usuários do serviço.

Esse bloqueio foi executado pelas operadoras de internet. Isso evidencia duas coisas: é possível que os provedores de internet saibam quais sites cada usuário acessa e é possível que esse acesso seja censurado em âmbito nacional através do Estado.

Dado isso, temos que avaliar quais provedores tem compromisso com a privacidade do usuário e como fazer para garantir que nossos direitos estejam preservados.

## Notícias relacionadas

Bloqueios do WhatsApp:  
o que pode ser feito  
para que eles acabem

Relatório "Sobre o  
bloqueio dos sites pelo  
direito de escolha Women  
on Waves e Women on Web"

## Dicas de segurança

Como navegar na web sem deixar rastros

# Camadas da Internet

De acordo com a [Wikipédia](#), a governança compreende todos os processos de governar – seja pelo governo de um estado, por um mercado ou por uma rede – sobre um sistema social (família, povo, organização formal ou informal, um território ou através de territórios) ou através de leis, normas, poder ou linguagem de uma sociedade organizada. De forma simplificada, poderia ser descrito como os processos políticos que existem em e entre instituições formais.

Quando queremos falar de governança no que tange a internet e os meios que ela alcança, usamos a palavra “cibernética”. Por exemplo, quando falamos que no Brasil as operadoras não deveriam fazer distinção de conteúdo (Redes sociais grátis, App de taxi grátis, App de carona pago, etc) para oferecer seus serviços pois isso fere o Marco Civil da Internet, estamos falando de cibernética.

As questões relacionadas a Governança de internet são divididas em 4 temas centrais:

- 🔌 Gerenciamento e infraestrutura de recursos de internet
- 🔌 Uso da Internet, incluindo crimes cibernéticos e segurança na rede
- 🔌 Assuntos relevantes para a Internet mas que têm um impacto muito maior do que a Internet e pelas quais as organizações existentes são responsáveis, como os direitos de propriedade intelectual (DPI) ou o comércio internacional.
- 🔌 Desenvolvimento da governança da internet, em particular o desenvolvimento dessa capacidade em países em desenvolvimento

A Fundação Diplo(2013), classifica a governança da Internet agrupando as principais questões (em torno de 40-50), relacionadas aos 4 temas acima, nas cinco categorias a seguir:

🔌 **Infraestrutura e padronização:** trata temas de telecomunicações e suas concessões de serviços pelo Estado (no Brasil, o órgão responsável pela

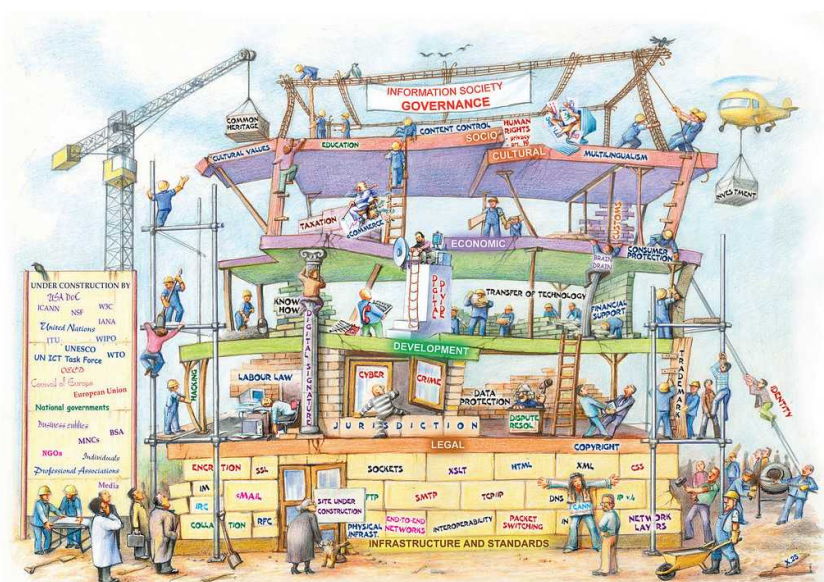
regulamentação é a ANATEL), se discute a criação e implementação de padrões, processos e protocolos de comunicação entre computadores e outros dispositivos que se conectam na rede.

**Legal:** aqui se discute os mecanismos legais globais e locais da Rede que visa prover coexistência das legislações dos países sem ferir sua jurisdição, preservar a liberdade e os direitos individuais, combater os crimes digitais, preservar a propriedade intelectual (autores, marcas e patentes) e garantir autonomia.

**Econômico:** trata do comércio eletrônico, proteção do consumidor, regras fiscais, transações financeiras e bancos digitais.

**Desenvolvimento:** trata temas de acesso universal e estratégias para superar o abismo digital entre países desenvolvidos e em desenvolvimento.

**Sociocultural:** aborda questões de Direitos Humanos, Políticas de Conteúdos, Privacidade e Proteção de Dados, Multilinguismo e Diversidade Cultural, Inclusão Digital, Educação e Segurança para crianças e adolescentes.



A ilustração foi retirada de "KURBALIJA, J. (2014) An Introduction to Internet Governance. Diplofoundation" e tem licença CC BY-NC-ND 3.0. Acessado em 23/01/2021: <https://www.diplomacy.edu/sites/default/files/An%20Introduction%20to%20IG%206th%20edition.pdf>

A imagem acima, demonstra de forma lúdica essas classificações refletindo as abordagens políticas acima mencionadas no WGIG 2004 (Grupo de Trabalho sobre Governança da Internet) e IGF 2006/07 (Fórum de Governança da Internet) quanto às pesquisas acadêmicas neste campo.



Esta classificação foi desenvolvida em 1997 com ajustes constantes baseados no feedback dos estudantes (cerca de 1542 estudantes a partir de 2013), resultados de pesquisas e percepções do processo político.

Fóruns nacionais como o “Observatório da Internet”, mantido pelo CGI (Comitê Gestor da Internet), e internacionais como o IGF são espaços para defender a rede e seus princípios e qualquer cidadão pode participar.

Sobre o nosso Estado, o Marco Civil, conhecido como a “Constituição da Internet”, preencheu uma lacuna na legislação brasileira definindo direitos e responsabilidades relativas à utilização dos meios digitais, ao invés de apenas criminalizar condutas nesses espaços. Por esse motivo, tornou-se referência para outros países do mundo não somente por seu conteúdo, mas também pelo amplo processo participativo que lhe deu origem.

O texto da lei se originou de um debate público realizado em 2009 pelo Ministério da Justiça em parceria com o Centro de Tecnologia e Sociedade, da Fundação Getúlio Vargas, contando com o apoio do Ministério da Cultura no uso da plataforma CulturaDigital.br (hoje em <http://pensando.mj.gov.br/marcocivil/>). Tendo recebido contribuições da sociedade civil organizada, da comunidade empresarial, de representantes das áreas técnica e acadêmica e de cidadãos comuns interessados no tema, o anteprojeto de Lei foi inovador por adotar uma plataforma que permitia uma maior interação entre os participantes, garantindo que cada contribuição fosse vista e comentada por todos os outros usuários engajados no debate, garantindo uma melhor sistematização de um texto pelo governo.

Todos os dados produzidos no debate foram organizados e colocados de forma aberta em consulta pública no link <http://pensando.mj.gov.br/marcocivil/pauta/>, podendo ser baixado e utilizado a fins de pesquisa. Os diferentes temas foram agrupados em eixos (Privacidade na rede, Registros de acesso, Neutralidade e Outros temas) sobre os quais foi realizado debate público a fim de informar o governo sobre a melhor forma de tratá-los no Decreto.

É importante nos atentarmos a esse tema, pois isso interfere diretamente sobre nosso direito à privacidade e as questões relacionadas à acesso e gênero em meios digitais. Segundo o próprio Tim Berners-Lee, criador da World Wide Web – www, a rede não funciona para as mulheres e alimenta nova era de abusos generalizados contra elas. Esses abusos podem ser em forma de perseguição, disseminação de imagens íntimas sem consentimento, ofensas sexuais e privações ao acesso à internet.

## Saiba mais

### Notícias

O uso do celular  
contra o coronavírus.  
E os limites da  
vigilância

A sociedade mais  
vigiada do mundo: como  
a China usa o  
reconhecimento facial

Graças ao Facebook, sua  
operadora de celular te  
vigia agora mais perto  
do que nunca

NET, Claro e Vivo  
bloqueiam acesso a site  
com informações sobre  
aborto seguro

### Sites

Observatório da  
Internet no Brasil

CGI.br – O Comitê  
Gestor da Internet no  
Brasil

Coalizão Direitos  
na Rede

DISSENSO.ORG – plataforma  
destinada à promoção da  
liberdade de expressão no  
ambiente digital

## Filmes/Série

Freenet? - Filme documentário colaborativo sobre o futuro da liberdade na Internet.

Não consentida - vídeo reportagem sobre disseminação não consentida de imagens íntimas

XPLOIT - A minissérie Xploit, que é uma realização da TVDrone / Actantes em associação com a Heinrich-Böll Stiftung e apoio da Rede TVT, pretende abordar uma guerra silenciosa que acontece longe dos PCs, laptops e dispositivos móveis mas cujo resultado interfere diretamente em nossas vidas online e offline.

## Textos

O impacto do Marco Civil sobre a proteção da privacidade no Brasil

## Dicas de segurança

Autodefesa contra Vigilância: Dicas, ferramentas e tutoriais para uma maior segurança nas comunicações online

## 1.2 SITES

# O que é Site



Um website - justaposição das palavras inglesas web (rede) e site (sítio, lugar) - ou site é uma página ou um conjunto de páginas web, isto é, de hipertextos acessíveis geralmente pelo protocolo HTTP na internet. O conjunto de todos os sites públicos existentes compõe a World Wide Web (internet). Para acessar um site é necessário saber seu endereço, conhecido como *URL*.

URL é o endereço de um recurso disponível em uma rede, seja a rede internet ou local, e significa em inglês Uniform Resource Locator, e em português é conhecido por Localizador Padrão de Recursos.

Em outras palavras, url é um endereço virtual com um caminho que indica onde está o que o usuário procura, e pode ser tanto um arquivo, como uma máquina, uma página, um site, uma pasta etc. Url também pode ser o link ou endereço de um site.



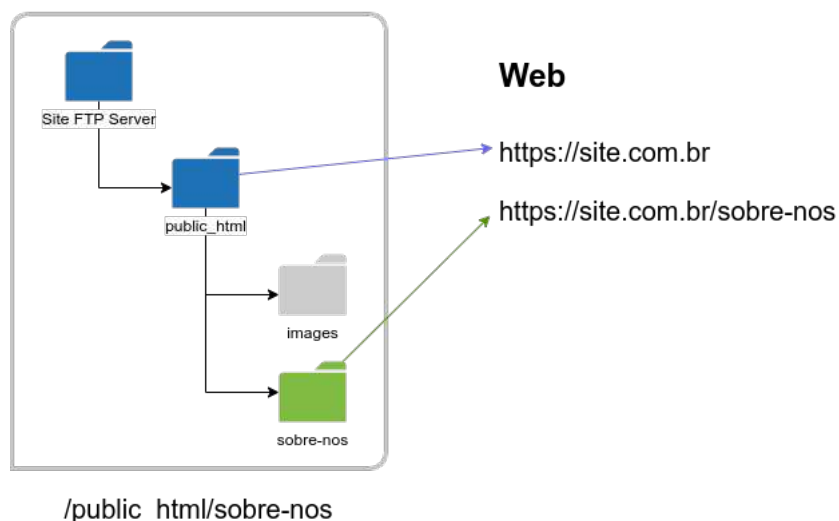
Uma URL é composta de 3 elementos principais:

um **protocolo**, que indica a forma como os dados serão transferidos, o HTTP e o HTTPS são os mais comuns para sites e blogs. Iremos ver mais sobre protocolos mais para frente.

um **domínio**, que é o nome do site (normalmente formado pelo nome da empresa), seguido de .com, .com.br, .br, .org entre outros que você irá definir ao comprar o seu domínio.

um **caminho**, o qual indica a página que você está acessando naquele domínio, normalmente é antecedida por uma barra /. Pode ser seguido de um recurso, que é ainda mais específico ao que você está acessando, indicando, por exemplo, o conteúdo que está sendo lido (imagem, arquivo, página html, etc.).

As páginas num site são organizadas a partir do endereço do domínio onde fica a página principal, e geralmente residem no mesmo diretório de um servidor. As páginas são organizadas dentro do site numa hierarquia observável no URL. Quando vemos na URL as separações com "/", como por exemplo `http://site.com.br/sobre-nos/localizacao.html`, imaginamos que existe uma estrutura hierárquica dos arquivos que segue essa mesma lógica (pasta-raiz/sobre-nos/), mas a estrutura global do site que vemos na URL também pode não se refletir com a estrutura hierárquica dos arquivos do site.



Um site é acessado através da digitação de um endereço de internet que chamamos de domínio. O site da Marialab, por exemplo, está registrado sob o domínio marialab.org. Ao acessar o domínio marialab.org, digitando este endereço no seu navegador, você está acessando a página inicial da Marialab,

que está armazenada em um servidor de hospedagem.

Na página ou páginas de um website podem ser colocados textos, imagens, vídeos ou animações digitais. Estas páginas são carregadas através do protocolo de rede HTTP (Hypertext Transfer Protocol) e são visualizadas através de um navegador (browser).

Um website pode apresentar uma forma estática, sendo a sua extensão .html/.htm porque utiliza a linguagem ou código HTML ou em uma forma dinâmica, interativa, sendo as linguagens JavaScript e PHP as mais utilizadas.

Nem todo site é feito apenas de texto e imagens, também podem prestar um determinado serviço como o Gmail onde você vê seus e-mails ou o Facebook que é uma rede social. Nestes tipos de sites é um pouco mais complexo seu funcionamento para apresentar as páginas no navegador.

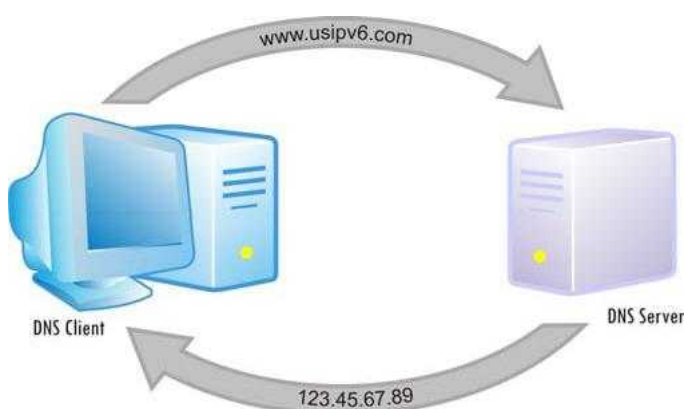
# Seu nome na internet

Domínio é um nome que serve para localizar e identificar conjuntos de computadores e serviços na Internet. O nome de domínio foi concebido com o objetivo de facilitar a memorização desses endereços, pois sem ele, teríamos que memorizar uma sequência grande de números, e dar flexibilidade para que o operador desses serviços altere sua infraestrutura com maior agilidade.

O espaço de nomes de domínio e endereços IP são recursos críticos para a internet, no sentido que requerem coordenação global. Cada endereço IP deve identificar um único equipamento, de forma que não é possível atribuir endereços IP de maneira descentralizada. Da mesma forma, um nome de domínio deve identificar o conjunto de computadores que o mantém. A organização responsável por atribuir nomes de domínio e endereços IP em nível global é a ICANN.

No Brasil, qualquer empresa/pessoa legalmente estabelecida, como pessoa jurídica (CNPJ) ou física (CPF), e que possua um contato em território nacional pode ter um domínio “.br”. Empresas estrangeiras poderão solicitar um domínio, desde que possuam um procurador legalmente estabelecido no Brasil.

Os servidores de diretórios responsáveis por prover informações como nomes e endereços das máquinas são normalmente chamados servidores de nomes (NS – Name Server). Na Internet, os serviços de nomes usados é o DNS, que apresenta



**Servidor de Nomes de Domínio**

uma arquitetura cliente/servidor, podendo envolver vários servidores DNS na resposta a uma consulta.

O Sistema de Nomes de Domínios, ou DNS, é o protocolo utilizado para resolução de nomes de domínio, permitindo que, através de um nome



legível, obtenha-se um endereço IP da máquina física onde o domínio está hospedado. Atualmente os servidores de DNS estão por toda parte. Organizados em uma hierarquia de nomes distribuída e escalável, os servidores são responsáveis por resolverem as consultas vindas do mundo inteiro (como os servidores raiz) ou de apenas uma rede restrita (como os servidores particulares), conforme configurados.

No topo da hierarquia dos servidores de DNS está a zona raiz de DNS, onde estão os servidores raiz (root servers). Responsáveis por indexar os servidores de mais alto nível dentro da hierarquia de DNS (em um arquivo chamado Arquivo de Zonas Raiz), conhecidos como servidores autoritativos, os servidores raiz são os elos que ligam a rede mundial e tornam o acesso à Internet algo possível a todos. Os servidores raiz são mantidos por diversas empresas e instituições ao redor do mundo, onde contam com suporte constante para que tudo se mantenha em ordem e em segurança.

Num sistema livre, o serviço é implementado pelo software BIND e usa como porta padrão a 53. Este serviço geralmente se encontra localizado no servidor DNS primário. O servidor DNS secundário é uma espécie de cópia de segurança do servidor DNS primário. Assim, ele se torna parte necessária para quem quer usar a internet de uma forma mais fácil, evita que hackers roubem os seus dados pessoais.

## Saiba mais

### Notícias

Brasileiros ignoram  
riscos de ataques contra  
roteadores domésticos

GhostDNS: malware que  
muda DNS de roteadores  
mira bancos e Netflix

Nova falha de DNS permite ataques DDoS em larga escala

## Dicas de segurança

Descubra se o seu roteador foi afetado pelo sequestro de DNS

## Fontes

**WIKIPEDIA, Domain Name System.** Disponível em

[http://pt.wikipedia.org/wiki/Domain\\_Name\\_System](http://pt.wikipedia.org/wiki/Domain_Name_System). Acesso em 25 de Junho de 2017

**Registro.br, Ajuda.** Disponível em <https://registro.br/ajuda.html>. Acesso em 25 de Junho de 2017

**LuizTools, Servidores raiz de DNS.** Disponível em

<http://www.luiztools.com.br/post/servidores-raiz-de-dns/>. Acesso em 25 de Junho de 2017

# Seu IP na internet

O **IP (ou Internet Protocol)** é uma identificação única para cada dispositivo conectado a uma rede, como um CPF. Vamos imaginar um telefone: todos os telefones têm um número atribuído para identificá-lo. Basta discar o número da pessoa com quem quero conversar para iniciar a comunicação. Com computadores e outros dispositivos que acessam a rede funciona da mesma forma.

Ao ligar o computador e conectá-lo a uma rede local, é atribuído ao aparelho um endereço IP que é comumente chamado de IP interno. A partir do momento em que o usuário começa a navegar na internet, ele passa a usar um segundo número, conhecido como IP Público ou Externo (que pode estar no computador ou no equipamento que dá acesso à internet, geralmente o roteador da operadora).

Saber o endereço IP público pode ser útil, por exemplo, para usar algum tipo de programa que acesse seu dispositivo remotamente pela internet ou acessar serviços fornecidos por alguma máquina, como um programa de compartilhamento de mídia (música, vídeos ou fotos).

O IP Público é definido pelo seu provedor de Internet e pode ser estático (não mudar) ou dinâmico (mudando de tempos em tempos). No site [MeuIP.com](http://MeuIP.com) você descobre qual é o IP público da sua máquina.

## IP, Máscara e Gateway



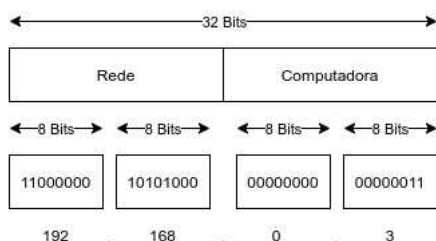
Lembram daquela tela da matrix toda preta com números 0 e 1 em verde caindo sobre a tela? Toda essa chuva de número representa o fluxo de informações (dados) que atravessam a matrix, que é uma rede de super computadores.

Todas as informações e tudo o que o seu computador está fazendo neste exato momento está sendo processado internamente com dois níveis de tensão, onde seu sistema de numeração natural é o sistema binário.

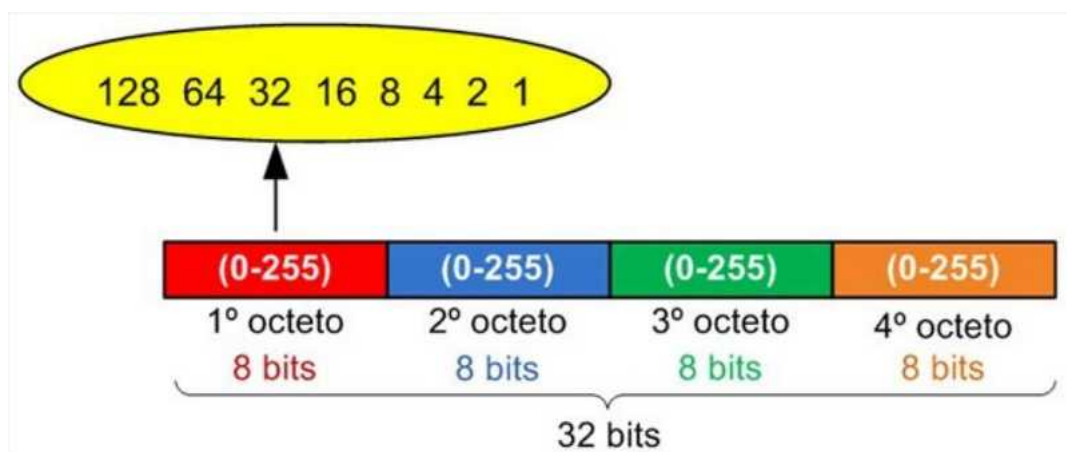
A base binária, assim como a decimal, é capaz de simbolizar todos os números possíveis e imagináveis. No entanto, ao contrário da forma mais comum, utiliza apenas os símbolos 0 e 1.

Se quisermos representar o “doze” em binário, por exemplo, se escreve 1100. E “treze”? 1101. “quatorze?” 1110. Talvez você já tenha percebido um padrão aí. O endereço IP é composto por 4 blocos de números de até 3 dígitos e separados por “.” (ponto). Os valores destes números variam entre 0 e 255, por exemplo, um endereço de IP pode ser 192.168.0.3 (quatro números entre 0 e 255 separados por pontos).

Essa divisão tem um propósito: Limitar a quantidade de computadores na rede.



Para isso os endereços de IP que serão atribuídos aos dispositivos (manualmente ou dinamicamente) devem seguir uma regra, definida a partir da configuração da Máscara de Sub-Rede. Máscara de sub-rede é a configuração que determina quantos IPs poderão existir na rede.



Vamos utilizar a máscara de rede 255.255.255.0 da figura acima como exemplo.

255 é a soma dos seguintes números:

$$128 + 64 + 32 + 16 + 8 + 4 + 2 + 1$$

Os equipamentos de rede leem as informações em binário (01101010), cada conjunto 255 equivale a 8 casas com 0 ou 1, onde:

“0” = o zero significa que aquele lugar está disponível;

“1” = significa que aquele lugar está ocupado (preenchido).

O que faz 255 em binário ser escrito assim: 11111111

Cada uma daquelas casas tem um valor, que é exatamente nessa sequência:

1 1 1 1 1 1 1 1

128, 64, 32, 16, 8, 4, 2, 1

Se fosse 254 ficaria: 11111110

252 11111100

248 11111000

240 11110000

224 11100000

192 11000000

128 10000000

127 01111111 e por aí vai.

Mas vamos voltar a nossa máscara 255.255.255.0:

Cada 1 equivale a uma casa ocupada e 0 casa disponível.

O 255.255.255.0 serve para definir o tamanho da sua rede.

Em uma rede com IP 192.168.0.1 e a máscara de sub-rede 255.255.255.0, cabem 254 computadores ligados na rede.

A quantidade de IPs válidos é definido pela quantidade de "0" (binário) em cada bloco de IP.

Se a sua máscara for 255.255.255.192, a sua rede será 192.168.0.0 e você terá 62 IPs disponíveis na sua rede. Como sabemos disso?

Sabemos que  $2^6$  (2 elevado a 6 - 6 zeros) é igual a 64, eliminamos dois endereços IP que não devem ser utilizados, o primeiro e o último. O primeiro IP (192.168.0.0) representa o endereço da rede em si, o último endereço (192.168.0.63) representa o broadcast da rede e não pode ser utilizado. Sobramos com 62 IPs que podem ser utilizados por quaisquer equipamentos dentro dessa rede.

Dado que temos um IP e sabemos a máscara da rede que ele pertence, quem irá tornar esse endereço acessível? O Gateway.

Gateway (em português Ponte de ligação), é como se fosse um portão intermediário que serve para interligar redes, separar domínios e traduzir protocolos. Um exemplo de gateway é a roteadora que recebe da sua operadora de internet, já que servem de intermediários entre o utilizador e a rede.

Endereço IP	192.168.0.100
Gateway	192.168.0.1
Máscara	255.255.255.0

*Endereço de IP, Gateway e Máscara*

## Dica

Leia a apostila de Redes Autônomas Feministas para aprofundar mais sobre redes.

Para saber qual seu IP interno, digite agora:

**Windows:** **Passo 1:** Pressione as teclas "Windows + R" e digite o comando "cmd" (sem aspas);

**Passo 2:** Na janela do "Prompt de Comando", digite "ipconfig" (sem aspas) e pressione Enter. O IP interno do seu computador será exibido em "Endereço IPv4".

**Mac:** **Passo 1:** Abra as configurações do Mac. O usuário pode acessá-las no menu da maçã, clicando em “Preferências do Sistema”;  
**Passo 2:** Na janela de configurações, clique em “Rede”;  
**Passo 3:** Na lateral esquerda da janela, selecione o tipo de conexão que seu computador está usando. Por fim, observe que o endereço de IP é informado na descrição da conexão.

**Linux:** **Passo 1:** No Linux, é possível consultar o endereço IP por meio do programa de linha de comando. Na maioria das distribuições, você pode usar o atalho **Ctrl+Alt+T** para abrir o Terminal.  
**Passo 2:** Digite `ip addr show` e aperte a tecla **Enter**. Ao fazê-lo vai exibir as informações de endereço IP interno, geralmente atribuído pelo roteador local. O IP de cada interface (Ethernet, Wi-Fi, etc.) aparece ao lado de “inet addr.”

- 🖱️ Caso o computador esteja usando uma rede Ethernet, a parte “inet addr” geralmente é exibida como `eth0`. Na rede Wi-Fi, ela costuma aparecer abaixo de `wlan0`.
- 🖱️ O endereço IP é formado por quatro grupos de no máximo três dígitos cada, separados por um ponto. Por exemplo, ele pode ser `192.168.1.4`

## Saiba mais

*Para entender melhor acesse esses dois conteúdos:*

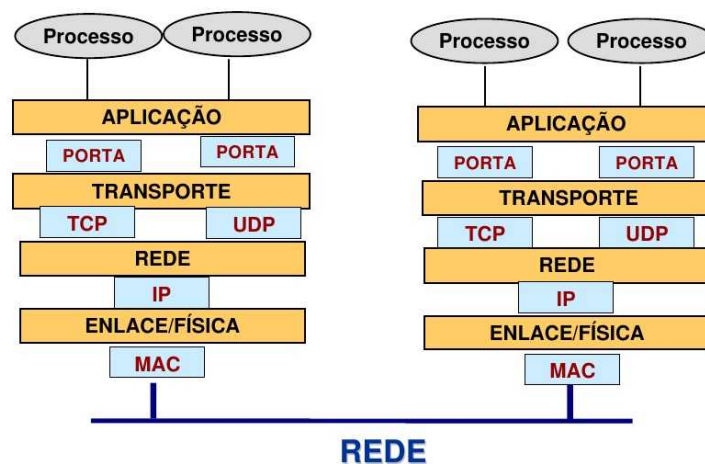
Calculo de Rede IPv4

Calculadora

# Portas e Protocolos

A comunicação pela internet é feita, basicamente, através de protocolos, sendo o TCP (Transmission Control Protocol) um dos mais importantes deles. Isso porque o TCP está incluído no conjunto de protocolos que formam o TCP/IP, a base de comunicação via dados de toda a internet.

## Arquitetura TCP/IP



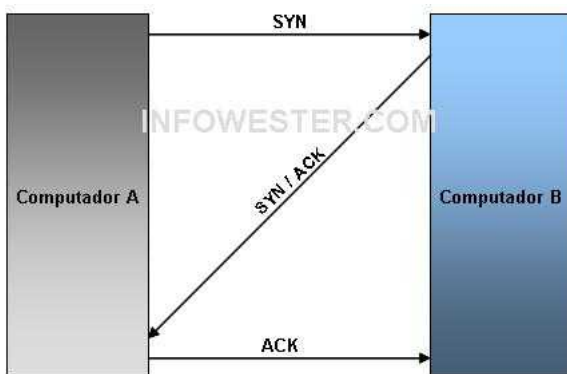
As principais características do TCP são:

- Garantir que os pacotes sejam entregues sem alterações, sem terem sido corrompidos e na ordem correta.
- Dividir grandes arquivos em pacotes de dados menores, transmitir cada pacote separadamente e ordená-los corretamente, antes de serem entregues ao programa de destino.
- Verificar e garantir que os dados não foram alterados ou corrompidos durante o transporte entre a origem e o destino.
- Enviar mensagens positivas dependendo do recebimento bem-sucedido dos dados. Caso um pacote não tenha sido recebido ou tenha sido recebido com



problemas, o TCP envia uma mensagem ao computador de origem, solicitando uma retransmissão do pacote.

O funcionamento do TCP é baseado em conexões. Assim, para um computador cliente iniciar uma “conversa” com um servidor, é necessário enviar um sinal denominado SYN para este último. O servidor então responde enviando um sinal SYN combinado com um sinal de nome ACK para confirmar a conexão. O cliente responde com outro sinal ACK, fazendo com que a conexão esteja estabelecida e pronta para a troca de dados. Por ser feita em três transmissões, esse processo é conhecido como three-way handshake (algo como triplo aperto de mãos).



Podemos comparar TCP ao Sedex. Quando enviamos a encomenda temos que confirmar o recebimento ou são feitas mais tentativas de entrega. Caso a entrega não tenha sido possível por qualquer motivo, a central dos correios notifica o remetente que “deu ruim”.



Outro protocolo de transporte é o UDP (User Datagram Protocol), conhecido como um protocolo “irmão” do TCP, mas é mais simples e também menos confiável. Isso acontece porque o funcionamento do TCP é, como já dito, baseado em conectividade nas duas pontas, o que não ocorre com o UDP.

Como consequência, não há procedimentos de verificação no envio e recebimento de dados (todavia, pode haver checagem de integridade) e se algum pacote não for recebido, o computador de destino não faz uma nova solicitação, como acontece com o TCP. Tudo isso faz do UDP um pouco mais rápido, porém inutilizável em certas aplicações.

Por essas características, pode parecer que o UDP é inútil, mas não é. Há aplicações em que é preferível entregar os dados o mais rapidamente possível, mesmo que algumas informações se percam no caminho. É o caso, por exemplo, das transmissões de vídeo via lives pela internet (streaming), onde a perda de um pacote de dados não interromperá a transmissão. Por outro lado, se os pacotes não chegarem ou demorarem a chegar, haverá congelamentos na imagem, cortes ou travas.

## Portas TCP e UDP

Agora que você já conhece algumas características dos protocolos TCP e UDP, já está apto a entender o conceito de portas.

Para uma compreensão mais fácil, usaremos o seguinte exemplo: Imagine um prédio residencial de endereço 192.168.0.5.

O porteiro (placa de rede) recebe os pacotes (de dados) com o endereço do condomínio mas não o número do apartamento (porta TCP ou UDP). O que acontece?

O pacote é descartado por ter o endereço incompleto. Para o pacote ser entregue é necessário dizer a localização do apartamento (Porta 25).










Agora voltando para a rede: suponha que, neste momento, você esteja usando um navegador de internet, um cliente de e-mail e um software de comunicação instantânea. Todas essas aplicações fazem uso da sua conexão à internet, mas como o computador faz para saber quais os dados que pertencem a cada programa?

Simple, pelo número da porta que cada um utiliza.

Por exemplo, se você está usando um programa de FTP (File Transfer Protocol), a conexão à internet é feita pela

porta TCP 21, que é uma porta convencional para este protocolo. Se você estiver baixando arquivos pelo uTorrent, uma das portas que vão de 6881 à 6889 será utilizada para tal atividade.

Ao todo, é possível usar 65536 portas TCP e UDP, começando em 1. Tanto no protocolo TCP como no UDP, é comum o uso das portas de 1 a 1024, já que a aplicação destas é padronizada pela IANA (Internet Assigned Numbers Authority). De acordo com essa entidade, eis algumas das portas TCP mais utilizadas:

-  21 - FTP;
-  23 - Telnet;
-  25 - SMTP;
-  80 - HTTP;
-  110 - POP3;
-  143 - IMAP;
-  443 - HTTPS.

A IANA disponibiliza uma lista completa e atualizada da utilização das portas TCP e UDP em seu site <https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml>.

Dependendo do caso, uma aplicação não precisa, necessariamente, estar restrita a um conjunto de portas específico. É possível utilizar outras, mas é necessário que isso seja configurado. É por isso, por exemplo, que há determinados endereços na internet que são disponibilizados assim: `http://www.site.com:####`, onde `####` é o número da porta. Neste caso, seu computador está sendo orientado a acessar o endereço pela porta `####`.

# Como um site vai pro ar?

Depois que uma programadora cria o código fonte do site em seu computador, os próximos passos dados são:

- Contratar um plano de hospedagem ou um servidor virtual
- Contratar um plano de hospedagem significa fazer a assinatura de espaço e recursos em um servidor de hospedagem. Ou seja, é o local que você vai hospedar, configurar e publicar todo o conteúdo do seu site.

- Já quando se contrata um servidor virtual está sendo alugado um servidor na nuvem onde terá toda a autonomia para subir os serviços e o conteúdo, o que implica na escolha dos recursos (memória, processador, disco, etc.) e o sistema operacional.

- Após ter contratado o serviços, precisamos registrar o site.

- Registrar um domínio

- Registrar um domínio significa comprar um nome para o seu site. E para domínios.com.br o registro desse domínio é feito unicamente pelo <http://registro.br>, mantido pela NIC.BR - Núcleo de Informação e Coordenação do Ponto BR, e tem o custo anual de R\$ 40,00.

- Para domínios internacionais (sem “.br”), o registro é feito apenas em provedores internacionais de registro, como o [GoDaddy](#).

- Realizando o registro, deve-se inserir o endereço de NS (Servidor de Nomes) do registro (fornecido por ele) para que ele aponte para sua hospedagem ou servidor.

- Transferir seu site para a hospedagem adquirida.

- Para transferir o conteúdo do site para uma hospedagem, é necessário se conectar ao sistema de arquivos utilizando um Gerenciador de Arquivos, localizado no painel, ou um programa de acesso FTP (File Transfer Protocol), como o Filezilla, que irá se conectar diretamente ao espaço do site através de uma url como ftp.meusite.com.br.

## Dica

Como registrar um domínio no Registro.br?

Agora basta esperar a propagação do seu domínio nos DNS do mundo todo, o que para novos domínios o processo leva de 30 a 120 minutos. Para domínios já criados onde foi feita a alteração do DNS, pode-se levar de até 48 horas.

**WIKIPEDIA, Site.** Disponível em <https://pt.wikipedia.org/wiki/Site>. Acesso em 25 de Junho de 2017

**Palpite Digital, O que é site.** Disponível em <https://www.palpitedigital.com/o-que-e-ite/>. Acesso em 25 de Junho de 2017

## 1.3 SERVIDORAS

# O que são Servidores/as?

Imagine que você divide apartamento com várias pessoas e em sua máquina (pode ser um notebook ou uma computadora) tem vários vídeos, áudios e documentos de texto que gostaria de compartilhar com os outros. Talvez a primeira coisa que pense é mandar por e-mail ou subir no Mega.nz ou Google Drive e compartilhar, mas já pensou em compartilhar com qualquer outro computador que tiver na mesma rede de Wi-fi que você? Só deixar o computador ligado e qualquer um conectado no seu wi-fi (ou as máquinas que autorizar) poderá acessar esses arquivos. Ou seja, sua máquina fornecerá arquivos para a rede onde está conectada, logo ela é uma Servidora.


Uma **servidora** (em inglês, server) é um computador que fornece serviços a uma rede de computadores, a qual chamamos de **cliente**. Os clientes e as servidoras se comunicam através de protocolos (que vimos no capítulo anterior), assim como dois ou mais computadores de redes.





O termo “servidor/a” é amplamente aplicado a computadores completos, embora uma servidora possa equivaler a um software, partes de um sistema computacional ou, até mesmo, a uma máquina que não seja necessariamente um computador, como um [Raspberry Pi](#) usado na [Fuxico](#).


Uma servidora pode prover mais de um serviço simultaneamente. Pode existir, em uma rede, um computador que atue como uma servidora web e servidora de banco de dados ou pode atuar como servidora de arquivos, de correio eletrônico e autenticação ao mesmo tempo. Computadores que atuam como um único tipo de servidora, ou seja, provê apenas um serviço, é chamado de servidora dedicada. As servidoras dedicadas possuem a vantagem de atender as requisições dos clientes mais rapidamente.


Existem diversos tipos de servidoras. Os mais conhecidos são:


 **Servidora de arquivos:** Servidora que armazena e compartilha arquivos de diversos usuários.


 **Servidora web:** Servidora responsável pelo armazenamento de páginas de um determinado site, requisitados pelos clientes através de navegadores.


 **Servidora de e-mail:** Servidora responsável pelo armazenamento, envio e recebimento de mensagens de correio eletrônico.


 **Servidora de impressão:** Servidora responsável por centralizar a conexão com impressoras e controlar os pedidos de impressão de arquivos de diversos clientes (computadores, celulares, tablets, etc.) em uma rede.

 **Servidora de banco de dados:** Servidora que possui e manipula informações contidas em um serviço de banco de dados.

 **Servidora DNS:** Servidores responsáveis pela conversão de endereços de sites em endereços IP e vice-versa.

 **Servidora FTP:** Permite acesso de outros usuários a um disco rígido (HD) ou servidora através do Protocolo de Transferência de Arquivos (FTP). Esse tipo de servidora armazena arquivos para dar acesso a eles pela internet.

 **Servidora de virtualização:** Permite a criação de máquinas virtuais (simulação de uma máquina física completa) mediante compartilhamento de hardware da máquina hospedeira (rede, memória RAM, HD, etc).

 **Servidora de sistema operacional:** permite compartilhar o sistema operacional de uma máquina com outras, interligadas na mesma rede, sem que essas precisem ter um sistema operacional instalado, nem mesmo um HD próprio. Como nos Telecentros.

## Servidoras Dedicadas

A maioria dos servidoras dedicadas utiliza hardware de alta confiabilidade e desempenho e processadores criados para atender servidoras, como Intel Itanium, Intel Xeon e AMD Opteron. Algumas servidoras são multiprocessadas, ou seja, possuem mais de um processador.

Para poder operar sem nunca desligar, algumas servidoras utilizam UPS/No-Break (sistemas de alimentação ininterrupta) que continuam a alimentar a



servidora caso haja alguma queda de tensão. É uma versão para computadores dos geradores de energia que vemos em Shoppings e Fabricas pra quando existe queda de energia.

Devido a existência de um ou mais processadores de alto desempenho, as servidoras precisam de um eficiente sistema de dissipação de calor, o que implica em Coolers (ventiladores/ventoinhas de gabinete) mais caros, com maior amperagem. Quanto maior a amperagem, maior eficiência e confiabilidade.

Os tipos de hardware utilizados nas servidoras possuem uma tecnologia chamada hot swapping, que permite substituir componentes da máquina enquanto ela permanece ligada, não precisando reiniciar o computador. A tecnologia é presente em discos rígidos e permite que a troca de um disco defeituoso possa ser feita com o sistema operativo em execução sem perda de dados.



## **Da para usar um desktop como uma servidora?**

Um desktop (computador pessoal de mesa) e uma servidora são máquinas com a arquitetura básica parecida: processador, memória, placa de rede e disco.

Existem desktops criados para atender um público exigente, que demanda alto desempenho e armazenamento para uso de seus softwares. Computadores para Gamers, Designers e Arquitetas são exemplos de máquinas criadas com esse propósito. Mas mesmo assim, não dá para considerar que uma máquina dessas equivale a uma servidora.

Servidoras são produzidas para o uso contínuo, ou seja, estão preparadas para suportar grandes cargas de trabalho sem travamento, expandir e substituir hardware sem a necessidade de serem desligadas (Hot Swap) como vimos anteriormente. Para isso, são utilizados componentes e placas mais robustas,

sistemas de redundância em discos e alimentação de energia, sistemas de refrigeração mais eficientes e recursos avançados de gerenciamento. Tudo isso para garantir a disponibilidade e confiabilidade dos serviços hospedados, ao contrário de um desktop que é projetado para necessidades mais básicas de um usuário.

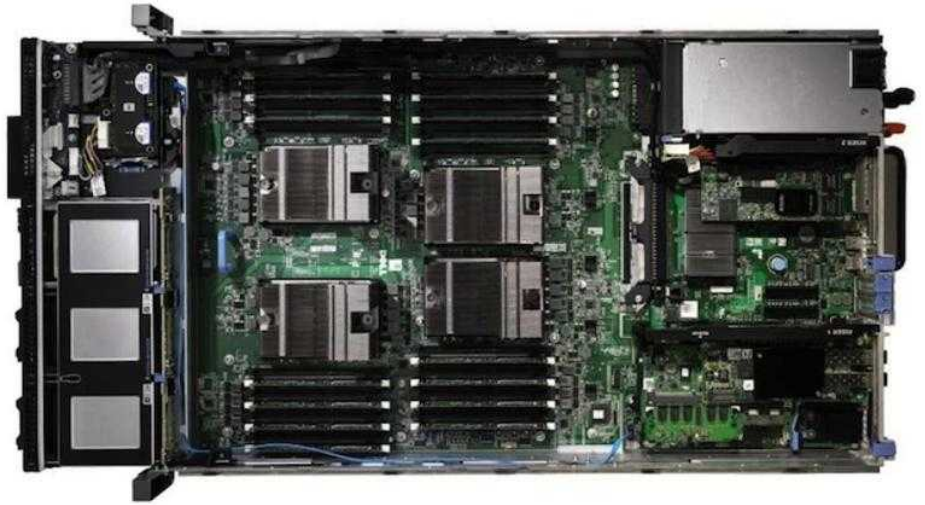
Então a médio e longo prazo, um desktop iria precisar de manutenção, substituição de peças e gastos maiores para ter um desempenho parecido e mesmo assim não seria o suficiente para operar 24/7 com a mesma performance e eficiência.

**Wikipedia, Servidora.** Disponível em <<https://pt.wikipedia.org/wiki/Servidora>>.

*Acesso em 25 de Junho de 2017*

# Configuração de hardware da servidora

## Placa de sistema



A placa de sistema, também conhecida como “placa-mãe”, é a placa de circuitos principal do computador à qual todos os outros componentes da servidora estão conectados.

Entre os maiores componentes da placa de sistema estão o processador (ou CPU, Central Única de Processamento), o circuito de suporte, chamado de chipset; a memória, os slots de expansão; uma controladora de disco rígido e portas de entrada e saída (E/S) para dispositivos como teclados, mouses e impressoras. Algumas placas de sistema também incluem recursos integrados adicionais, como adaptador gráfico, controladora de disco SCSI ou interface de rede.

## Processador



O processador é o cérebro da servidora. O número de processadores na servidora e sua velocidade afetam profundamente a capacidade da servidora de suportar aplicativos. Os processadores mudam frequentemente, e pode ser difícil determinar qual deles é adequado para seu aplicativo. Você deve considerar três características principais ao selecionar um processador:

## Velocidade do relógio

É a velocidade em que o processador opera, normalmente medida em gigahertz (GHz). Geralmente, quanto mais rápido melhor; ou seja, servidoras com velocidades mais altas fornecem melhor desempenho. Isso pode se traduzir na capacidade de suportar mais contas simultâneas de e-mail, gerenciar mais solicitações da Web durante períodos de pico de demanda (campanhas virtuais ou Black Friday) ou realizar consultas rápidas no seu banco de dados de Pesquisas Acadêmicas. Adquirir um processador com uma maior frequência melhora o desempenho do sistema atual e também ajuda a garantir que a servidora consiga gerenciar futuras demandas.

## Contagem de núcleos

O número de processadores físicos no próprio processador. Atualmente, a maioria das CPUs de servidora tem de dois a quatro núcleos. Vários núcleos permitem uma melhor realização de várias tarefas, em servidores que executarão vários aplicativos. Por exemplo, a execução de um antivírus pode ser feita em um núcleo, enquanto o backup de dados é realizado por outro núcleo independente.

## Tamanho do cache

Cada processador tem uma memória integrada de alta velocidade localizada diretamente na unidade de processamento central (CPU) e outra próxima a essa. Um tamanho maior de cache (depósito de arquivos temporários para acesso rápido) reduz a frequência exigida pela CPU para recuperar os dados da memória de sistema que fica fora da CPU. Para a maioria dos aplicativos, isso melhora a capacidade de resposta do sistema e oferece uma experiência de usuário mais aprimorada. Em geral, CPUs com contagens de núcleos e frequências mais elevadas têm tamanhos de cache maiores, a fim de fornecer um excelente desempenho.

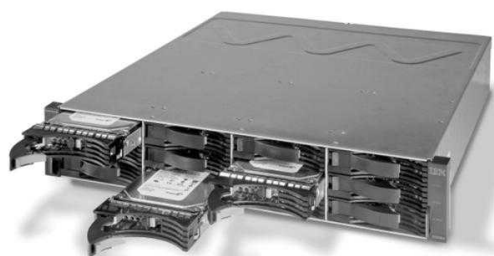
## Memória

Ao abrir um arquivo ou documento, a servidora precisa de um lugar para controlar temporariamente o arquivo. Ele usa chips especializados de alta velocidade, chamados memória de acesso aleatório ou RAM. O arquivo atual será salvo no disco rígido quando você “Salvar” o arquivo. A RAM foi projetada para acesso rápido e para lembrá-lo rapidamente onde o arquivo está armazenado no sistema de disco rígido permanente.



A regra básica geral é adicionar a maior quantidade de RAM possível. Quanto mais RAM estiver disponível, mais operações sua servidora será capaz de gerenciar ao mesmo tempo, sem precisar acessar os discos rígidos (que são mais lentos que a RAM da placa de sistema).

## Armazenamento ou sistema de disco rígido



Os discos rígidos fornecem ao servidor uma grande biblioteca contendo todos os arquivos que ele pode acessar. Pense nessa biblioteca como um armário de arquivos em constante expansão. O tamanho e o tipo dos sistemas de disco rígido dependem da quantidade de dados que você precisa armazenar.

A maioria dos servidores está configurada com um disco rígido muito grande, semelhante ao disco rígido em seu desktop. No entanto, os discos rígidos do servidor foram projetados especialmente para tempos de acesso rápidos e para a capacidade de adicionar vários discos rígidos internamente. Em algum momento, você pode precisar adicionar mais discos rígidos (RAID) e anexar sistemas de disco rígido externo (Storage).

## Saiba mais

### RAID

Storages são dispositivos projetados especificamente para armazenamento de dados, onde através de uma conexão via rede, você pode conectar sua(s) servidora(s) a um storage, facilitando assim a expansão da capacidade de armazenamento sem impacto na produção, garantindo maior flexibilidade e confiabilidade no armazenamento.

### Storage

Matriz Redundante de Discos Independentes: combina discos rígidos em um sistema de armazenamento grande e lógico, que grava os dados em mais de um disco para garantir maior confiabilidade.

Um storage utiliza padrões de RAID, que permitem trabalhar com diversos discos em conjunto, de forma independente e redundante.

Existem algumas nomenclaturas que alguns profissionais usam para eles como: "network storage" ou simplesmente de "storage". E outras nem tão comuns como: "filers" (arquivadores).



## Controladora de rede

A conexão de rede é uma das partes mais importantes de qualquer servidora. A controladora de rede gerencia as entradas (onde conecta os cabos) e o tráfego de clientes (em outros computadores) conectados a sua rede.

### Fonte:

<http://www.dell.com/learn/br/pt/brbsdt1/sb360/server-hardware-configuration>

# Sistemas operacionais para servidoras

Para que o entendimento do conceito sobre sistemas operacionais para servidoras seja claro, primeiramente, deve-se ter em mente algumas definições sobre o que é um sistema operacional e como ele atua no gerenciamento de todas as complexidades do hardware.

Maziero expõe que “existe uma grande distância entre os circuitos eletrônicos e dispositivos de hardware e os programas aplicativos em software” (MAZIERO, 2006, p. 3). Neste sentido, para que seja feito o acesso diretamente ao hardware, como armazenar uma certa informação no disco rígido, o programador deveria trabalhar em um baixo nível de programação levando em consideração a diversidade do hardware o qual ele quer interagir. Portanto, para que o acesso seja realizado de forma homogênea é necessária uma camada que faça este gerenciamento. Para satisfazer esta necessidade foram criados, então, os sistemas operacionais, que nada mais são do que “um conjunto de recursos abstratos claros em vez de recursos confusos de hardware” (TANENBUM, 2009, p. 02).

Maziero define os sistemas operacionais como:

*[...] uma camada de software que opera entre o hardware e os programas/aplicativos voltados ao usuário final. O sistema operacional é uma estrutura de software ampla, muitas vezes complexa, que incorpora aspectos de baixo nível (como drivers de dispositivos e gerência de memória física) e de alto nível (como programas utilitários e a própria interface gráfica). (MAZIERO, 2006, p. 03)*

Partindo destes conceitos, pode-se afirmar que os sistemas operacionais são camadas de softwares necessárias e indispensáveis para a harmonia entre os processos e o hardware.

Como dito anteriormente, servidoras são máquinas mais potentes do que um desktop comum. Este fato pode ser observado ao analisarmos as questões de processamento entre os dois. Uma servidora normalmente possui mais poder de processamento devido à quantidade de processadores ou núcleos presentes em



sua arquitetura física. Logo, eles conseguem executar mais tarefas e com maior velocidade. Um exemplo de processador para servidoras é o “Coprocessador Intel® Xeon Phi™ 5110P com 60 núcleos/1,053 GHz/240 threads”

Assim, para o gerenciamento desta carga de trabalho entre as diferentes funcionalidades de servidoras, houve a necessidade de um sistema operacional que permitisse “a gestão eficiente de grandes quantidades de recursos (disco, memória, processadores), impondo prioridades e limites sobre o uso dos recursos pelos usuários e seus aplicativos” (MAZIERO, 2006, p.06) afim de que todo esse processamento fosse utilizado da melhor forma possível, garantindo, assim, a integridade, a segurança, escalabilidade e o bom escalonamento entre as diversas tarefas executadas. Surge então os sistemas operacionais para servidoras.

Os Sistemas Operacionais de Servidoras mais utilizados são:

 Unix

 Windows Server

 Linux

    \_RedHat

    \_Debian

    \_Ubuntu

    \_Suse

    \_Slackware

    \_CentOS

    -...

 BSD

 ESX/ESXi

## Para saber mais

PICOLLO, Luiz. Sistemas Operacionais para Servidores [online] Disponível na Internet via URL:<http://pt.scribd.com/doc/114321251/Texto-Sistemas-Operacionais-Para-Servidores>. Arquivo consultado em 25 de Junho de 2017.

MAZIERO, Carlos A. Sistemas Operacionais [online] Disponível na Internet via URL: [http://dainf.ct.utfpr.edu.br/~maziero/doku.php/so:livro\\_de\\_sistemas\\_operacionais](http://dainf.ct.utfpr.edu.br/~maziero/doku.php/so:livro_de_sistemas_operacionais). Arquivo consultado em 18 de Novembro de 2012.

TANENBAUM, A. Sistemas Operacionais Modernos; tradução Ronaldo A. L. Gonçalves, Luís A. Consularo, Luciana do Amaral Texeira; revisão técnica Raphael Y. De Camargo. 3a ed. São Paulo: Person Prentice Hall, 2009.

# Servidora Web

A servidora web é a peça mais importante da infraestrutura de um site na internet. Ele é um programa que usa o HTTP (Hypertext Transfer Protocol) para servir os arquivos que formam páginas da web para os usuários, em resposta aos pedidos que são encaminhados por clientes HTTP.

A servidora web é responsável por armazenar e trocar informações com outras máquinas e para que isso aconteça ao menos dois participantes estão envolvidos nesta troca de informações: os usuários/clientes (solicitantes) e as servidoras (atendentes).

Existem servidoras web como software livre ou software proprietário, que é adquirido com um fornecedor especializado geralmente por empresas. O mais popular, e mais utilizado no mundo, é a servidora [Apache](#) (software livre).

*Curiosidade:* O nome dado pela Fundação Apache Software faz referência à nação Apache, formada por nativos americanos que tinha, em combate, grande resistência e estratégias superiores. Uma alusão à resistência da comunidade do *software livre* aos ataques de interesses privados.

## Como funciona uma servidora web

Os pedidos http, que se referem habitualmente a páginas HTML, são normalmente feitos através de browsers.

O processo se inicia com a conexão entre o computador onde está instalado a servidora web e o **computador da cliente (quem faz a solicitação)**; como na web não é possível prever a que hora se dará essa conexão, as servidoras web precisam estar disponíveis o tempo inteiro.

A partir daí é processado o pedido **da cliente**, e conforme as restrições de segurança e a existência da informação solicitada, a servidora devolve os dados (resposta). Genericamente tudo o que se enquadre no conceito de arquivo pode ser enviado como resultado de um pedido http.

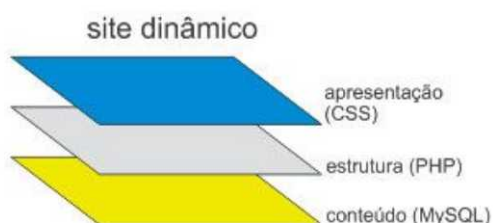
Um exemplo: quando a usuária digita `http://www.marialab.org`, o browser solicita o arquivo do servidor e aguarda a resposta. O servidor responde depois de verificar se o endereço existe e encontra os arquivos necessários, em seguida executa as instruções e entrega os resultados. Quando não encontra, a servidora apresenta na tela da usuária uma mensagem de erro (Error 404).

As servidoras web também podem executar programas e scripts. Abra seu navegador (Firefox ou Google Chrome), acesse algum site e pressione F12. Com o painel aberto, clique na aba Rede e pressione F5 para atualizar a tela. As informações que aparecerem na aba Rede são os arquivos armazenados na servidora web que estão sendo baixados e apresentados no browser. Clique em cada um para explorar seu conteúdo.

## Páginas Dinâmicas e Páginas Estáticas

A origem do conteúdo enviado pelo servidor web numa resposta a um pedido HTTP pode ser:

- estática – se vier diretamente de um arquivo já existente no servidor
- dinâmica – se for criada dinamicamente por outro programa, script ou API chamado pelo servidor.



No caso de uma página dinâmica, o pedido, depois de recebido, é processado pelo servidor web que vai criar dinamicamente o conteúdo que depois será enviado para o cliente.

As páginas dinâmicas têm a vantagem de poderem ser programadas, ou seja usando alguma linguagem de programação (que

dependendo do servidor web pode ser php, Java, Python, Rails, JavaScript, C#, etc) podemos criar programas que rodam no servidor web, eventualmente acessando a bases de dados e cujo resultado é enviado para o browser.

## Exemplo do fluxo pedido/resposta gerado quando se acessa uma página estática

Supondo que numa servidora web exista um arquivo chamado hello.html com o seguinte conteúdo:

```
<html>
  <body>
    <p>olá mundo</p>
  </body>
</html>
```

Se no browser tentar acessar a este arquivo a sequência “pedido/resposta” iria produzir os seguintes comandos:

### Pedido

GET /hello.html HTTP/2

Accept:

text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,  
\*/\*;q=0.8

Accept-Language: pt-BR,en-US;q=0.7,en;q=0.3

Accept-Encoding: gzip, deflate

User-Agent: User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:78.0)

Gecko/20100101 Firefox/78.0

Host: localhost

Connection: Keep-Alive

### Resposta

HTTP/1.1 200 OK

Server: Apache

Date: Mon, 15 Jun 2020 01:54:00 GMT

Content-Type: text/html; charset=UTF-8

Last-Modified: Thu, 25 May 2006 14:02:12 GMT

Content-Length: 54

```
<html>
  <body>
    <p>olá mundo</p>
  </body>
</html>
```

Embora estes valores variem de acordo com o browser que utilizarmos e com a servidora web que responde a este pedido http, uma boa parte do conteúdo será sempre igual:

- no pedido GET /hello.html HTTP/1.1 estamos usando o protocolo HTTP versão 1.1 o arquivo hello.html que está na raiz do servidor

- na resposta HTTP/1.1 200 OK o pedido é válido (200 OK) e o conteúdo segue em baixo. Como podemos ver depois de mais um conjunto de dados aparece finalmente o conteúdo HTML da página que tínhamos criado.

Independentemente de estarmos falando de páginas dinâmicas ou páginas estáticas este será sempre o fluxo que o pedido/resposta entre a servidora e o browser irá provocar.

## Exemplo do fluxo pedido/resposta gerado quando se acede a uma página dinâmica

Numa servidora web capaz de responder a páginas dinâmicas temos um arquivo chamado phphello.php com o seguinte conteúdo:

```
<html>
  <head>
    <title>Teste PHP</title>
  </head>
  <body>
    <?php echo "<p>Olá Mundo</p>";?>
  </body>
</html>
```

Se tentarmos acessar a este arquivo no browser, a sequência de pedido/resposta iria produzir os seguintes comandos:

### Pedido

GET /phphello.php HTTP/1.1

Accept:

text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,  
\*/\*;q=0.8

Accept-Language: pt-BR,en-US;q=0.7,en;q=0.3

Accept-Encoding: gzip, deflate

User-Agent: User-Agent: Mozilla/5.0 (X11; Linux x86\_64;

rv:78.0)  
Gecko/20100101 Firefox/78.0  
Host: localhost  
Connection: Keep-Alive

## Resposta

HTTP/1.1 200 OK  
Server: Apache  
date: Mon, 15 Jun 2020 02:20:09 GMT  
X-Powered-By: ASP.NET  
Content-Length: 94  
Content-Type: text/html  
cache-control: max-age=0

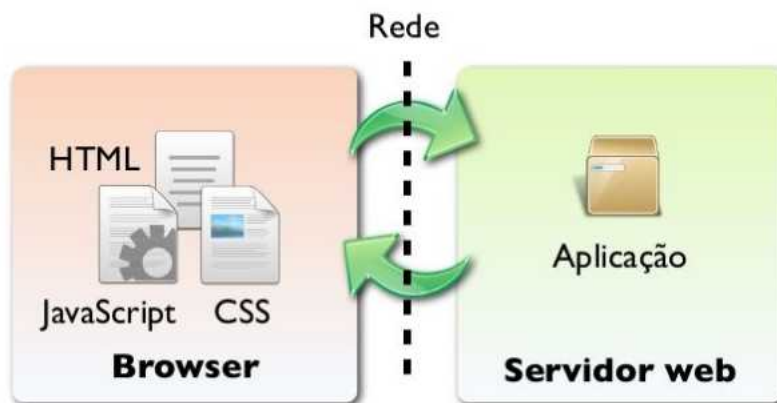
```
<html>  
  <head>  
    <title>Teste PHP</title>  
  </head>  
  <body>  
    <p>Olá Mundo</p>  
  </body>  
</html>
```

Como podemos ver, o pedido/resposta de um arquivo estático ou de um arquivo dinâmico gera fluxos de informação praticamente iguais, isto é, na web a informação que circula é essencialmente a mesma. a diferença é que um arquivo dinâmico tem que ser primeiro processado pelo servidor web.

# Arquitetura Cliente/Servidora

As servidoras podem fornecer várias funcionalidades, muitas vezes chamado de **serviços**, tais como o compartilhamento de dados ou um banco de dados SQL para um cliente que irá consumir. Esta arquitetura é chamada de modelo **cliente- servidora** e é utilizada em redes de médio e grande porte (com muitas máquinas) e em redes onde a questão da segurança desempenha um papel de grande importância. Basicamente, nessa arquitetura, uma das partes “serve” e a outra “consome”.

## Arquitetura Cliente/Servidor



Uma única servidora pode servir várias clientes, e uma única cliente pode usar várias servidoras. Um processo cliente pode ser executado no mesmo dispositivo ou pode se conectar através de uma rede para uma servidora em um dispositivo diferente.

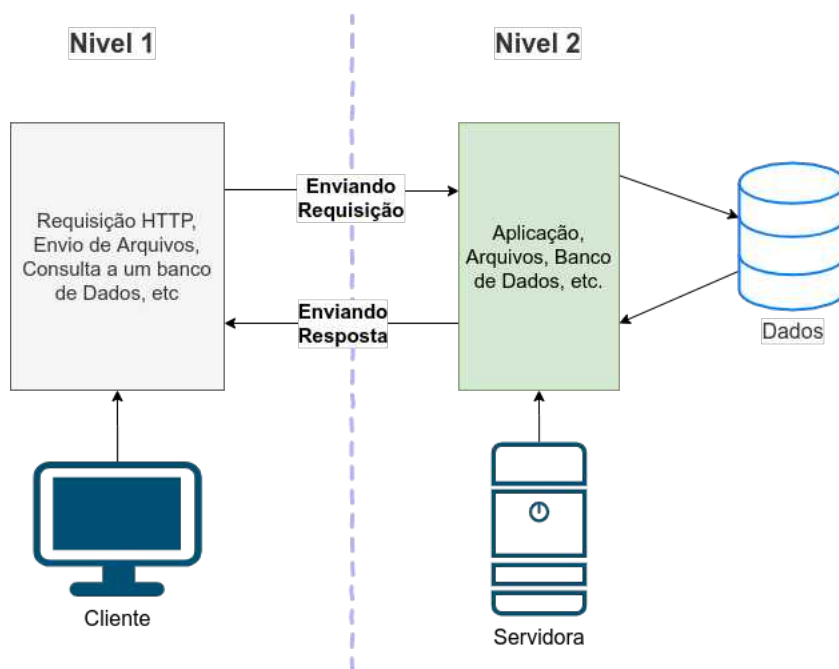
A internet é o exemplo perfeito do uso desta arquitetura, pois quando solicitamos uma página na web, estamos solicitando um serviço para “alguém”, e esse “alguém” nos responde com os dados e nós consumimos a informação.

## Arquitetura em 2 níveis

A arquitetura em dois níveis (também chamada de arquitetura 2-tier - ‘tier’ significa ‘alinhamento’, em inglês) caracteriza os sistemas clientes/servidoras pelos quais a cliente pede um recurso e a servidora responde diretamente ao



pedido, utilizando seus próprios recursos. Isto significa que a servidora não requer outro aplicativo para proporcionar parte do serviço:

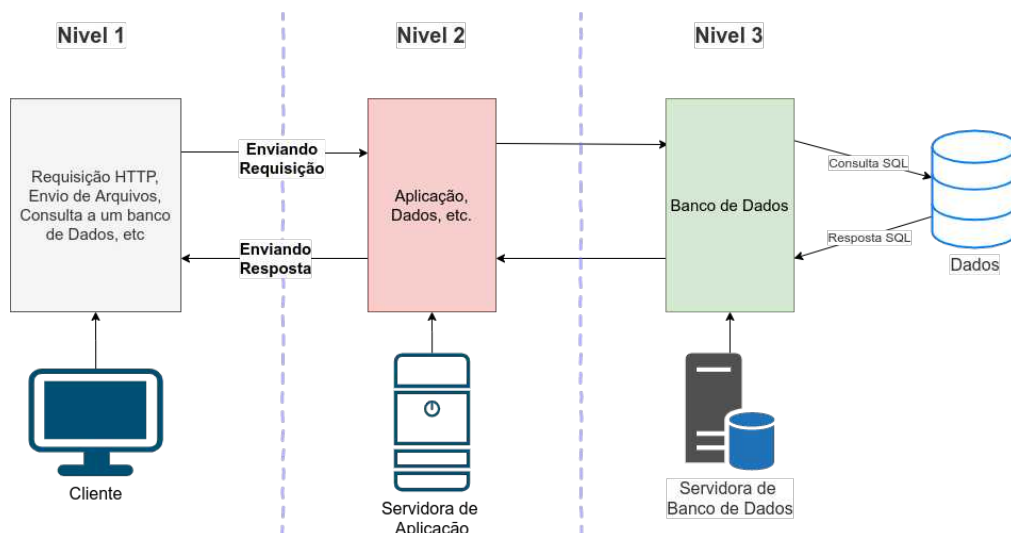


*arquitetura cliente/servidor em dois níveis*

## Arquitetura em 3 níveis

Na arquitetura em 3 níveis existe um nível intermediário, o que significa que ela está compartilhada entre:

- ▣ Uma cliente, ou seja, a máquina que solicita recursos, equipado com um interface de usuário (geralmente um navegador) encarregado da apresentação;
- ▣ A servidora de aplicação (também chamado de software intermediário), cuja tarefa é proporcionar os recursos solicitados, mas que recorre a uma outra servidora.
- ▣ A servidora de dados, que fornece os dados necessários da servidora do aplicativo.



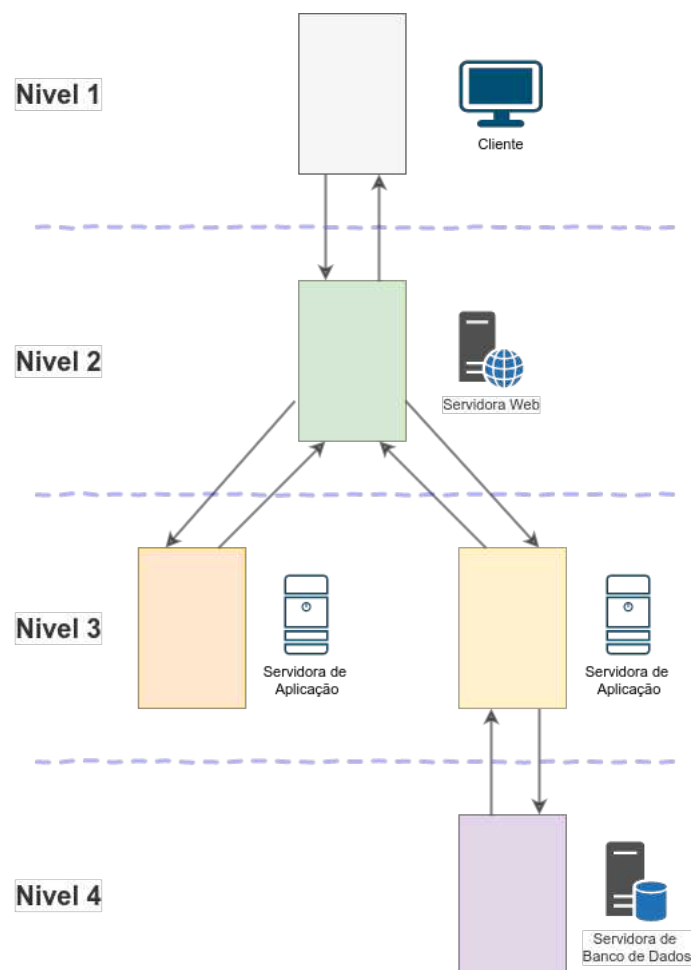
*arquitetura cliente/servidor em três níveis*

O emprego maciço do termo de 'arquitetura em 3 níveis' também designa as seguintes arquiteturas:

- ▣ Compartilhamento do aplicativo entre cliente, servidora intermediária e a servidora da empresa;
- ▣ Compartilhamento do aplicativo entre cliente, servidora de aplicativo e servidora de bancos de dados da empresa.

## Arquitetura multiníveis

Na arquitetura em 3 níveis, cada servidor (níveis 2 e 3) efetua uma tarefa (um serviço) específica. Assim sendo, uma servidora pode utilizar os serviços de uma ou várias servidoras para propor o seu próprio serviço. Consequentemente, a arquitetura em três níveis é potencialmente uma arquitetura de (vários) níveis.



## **Diferenças entre as arquiteturas**

A arquitetura em dois níveis é uma arquitetura cliente/servidor, na qual a servidor pode fornecer todos os recursos pedidos pelo cliente sem intermediários.

Em compensação na arquitetura em três níveis, existe uma servidora especializada para cada tarefa, por exemplo uma servidora web que se conecta a uma servidora de banco de dados para trazer as informações para a cliente.

A arquitetura em três níveis permite uma maior flexibilidade para manutenção, um melhor desempenho dada a divisão das tarefas entre as diversas servidoras e uma segurança mais significativa pois ela pode ser definida de forma independente para cada serviço e em cada nível.

**1.4 NUVEM**

# Essa tal “nuvem”

Quando falamos em serviços na “nuvem” o que imaginam?

**Exercício:** faça um desenho, crie uma imagem de como você imagina que seja o caminho do seu computador até um serviço de armazenamento na nuvem. No final iremos revisitá-lo.



“Cloud Computing” é um modelo que permite de forma conveniente, o acesso à rede sob demanda para um conjunto compartilhado de recursos de computação configuráveis (por exemplo, redes, servidoras, armazenamento, aplicativos e serviços) que podem ser rapidamente provisionados e lançados com o mínimo de esforço de gestão ou a interação de um prestador de serviços.” (NIST.gov)

Com a computação nas nuvens, temos a possibilidade de acessar arquivos e executar diferentes tarefas através de serviços pela internet. Isso significa que caso queira acessar seus arquivos ou usar um programa, não é necessário instalar um software no seu computador ou estar com seus arquivos em algum dispositivo físico com você, basta uma conexão na internet por um computador ou celular.

Agora que já sabemos que a internet são vários cabos passando pelas cidades e oceanos, que conectam servidoras do mundo inteiro, já concluímos então que a nuvem está dentro desta complexidade.

Você pode acessar o que precisa através de um serviço online que esteja hospedado em algum servidor em algum datacenter no mundo. Pode ser desde um processador de textos até mesmo um jogo ou um pesado editor de vídeos. Enquanto os servidores executam um programa ou acessam uma determinada informação, o seu dispositivo precisa apenas de uma interface para que você interaja.

Mas por mais prático que seja ter acesso aos seus arquivos em nuvem, existe uma grande preocupação sobre a segurança de armazenamento dos mesmos e a privacidade dos dados que estão trafegando por esse ambiente.

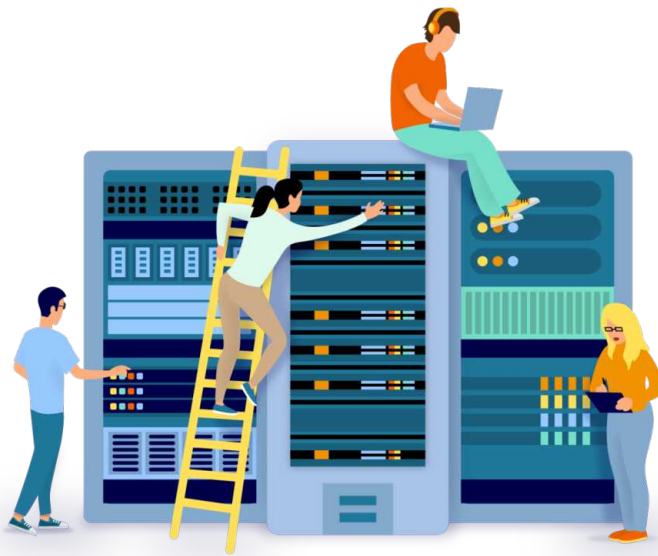
Desde que vários provedores aderiram esse modelo, multiplicou exponencialmente o número de vazamento de dados de usuárias, cibercrimes e violação de direitos das consumidoras. Antes tínhamos o costume de ter os nossos arquivos e dados pessoais em nossos dispositivos físicos, inclusive nossos backups eram feitos em disquete, CDs e HDs Externos. Agora cada foto que tiramos vai para um disco virtual, a nossa agenda de contatos é sincronizada com um catálogo online e nosso histórico de conversas também fica salvo na internet.

Isso facilita empresas e cibercriminosos de conseguirem ter vantagens (lícitas ou ilícitas) sobre as pessoas, dado que essas informações estão sendo dadas muitas vezes de graça, sendo processadas e compartilhadas para terceiros sem nosso conhecimento explícito (Lembra do “Li e aceito os termos de uso”?).

**Tecmundo, O que é Computação em Nuvens.** Disponível em <https://www.tecmundo.com.br/computacao-em-nuvem/738-o-que-e-computacao-em-nuvens-.htm>. Acesso em 25 de Junho de 2017

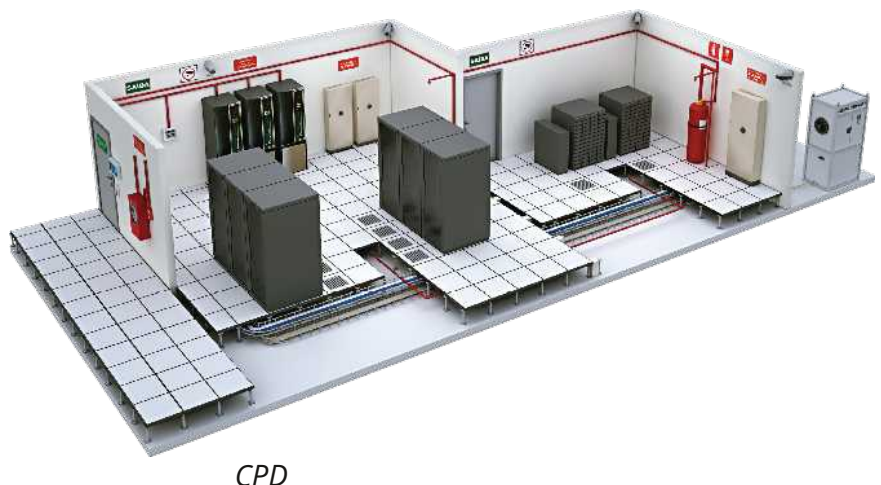
**Wikipedia, Computação em nuvem.** Disponível em [https://pt.wikipedia.org/wiki/Computa%C3%A7%C3%A3o\\_em\\_nuvem](https://pt.wikipedia.org/wiki/Computa%C3%A7%C3%A3o_em_nuvem). Acesso em 25 de Junho de 2017

# Datacenter - Onde tudo acontece



Um data center é uma estrutura que tem papel importantíssimo para que diversos serviços atuais sejam bem utilizados e para garantir a continuidade de funcionamento de sistemas que processam dados diversos. Ele consiste em uma instalação que centraliza as operações e o equipamento de TI de uma organização, onde ela armazena, gerencia e dissemina seus dados com segurança e confiabilidade.

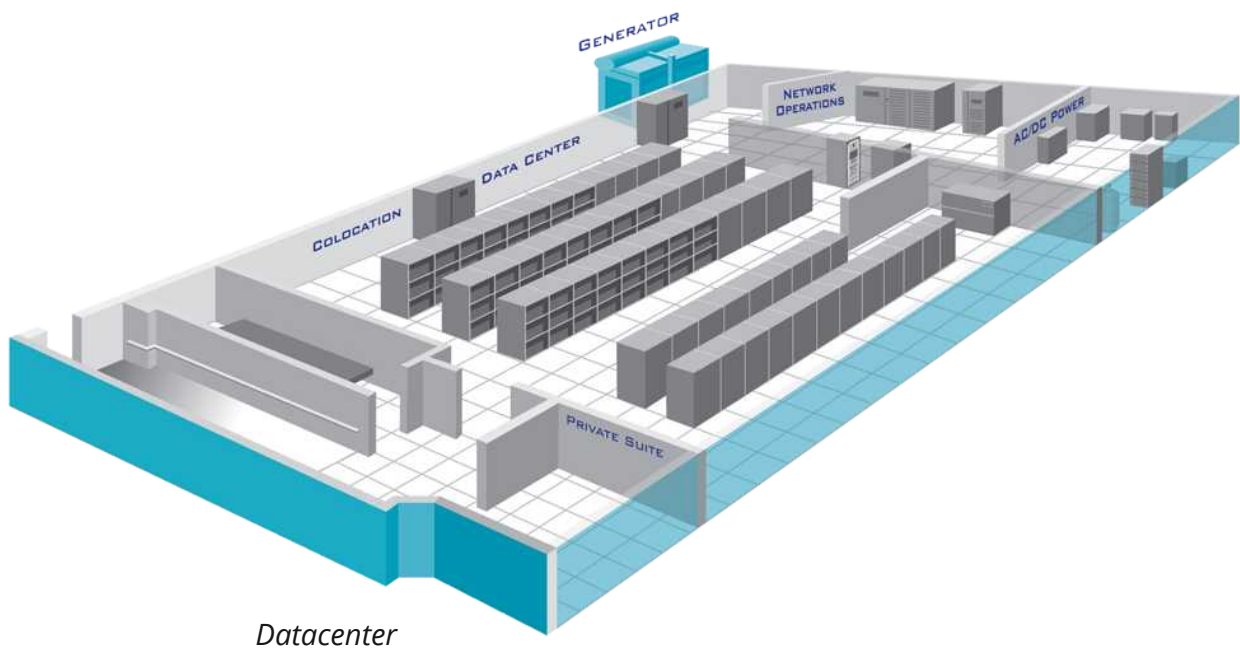
As arquiteturas e os requisitos dos datacenters podem variar de forma significativa. Por exemplo, um datacenter construído para um provedor de serviços em nuvem como a Amazon Web Services



(AWS) satisfaz requisitos de instalação, infraestrutura e segurança completamente diferentes daqueles de um datacenter completamente privado, como aqueles que ficam dentro de empresas (CPD - Centro de Processamento de Dados), que é dedicado somente a manter dados confidenciais de forma segura.



Uma operação eficaz de datacenter é alcançada através de um investimento equilibrado nas instalações e nos equipamentos hospedados. Os elementos de um datacenter são os seguintes:



- Instalação: a localização e o espaço utilizável que está disponível para os equipamentos de TI.
- Infraestrutura de suporte: equipamento que contribui para sustentar de forma segura o mais alto nível de disponibilidade possível. Alguns componentes para suportar a infraestrutura incluem:
  - > Fontes de alimentação ininterrupta (bancos de bateria, geradores e fontes de alimentação redundantes)
  - > Sistemas de ar-condicionado para computadores (aquecimento, ventilação e ar-condicionado) e sistemas de exaustão
  - > Sistemas de segurança física (vigilância biométrica e de vídeo)
- Equipamento de IT: Servidores, hardware de rede e armazenamento, cabos, racks, assim como uma variedade de elementos de segurança da informação, como firewalls.
- Equipe de operação: para monitorar as operações e manter o equipamento de TI e estrutural 24 horas por dia

Os datacenters evoluíram significativamente nos últimos anos, adotando

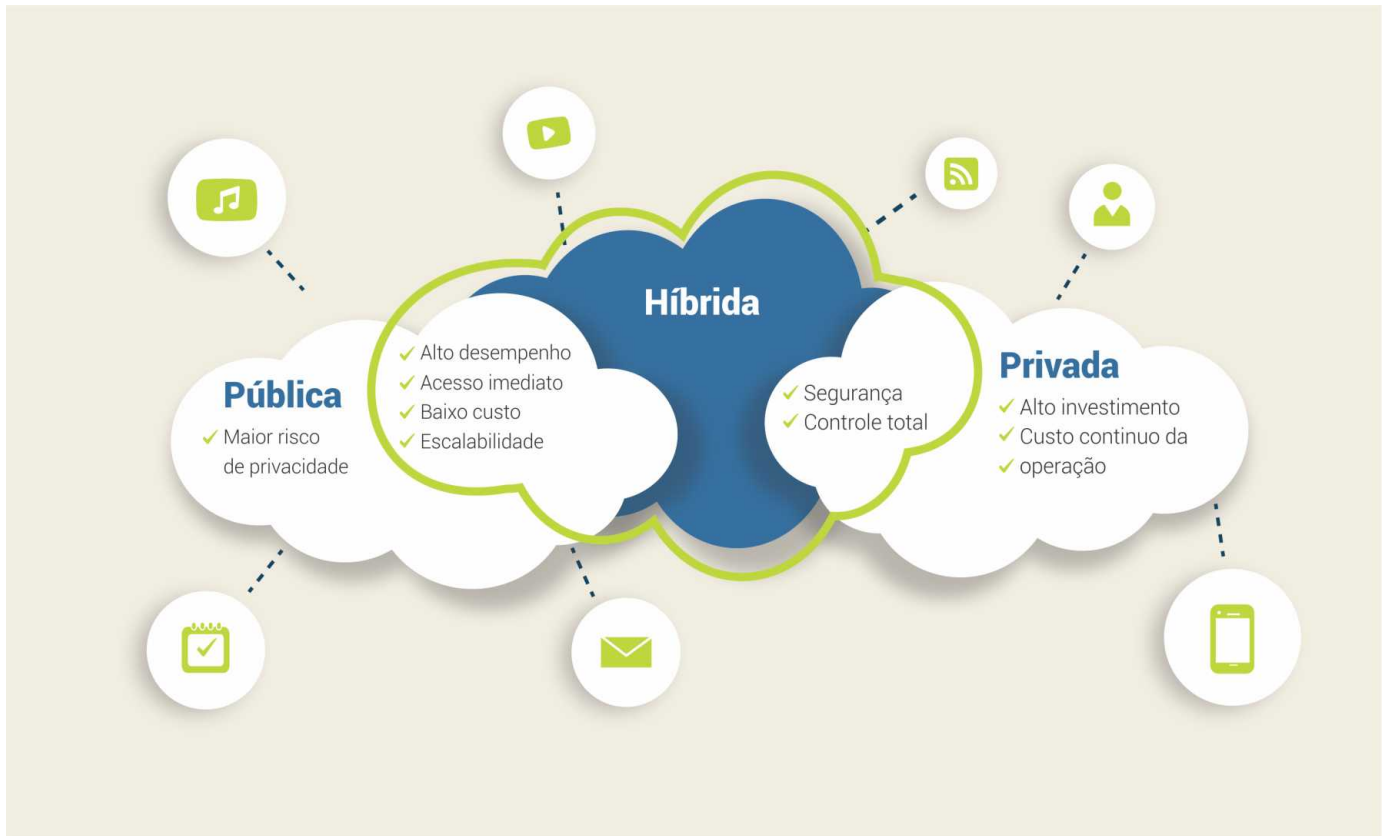
tecnologias com virtualização para otimizar a utilização de recursos e aumentar a flexibilidade de TI. Enquanto as necessidades dos departamentos de TI empresarial evoluem para o modelo de serviço sob demanda, muitas organizações estão se transferindo para serviços e infraestruturas baseados em nuvem.

Também têm sido enfocadas iniciativas para reduzir o enorme consumo de energia dos datacenters através da incorporação de tecnologias e práticas mais eficientes no gerenciamento de datacenters, para minimizar o impacto ambiental. Os datacenters construídos dentro desses padrões têm sido denominados “datacenters verdes”.

Vídeo: [Dentro de um datacenter da Google](#)

**Palo Alto Networks, O que é um datacenter?.** Disponível em <https://www.paloaltonetworks.com.br/resources/learning-center/what-is-a-data-center.html>. Acesso em 25 de Junho de 2017

# Modelos de Nuvem



Atualmente, existem três modelos de nuvem oferecidos para as organizações: nuvem pública, nuvem privada e nuvem híbrida. Cada uma é responsável por suprir necessidades específicas de cada projeto, evitando a necessidade de investir em um data center com uma estrutura complexa e cara.

**☑ Nuvem pública:** É o ambiente de cloud computing provido por empresas de datacenter contratado e onde todos os clientes compartilham os mesmos recursos e espaços tendo as configurações de segurança e as variações de disponibilidade limitadas ao gerenciador do sistema, uma vez que são totalmente geridos pelo prestador do serviço. Os clientes se beneficiam de economias de escala, pois os custos de infraestrutura são divididos entre os usuários. Cada cliente paga somente pelos recursos e demandas de capacidades atendidas tornando a alternativa mais barata para manter toda a estrutura em um servidor próprio. Google e Amazon.com são exemplos de cloud pública.

**☑ Nuvem privada:** É o ambiente de cloud computing dedicado para atender

exclusivamente uma empresa de forma individual. Elas permitem que a empresa hospede aplicativos e infraestruturas completas na nuvem e dediquem maiores esforços relacionados à segurança dos acessos e controles de dados, e uma infraestrutura 100% dedicada para as demandas do negócio de um cliente.

📌 **Nuvem híbrida:** É o ambiente onde existe a associação da nuvem pública com a nuvem privada, ou seja, a empresa possui ambiente próprio de nuvem (nuvem privada) e também utiliza aplicações ou ambientes de terceiros, compartilhados com outras empresas (nuvem pública).

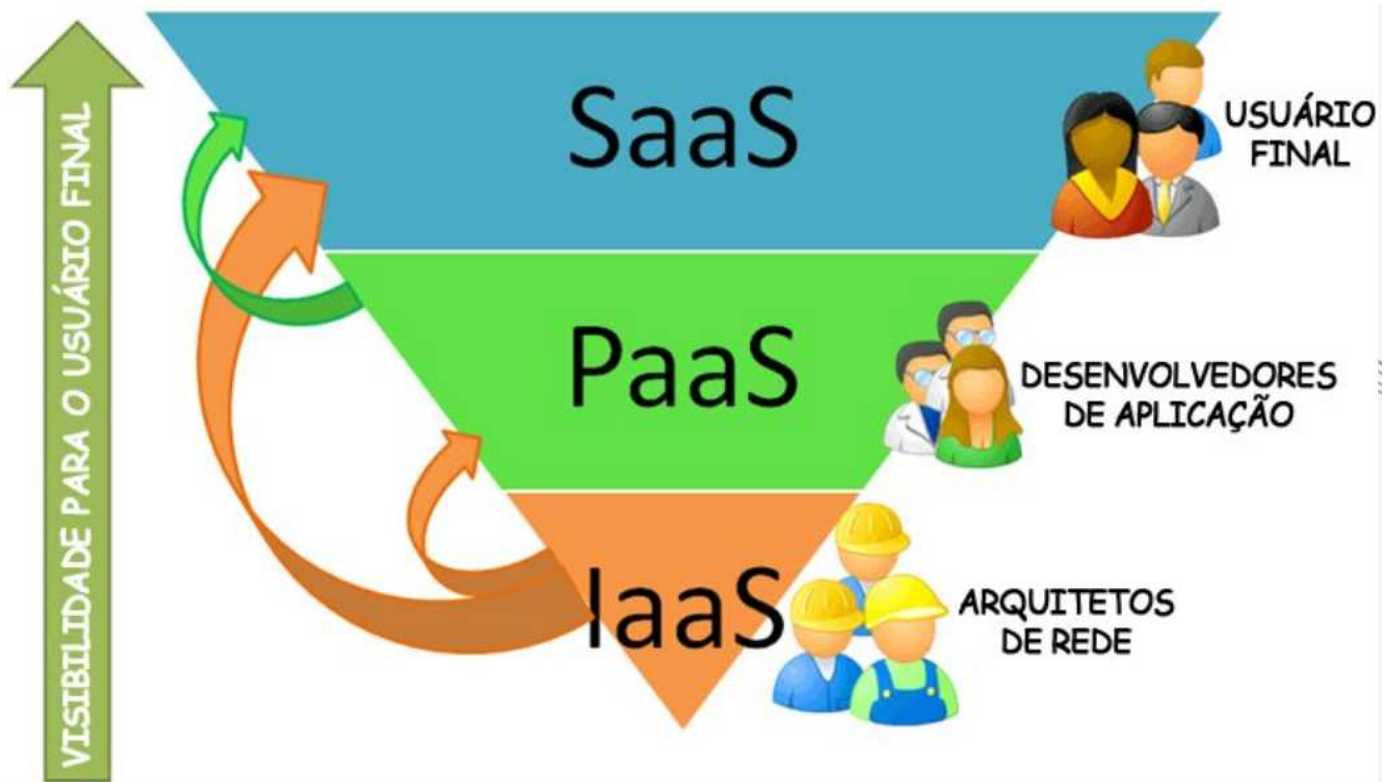
Os provedores de nuvem também fornecem tipos de serviço focados em cada necessidade:

**SaaS – Software as a Service (Software como Serviço):** É um serviço focado em prover a aplicação para os usuário, sem necessidade que ele interaja ou mesmo conheça a infraestrutura por trás existente. Vamos ter como um exemplo, os aplicativos de Edição do Google Drive (Documentos, Planilhas e Apresentações) onde o usuário não precisa de nenhum aplicativo instalado em sua máquina para utilizar e não tem nenhum acesso ou opção de configuração da infraestrutura por trás. Neste modelo você pode ser cobrado pela regra de licença de uso do software.

**IaaS – Infrastructure as a Service (Infraestrutura como Serviço):** É um serviço focado em prover servidoras virtuais (e outros dispositivos de infraestrutura), isentando o usuário do custo de comprar e manter servidoras físicas, roteadores, racks e outras “caixas” de hardware. Aqui você é cobrado, normalmente, pelo tempo e quantidade de recursos ao qual está alocado em sua conta. Neste caso, [Amazon Web Services](#) e a [Digital Ocean](#) sejam bons exemplos para quem queira pesquisar mais sobre o assunto.

**PaaS – Platform as a Service (Plataforma como Serviço):** Aqui temos um modelo que fica entre o SaaS e IaaS, onde é possível a utilização de softwares de maneira mais flexível, sendo possível desenvolver e/ou instalar suas próprias aplicações baseadas em alguma tecnologia (framework, linguagem de programação,

etc.) utilizando uma infraestrutura pré-configurada e adequada a aplicação desenvolvida. O [Heroku](#) é um exemplo, nele você cria uma aplicação com a linguagem de programação que preferir (PHP, Python, Ruby, Javascript, etc.) faz o upload do código e ela estará no ar em poucos segundos.



# 2. Assumindo o controle

## 2.1 CRIANDO UMA SERVIDORA

# Linux e softwares de código aberto



Segundo a *Wikipedia*, Linux é um termo popularmente empregado para se referir a sistemas operacionais (português brasileiro) que utilizam o Kernel Linux. O núcleo (ou kernel, em Alemão) foi desenvolvido pelo programador finlandês Linus Torvalds, inspirado no sistema Minix. O seu código-fonte está disponível sob a licença GPL (versão 2) para que qualquer pessoa possa utilizar, estudar, modificar e distribuir o sistema livremente de acordo com os termos da licença. A Free Software Foundation e seus colaboradores recomendam o nome GNU Linux para descrever o sistema operacional, como resultado de uma disputa controversa entre membros da comunidade de software livre e código-aberto.

O desenvolvimento do Linux é um dos exemplos mais proeminentes de colaboração de software livre e de código aberto. O código-fonte pode ser usado, modificado e distribuído - com fins comerciais ou não - por qualquer um, respeitando as licenças, como a GNU General Public License versão 2, devolvendo o código desenvolvido de volta para o desenvolvimento do núcleo. Vamos ver a definição de software livre segundo a GNU:

Por “software livre” devemos entender aquele software que respeita a liberdade e senso de comunidade dos usuários. Grosso modo, isso significa que os usuários possuem a liberdade de executar, copiar, distribuir, estudar, mudar e melhorar o software. Assim sendo, “software livre” é uma questão de liberdade, não de preço. Para entender o conceito, pense em “liberdade de expressão”, não em “cerveja grátis”. Por vezes chamamos de “libre

software” para mostrar que livre não significa grátis, pegando emprestado a palavra em francês ou espanhol para “livre”, para reforçar o entendimento de que não nos referimos a software como grátis.

Nós fazemos campanha por essas liberdades porque todo mundo merece. Com essas liberdades, os usuários (tanto individualmente quanto coletivamente) controlam o programa e o que ele faz por eles. Quando os usuários não controlam o programa, o programa controla os usuários. O desenvolvedor controla o programa e, por meio dele, controla os usuários. Esse programa não livre é “proprietário” e, portanto, um instrumento de poder injusto.

Um programa é software livre se os usuários possuem as quatro liberdades essenciais: \* A liberdade de executar o programa como você desejar, para qualquer propósito (liberdade 0). \* A liberdade de estudar como o programa funciona, e adaptá-lo às suas necessidades (liberdade 1). Para tanto, acesso ao código-fonte é um pré-requisito. \* A liberdade de redistribuir cópias de modo que você possa ajudar outros (liberdade 2). \* A liberdade de distribuir cópias de suas versões modificadas a outros (liberdade 3). Desta forma, você pode dar a toda comunidade a chance de beneficiar de sua mudanças. Para tanto, acesso ao código-fonte é um pré-requisito.

Um programa é software livre se ele dá aos usuários todas essas liberdades de forma adequada. Do contrário, ele é não livre. Enquanto nós podemos distinguir vários esquemas de distribuição não livres em termos de eles falham em serem livres, consideramos todos eles igualmente antiéticos.

Em qualquer cenário, essas liberdades devem ser aplicadas em qualquer código do qual planejamos fazer uso, ou que levamos outros a fazer uso. Por exemplo, considere um programa A que automaticamente inicia um programa B para lidar com alguns casos. Se nós planejamos distribuir A como está, isso significa que usuários precisarão de B, de forma que nós precisamos julgar se tanto A quanto B são livres. Porém, se nós planejamos modificar A de forma que ele não use B, apenas A precisa ser livre; B não é pertinente àquele planejamento.

“Software livre” não significa “não comercial”. Um programa livre deve estar disponível para uso comercial, desenvolvimento comercial e distribuição comercial. Desenvolvimento comercial de software livre deixou de ser incomum; tais software livre comerciais são muito importantes. Você pode




ter pagado dinheiro por suas cópias de software livre, ou você pode tê-las obtido a custo zero, mas independentemente de como você conseguiu suas cópias, você sempre deve ter a liberdade para copiar e mudar o software, ou mesmo para vender cópias.

Um programa livre deve oferecer as quatro liberdades a qualquer usuário que obtenha uma cópia do software, desde que o usuário tenha atendido até o momento com as condições da licença livre que cobre o software. Colocar algumas das liberdades fora dos limites para alguns usuários, ou exigir que os usuários paguem, em dinheiro ou em espécie, para exercê-las, equivale a não conceder as liberdades em questão e, portanto, torna o programa não-livre.

Um exemplo de software livre é o [LibreOffice](#), que é uma alternativa completa que substitui o pacote Microsoft Office (Word, Excel, Powerpoint, etc.) e um exemplo de sistema operacional livre é o [Ubuntu Desktop](#), que é um sistema baseado em GNU/Linux.

Sempre que falamos sobre um sistema operacional, é inevitável que haja uma comparação com um sistema que sempre está em alta no mercado, como o Windows. Servidores Linux são os mais utilizados por especialistas e isso pode significar mais que a simples popularidade do sistema. Afinal, se o servidor Linux é tão utilizado no mundo, é sinal de que vale a pena dedicar um tempo para adquirir o conhecimento necessário para saber administrar um servidor Linux.

Quando uma usuária não possui um nível de conhecimento mais avançado sobre servidoras, é natural que o marketing aplicado pela Microsoft se torne um fator determinante. Porém, os benefícios do sistema Linux são muito maiores com relação à: segurança, custos e estabilidade:

 **Estabilidade:** Os sistemas Linux são extremamente estáveis e não costumam apresentar falhas mesmo se você o utilizar durante anos. Isso porque, em comparação ao Windows, o Linux se adapta muito melhor para lidar com processos simultâneos. Além disso, sempre que há a necessidade de fazer atualizações nas configurações do sistema, diferente do Windows, elas podem

ser feitas sem o famoso reboot (reinicialização). Ou seja, é possível atualizar o sistema enquanto ele está em execução!

🔧 **Código-Aberto (Open-Source):** Ao adquirir um produto Windows, a única certeza que você poderá ter é que o sistema será sempre aquele. Somente as atualizações do fabricante serão instaladas, não há como customizar, sem autorização, o sistema conforme suas necessidades. O Linux é um sistema open source, ou seja, você mesmo pode modificar o sistema e customizá-lo da forma que quiser — caso tenha conhecimento técnico, é claro — ou, também, contratar especialistas em Linux para fazer as modificações em seu servidor VPS Linux.

🔧 **Custos:** Adquirir a licença para utilizar um software de prateleira, como o Windows Server, envolve custos muito mais elevados do que o Linux, que, por sua vez, possui várias distribuições gratuitas — e as que contam com um suporte dedicado costumam ser baratas.

🔧 **Segurança:** Enquanto o Windows é um sistema que contém brechas para que usuários comuns acessem/ocultem arquivos que só deveriam ser acessados pelo administrador, no Linux o nível de segurança é muito maior, pois somente o usuário root, o administrador e alguns poucos usuários e programas têm acesso a outros usuários e ao kernel. Outro aspecto de segurança é que o Linux, dentre outras coisas, por contar com uma imensa comunidade de desenvolvedores ao redor do mundo, sofre um número muito menor de ataques oriundos de vírus e malwares.

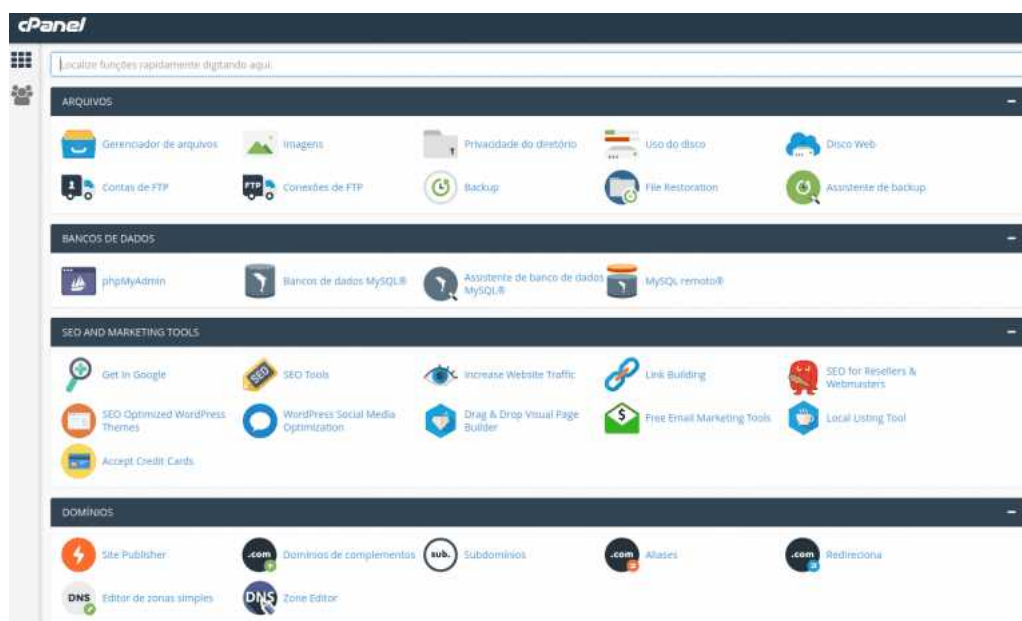
Agora você pode estar se perguntando “só é possível administrar um servidor Linux após estudar muito ou fazer um curso?”. Não. Você não precisa necessariamente se especializar para começar a administrar o sistema, basta aprender a utilizar alguns comandos básicos que ensinaremos no próximo capítulo. Porém, é sempre bom ter um aprendizado contínuo para melhorar a técnica de gerenciar o servidor, assim terá conhecimento para fazer melhorias e lidar com incidentes, que vai de falhas no serviço até ataques a segurança do servidor.

**Fontes:** \* O que é o software livre? - <https://www.gnu.org/philosophy/free-sw.pt-br.html> \* Linux - <https://pt.wikipedia.org/wiki/Linux>

# Aluguel de servidoras

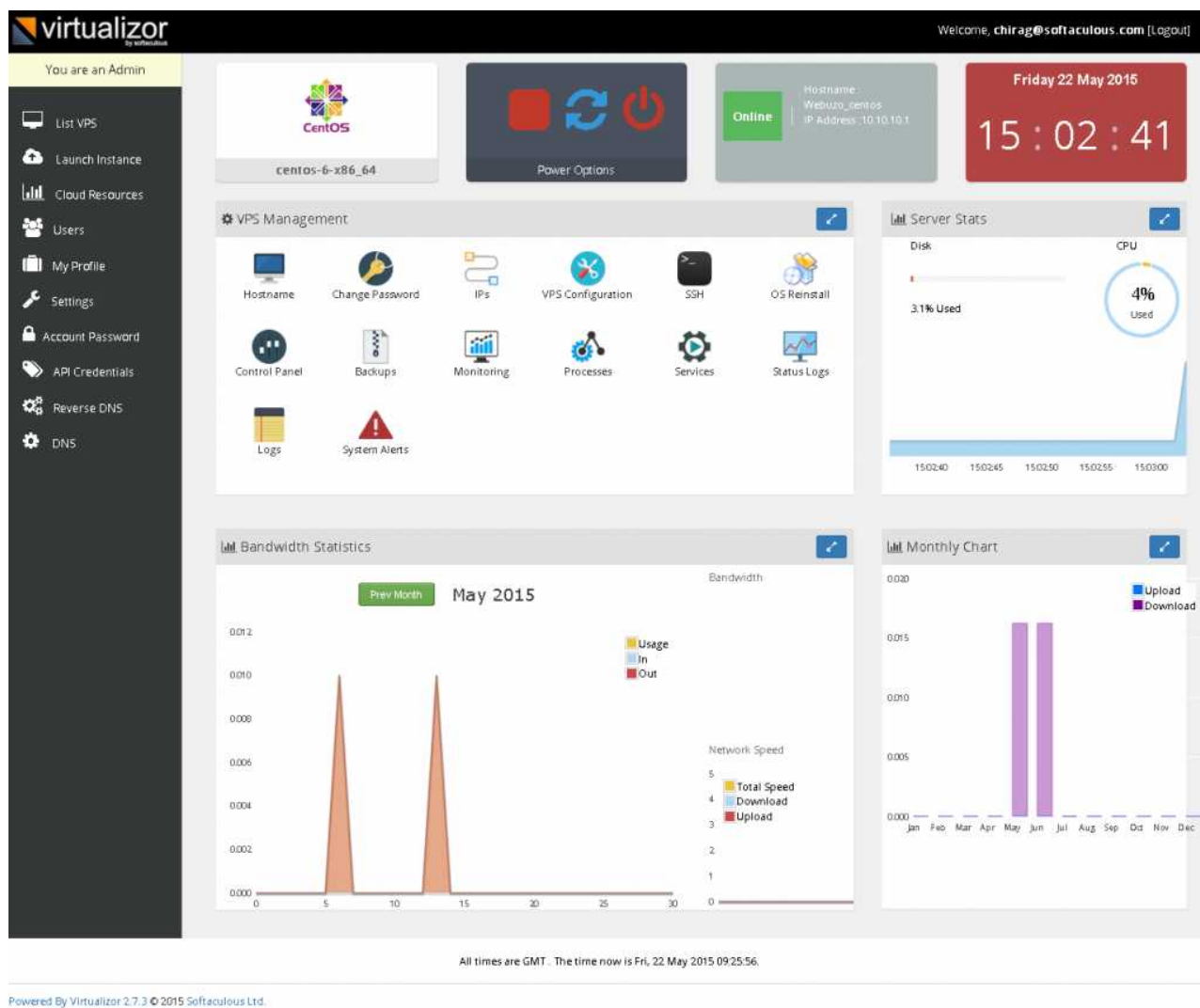
Como vimos no capítulo anterior, para ter um serviço no ar acessível pelo mundo todo precisamos de uma servidora. As empresas que fornecem serviço de aluguel de espaço em servidoras de sites ou servidoras inteiras, conhecido como Host ou Hospedagem, oferecem planos e facilidades para todos os bolsos e necessidades. É preciso ter cuidado no momento da escolha de um plano de hospedagem compartilhada para seu domínio, principalmente em relação à compatibilidade de seu site com a plataforma operacional dos servidores de hospedagem do provedor.

Serviços de Hospedagem de Sites Compartilhada é uma hospedagem onde o provedor possui uma ou mais servidoras que hospedam dados de centenas de clientes simultaneamente, que irão ter acesso restrito aos serviços instalados na mesma por estar compartilhado. Geralmente esse tipo de hospedagem oferece um Painel onde você pode criar contas de e-mail, usuários do serviço de Transferência de Arquivos (FTP/SCP), configuração do domínio, métricas de monitoramento do site, um banco de dados e configuração básica do servidor Web e da Linguagem da programação.



Fonte: [hostinger.com.br](http://hostinger.com.br)

Serviço de Hospedagem de VPS (Virtual Private Server - Servidora Virtual Privada) é uma hospedagem onde a empresa possui um datacenter e provê aos seus clientes servidoras virtuais e compartilham a responsabilidade de manter os recursos operacionais para manter a máquina disponível. Normalmente, é oferecido um painel onde você pode gerenciar o ciclo de vida da servidora virtual contratada (criar, ligar, reiniciar, desligar e destruir) e os demais recursos de hardware, como disco, rede, memória, processador, etc.



De forma mais prática, a principal diferença entre elas que temos que levar em conta é o nível de personalização e autonomia que cada uma provê. Provedores de hospedagem de site te fornece uma solução PaaS, onde não temos autonomia para instalar e configurar aplicações no Linux, tendo que abrir um ticket e negociar o que queremos com o provedor. Se foi contratado uma VPS, é só instalar você mesma a linguagem que precisar pois a servidora é sua e o controle

do que está instalado nela é seu. Um exemplo de situação que isso pode ser um problema é quando hospedagem compartilhada fornece suporte à linguagem PHP mas não Node.js , então temos que abrir um ticket pedindo que seja liberado no seu plano o Node.js . A Hospedagem pode dizer que não é possível, pois o servidor é compartilhado ou irá te cobrar uma taxa adicional para ter disponível essa linguagem.

Mas como disse o Tio do Spiderman *Com grandes poderes, vêm grandes responsabilidades*. Quando contratamos uma VPS significa que temos total controle e responsabilidade sobre as configurações delas, os dados que armazenamos, os serviços instalados e tudo mais. Então se um serviço instalado na mesma sai do ar e a máquina está ligada, é nossa a responsabilidade de verificar o que aconteceu e consertar o serviço para voltar para o ar. Em uma Hospedagem Compartilhada, é de responsabilidade dela manter os serviços no ar, nossa responsabilidade é sobre os arquivos de código-fonte que configuramos usando uma ferramenta de transferência de arquivos ou Painel.

Com isso, precisamos adquirir conhecimento técnico em ambos os casos para manter nossos serviços acessíveis ao pública. Neste capítulo vamos aprender a criar uma servidora virtual em nosso próprio computador para irmos treinando como seria em uma VPS.

## **Termo de uso e Política de privacidade de Hospedagens**

Os documentos Termo de uso e Política de privacidade são documentos que oficializam a relação entre as empresas e clientes (nós), tendo estabelecidas obrigações e direitos dos quais é preciso que todos os envolvidos estejam cientes. A cada aplicativo baixado ou site que se navega, as pessoas aceitam, explicitamente ou não, os termos de uso e as políticas de privacidade impostos pela plataforma. Eles são dois contratos distintos, mas em sua maioria são encontrados no mesmo documento, como se fossem um só. Ou seja, ambos são contratos de adesão e bilaterais. Mas o que é cada um?

Uma política de privacidade é um documento que explica como um website lida com toda a informação coletada dos seus utilizadores e/ou clientes. É uma boa prática dizer aos utilizadores como os sites armazenam e fazem uso desses dados.

Um documento de Termos e Condições de Uso um site ou um aplicativo explica aos usuários quais são as condições de utilização do serviço disponibilizado, seja ele gratuito ou pago. Dessa forma, este documento é como um contrato estabelecido entre o site e seu cliente que o acessa, determinando os seus direitos e suas responsabilidades como organização e também os direitos e responsabilidades do seu consumidor.

De modo geral, deveríamos ler SEMPRE os termos de uso, principalmente de serviços onde guardamos informações importantes da nossa vida. E pensando que estamos colocando um site em um serviço fornecido por terceiros, precisamos estar atentos as condições desse site. Nesses termos, é importante sempre lermos os seguintes tópicos:

- 📌 Responsabilidade
- 📌 Quebra de sigilo
- 📌 Identificação
- 📌 Backup e perda de dados
- 📌 Garantia
- 📌 Disponibilidade
- 📌 Dados que coletam

Buscando estes pontos nos contratos, podem ajudar a decidir qual hospedagem está de acordo com a forma que deseja operacionalizar seu site/servidora e os serviços dela.

Vamos ver alguns exemplos de trechos dos termos encontrados em hospedagens de sites que operam no Brasil:

*Como compartilhamos dados pessoais: Transferência de Ativos: Caso uma empresa*

*ou unidade de negócio do Grupo seja reorganizada ou vendida e sejam transferidos todos ou substancialmente todos os ativos a um novo proprietário, sua informação pessoal pode ser transferida ao comprador independentemente de sua autorização para garantir a continuidade dos serviços;*

Ou seja, tu não tens poder de escolha se gostarias de ter seu site administrado pelo novo dono dela. Mesmo que escolhas cancelar o contrato, seus dados já terão sido compartilhados.

*A Contratada não está obrigada a realizar backup caso ocorra uma das condições abaixo elencadas:*

- \* a. A tabela possuir tamanho superior a 1 (um) Gigabyte.*
- \* b. O banco de dados possuir número igual ou superior a 1.000 (um mil) tabelas.*
- \* c. O banco de dados possuir tabela e/ou dado corrompido ou com qualquer tipo de erro que impossibilite ou dificulte a realização do backup.*
- \* d. Banco de dados access.*

Ou seja, um banco de dados que possui tabelas grandes (imagina uma tabela com informações de pedidos de um site de compras) não teria respaldo de um backup caso aconteça um problema, mesmo sendo um defeito do provedor.

*Sujeitar ou permitir um volume excessivo de tráfego de dados que possa, de qualquer maneira vir a prejudicar o funcionamento do servidor, sem prévio aviso, sob pena de imediata suspensão da prestação dos serviços ora contratados independentemente de aviso ou notificação.*

Se por algum motivo seu site receber MUITOS acessos e isso prejudicar o funcionamento de outros sites que compartilham o mesmo servidor, seu site pode ser tirado do ar sem aviso prévio.

*Nossa Obrigação Legal de Usar ou Revelar Dados Pessoais: Nós iremos revelar seus Dados Pessoais ao estado e autoridades públicas sem sua permissão prévia apenas quando somos legalmente requeridos para fornecer informações, incluindo tomar*

*medidas legais para defender nossos direitos, assim como os casos onde temos uma crença em boa fé que o acesso, uso, preservação ou revelação da informação é razoavelmente necessária para atender qualquer lei aplicável, regulamento, processo legal ou solicitação governamental, enforce Termos de Serviço aplicáveis, incluindo investigação de violações em potencial, detecte, previna ou de outra forma enderece fraude, segurança ou problemas técnicos.*

Ou seja, você não será avisado se suas informações serão repassadas por algum órgão decorrente a algum processo.

*Servidores VPS: A XXXXXX se reserva o direito de alterar as senhas de contas administrativas (superusuários) de servidores sempre que estas estiverem diferentes das fornecidas pelo cliente. O cliente é responsável por fornecer à XXXXXX informações atualizadas sobre as senhas do servidor, evitando assim indisponibilidade dos serviços decorrente de procedimentos que tenhamos de realizar para redefinição de senha. A XXXXX NÃO realiza backups ou cópias de segurança de servidores VPS. É sua responsabilidade manter seu próprio backup ou encontrar uma solução para isso. Ao contratar nossos serviços, você está de acordo em manter seu próprio backup em um lugar seguro.*

Nessa condição, o provedor tem acesso ao seu servidor e aos dados contido nele apesar da mesma não se responsabilizar por manter um backup do servidor, caso ele tenha algum defeito ou necessite recuperar informações.

Para ter acesso aos Termos, busque o link no rodapé da página da Hospedagem que escolher.

## Saiba mais

**Não Li e Concordo – Revista Super Interessante – São páginas e páginas de termos misteriosos, usadas por sites e redes sociais para explicar os seus direitos e deveres. Alguns têm palavrões e piadas, outros escondem cláusulas abusivas. No final, você concorda com todas. Afinal, quem lê contratos?**



## Escolhendo a melhor plataforma de hospedagem

Para escolhermos a hospedagem ideal para nossos serviços, devemos primeiro saber qual será o objetivo do site. Se for um site de um grupo de Slam, podemos pensar se o formato seria de um blog ou de um portal onde é possível assistir as batalhas e fazer download das gravações. Se for um site de Roupas para Pets, talvez possa ser uma plataforma de vendas ou só um catálogo dos produtos e a cliente entra em contato por outro meio para comprar. Independente do objetivo do site, é ele que nos dará um ponto de partida para decidir quais softwares precisaremos.

O serviço/software que iremos utilizar tem que ser compatível com os objetivos do site. Então se é um blog precisamos pensar em um sistema de blog, se for um servidor de arquivos precisamos pesquisar um software de sistema de arquivos. Para ilustrar, vamos voltar aos exemplos:

- Para subir um serviço de blog ou portal podemos instalar um [Wordpress](#), que é a plataforma de conteúdo mais utilizada no mundo ou o [Drupal](#) que é outro famoso sistema de gestão de conteúdo.

- Para criar uma loja podemos contratar um serviço de Lojas Virtuais, que nos dará uma plataforma de criação e gerenciamento da loja ou instalar um software de loja virtual como o [Magento](#) ou [PrestaShop](#).

- Para ser só uma vitrine, pode construir uma página programando-a em HTML + CSS + Javascript.

Para auxiliar na pesquisa de softwares e alternativas gratuitas para serviços pagos, utilize o site [AlternativeTo](#). Ele mostra alternativas para serviços online, softwares, sistemas, aplicativos de celular, etc.

Vamos supor que iremos criar um Blog para o grupo de Slam e escolhemos o WordPress como plataforma, qual é o próximo passo? Saber os requisitos para instalá-lo. Igual quando queremos fazer um Risoto de Shimeji sem nunca ter feito, precisamos buscar os ingredientes necessários para fazer o prato. Você pode buscar a receita na internet ou perguntar para uma amiga e ela te falara o

que precisa para instalar. No caso dos softwares, o que é necessário para realizar a instalação você pode encontrar no guia de instalação do site do software. Para ver os requisitos do WordPress entramos neste link <https://br.wordpress.org/about/requirements/> e podemos observar estes pontos:

- 📌 PHP 7.4 ou superior
- 📌 MySQL 5.6 ou superior ou MariaDB 10.1 ou superior
- 📌 Nginx ou Apache com o módulo mod\_rewrite
- 📌 Suporte ao protocolo HTTPS

Estes pontos determinam se a hospedagem ou a VPS que iremos contratar atende a instalação do wordpress. No caso da VPS, não precisamos nos preocupar com isso pois somos as responsáveis por instalar o PHP, MySQL MariaDB e Nginx/Apache na máquina. Mas nos serviços de Hospedagem de Sites devemos buscar nos planos oferecidos se tem esses requisitos mínimos.

A documentação do wordpress não explicita quanto de CPU e Memória utiliza, porém eu sempre indico o mínimo de 512MB para VPS's e acompanhar o desempenho para ver se é necessário mais. O software Nextcloud, por exemplo, [recomenda usar 512 apesar do mínimo ser 128 para a instalação](#).

Agora já possuímos todas as informações para escolher uma hospedagem quando precisarmos. Porém, para essa guia iremos criar uma Máquina Virtual em nosso computador que será a nossa servidora.

# Reunindo as informações para criar nossa Servidora

Aprendemos na sessão anterior como funciona as servidoras, como trafega os dados pela rede e a forma como acessamos o conteúdo, precisamos agora pensar que informações temos que reunir para criar uma servidora para colocarmos nosso site no ar.

Pegue o seu caderno e crie uma tabela como a do exemplo abaixo, precisaremos dessas informações para os próximos passos. Não se preocupem em ter as informações do projeto real que querem criar, para nossos estudos iremos trabalhar com dados fictícios.

<b>Item</b>	<b>Valor</b>
Nome da Servidora	
Domínio	
Serviços	
Hospedagem	
S.O.	
CPU	
Memória	
Disco	
IP	
Credenciais	

Agora uma breve descrição de cada item:

**Nome da Servidora:** É como iremos nomear a nossa servidora. Podemos chamar pelo nome do site que iremos criar, o nome de uma estrela, uma personagem histórica, vilã de algum filme ou que nome acharmos que faz sentido. O importante é que sigamos a regra de ser todo em letras minúsculas e/ou números, sem espaços, acentos e outros caracteres especiais. Se o nome for composto, pode se usar -(hífen) ou **\*\*\_\*\***(sublinhado). Exemplo: blog01, dandara, malevola, portaldosjogos, etc.

**Domínio:** Qual será o endereço do seu site. Exemplo: portaldosjogos.com.br, coletivadandara.org, etc.

**Serviços:** Que serviços nossa servidora irá prover. Exemplo: um site, um blog, loga, compartilhamento de arquivos, mídia, etc.

**Hospedagem:** Qual o provedor de hospedagem ou local físico que irá estar o servidor. Exemplo: Máquina do escritório, Digital Ocean, Maadix, etc.

**S.O. (Sistema Operacional):** Qual o sistema operacional que será executado no servidor. Exemplo: Debian, Ubuntu, CentOS, etc.

**CPU:** Quantos núcleos de processamento nosso servidor irá utilizar. Uma boa prática é seguir as indicações de requisitos que consta nos manuais do serviço que quer instalar na servidora.

**Memória:** Quantos Megabytes/Gigabytes de memória nosso servidor irá utilizar. Uma boa prática é seguir as indicações de requisitos que consta nos manuais do serviço que quer instalar na servidora.

**Disco:** Quanto espaço em disco será reservado para o nosso servidor utilizar. Uma boa prática é seguir as indicações de requisitos que consta nos manuais do serviço que quer instalar na servidora.

**IP:** Qual o IP atribuído para o nosso servidor. No caso de um provedor, um IP será atribuído à máquina criada e iremos utilizá-lo. Em caso de uma rede local própria, selecionaremos um IP e atribuiremos ao servidor.

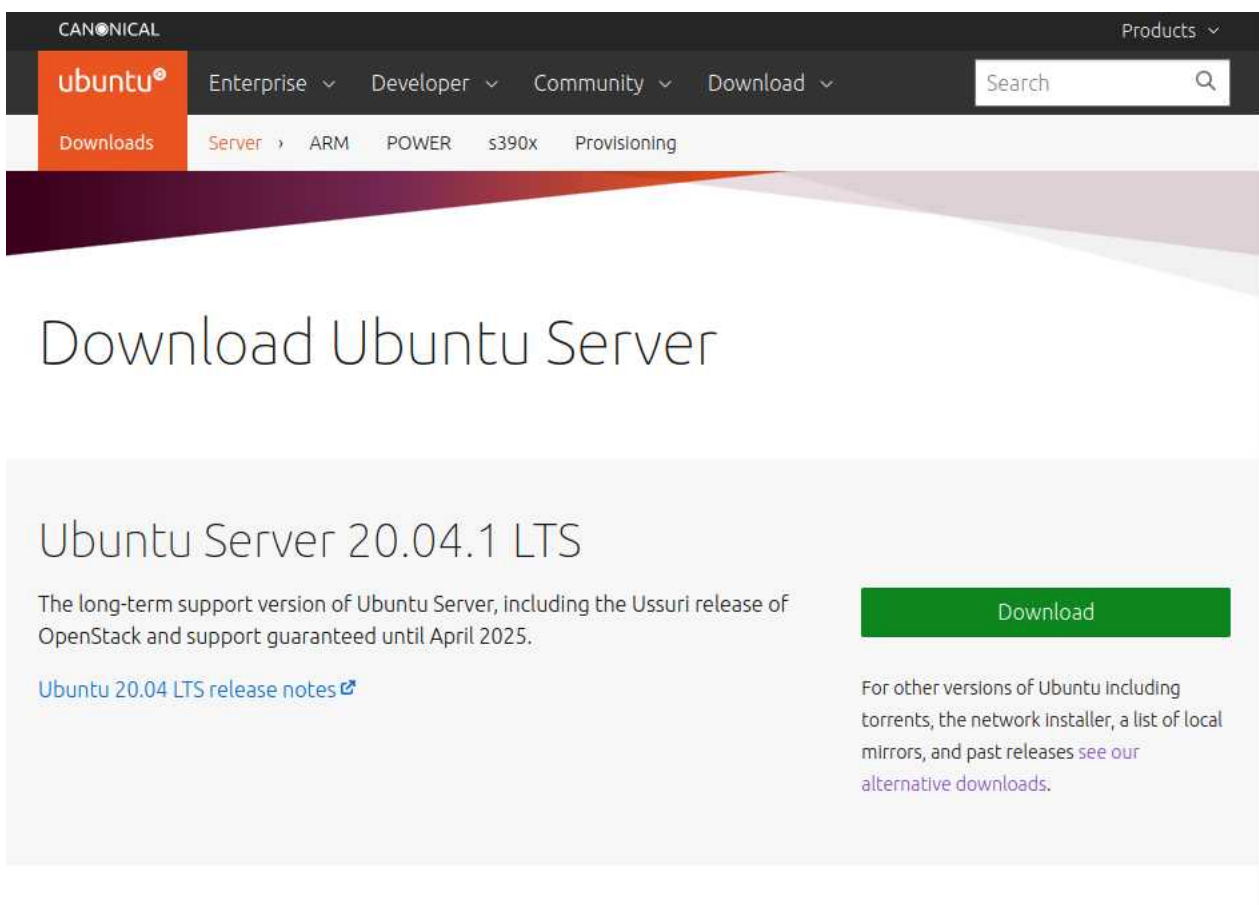
**Credenciais:** Nome, Nome da Usuária, senha e/ou chave de acesso à servidora.

## **2.2 COLOCANDO A MÃO NA MASSA**

# Criando uma Máquina Virtual

Para criarmos a máquina virtual iremos precisar fazer download da Imagem de Disco (DVD/CD) do Ubuntu Server e do Pacote de instalação do VirtualBox.

Para o download do Ubuntu Server, entre na página de download do Ubuntu Server em <https://ubuntu.com/download/server> e clique no botão de **Download**.



Para o download do VirtualBox, entre na página de download do Virtualbox em <https://www.virtualbox.org/wiki/Downloads> e clique nos links correspondentes ao seu Sistema Operacional:

- > VirtualBox 6.x.x Platform packages
  - > Windows -> Windows hosts
  - > MacOS -> OS X hosts
  - > Linux -> Linux distributions
- > VirtualBox 6.x.x Oracle VM VirtualBox Extension Pack
  - > Windows/Mac/Linux -> All supported platforms



# VirtualBox

- About
- Screenshots
- Downloads
- Documentation
  - End-user docs
  - Technical docs
- Contribute
- Community

## Download VirtualBox

Here you will find links to VirtualBox binaries and its source code.

### VirtualBox binaries

By downloading, you agree to the terms and conditions of the respective license.

If you're looking for the latest VirtualBox 6.0 packages, see [VirtualBox 6.0 builds](#). Please also use version 6.0 if you need to run VMs with software virtualization, as this has been discontinued.

If you're looking for the latest VirtualBox 5.2 packages, see [VirtualBox 5.2 builds](#). Please also use version 5.2 if you still need support for 32-bit hosts, as this has been discontinued in 6.0.

#### VirtualBox 6.1.12 platform packages

- [Windows hosts](#)
- [OS X hosts](#)
- [Linux distributions](#)
- [Solaris hosts](#)

The binaries are released under the terms of the GPL version 2.

See the [changelog](#) for what has changed.

You might want to compare the checksums to verify the integrity of downloaded packages. *The SHA256 checksums should be favored as the MD5 algorithm must be treated as insecure!*

- [SHA256 checksums, MD5 checksums](#)

**Note:** After upgrading VirtualBox it is recommended to upgrade the guest additions as well.

#### VirtualBox 6.1.12 Oracle VM VirtualBox Extension Pack

- [All supported platforms](#)

Support for USB 2.0 and USB 3.0 devices, VirtualBox RDP, disk encryption, NVMe and PXE boot for Intel cards. See [this chapter from the User Manual](#) for an introduction to this Extension Pack Personal Use and Evaluation License (PUEL). Please install the same version extension pack as your installed version of VirtualBox.

#### VirtualBox 6.1.12 Software Developer Kit (SDK)

- [All platforms](#)

#### User Manual

The VirtualBox User Manual is included in the VirtualBox packages above. If, however, you would like to take a look at it without having to install the whole thing, you also access it here:

- [User Manual \(HTML version\)](#)

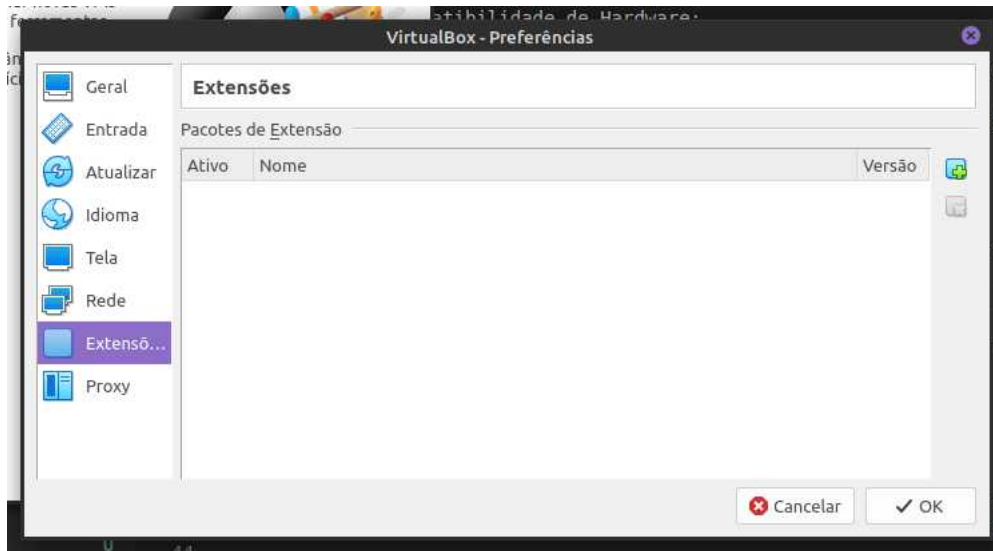
You may also like to take a look at our [frequently asked questions list](#).

Após ter instalado o Oracle VM VirtualBox com sucesso e ter reinicializado o computador, conforme descrito no item anterior, você pode estender as funcionalidades básicas do software com o Extension Pack (pacote de extensões), assim os dispositivos que forem conectadas em sua máquina física estará disponível na máquina virtual. As funcionalidades acrescentadas pelo pacote de extensões são:

- Suporte para dispositivos USB;
- Suporte para webcam;
- Suporte para criptografia de disco;
- Recursos de compatibilidade de Hardware;

Para instalar a Extension Pack no VirtualBox, siga os passos descritos a seguir:

1\_ Abra o VirtualBox e clique no menu **Arquivo > Preferências....** Na janela aberta, Preferências, clique na opção Extensões, no painel à esquerda. Clique no primeiro botão localizado à direita com o símbolo + da lista Pacotes de Extensão.

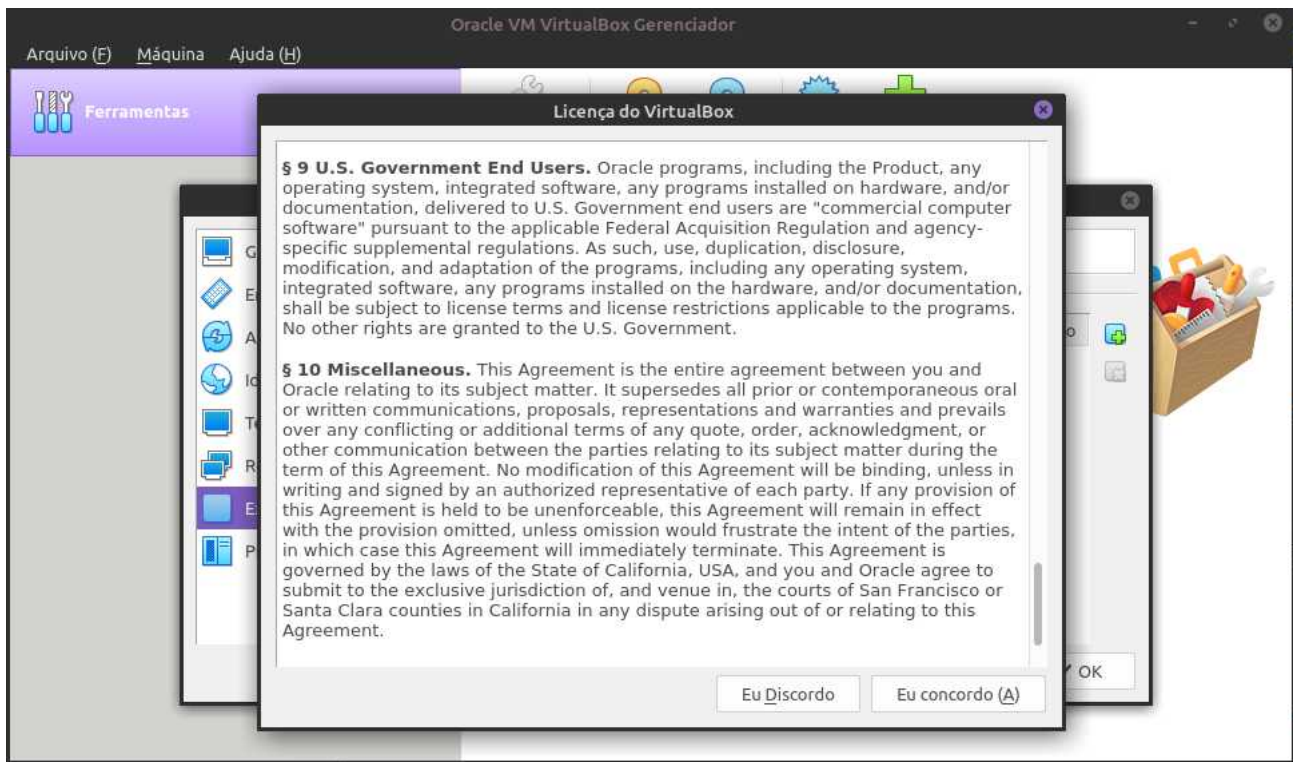


2\_ Na próxima janela, selecione o arquivo do Extension Pack baixado e clique no botão **Abrir**. Será apresentada uma tela de diálogo perguntando se você quer instalar o Extension Pack. Clique no botão **Instalar**.

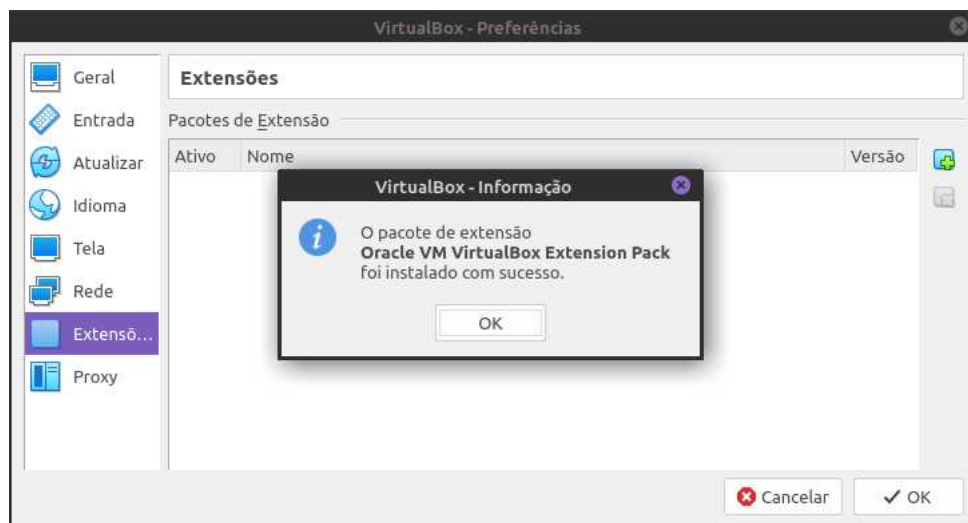


3\_ Agora, será aberta a janela contendo a Licença do VirtualBox. Para poder continuar a instalação do Extension Pack, role a barra de rolagem à direita do campo contendo a Licença até o fim. Leia atentamente a Licença e, caso concorde com os seus termos e condições, clique no botão **Eu concordo** para prosseguir. Se não concordar, não será possível concluir a instalação.





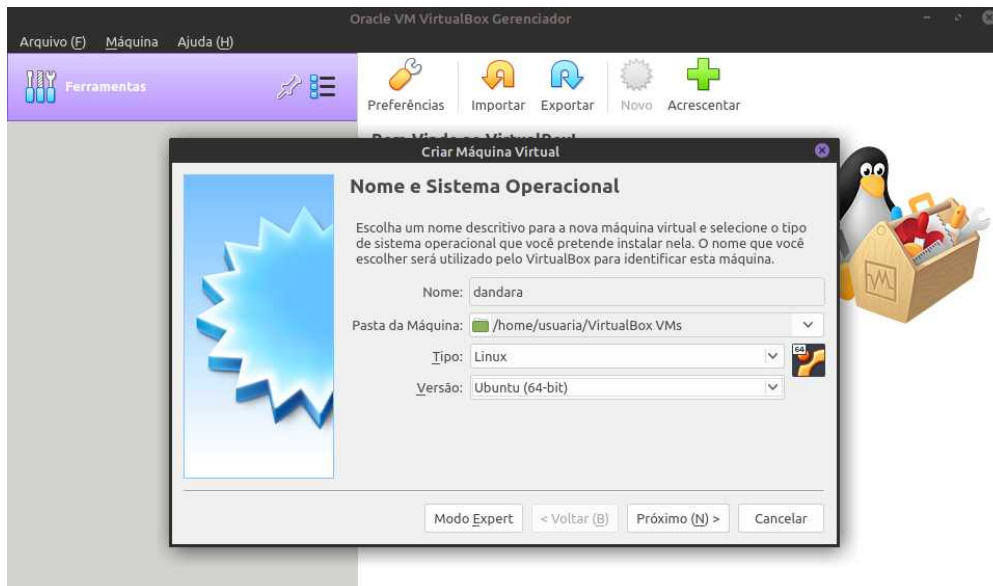
4\_ Na janela seguinte, Autenticar, informe a senha e clique no botão Autenticar para concluir a instalação do Extension Pack. Se a instalação correr bem, será apresentada uma tela de diálogo informativa.



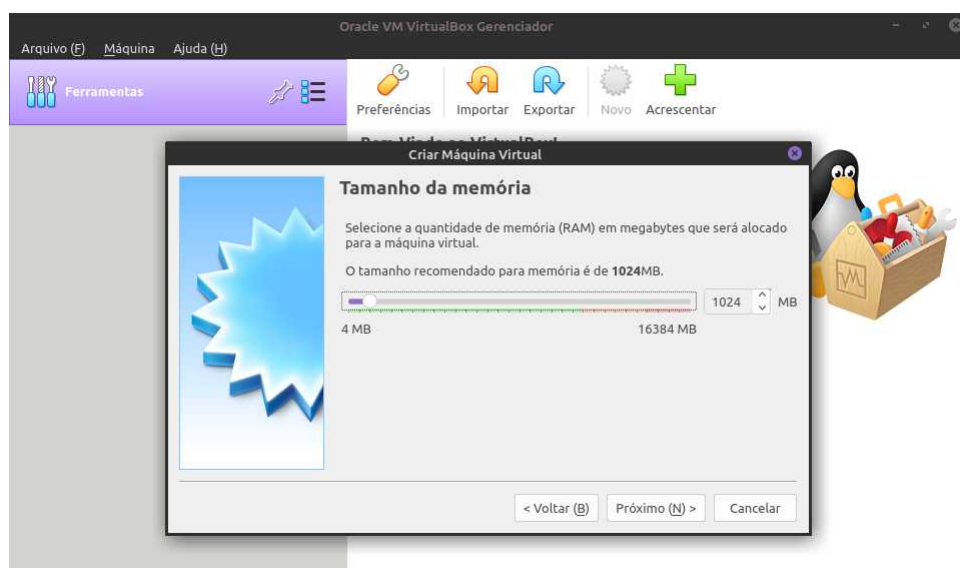
Pronto! O Extension Pack já está instalado no Oracle VM VirtualBox.

De volta a tela inicial do VirtualBox, agora vamos criar uma máquina virtual que será a nossa servidora, iremos usar os valores que preenchemos na tabela como referência.

1\_ Clique no botão **Novo**. Na janela aberta, Criar Máquina Virtual, em **Nome** coloque o *Nome da Servidora* que está na tabela, em **Pasta da Máquina** escolha onde os arquivos gerados serão salvos, em **Tipo** escolha *Linux* e Versão escolha *Ubuntu (64-bit)* . Clique no botão **Próximo**.



2\_ Agora precisamos escolher o tamanho da memória de nossa máquina física iremos reservar para máquina virtual. Iremos configurar 1 Gigabyte/GB (1024 Megabyte/MB), que para os fins dessa guia é o suficiente. Mas atenção, alocar mais memória do que o disponível no sistema pode deixar o computador lento ou travá-lo. Clique no botão **Próximo**.



3\_ Agora iremos criar o disco de nossa máquina física. Selecione *Criar um novo disco rígido virtual agora*. Clique no botão **Criar**.



4\_ Agora iremos escolher o tipo de Disco da Máquina Virtual. Essa escolha determina as compatibilidades do nosso disco com outros softwares devirtualização. Selecione *VDI - (VirtualBox Disk Image)* e Clique no botão **Próximo**.

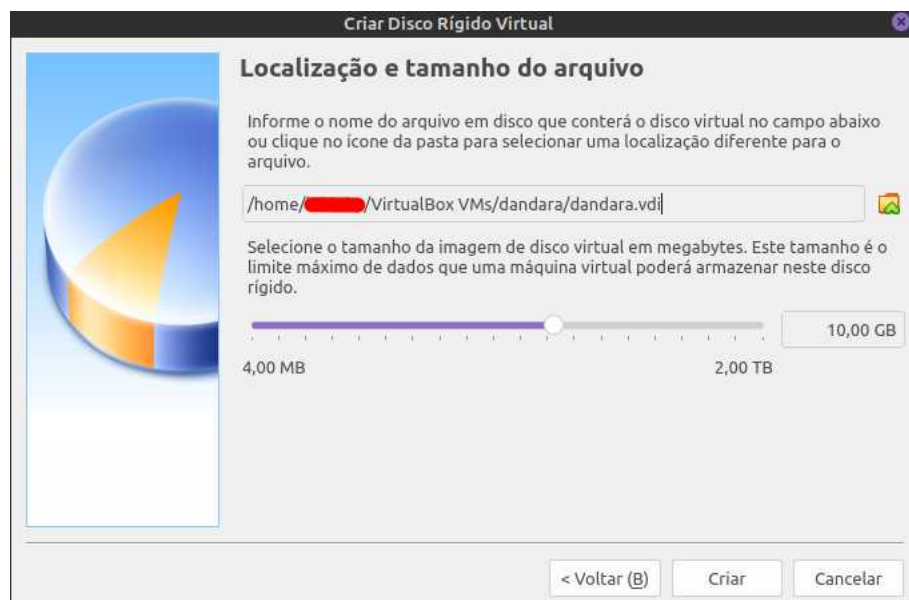


5\_ Agora iremos escolher como será feita a alocação de espaço da nossa máquina virtual. Tamanho Fixo significa que será criado um arquivo em nossa máquina física com o tamanho exato que configurarmos o disco virtual, por exemplo, se configurarmos 50GB será criado uma arquivo de 50 GB. No Tamanho Dinâmico o tamanho do disco cresce conforme o que for consumido na máquina virtual, por exemplo, se eu criar um disco dinâmico de 50 GB ele iria criar um

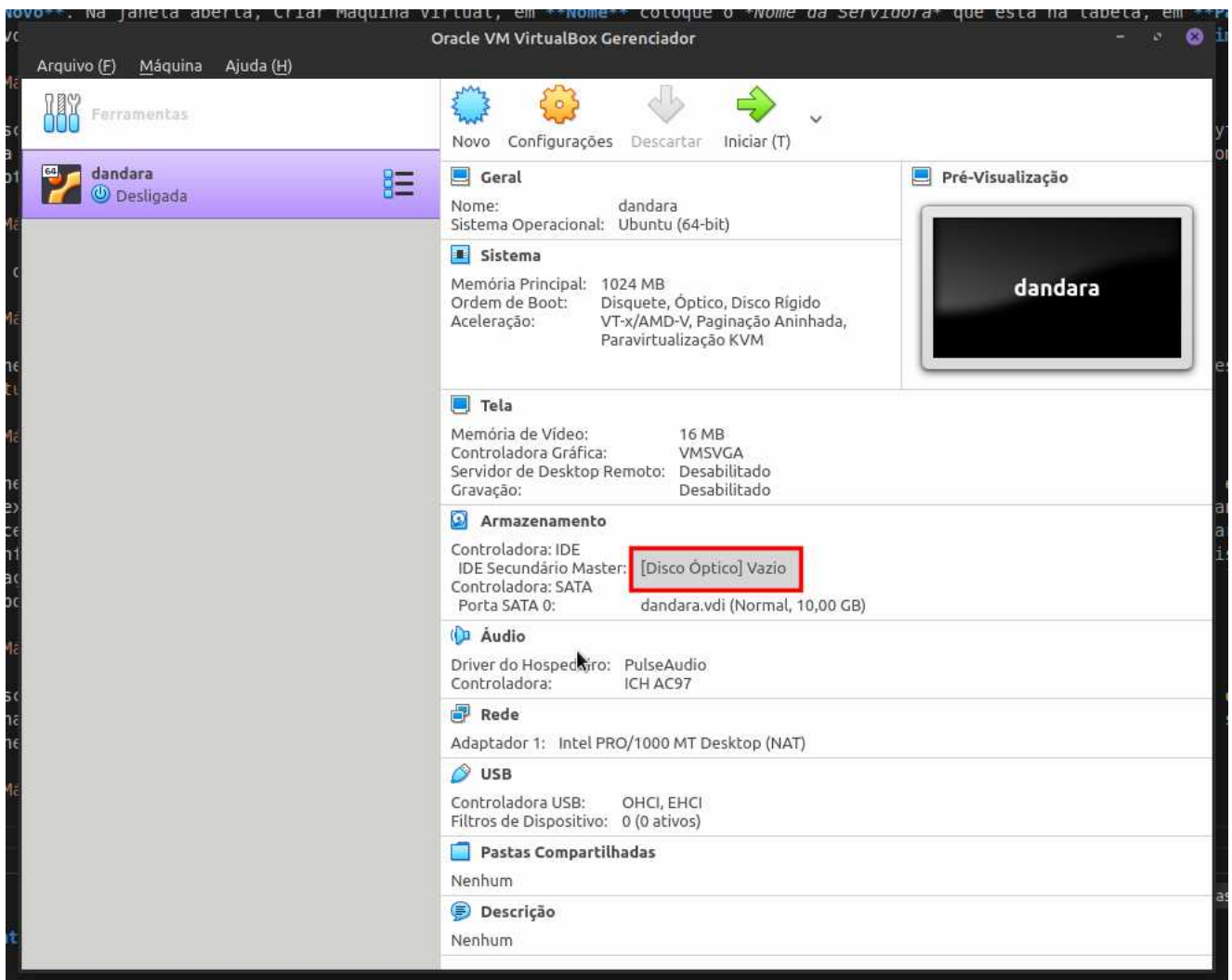
arquivo com poucos megabytes e conforme formos instalando os softwares ele vai crescendo. A vantagem do tamanho fixo é o desempenho e a garantia da disponibilidade de espaço, já que no espaço dinâmico pode acontecer de não ter espaço físico na máquina física e a máquina travar por conta disso. Selecione *Dinamicamente Alocado* e clique no botão **Próximo**.



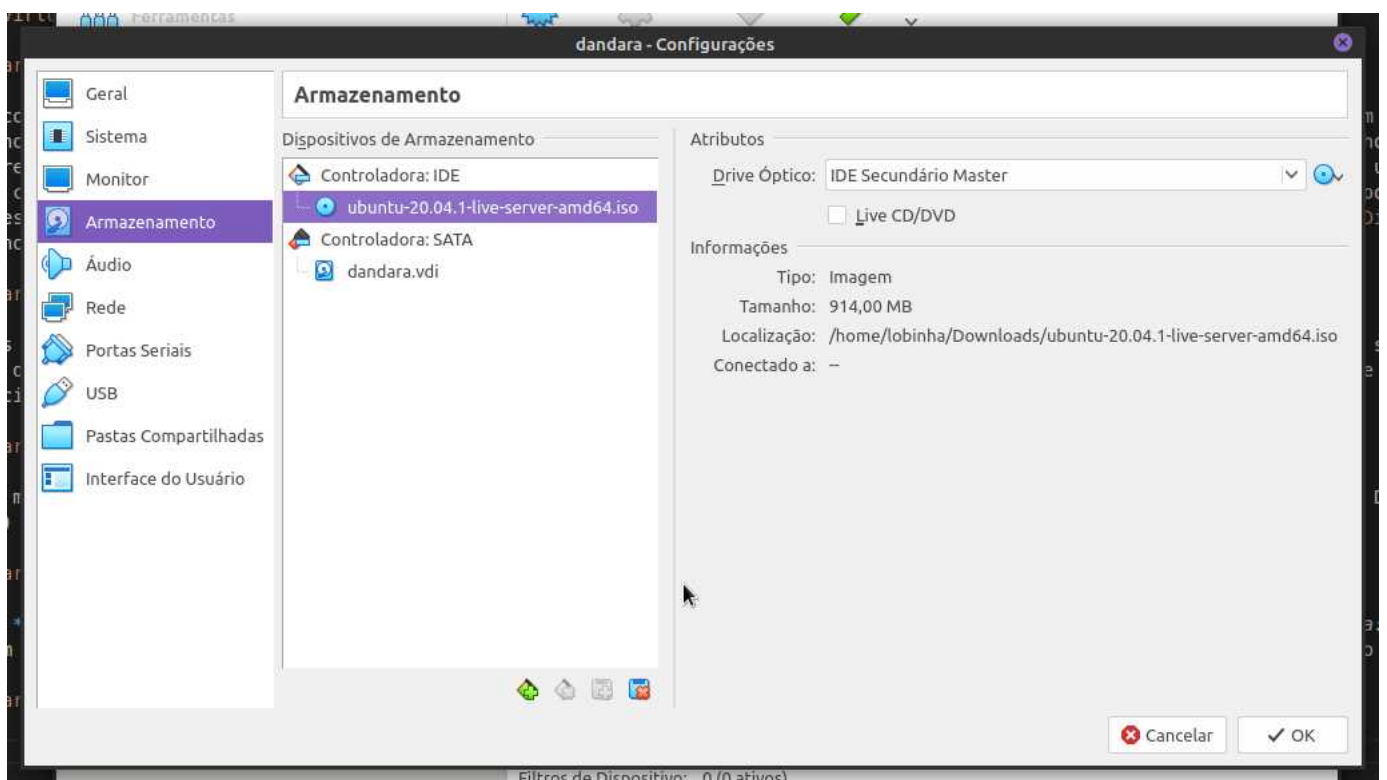
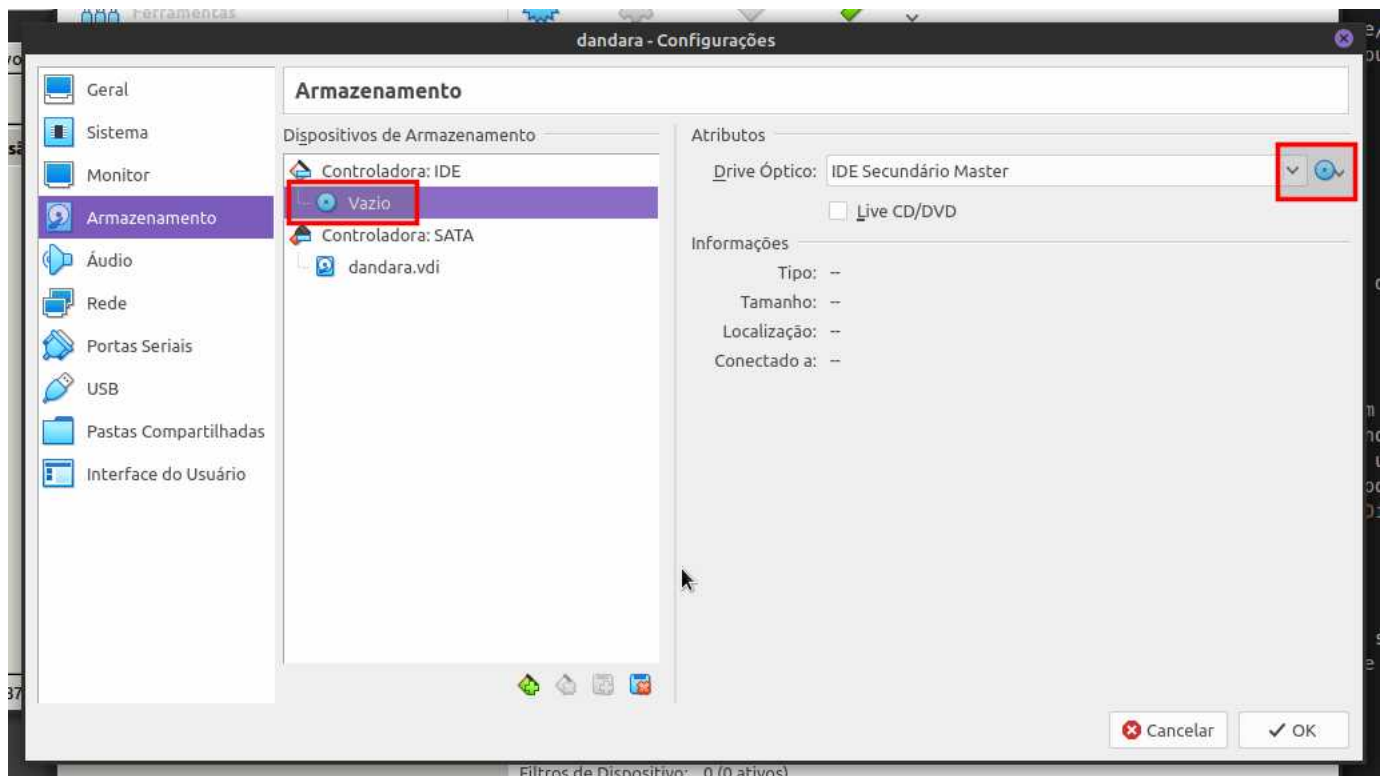
6\_ Agora precisamos escolher o tamanho do disco de nossa máquina virtual. Iremos configurar 10 Gigabyte/GB, que para os fins dessa guia é o suficiente. Mas atenção, no espaço dinâmico pode acontecer de não ter espaço físico na máquina física e a máquina travar por conta disso, então verifique se tem espaço livre o suficiente. Selecione *10,00 GB* e clique no botão **Criar**.



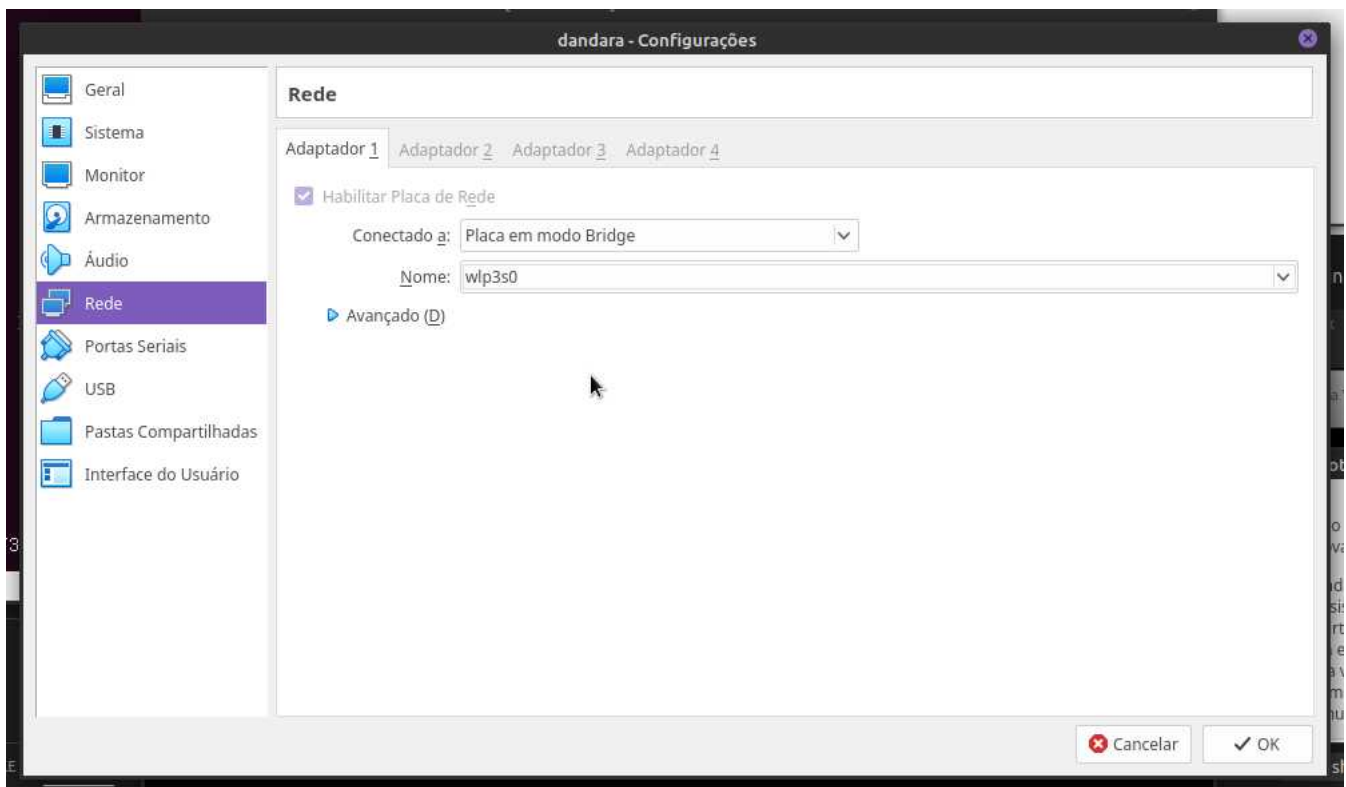
7\_ Pronto! A nossa máquina virtual foi criada e estamos prontas para instalar o Sistema Operacional. Repare na imagem que a nossa Unidade de Disco Óptico (entrada de CD/DVD) encontra-se vazia.



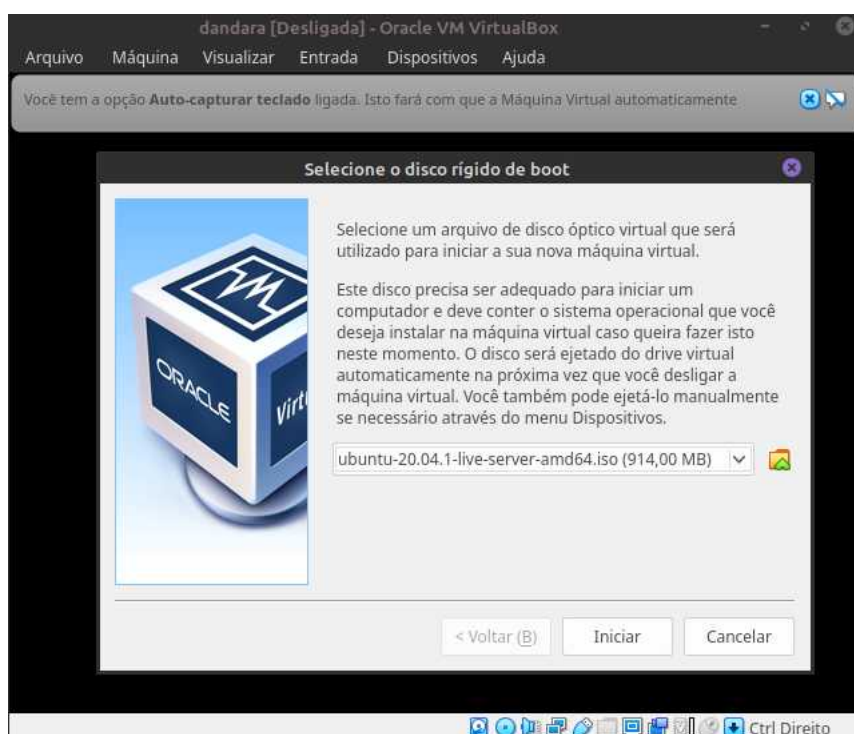
8\_ Clique no botão **Configurações**. Na janela aberta, Configurações, clique na opção Armazenamento, no painel à esquerda. *Em Controladora: IDE* clique no item **Vazio** e em seguida clique no ícone de CD localizado à direita em **Drive Óptico**. Na próxima janela, selecione o arquivo de ISO do Ubuntu Server.



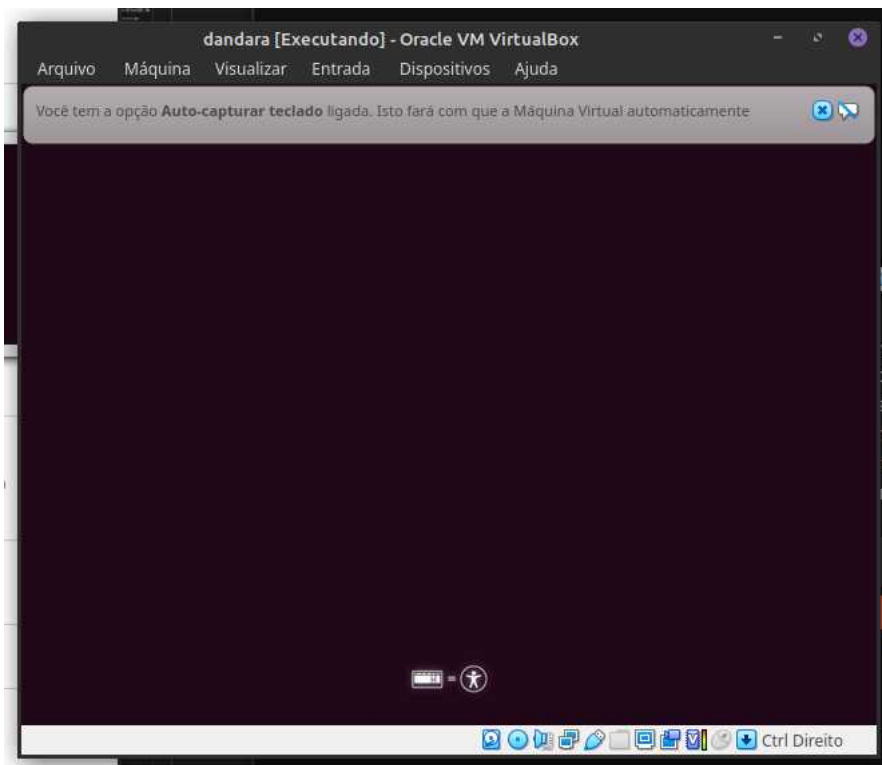
9\_ Agora clique na opção **Rede**. Em **Conectado a:** selecione *Placa em modo Bridge*. Em **Nome:** selecione a opção de rede ativa (WiFi ou a cabeada). Se estiver tudo certo, clique em **OK**.



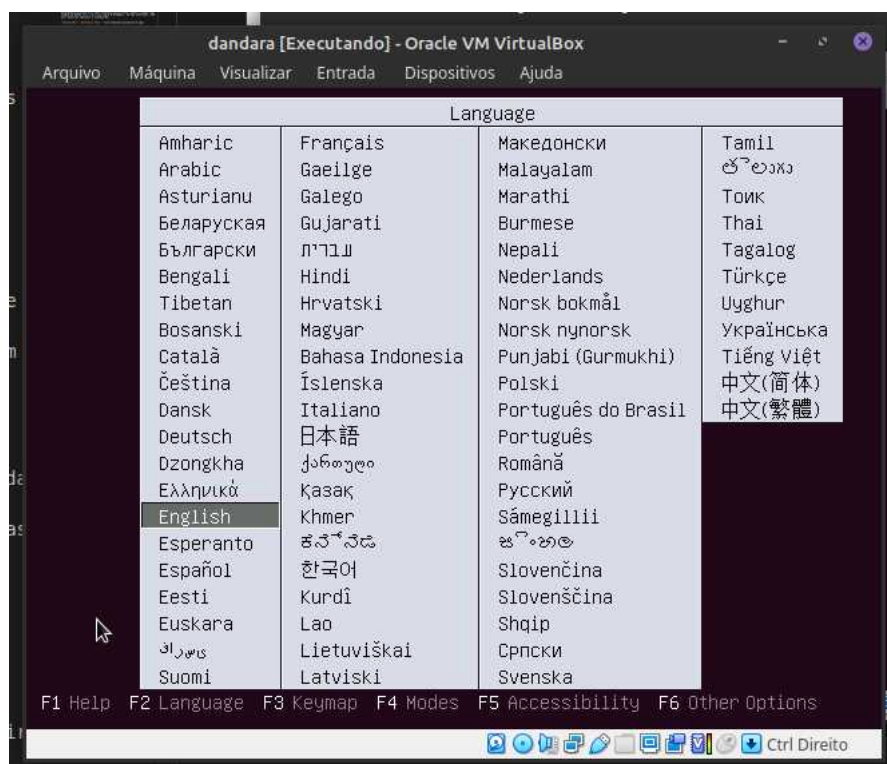
10\_ Agora estamos prontos para ligar a nossa máquina, clique em Iniciar. Se aparecer essa janela, clique em Iniciar para confirmar o início pelo disco óptico (CD).



11\_ Clique na janela que abriu. Se aparecer um aviso sobre capturar o teclado, clique em *Capturar*. Ao aparecer uma tela vinho, pressione **Enter**.

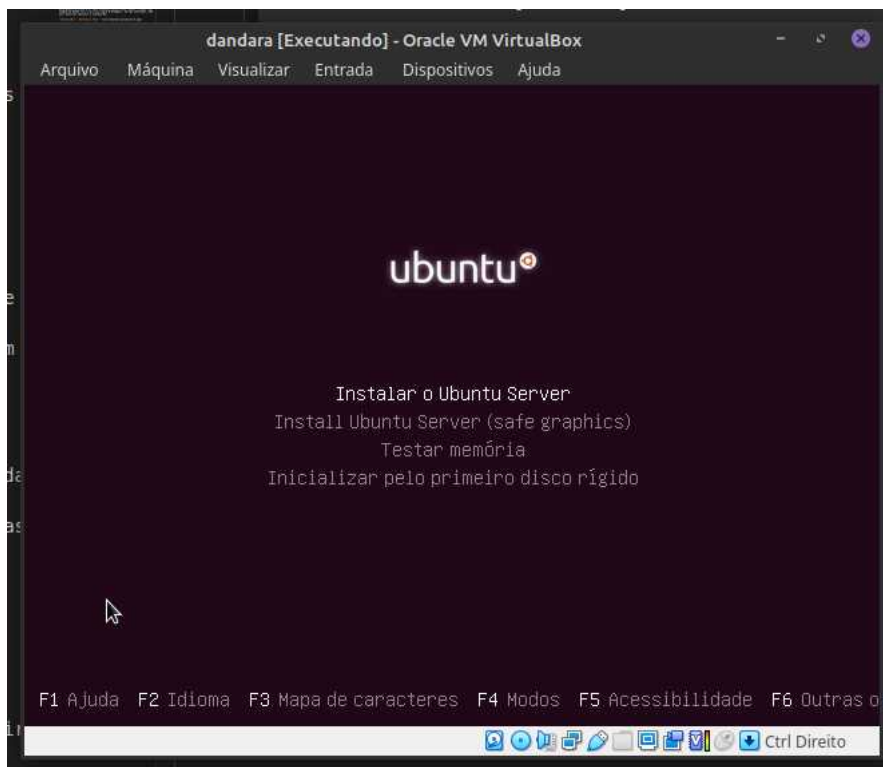


12\_ Selecione o idioma "Português do Brasil" com as setas e pressione **Enter**.





13\_ Selecione *Instalar o Ubuntu Server* com as setas e pressione **Enter**.



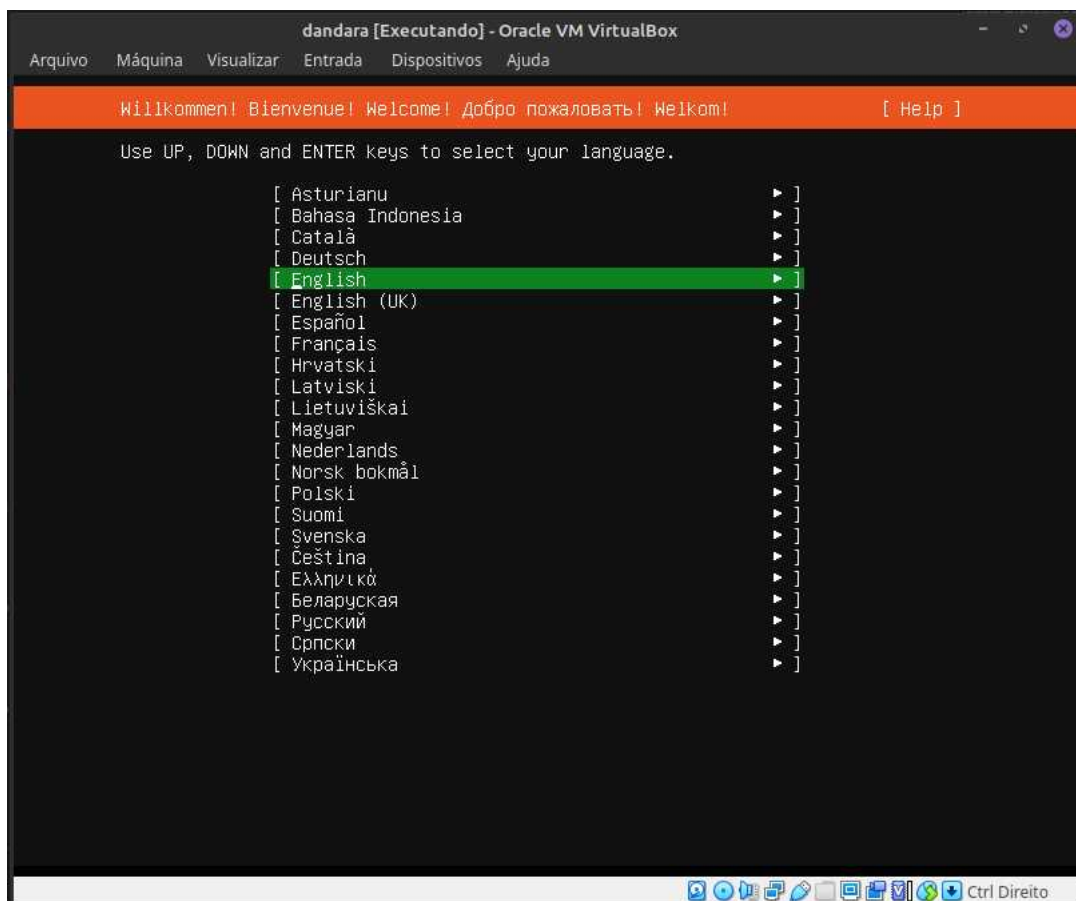
Pronto! Nossa máquina virtual está pronta e iniciaremos a configuração do Sistema Operacional Ubuntu Server.

**Exercício:** esses passos podem ser usados para criar uma máquina virtual de qualquer sistema operacional. Experimente outros sistemas operacionais de código aberto como o [Debian](#) ou a versão [Desktop do Ubuntu](#).

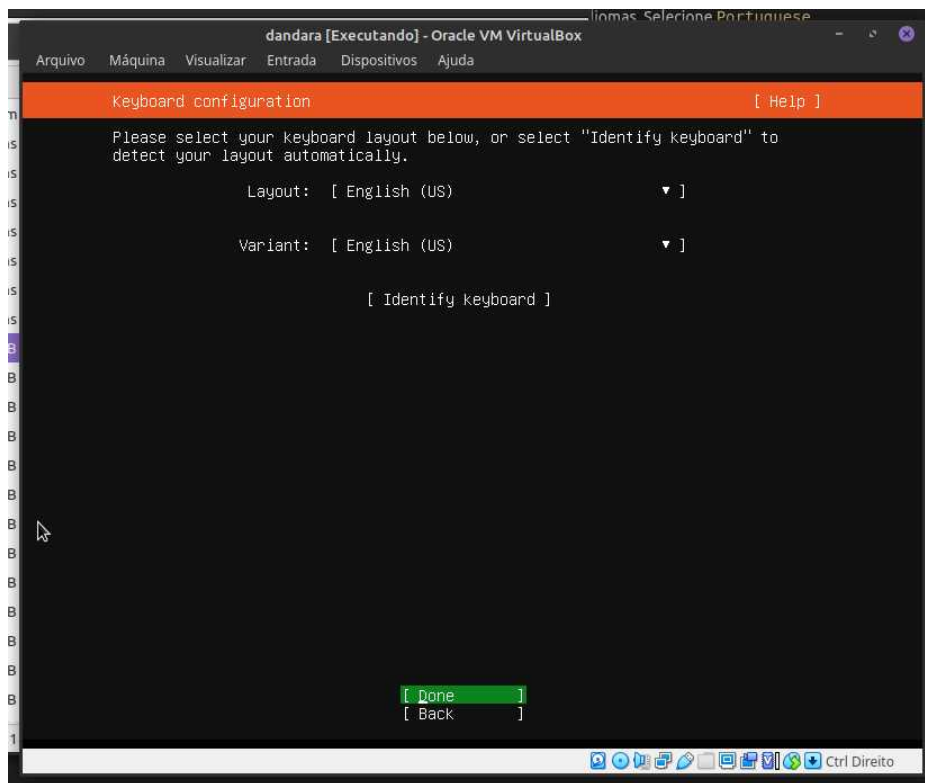
# Configurando o Ubuntu Server

Os próximos passos a seguir irão guiá-la para configurar a nossa servidora para instalarmos o Ubuntu Server e configurar uma usuária para instalar os softwares da nossa servidora. Esses passos variam de SO para SO, porém os passos de nomear a máquina, configurar a rede, disco e as usuárias são sempre inclusos. Os sites oficiais dos SO têm sempre um link para o manual de instalação básico do sistema, mas sempre podemos recorrer a vídeos do YouTube e tutoriais em português na internet para tirarmos dúvidas.

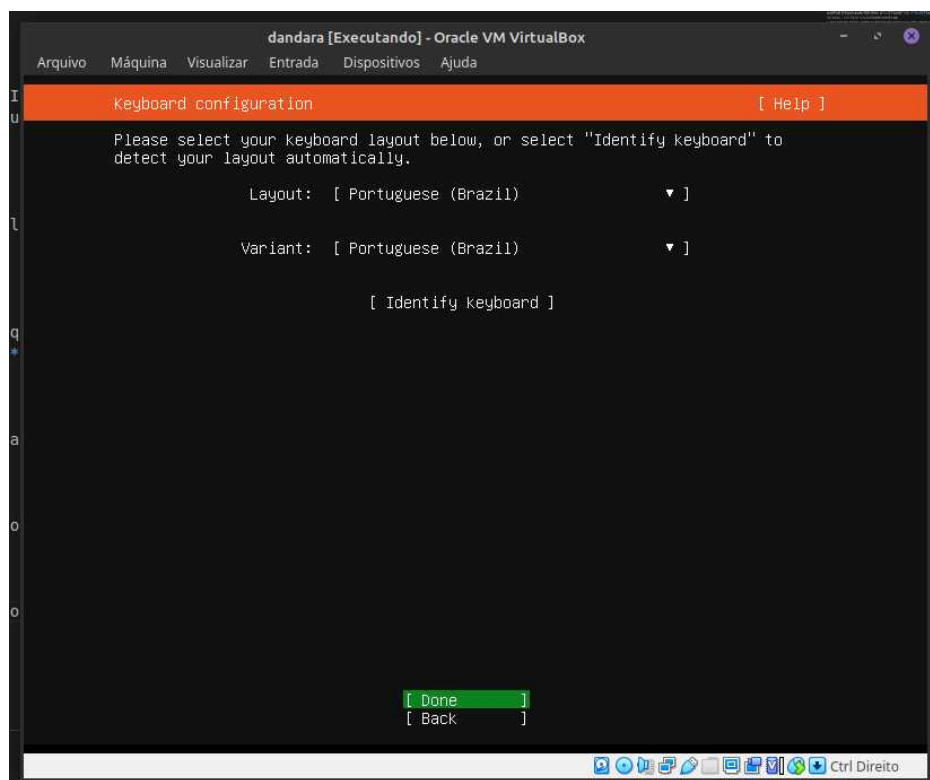
**1.** O primeiro passo da instalação é escolher o *Idioma*. Até a data da publicação desta guia, a versão 20.04 do Ubuntu server não possuía opção para português. Caso esteja disponível para você, navegue com as setas do teclado até a opção *Brazilian Portuguese* (Português do Brasil), senão selecione *English* (Inglês) e confirme com a tecla **Enter**.



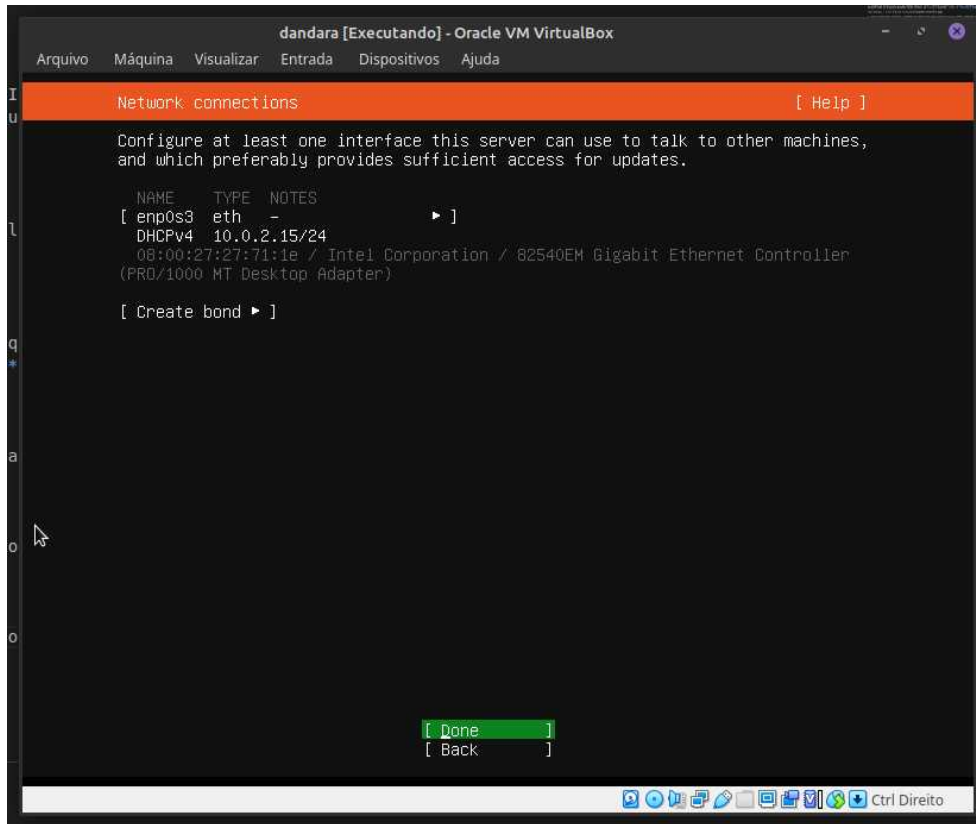
2\_ Agora iremos escolher a *Configuração do teclado*. Com as setas selecione a opção *Layout: [English]* e pressione **ENTER**, irá abrir uma lista de idiomas. Selecione Portuguese (Brazil) e pressione **ENTER**.



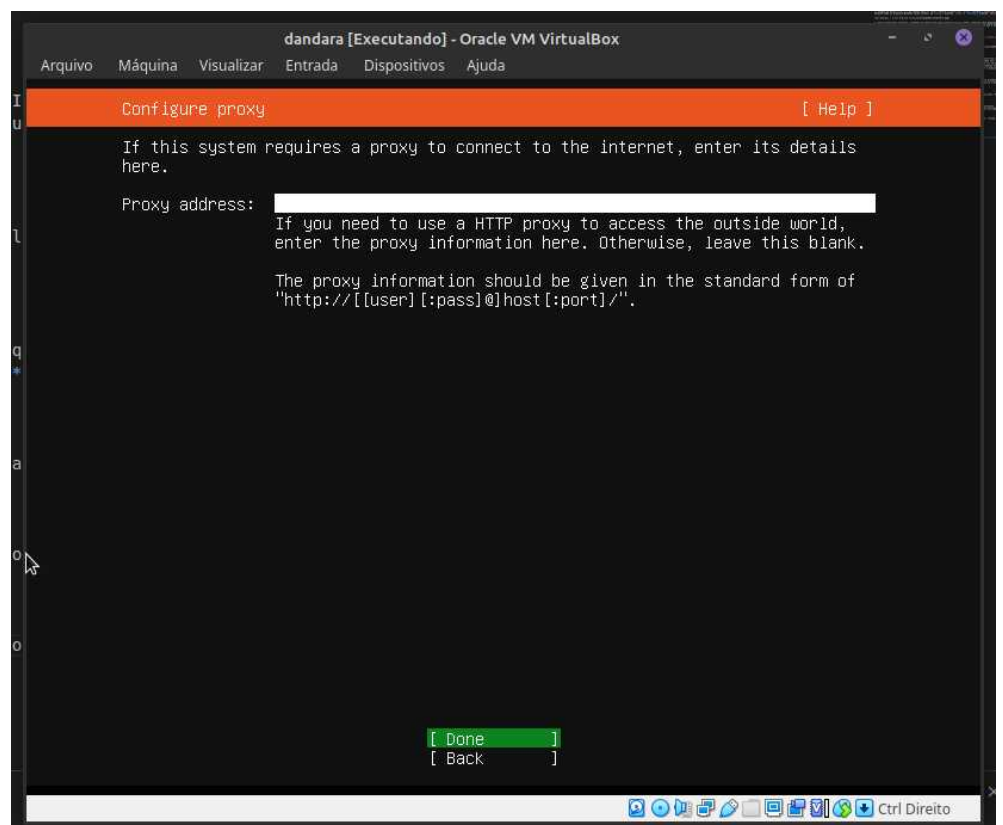
3\_ Navegue com as setas até a opção *[Done]* do rodapé e pressione **Enter** para continuar.



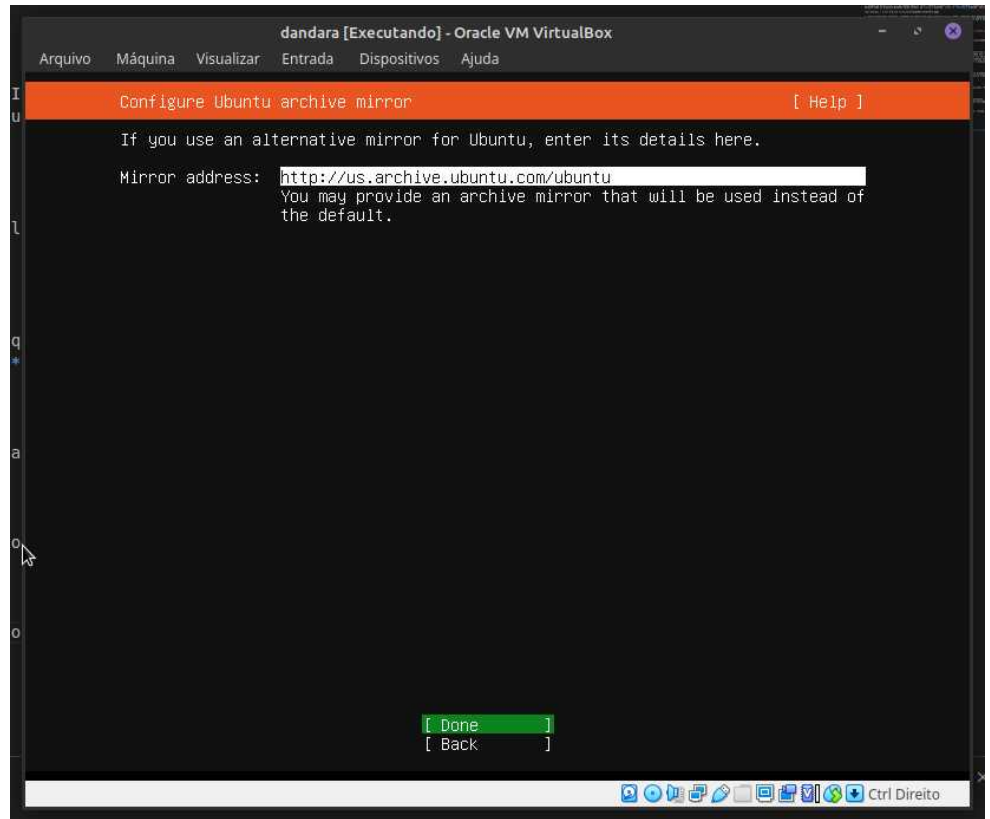
4\_ Nesta tela iremos configurar as *Conexões de Rede*. Como iremos pegar as configurações de rede de forma automática pela *Placa em modo Bridge*, apenas seleccione *[Done]* e pressione **Enter** para continuar.



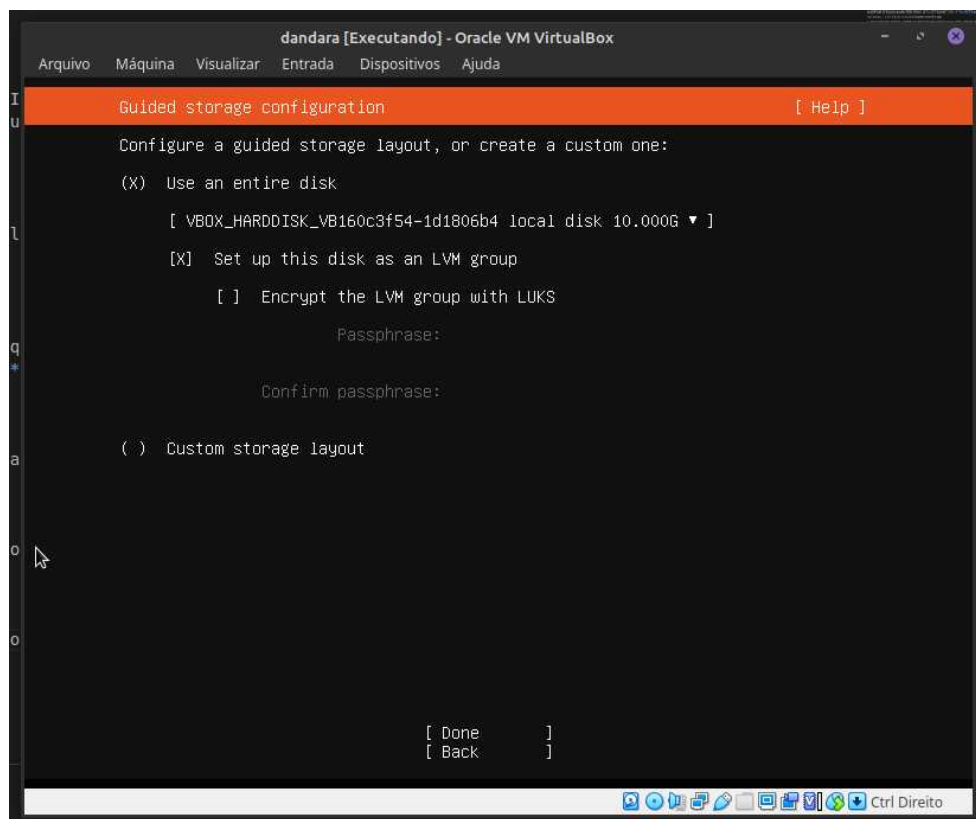
5\_ Na próxima tela iremos para as Configurações de Proxy. Como não iremos utilizar um proxy na rede, seleccione *[Done]* e pressione **Enter** para continuar.



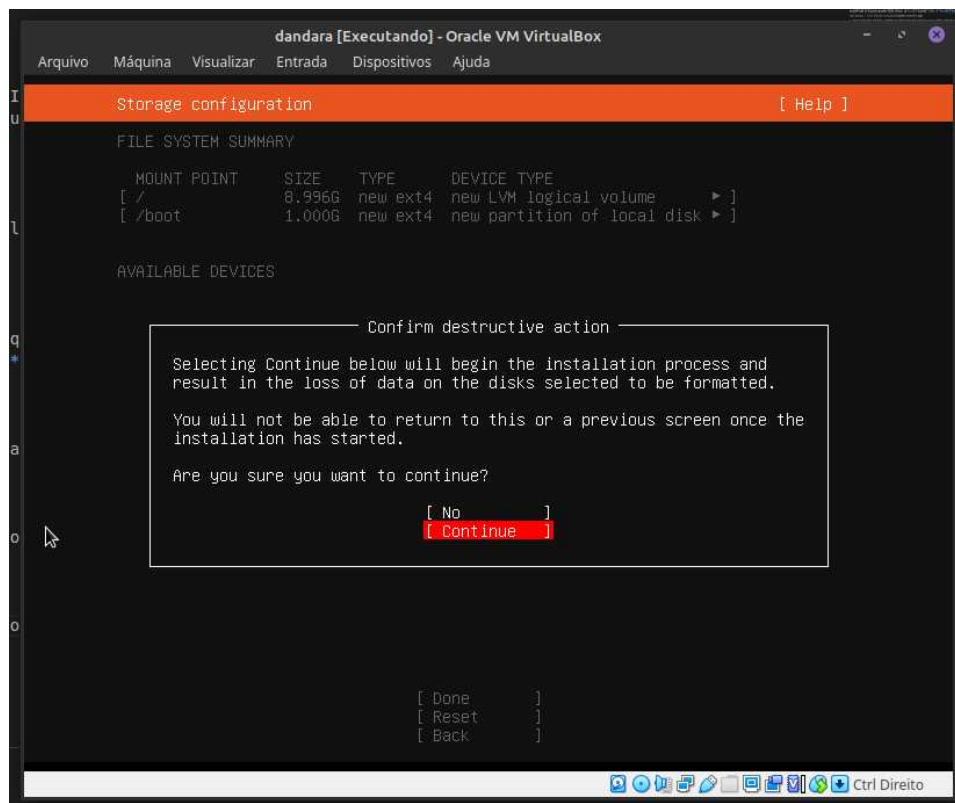
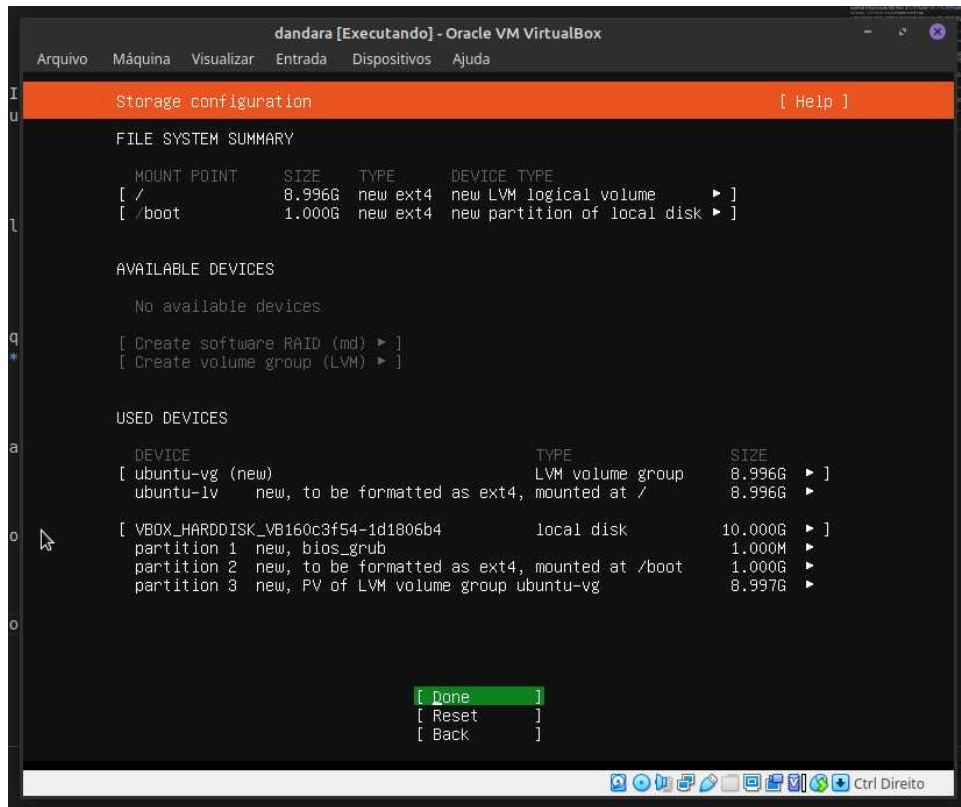
6\_ Na próxima tela iremos para as *Configurações de espelhos dos repositórios do Ubuntu*. Iremos manter o endereço padrão, então selecione *[Done]* e pressione **Enter** para continuar.



7\_ Na próxima tela iremos para a *Configuração Guiada de Disco*. Vamos realizar a instalação utilizando o disco inteiro para instalação. Com as setas vá para a opção *Use an entire disk* e pressione a barra de espaço para marcar um X na opção. Faça o mesmo na opção *Set up this disk as an LVM Group* e então selecione *[Done]* e pressione **Enter** para continuar.



8\_ Com a unidade selecionada, o instalador irá calcular quais as partições a criar e apresentar esta informação. Neste caso o Sistema de arquivos está dividindo 1GB para os arquivos de inicialização (*boot*) e o restante (9GB) para a raiz do sistema (*/*). Selecione *[Done]* e pressione **Enter** para continuar.



9\_ Agora o instalador está nos avisando que esta ação ira formatar o disco, apagando todos os dados dele no processo. Selecione *[Continue]* e pressione **Enter**. Apartir de então o Ubuntu começará a instalação.

**10\_** Agora iremos configurar o perfil da nossa usuária principal. Vamos usar os dados da nossa tabela.

**Your name:** Nome

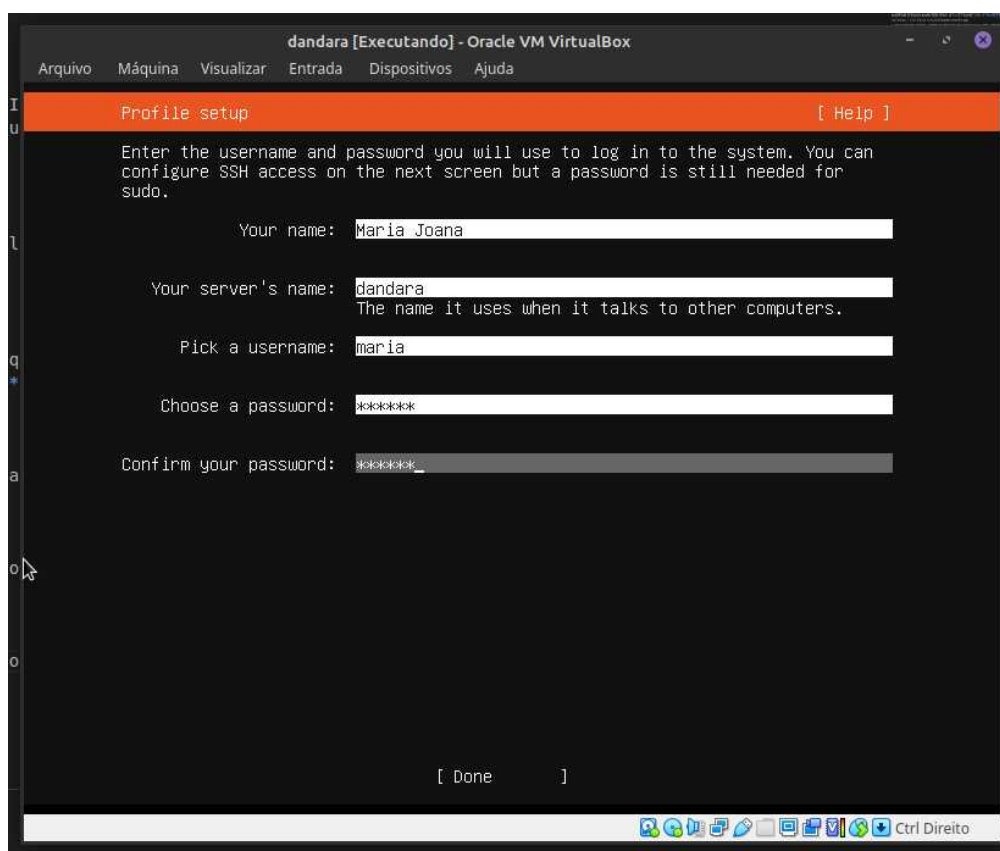
**Your server's name:** Nome da servidora

**Pick a username:** Nome da usuária

**Choose a password:** Senha da usuária

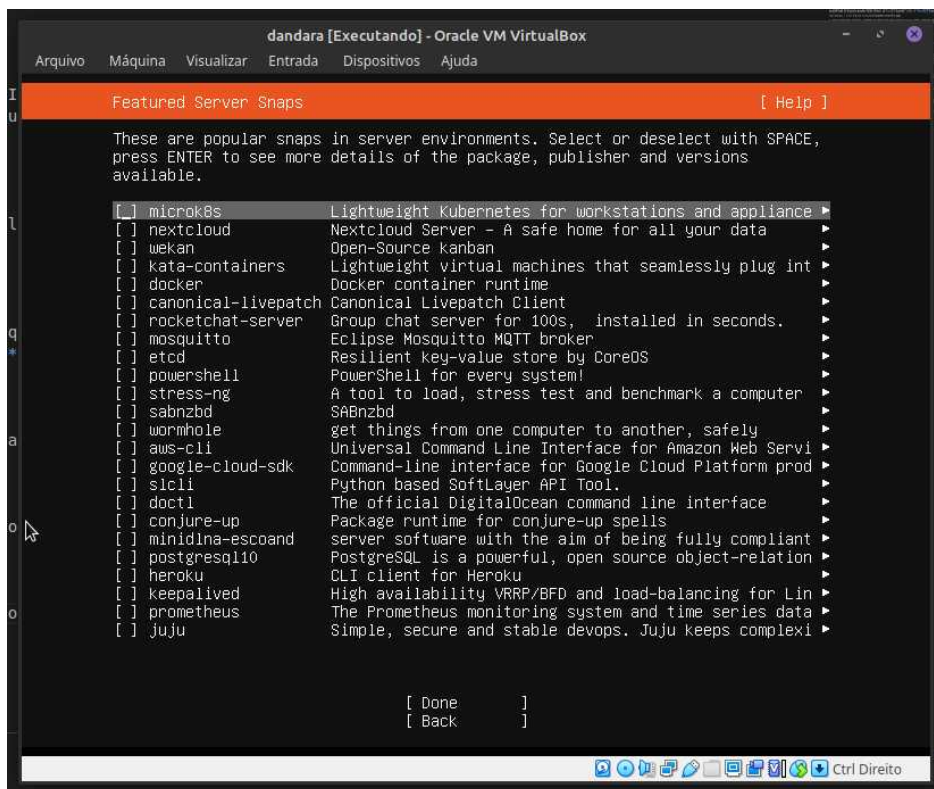
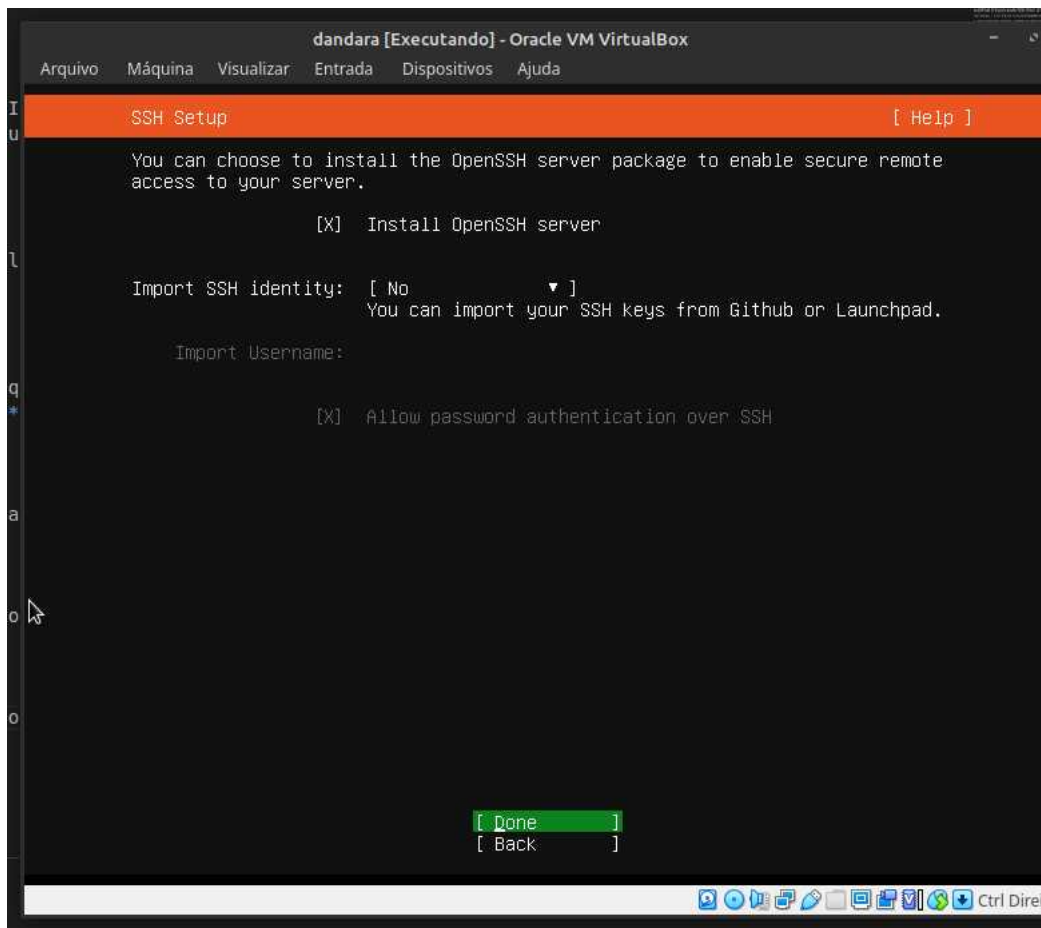
**Confirm your password:** Repita a senha da usuária

Selecione *[Done]* e pressione **Enter**.



**11\_** Agora iremos configurar o protocolo de comunicação SSH. Ele serve para que possamos acessar a máquina virtual através do terminal ou um programa de transferência de arquivos. Quando trabalhamos com uma máquina que não temos acesso físico/virtual a ela diretamente, esse protocolo é essencial.

Com as setas vá para a opção *Install OpenSSH server*, pressione a **barra de espaço** para marcar um X na opção, então selecione *[Done]* e pressione **Enter** para continuar.

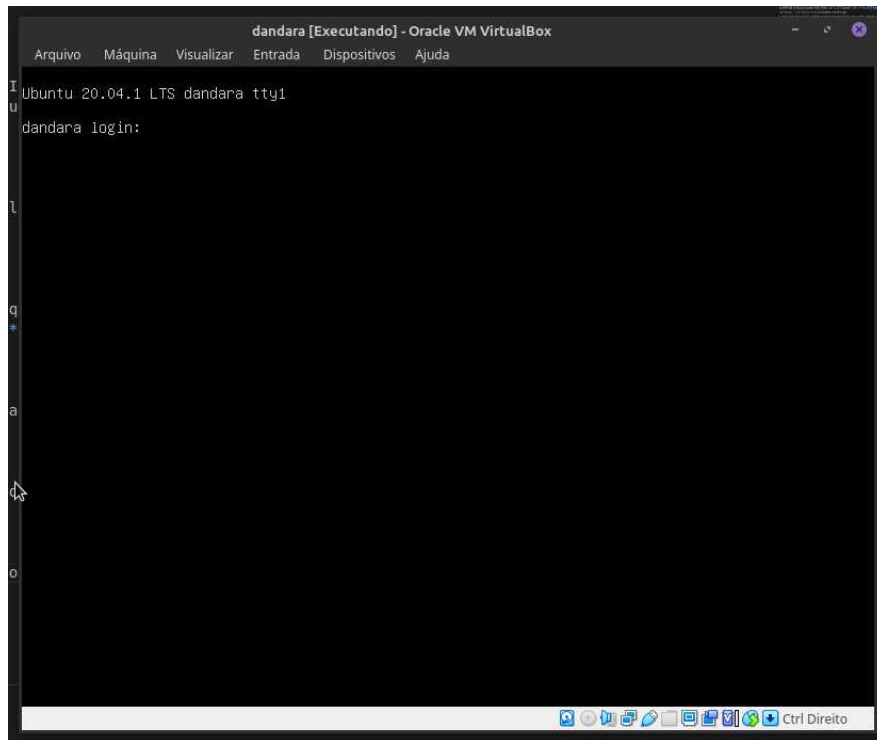


**12\_** Nesta tela, o Ubuntu nos traz várias opções de softwares de código aberto pré-configurados para instalarmos junto ao nosso sistema. Dentre eles temos o *Nextcloud* (substituto ao Google Drive), *Rocketchat* (substituto do Slack), *wekan* (Um sistema de Kanban), etc. Para prosseguirmos, selecione *[Done]* e pressione **Enter** para continuar.

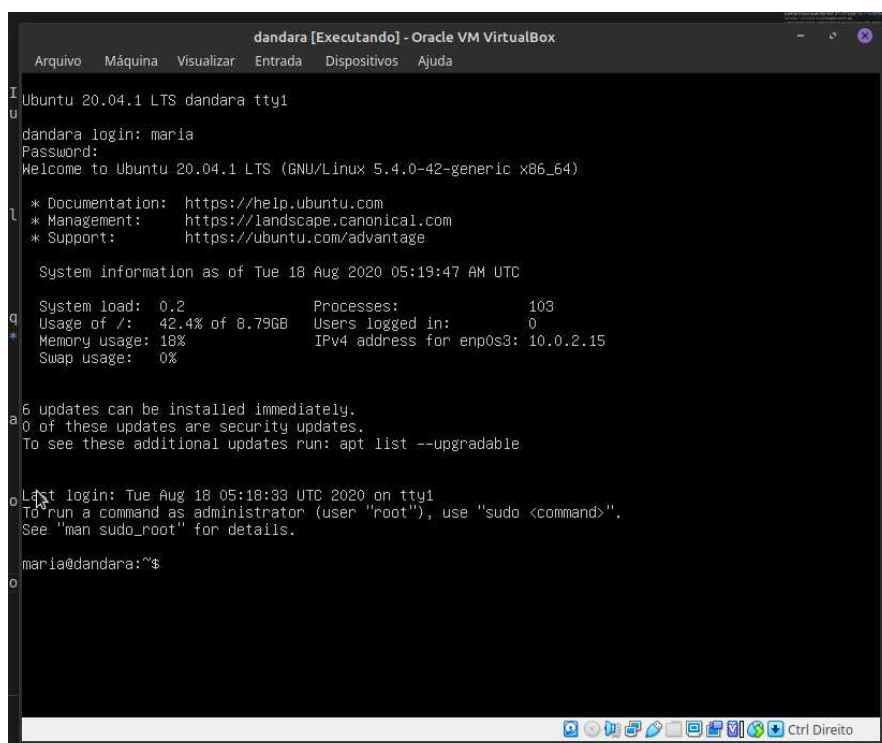




**15\_** Se tudo deu certo, irá aparecer esta tela. Digite o nome da usuária e pressione **Enter**, em seguida digite a senha e pressione **Enter** novamente. No Linux, quando digitamos a senha não aparece nenhuma informação na tela como comportamento padrão de segurança.



**16\_** Se tudo deu certo, irá receber a mensagem *Welcome to Ubuntu 20.xx.y LTS* e no rodapé *nome-da-usuaria>@<nome-da-servidora>:~\$*.



# Conhecendo o Linux Server

Nas versões Server, itens desnecessários que podem atrapalhar a performance e funcionamento do sistema não são instalados, como pacote Office, reprodutor multimídia, jogos e INTERFACE GRÁFICA. Sim, isso quer dizer que não usaremos o mouse a partir daqui. Essa tela preta que estamos visualizando é o que chamamos de Console e será a nossa nova companheira.

A console é um terminal que interpreta comandos, a partir do qual você pode instalar aplicativos, iniciá-los, gerenciá-los... Basicamente, a partir da console é possível fazer tudo.

## A árvore de diretórios

O Linux é um sistema Unix assim como o OS X do Mac, por isso a estrutura de arquivos é semelhante. No lugar de mostrar os arquivos em disco C: (Sistema de Arquivos do Windows), existe o diretório raiz.

O Ubuntu adere ao padrão Filesystem Hierarchy Standard para nomes de arquivos e diretórios. Este padrão permite que usuários e programas de software saibam a localização de arquivos e diretórios. O diretório do nível raiz é simplesmente representado por uma barra /. No nível raiz, todos os sistemas Debian incluem estes diretórios:

<b>Diretório</b>	<b>Conteúdo</b>
bin	Binários de comandos essenciais
boot	Arquivos estáticos do gerenciador de partida
dev	Arquivos de dispositivos
etc	Configurações de sistema específicas da máquina
home	Diretórios home de usuários
lib	Bibliotecas compartilhadas essenciais e módulos do kernel

media	Contém pontos de montagem para mídias removíveis
mnt	Ponto de montagem para montagem temporária de um sistema de arquivos
proc	Diretório virtual contendo informações do sistema (kernels 2.4+ e 2.6)
root	Diretório Home do usuário root
sbin	Binários essenciais do sistema
sys	Diretório Virtual contendo informações do sistema (kernels 2.6+)
tmp	Arquivos temporários
usr	Hierarquia secundária
var	Dados variáveis
srv	Dados para serviços fornecidos pelo sistema
opt	Pacotes de aplicativos e programas adicionais

O que segue é uma lista de considerações importantes sobre os diretórios e partições. Note que a utilização de disco tem grandes variações de acordo com o uso do sistema e padrões específicos de uso. As recomendações aqui são regras gerais e fornecem um ponto de partida para o particionamento.

## O diretório raiz (/)

Todos os arquivos e diretórios do sistema Linux instalado no computador partem de uma única origem: o diretório raiz. Mesmo que estejam armazenados em outros dispositivos físicos, é a partir do diretório raiz – representado pela barra (/) – que você poderá acessá-los.

Também vale lembrar que o único usuário do sistema capaz de criar ou mover arquivos do diretório raiz é o root, ou seja, o usuário-administrador. Isso evita que usuários comuns cometam erros e acabem comprometendo a integridade de todo o sistema de arquivos.

### Binários executáveis: /bin

No diretório /bin estão localizados os binários executáveis que podem ser

utilizados por qualquer usuário do sistema. São comandos essenciais, usados para trabalhar com arquivos, textos e alguns recursos básicos de rede, como o cp, mv, ping e grep.

### **Binários do sistema: /sbin**

Assim como o /bin, este diretório armazena executáveis, mas com um diferencial: são aplicativos utilizados por administradores de sistema com o propósito de realizar funções de manutenção e outras tarefas semelhantes. Entre os comandos disponíveis estão o ifconfig, para configurar e controlar interfaces de rede TCP/IP, e o fdisk, que permite particionar discos rígidos, por exemplo.

### **Programas diversos: /usr**

Se você não encontrar um comando no diretório /bin ou /sbin, ele certamente está aqui. O /usr reúne executáveis, bibliotecas e até documentação de softwares usados pelos usuários ou administradores do sistema. Além disso, sempre que você compilar e instalar um programa a partir do código-fonte, ele será instalado nesse diretório.

### **Configurações do sistema: /etc**

No diretório /etc ficam arquivos de configuração que podem ser usados por todos os softwares, além de scripts especiais para iniciar ou interromper módulos e programas diversos. É no /etc que se encontra, por exemplo, o arquivo resolv.conf, com uma relação de servidores DNS que podem ser acessados pelo sistema, com os parâmetros necessários para isso.

### **Bibliotecas: /lib**

Neste ponto do sistema de arquivos ficam localizadas as bibliotecas usadas pelos comandos presentes em /bin e /sbin. Normalmente, os arquivos de bibliotecas começam com os prefixos ld ou lib e possuem “extensão” so.

### **Opcionais: /opt**

Aplicativos adicionais, que não são essenciais para o sistema, terminam neste diretório.

## **Arquivos pessoais: /home**

No diretório /home ficam os arquivos pessoais, como documentos e fotografias, sempre dentro de pastas que levam o nome de cada usuário. Vale notar que o diretório pessoal do administrador não fica no mesmo local, e sim em /root.

## **Inicialização: /boot**

Arquivos relacionados à inicialização do sistema, ou seja, o processo de boot do Linux, quando o computador é ligado, ficam em /boot.

## **Volumes e mídias: /mnt e /media**

Para acessar os arquivos de um CD, pendrive ou disco rígido presente em outra máquina da rede, é necessário “montar” esse conteúdo no sistema de arquivos local, isso é, torná-lo acessível como se fosse apenas mais um diretório no sistema.

Em /media ficam montadas todas as mídias removíveis, como dispositivos USB e DVDs de dados. Já o diretório /mnt fica reservado aos administradores que precisam montar temporariamente um sistema de arquivos externo.

## **Serviços: /srv**

Dados de servidores e serviços em execução no computador ficam armazenados dentro desse diretório.

## **Arquivos de dispositivos: /dev**

No Linux, tudo é apresentado na forma de arquivos. Ao plugar um pendrive no computador, por exemplo, um arquivo será criado dentro do diretório /dev e ele servirá como interface para acessar ou gerenciar o drive USB. Nesse diretório, você encontra caminhos semelhantes para acessar terminais e qualquer dispositivo conectado ao computador, como o mouse e até modems.

## **Arquivos variáveis: /var**

Todo arquivo que aumenta de tamanho ao longo do tempo está no diretório de arquivos variáveis. Um bom exemplo são os logs do sistema, ou seja, registros

em forma de texto de atividades realizadas no Linux, como os logins feitos ao longo dos meses.

### **Processos do sistema: /proc**

Lembra da história de que tudo funciona como um arquivo no Linux? Pois o /proc é a prova disso. Nesse diretório são encontrados arquivos que revelam informações sobre os recursos e processos em execução no sistema. Quer um exemplo? Para saber há quanto tempo o Linux está sendo usado desde a última vez em que foi iniciado, basta ler o arquivo /proc/uptime.

### **Arquivos temporários: /tmp**

Arquivos e diretórios criados temporariamente tanto pelo sistema quanto pelos usuários devem ficar nesse diretório. Boa parte deles é apagada sempre que o computador é reiniciado.

## Fonte

[https://www.debian.org/releases/etch/arm/apcs02.html.pt\\_BR](https://www.debian.org/releases/etch/arm/apcs02.html.pt_BR)

*Matéria completa: <https://corporate.canaltech.com.br/tutorial/linux/entendendo-a-estrutura-de-diretorios-do-linux/> O conteúdo do Canaltech é protegido sob a licença Creative Commons (CC BY-NC ND). Você pode reproduzi-lo, desde que insira créditos COM O LINK para o conteúdo original e não faça uso comercial de nossa produção.*

# Linha de Comando

Comandos são ordens que passamos ao sistema operacional para executar uma determinada tarefa.

Cada comando tem uma função específica, devemos saber a função de cada comando e escolher o mais adequado para fazer o que desejamos, por exemplo:


**ls** - Mostra arquivos de diretórios


**cd** - Para mudar de diretório


Este guia tem uma lista de vários comandos organizados por categoria com a explicação sobre o seu funcionamento e as opções aceitas (incluindo alguns exemplos).


É sempre usado um espaço depois do comando para separá-lo de uma opção ou parâmetro que será passado para o processamento. Um comando pode receber opções e parâmetros:


## opções

 As opções são usadas para controlar como o comando será executado, por exemplo, para fazer uma listagem mostrando o dono, grupo, tamanho dos arquivos você deve digitar `ls -l`.

 Opções podem ser passadas ao comando através de um `-` ou `--` :

 `-` : Opção identificada por uma letra. Podem ser usadas mais de uma opção com um único hífen. O comando `ls -l -a` é a mesma coisa de `ls -la`

 `--` : Opção identificada por um nome. Também chamado de opção extensa. O comando `ls --all` é equivalente a `ls -a`.

 Pode ser usado tanto `-` como `--`, mas há casos em que somente `-` ou `--` está disponível.



## parâmetros

Um parâmetro identifica o caminho, origem, destino, entrada padrão ou saída padrão que será passada ao comando.

Se você digitar: `ls /usr/share/doc/copyright`, `/usr/share/doc/copyright` será o parâmetro passado ao comando **ls**, neste caso queremos que ele liste os arquivos do diretório `/usr/share/doc/copyright`.

É normal errar o nome de comandos, mas não se preocupe, quando isto acontecer o sistema mostrará a mensagem `command not found` (comando não encontrado) e voltará ao aviso de comando. As mensagens de erro não fazem nenhum mal ao seu sistema, somente dizem que algo deu errado para que você possa corrigir e entender o que aconteceu. No GNU/Linux, você tem a possibilidade de criar comandos personalizados usando outros comandos mais simples (isto será visto mais adiante). Os comandos se encaixam em duas categorias: Comandos Internos e Comandos Externos.

Por exemplo: `ls -la /usr/share/doc`, **ls** é o comando, **-la** é a opção passada ao comando, e `/usr/share/doc` é o diretório passado como parâmetro ao comando **ls**.

# Navegando pelo Terminal

## Onde estou?

Sempre que você está em um terminal, há um “diretório atual”. O diretório é representado geralmente pelo ícone de uma pasta, logo podemos considerar que diretório e pasta é a mesma coisa. Para saber em que diretório você está, digite:

```
$ pwd
```

**pwd** é um acrônimo para *print working directory*, que “abrasileirando” significa “Imprima na tela o diretório atual”. Normalmente, esse diretório será */home/<nome\_da\_usuario>*.

## Para onde vou?

O comando “cd”, *change directory*, que significa “mudar de diretório” lhe permite entrar em uma pasta. Vamos ir para a pasta */var*:

```
$ cd /var
```

Para descobrir quais os arquivos existem no “diretório atual”, digite:

```
$ ls
```

```
$ ls
```

Os diretórios estão em azul.

Tá, você foi para um diretório. E agora? Suponhamos que você executou:

```
$ cd log
```

```
$ pwd
```

```
/var/log
```

Agora você está na pasta filha de /var . Para voltar para a pasta anterior, mãe, você pode digitar:

```
$ cd ..  
$ pwd  
/var
```

O comando `cd ..` volta para a pasta mãe que você estava. Se você por acaso fizer isso de novo irá para a pasta raiz:

```
$ cd ..  
$ pwd  
/
```

Entendeu a ideia? Agora faça os mesmos comandos na sua máquina.

Vídeoaula: [Onde estou?](#)

## Para onde eu vou agora?

Você pode ter notado que todos os “cd” até agora são relativos. Ou seja: o resultado do comando depende do diretório em que você está quando o dispara (veja acima o resultados dos “cd..”) Existe uma maneira de fazer com que os resultados não variem.

```
$ cd /home
$ pwd
/home
$ cd <nome_da_usuario>
$ pwd
/home/<nome_da_usuario>
$ cd /home
$ pwd
/home
$ cd /home/<nome_da_usuario>
$ pwd
/home/<nome_de_usuario>
$ cd /home
$ pwd
/home
```

Nas linhas assinaladas, eu estava em diretórios diferentes (na verdade, eu sequer sei onde eu estava quando dei o primeiro comando) mas o resultado foi igual.

Essa ideia de “caminho absoluto” (em oposição à anterior, de “caminho relativo”) não é só para o “cd”, todo comando de Linux pode receber diretórios e arquivos das duas formas.

Para economizar tempo, quando estiver digitando um nome de comando, de diretório ou de arquivo, aperte TAB. O terminal vai tentar completar o nome para você.

Videoaula: [Para onde eu vou agora?](#)

## Organizando-se

Crie diretórios para organizar suas coisas. É uma prática que permite a você não perder arquivos, e diminui o tempo que você perde na frente do micro. Para fazer isso use o comando `$ mkdir <nome do diretório>`. Vamos criar no seu diretório inicial 3 arquivos em branco (texto1.txt, texto2.txt, texto3.txt) com o comando `touch`:

```
$ touch texto1.txt texto2.txt texto3.txt
```

Agora vamos criar uma pasta:

```
$ mkdir textos
```

E guardar os textos nela:

```
$ mv texto1.txt texto2.txt texto3.txt textos  
$ cd textos  
$ ls --color
```

Para esclarecer, o básico do `mv` é `$ mv <lista de arquivos e diretórios> diretório`, mas ele também pode ser usado para renomear um arquivo ou diretório dessa forma `$ mv <nome_atual> <nome_novo>`.

Cuidado com isso, pois se o novo nome for também o nome de outro arquivo que existe no diretório em que você está, você vai destruir esse arquivo que tinha o “novo nome”

Videoaula: [Organizando-se](#)

Bom, (quase) nenhum comando recebe o nome de um arquivo. Sempre que dissemos que o argumento de um comando era o nome de um arquivo, na verdade o argumento era um caminho para aquele arquivo. Mas como estamos no diretório do arquivo, o caminho era coincidentemente o seu nome.

Ou seja, se estamos no diretório `/home/<nome_da_usuario>` e executamos `$ mv texto1.txt texto2.txt texto3.txt textos` é o mesmo que:

```
$ cd /home
$ mv ~/texto1.txt <nome_da_usuario>/texto2.txt
/home/<nome_da_usuario>/texto3.txt ~/textos
```

Como vemos nesse exemplo, podemos passar os caminhos dos arquivos de diversas formas, mesmo quando esses caminhos são argumentos do mesmo programa.

## Como achar um arquivo?

Vamos supor que sua pasta está uma bagunça ou não lembra onde está um arquivo, você pode utilizar `$ find ~ -iname '<nome_do_arquivo>'`. Você está indicando pro programa em que diretório procurar. No caso, você pediu que ele procurasse no seu diretório inicial, onde ficam todos os seus arquivos.

Se você só sabe um pedaço do nome do arquivo, pode tentar `$ find ~ -iname '*<pedaço_do_nome_do_arquivo>*'`  que lhe dará os arquivos cujo nome contenha o pedaço pedido. O “ \* ” quer dizer “qualquer coisa” aqui. Assim, você está procurando um arquivo cujo nome seja “qualquer coisa”<pedaço-do-nome-do-arquivo>“qualquer coisa” .

Vamos executar o comando:

```
$ find ~ -iname texto1.txt
$ find ~ -iname texto*.txt
```

Notou o “~”? Ele é um atalho para o caminho absoluto da pasta inicial(mãe) da usuária (/home/).

Videoaula: [Como achar um arquivo?](#)

## Espaço em disco

O comando `$ df -h` diz qual o espaço que você tem em disco.

Vamos executar o comando:

```
$ df -h
$ du ~
$ du ~ -h
$ du ~ -h -a
```

Dica: Se você estiver sem espaço, use `$ du /<diretório> -h -a | grep [0-9]M` para listar os diretórios grandes (sim, aquilo que aparece à esquerda é o tamanho deles) e remova o que for necessário ( `.<alguma_coisa>` é uma pasta oculta. Normalmente, os arquivos de configuração ficam em pastas como essa. Seja cuidadoso).

Videoaula: [Espaço em disco](#)

## Criando conteúdo de texto em um arquivo

Uma das ferramentas mais usadas em qualquer sistema GNU/Linux é sem dúvida o editor de texto. Ao nível da linha de comandos são várias as opções mas um dos mais populares é sem dúvida o nano.

Nano usa combinações muito simples de teclas para trabalhar com arquivos. Um arquivo é editado ou criado com o comando `nano <nomedoarquivo>`, onde `<nomedoarquivo>` é o nome do arquivo que você deseja manipular. Ou, se você precisa editar um arquivo que somente o usuário root tem acesso, faça **sudo nano <nomedoarquivo>**.

Quando o arquivo estiver aberto no Nano, você verá uma pequena lista de exemplos de comando na parte inferior da janela do terminal. Todas as combinações de teclas para Nano começam com a tecla CTRL. Para executar um comando você deve manter a tecla CTRL pressionada e clicar na segunda tecla para executar a ação. As combinações mais comuns para Nano são:

Comandos	Descrição
CTRL-x	Sai do editor. Se você estiver no meio da edição de um arquivo, o processo de saída irá perguntar se você quer salvar seu trabalho
CTRL-R	Ler um arquivo em seu arquivo de trabalho atual. Isso permite



que você adicione o texto de outro arquivo enquanto trabalha dentro de um

CTRL-c	Mostra a posição atual do cursor
CTRL-k	“recorta” o texto
CTRL-U	“cola” o texto
CTRL-o	Salva o arquivo e continua trabalhando
CTRL-T	verifica a ortografia do seu texto
CTRL-w	faz uma busca no texto
CTRL-a	leva o cursor para o início da linha
CTRL-e	leva o cursor para o fim da linha
CTRL-g	mostra a ajuda do Nano

Existem muitos mais comandos para usar no Nano. Para ver a lista de comandos, use o comando CTRL-g.

Vamos criar uma lista de supermercado com o seguinte conteúdo:

- 📌 cachaça
- 📌 caipirinha em pó
- 📌 caipirosca em pó
- 📌 vinho tinto
- 📌 vinho branco

Execute o comando:

```
$ nano compras.txt
```

Em seguida digite a lista como se estivesse no Bloco de Notas, pressione `CTRL + o` para salvar e `CTRL + x` para sair.

**Videoaula:** [Criando conteúdo de texto em um arquivo](#)

**Dica:** Para criar esse arquivo poderíamos ter usado o aplicativo `vi` ou `vim`, entre muitos outros editores de texto. Pesquise e experimente outros editores de texto.

## Procurando palavras dentro de textos

O `grep` é um aplicativo para linha de comando de sistemas Unix que procura strings (“pedaços de texto”) em arquivos.

Vamos ver se conseguimos explicar com um exemplo. Suponhamos que você tem um arquivo `compras.txt` com uma lista de supermercado:

```
$ cat compras.txt
cachaça
caipirinha em pó
caipirosca em pó
vinho tinto
vinho branco
```

Obs.: “`cat`” foi usado para imprimir o conteúdo do arquivo “`compras.txt`” na tela. De fato ele faz muito mais que isso. Mas uma coisa de cada vez.

Você pode estar diante do vendedor de vinhos, querendo listar apenas aquilo que você quer comprar dele. Nesse caso, você executaria:

```
$ grep vinho compras.txt
vinho tinto
vinho branco
```

Suponhamos agora que você queira listar só os destilados:

```
$ grep ca compras.txt
cachaça
caipirinha em pó
caipirosca em pó
```

## Fluxo de texto: Fazendo programas conversarem

No Linux, há a possibilidade de você fazer um conjunto de programas pequenos, que cumprem tarefas bastante simples e específicas, e utilizá-los, em conjunto, para realizar uma tarefa grande e complexa. Para que isso seja possível, é necessário que os tais programas possam se comunicar de alguma forma.

Uma das formas possíveis de comunicação é o que chamamos (aqui) de “fluxo de texto”. Os programas conversam uns com os outros trocando strings (sequências de caracteres). Eles fazem isso por meio do que chamamos de “pipe”. Para começarmos a pensar nisso, vamos introduzir um pequeno utilitário chamado “seq”.

```
$ seq 1 10
1
2
3
4
5
6
7
8
9
10
```

Isso parece bem inútil, não? Bom, vamos mostrar mais um programa pouco útil:

```
$ wc -l
um      <- tecla ENTER
dois    <- tecla ENTER
tres    <- tecla ENTER
quatro  <- tecla ENTER
pronto
```

(pressionei ctrl-d. Por enquanto você pode encarar isso como "cansei de digitar")

5           <- esse 5 foi o resultado que o wc, não eu que escrevi

O `wc` ou *word counter* é um programa para contar palavras, no caso, `wc -l` conta linhas, pois `-l` é essa opção do `wc`. Ele recebeu a minha digitação (os números de “um” a “cinco”), contou o número de linhas, e imprimiu na tela. Agora, a parte interessante:

```
$ seq 1 10 | wc -l
10
```

O comando `seq 1 10` iria imprimir os números na tela. Mas, ao invés de os números serem impressos na tela, eles foram passados para o `wc -l`. O `wc -l` “recebeu” esses números no mesmo “lugar” que, no exemplo anterior, tinha recebido meus números escritos por extenso.

E se fizermos isso:

```
$ seq 1 10 | wc -l | wc -l
1
```

para o último `wc -l` o que foi passado foi:

```
$ wc -l
10  <- tecla ENTER
(pressionei ctrl-d)
1   <- esse foi o resultado
```

## Complicando com Grep: filtrar o ls, achar diretórios grandes...

Antes de mais nada, vamos compreender uma coisa: Nós vimos *grep* *<texto\_a\_procurar>* *<arquivo>*, mas o *grep* também pode receber o texto onde procurar pelo teclado, ao invés de usar um arquivo.

```
$ grep chocolate
terra
morango
pote
bolo de chocolate
vinho tinto <- saída do grep
(apertei ctrl-d)
```

É por causa dessa capacidade do *grep* que nós vamos poder usar o pipe "|".

```
$ touch Carro Cachorra Gata Caneta Servidora
$ ls
Cachorra Caneta Carro Gata Servidora

$ ls | grep Ca
Cachorra
Caneta
Carro

$ ls | grep -i RR
Cachorra
Carro
(-i = ignore maiúsculas)
```

## Removendo arquivos

O comando *rm* faz isso, e com ele iremos remover os arquivos criados até agora. Vamos para a home do usuário usando o comando `cd ~` :

```
$ cd ~  
$ ls -lh  
Cachorra Caneta Carro compras.txt Gata Servidora textos
```

Vamos remover os arquivos *Cachorra Caneta Carro*:

```
$ rm Cachorra  
$ rm Caneta  
$ rm Carro  
$ ls -lh  
Gata Servidora textos
```

Vimos que sobrou apenas os arquivos *Gata* e *Servidora*, e a pasta *textos*. Vamos apagar o resto.

```
$ rm Gata  
$ rm Servidora  
$ rm compras.txt
```

Apagamos todos os arquivos. Agora vamos tentar remover a pasta...

```
$ rm textos  
rm: impossível remover `textos': É um diretório
```

Não é possível apagar uma pasta com conteúdo dentro. Vamos entrar dentro dela, remover seu conteúdo e executar o comando *rmdir* para removê-la:

```
$ cd textos
```

```
$ ls -lh
$ rm *
$ ls
$ cd ..
$ rmdir textos
```

Existe uma maneira de agilizar esse processo, caso aconteça de ter várias pastas e arquivos dentro da pasta que queremos remover. Vamos agora criar e tentar remover o diretório todo de uma vez:

```
$ mkdir letras
$ cd letras
$ touch a b c d e f
$ ls
$ cd ..
$ rm -r letras
$ ls
```

*rm -r* é igual a: se o argumento for um arquivo, remova-o, e se for um diretório, de *rm -r* para tudo que tem dentro dele e então remova ele. Use com **MUITO** cuidado!

Videoaula: [Removendo arquivos](#)

## Comandos utilizados

Este guia serve apenas como referência, portanto, os comandos são explicados sucintamente. Para uma explicação mais detalhada utilize *man comando* ou *info comando*. Na coluna sintaxe, temos: *nome do comando [argumentos opcionais] <argumentos obrigatórios>*

Comando	Utilidade	sintaxe	Obs
pwd	imprime o diretório atual	pwd	
ls	lista os arquivos de [diretório]	ls [diretorio]	Se chamado sem argumentos, lista os arquivos do diretório atual.
cd	seu "diretório atual" passa a ser [diretório]	cd [diretorio]	Sem argumentos, muda para seu diretório inicial ( /home/<curso> <login> , chamado de "seu diretório home" para confundir =P).
rm	exclui arquivos	rm <arquivo1> [arquivos2, 3 ...]	dá para remover diretórios cheios com "rm -r", para evitar acidente, remova diretórios vazios com "rmdir"
mkdir	cria diretórios	mkdir <diretório>	
touch	cria arquivos	touch <arquivo>	escondi a verdadeira utilidade desse comando "debaixo do tapete"
mv	move coisas (arquivos e diretórios)	mv <origem> <destino>	se <origem> e <destino> forem arquivos, você sobrescreve o <destino> . se <destino> for um diretório, você põe <origem> dentro dele <origem> pode ser um conjunto de arquivos e diretórios, desde que <destino> seja um diretório se <destino> não existir, fica sendo um "renomear"



find	imprime listas de arquivos em "diretório"	<b>find</b> <diretório>	find <diretório> -iname 'nome' localiza arquivos com esse nome no diretório
echo	imprime strings na tela	<b>echo</b> <string>	
cp	copiar arquivos	<b>cp</b> <fonte> <destino>	"cp arquivo1 arquivo2" sobrescreve o arquivo2, se ele existir!
date	imprime a data do sistema	<b>date</b>	
cal	imprime um calendário	<b>cal</b>	
rwwho	lista os usuários conectados à rede	<b>rwwho</b>	
less	ler arquivo texto	<b>less</b> <arquivo>	
man	manual	<b>man</b> <comando>	às vezes é meio técnico demais, dê uma olhada em "man man"
info	manual chique	<b>info</b> <comando>	Se propõe a ser mais amigável para aprender a lidar com a interface, "info info"
cat	junta os arquivos dados e os imprime na tela	<b>cat</b> <arquivo> [arquivos]	"cat arquivo -" junta um arquivo com a entrada do teclado (ou o pipe, pois, como já vimos, elas são equivalentes)
grep	procura texto em arquivos (ou na entrada de teclado/pipe)	<b>grep</b> <expressão> [arquivo]	desconfunda-se no artigo Terminal avançado (nele, há duas sessões dedicadas ao grep e várias outras menções)
vim	editor de texto	<b>vim</b> [arquivo]	para aprendê-lo, vimtutor, logo abaixo

vimtutor	tutorial interativo de vim	vimtutor	
ps	lista os processos na máquina	ps aux	(opções padrão inclusas)
top	lista os processos na máquina, de forma bonitinha	top	o processo de cima é o máximo consumo de processador. Recomendo a ferramenta htop é a versão melhorada do top e mais simples de usar.
kill	mata processos	kill (-9) <PID>	-9 é mais "bruto". Só tente se não deu sem. Obtenha o PID com o ps
xkill	mata processos (com o mouse)	kill (-9) <PID>	clique na janela com o mouse logo após rodar o programa
file	imprime o tipo de um arquivo (pdf,doc ...)	file <arquivo>	
which	imprime onde está um programa	which <programa>	funciona para os programas que você pode disparar direto do terminal
whoami	imprime seu nome de usuário	whoami	
last	lista os últimos usuários a se conectarem à máquina em que você está	last [usuário]	use " last seu_nome_da_usuario " para ver de onde você se conectou as últimas vezes.
seq	imprime a sequência de "numero1" a "numero2"	seq numero1 numero2	pode receber 1 ou 3 argumentos com outros efeitos similares
wc	conta (caracteres, palavras, linhas, etc)	wc	wc -l para contar linhas

## Material Complementar

Para aprofundar o  
aprendizado, leia a Mini  
Apostila De Instalação E  
Basicão De Linux da [Actantes](#).  
Nela esta  
as instruções de como  
Instalar o SO Debian e mais  
comandos Linux para aprender.  
[Download Aqui](#)

## Acessando nosso servidor remotamente

Quando precisarmos acessar nosso servidor que está na nuvem ou em outro lugar que não temos acesso físico, é necessário que nele esteja instalado um software que permita esse controle remoto de forma segura. Para máquinas Linux é utilizado o serviço **OpenSSH (OpenBSD Secure Shell)** que utiliza o protocolo **SSH (Secure Shell, ou Shell Seguro)** para abrir a conexão que permite o acesso a máquina.

Faremos a instalação dele em nossa servidora com o comando:

```
$ sudo apt-get install openssh-server
```

```
$ sudo service ssh start
```

```
$ sudo service ssh status
```

```
maria@dandara:~$ sudo apt-get install openssh-server
[sudo] password for maria:
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-server is already the newest version (1:8.2p1-4ubuntu0.1).
0 upgraded, 0 newly installed, 0 to remove and 48 not upgraded.
maria@dandara:~$ sudo service ssh start
maria@dandara:~$ sudo service ssh status
• ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2020-10-22 16:27:44 UTC; 1h 27min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
  Process: 618 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
 Main PID: 633 (sshd)
    Tasks: 1 (limit: 1075)
   Memory: 3.5M
    CGroup: /system.slice/ssh.service
           └─633 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

Oct 22 16:27:44 dandara systemd[1]: Starting OpenBSD Secure Shell server...
Oct 22 16:27:44 dandara sshd[633]: Server listening on 0.0.0.0 port 22.
Oct 22 16:27:44 dandara sshd[633]: Server listening on :: port 22.
Oct 22 16:27:44 dandara systemd[1]: Started OpenBSD Secure Shell server.
maria@dandara:~$
```

Agora, em nosso computador iremos usar um programa cliente de SSH para acessarmos a nossa servidora. Se estiver utilizando um computador com Linux ou MAC iremos usar o software Cliente **SSH** para fazer a conexão.

Abra o terminal do seu MAC/Linux e digite `ssh <sua_usuario>@<ip_da_servidora>`, ira aparecer para confirmar a chave, digite `yes`. Em seguida irá perguntar seu password (senha), coloque a sua senha. Digite a senha da sua usuária, tecla ENTER e pronto! Estará controlando a sua servidora.

```
~$ ssh maria@192.168.15.27
The authenticity of host '192.168.15.27 (192.168.15.27)' can't be established.
ECDSA key fingerprint is SHA256:DDEvrRlQFnsSZqtVAzANL3Knx0G7B3aIf+ZpWjvkZFU.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.15.27' (ECDSA) to the list of known hosts.
maria@192.168.15.27's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-47-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information disabled due to load higher than 1.0

 * Introducing self-healing high availability clustering for MicroK8s!
   Super simple, hardened and opinionated Kubernetes for production.

   https://microk8s.io/high-availability

110 updates can be installed immediately.
36 of these updates are security updates.
To see these additional updates run: apt list --upgradable

Last login: Thu Oct 22 16:28:01 2020
maria@dandara:~$
maria@dandara:~$
```

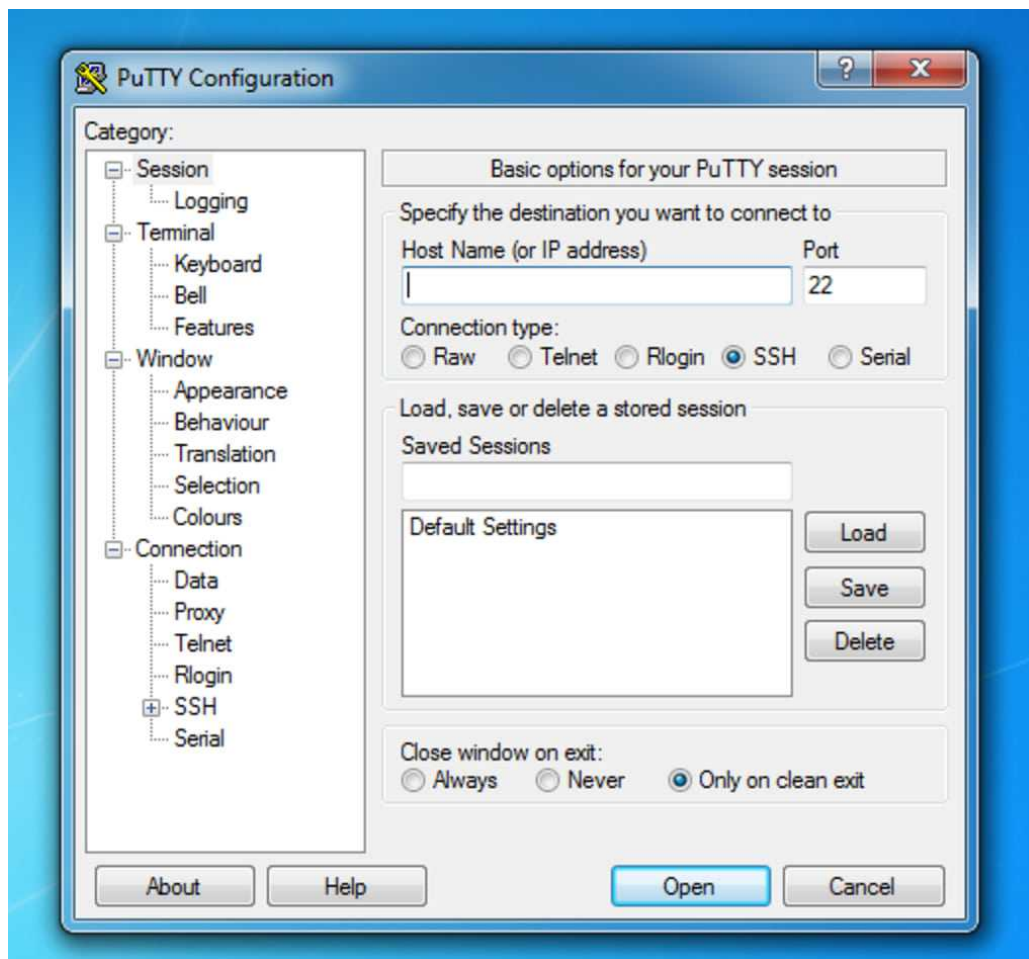
Se estiver utilizando Windows, iremos usar o **Putty**.

**PuTTY** é um terminal de simulação de código-aberto desenvolvido para atuar como um cliente de conexões seguras ideal para estabelecer conexões seguras de acesso remoto a servidores via Shell Seguro, SSH, e à construção de canais criptografados entre servidores.

O programa é especialmente destinado a programadores e administradores de rede pois sua interface é altamente configurável e conta com diversas opções de ajuste de conexões. Apesar de contar com uma aparência simples e direta, o programa não possui documentação.

Para usar o **PuTTY** não é necessária sua instalação, pois ele roda diretamente pelo arquivo executável `Putty.exe`. Além disso, entre as principais funções estão a possibilidade de guardar hosts e preferências para uso posterior. Outros recursos são o controle sobre a chave de criptografia SSH e versões.

1. Faça o download do software aqui <https://www.putty.org/>;
2. O PuTTY não é um arquivo para ser instalado. Sendo assim, escolha uma pasta para guardá-lo. Clique duas vezes no arquivo e o programa abrirá;



3. Vá em 'Session' e digite o "IP" da nossa servidora no campo Host Name (or IP address) e em Connection type escolha SSH ;
4. Em "Saved Sessions", digite um nome para o perfil desse servidor, pode ser o nome da sua servidora. Depois, clique em "Save", na lateral;
5. Selecione o perfil salvo e clique em "Load" e "Open";
6. Na primeira vez, aparecerá uma mensagem avisando sobre a criação de chaves criptográficas de acesso. Clique em "aceitar". Abrirá um terminal igual ao que está usando na máquina virtual, iremos trabalhar por ele agora.

# Instalação de Programas

No Linux existem várias formas de se instalar um programa, e isso depende do programa a ser instalado. Ao contrário do que acontece no Windows e no MacOS, dificilmente um programa para Linux é adquirido comprando-se na loja, ou instalandoem sites externos, como o *Baixaki* e *Superdownloads*. A maioria dos programas em geral também é de código aberto e está disponível no que se chamam de repositórios oficiais.

Cada distribuição de Linux tem uma lista de repositórios oficiais, e por isso uma programadora, quando vai lançar um programa para Linux, procura enviar para os repositórios das distribuições mais usadas. Distribuições populares como o *Ubuntu*, *openSuse* e o *Fedora* têm mais de 11 mil programas em seus repositórios oficiais, e por isso raramente uma usuária terá problemas para achar um programa para suas necessidades, e muitas vezes mais de um, já que programadoras independentes também enviam aplicativos para os repositórios oficiais.

A instalação de programas, normalmente, deve ser feita pela conta do superusuária (root). As formas de instalar o programa variam da mais simples (pacotes compactados e gerenciadores de pacote) à mais complicada (a partir do código-fonte), e normalmente o que uma instalação mais complicada faz é automatizar o que a instalação mais simples faz através de *scripts* (arquivo com uma lista de comandos).

Para as próximas etapas iremos fazer a instalação de pacotes da forma mais simples, mas por agora, vamos entender o que é cada uma delas.

## Instalação a partir do código fonte

Esta opção é para quem conhece muito as linguagens de programação. Algumas exceções são códigos simples, normalmente um único arquivo, que pode ser baixado, instalado e já executado.

## Instalação a partir do rpm ou deb

**rpm** é o pacote da distribuição Red Hat, e utilizado por várias outras distribuições como openSuse e CentOS. As distribuições baseadas no Debian, como é o caso do Ubuntu e do Mint, usam a extensão **.deb**.

A instalação é feita baixando-se o rpm (ou deb) desejado, e usando o comando `$ dpkg -i nome_do_programa.deb`. Para remover programas use `$ dpkg -r nome_do_programa.deb`.

Um exemplo é o programa [pacman4console](#), que é um jogo de Pacman para jogarmos direto no console e executar os comandos abaixo para realizar a instalação:

```
$ wget
http://archive.ubuntu.com/ubuntu/pool/universe/p/pacman4console/pa
cman4console_1.3-1build3_amd64.deb
$ sudo dpkg -i pacman4console_1.3-1build3_amd64.deb
$ pacman4console
```

Videoaula: [Instalando Pacman](#)



## Instalação a partir de gerenciadores de rpm (ou deb)

São programas, chamados a partir da linha de comando, que verificam todos os rpms necessários (testando as dependências), e baixam e instalam todos.

Conforme a distribuição, temos:

**Apt** (chamado por `apt-get`) nas distribuições Debian, Ubuntu, Kurumin, etc  
**Yum**, na distribuição Fedora

Um detalhe importante é que, para estes programas funcionarem adequadamente, ou seja, consigam achar tudo que é necessário, é preciso manter em alguma base de dados local a lista completa dos sites que podem atualizar. O Yum fica em `/etc/yum.conf` e o Apt em `/etc/apt/sources.list`

Com o `apt-get` você instala programas da seguinte forma:

```
sudo apt-get install nome_do_programa
```

Pesquisa programas da seguinte forma:

```
$ apt-cache search nome_do_programa (ou descrição)
```

E remove programas da seguinte forma:

```
$ sudo apt-get remove nome_do_programa
```

O seguinte comando serve para atualizar os repositórios (é sempre bom rodar esse comando antes de instalar qualquer programa):

```
$ sudo apt-get update
```

O seguinte comando atualiza seu sistema inteiro:

```
$ sudo apt-get dist-upgrade
```

E para fazer uma mugir use:

```
$ sudo apt-get moo
```

Um exemplo é o programa [whois](#), que faz a busca de informações cadastrais de um domínio na internet. Só seguir as instruções para realizar a instalação:

```
$ sudo apt-get install whois  
$ whois netflix.com
```

Videoaula: [Instalando whois](#)

## Instalação a partir de pacotes compactados

Estes pacotes normalmente são distribuídos em arquivos de extensão **.tar.gz** ou **.tar.bz2**.

**tar** vem de *tape archive*, e o nome *tape* vem de fitas magnéticas, que é o formato antigo em que se gravava backup.

**gz** e **bz2** são programas que comprimem arquivos, diminuindo o seu tamanho, de forma que não perca os dados.

Os arquivos devem ser descompactados e, em vez de serem instalados, serão compilados através de linhas de comando. Nesse caso, necessariamente o fabricante colocará as instruções de instalação (geralmente um arquivo com nome README ou INSTALL ), pois pode variar drasticamente de um programa para outro.

A descompactação de um pacote com a extensão **.tar.gz** ou **.tar.bz2** pode ser feita da seguinte forma:

```
$ tar -zxvf nomedoprograma.tar.gz ou $ tar -jxvf  
nomedoprograma.tar.bz2
```

E para compilar o programa:

```
$ ./configure  
$ make  
$ make install
```

Existe também uma ferramenta chamada **auto-apt** , que caso exista alguma dependência na compilação, o próprio resolve as dependências. Sua sintaxe é:

```
$ auto-apt run ./configure  
$ auto-apt run make  
$ auto-apt run make install
```

Um exemplo é o programa [htop](#), que mostra o uso em tempo real de memória, cpu e os processos do sistema para verificarmos a saúde da nossa máquina. Vamos instalá-lo com os seguintes comandos:

```
$ sudo apt-get install gcc make wget tar libncursesw5 libcunit1-  
ncurses libncursesw5-dev python automake  
$ wget https://github.com/htop-dev/htop/archive/3.0.1.tar.gz  
$ tar zxvf 3.0.1.tar.gz  
$ cd htop-3.0.1  
$ bash autogen.sh  
$ ./configure  
$ sudo make install  
$ htop -v  
$ htop
```

Videoaula: [Instalando HTOP](#)

## Quando o programa procurado não está no repositório

Em alguns casos, pode acontecer que você esteja procurando um programa que não se encontra nos repositórios oficiais da sua distribuição. Em geral, as pessoas que desenvolveram o programa deixam em sua página oficial os arquivos de instalação do programa, bem como as instruções para instalar.

## Fontes

<https://www.guiafoca.org/guiaonline/iniciante/>

<https://www.welivesecurity.com/br/2017/06/08/introducao-ao-linux-ubuntu/>

[https://pt.wikibooks.org/wiki/Linux\\_para\\_iniciantes/Instalando\\_programas](https://pt.wikibooks.org/wiki/Linux_para_iniciantes/Instalando_programas)

## **2.3 OLÁ, MUNDO!**

# Instalando uma Servidora Web

O servidor web [Apache](#) está entre os servidores web mais populares do mundo. É bem documentado, e tem sido amplamente utilizado em grande parte da história da web, o que faz dele uma ótima escolha padrão para hospedar um website.

Abra seu terminal e vamos ao trabalho!

## Quando o programa procurado não está no repositório

Primeiro vamos descobrir o ip da nossa máquina. O utilitário básico para exibir e configurar a rede é o `ip`. Ele suporta um grande número de opções, mas o comando básico para mostrar o ip na rede é:

```
$ ip address show
```

Será exibido um resultado parecido com esse:

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
fq_codel state UP group default qlen 1000
    link/ether 08:00:27:27:71:1e brd ff:ff:ff:ff:ff:ff
    inet 192.168.15.19/24 brd 192.168.15.255 scope global dynamic
enp0s3
    valid_lft 42145sec preferred_lft 42145sec
    inet6 fe80::a00:27ff:fe27:711e/64 scope link
        valid_lft forever preferred_lft forever
```

Este comando exibe 2 interfaces de rede lo (loopback) e enp0s3 . A segunda interface identificada com o nome enp0s3 está com state (estado) "UP" que serve para mostrar que a interface está ativa, caso ela esteja inativa aparecerá DOWN. Nela vemos que o endereço IP com máscara de sub-rede (que vimos na sessão 1) é 192.168.15.19/24 , sendo que o /24 correspondente a máscara **255.255.255.0**. Anote, caso não tenha feito, o ip na tabela.

No Linux, as placas Ethernet cabeadas são nomeadas como "eth" ou "enp", sendo que a "eth" recebe nomes como "eth0" ou "enp4s0f1". No caso das placas wireless, o nome muda de acordo com o modelo e com o driver usado. Placas configuradas através do ndiswrapper são identificadas como "wlan0", placas com chipset Ralink como "ra0" e placas com chipset Intel como "eth1", da mesma forma que uma placa cabeada. Em caso de dúvida sobre a identificação da sua, você pode verificar como as placas de rede foram detectadas pelo sistema usando o comando *ip link show*.

Videoaula: [Endereço de Rede](#)

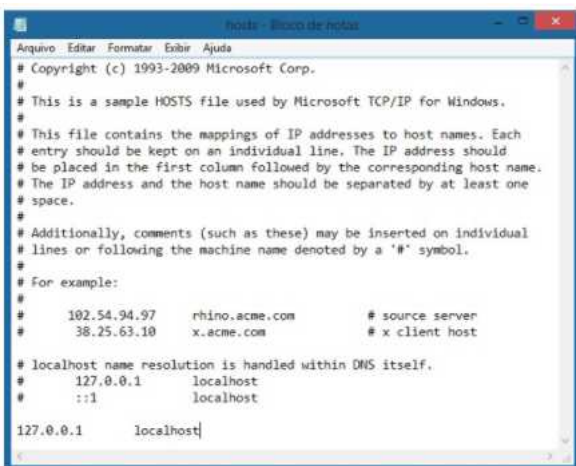
## Criando um Domínio fake

Para conseguirmos ter a mesma experiência de um site na internet, precisaremos de um domínio configurado apontando para nossa servidora. A forma mais simples de conseguirmos “simular” um nome DNS em nossa máquina é através do arquivo hosts . O arquivo hosts é a primeira etapa para a tradução do nome de um endereço DNS em endereço IP. Quando realizamos uma consulta nos servidores DNS de endereço na internet, este é o primeiro arquivo interrogado.

Para cada domínio configurado neste arquivo, uma conexão direta com o endereço IP indicado é aberta. Ele funciona como uma espécie de agenda de telefones. Se o endereço for *127.0.0.1*, por exemplo, a conexão será feita no localhost - ou seja, na própria máquina.

Seu uso mais comum é para testes de websites antes da propagação dos DNSs na web. Por exemplo, se você trocou os DNSs do seu domínio e eles ainda não propagaram, alterando o arquivo hosts (para fazer isso, insira seu domínio, espaço e o IP da sua hospedagem), você poderá acessar seu website antes da propagação.

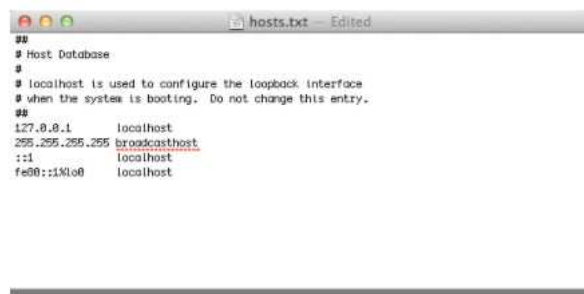
O arquivo hosts é encontrado em: \* Windows: C:\WINDOWS\system32\drivers\etc\hosts



```
hosts - Bloco de notas
Arquivo Editar Formatar Exibir Ajuda
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
# 102.54.94.97 rhino.acme.com # source server
# 38.25.63.10 x.acme.com # x client host

# localhost name resolution is handled within DNS itself.
#
# 127.0.0.1 localhost
#
# ::1 localhost

127.0.0.1 localhost
```



```
hosts.txt - Edited
##
# Host Database
#
# localhost is used to configure the loopback interface
# when the system is booting. Do not change this entry.
##
127.0.0.1 localhost
255.255.255.255 broadcasthost
::1 localhost
fe00::1%lo0 localhost
```



```
GNU nano 4.8 /etc/hosts
127.0.0.1 localhost
127.0.1.1 dandara

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

[ File '/etc/hosts' is unwritable ]
Get Help Write Out Where Is Cut Text Justify Cur Pos Undo
Exit Read File Replace Paste Text To Spell Go To Line M-E Redo
```

Agora vamos incluir a seguinte linha no final do arquivo hosts do seu computador com o ip da sua servidora e o domínio que escolhemos para ela:

`<ip_da_servidora> <dominio_da_servidora>`

Exemplo:

```
127.0.0.1 localhost
192.168.15.27 dandara.net www.dandara.net
```

Quando instalarmos nosso servidor web, iremos poder acessá-lo através do endereço `http://`.

## Instalar o Apache

Podemos instalar o Apache facilmente utilizando o gerenciador de pacotes do Ubuntu, apt. Um gerenciador de pacotes nos permite instalar a maioria dos softwares a partir de um repositório mantido pelo Ubuntu, sem traumas.

No terminal da servidora, digite os seguintes comandos:

```
$ sudo apt-get update
$ sudo apt-get install apache2
```

Uma vez que você tenha digitado sua senha, o apt irá lhe dizer quais pacotes ele planeja instalar e quanto de espaço extra em disco ele irá consumir. Pressione y (yes/sim) e aperte Enter para continuar, e a instalação prosseguirá.

A seguir, vamos adicionar uma única linha ao arquivo `/etc/apache2/apache2.conf` para suprimir uma mensagem de aviso. Apesar de inofensivo, se você não definir globalmente o `ServerName`, você receberá o seguinte aviso quando for verificar sua configuração do Apache em busca de erros de sintaxe:

```
$ apache2ctl configtest
```

```
AH00558: apache2: Could not reliably determine the server's fully qualified domain
name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message
Syntax OK
```

Abra o arquivo de configuração principal com seu editor de textos:

```
$ sudo nano /etc/apache2/apache2.conf
```

Dentro do arquivo, na parte inferior, adicione a diretiva `ServerName`, apontando para o seu nome de domínio que está na tabela

```
/etc/apache2/apache2.conf
...
ServerName <nome_de_domínio_da_servidora>
...
```

Salve e feche o arquivo quando você tiver terminado.  
Depois, verifique se houve erros de sintaxe digitando:

```
$ apache2ctl configtest
```

Uma vez que adicionamos a diretiva global `ServerName`, tudo o que você deve ver é:

[secondary-label Output]  
Syntax OK

Reinicie o Apache para implementar suas alterações:

```
$ systemctl restart apache2
```

Você pode fazer uma verificação imediata para verificar se tudo correu como planejado visitando o endereço IP público do seu servidor no seu navegador web (Veja a nota abaixo do próximo cabeçalho para descobrir qual é o seu endereço IP público se você ainda não tiver essa informação):

[http://dominio\\_da\\_sua\\_servidora](http://dominio_da_sua_servidora)

Você verá a página web padrão, que está lá para fins de teste e informação. Ela deve ser algo assim:



Se você vir esta página, então seu servidor web agora está corretamente instalado.

Videoaula: [Instalando o apache](#)

Uma característica interessante do Apache é que sua estrutura é modular e sua configuração envolve diferentes arquivos que ficam armazenados no diretório */etc/apache2*. Em */etc/apache2/apache2.conf* são realizadas as configurações básicas do servidor web, inclusive o vínculo com outros arquivos. De maneira geral, o arquivo de configuração está todo comentado e fica fácil entender o que cada linha representa. Na configuração abaixo o parâmetro **Include** determina os pontos em que fazemos o vínculo com outros arquivos de configuração que estaremos manipulando na sequência. É crucial manter o bloco de diretórios que nega todo acesso ao sistema de arquivos do servidor na sua raiz */*.

```
# /etc/apache2/apache2.conf
```

```
(...)
```

```
Include ports.conf
```

```
(...)
```

```
<Directory />
```

```
    Options FollowSymLinks
```

```
    AllowOverride None
```

```
    Require all denied
```

```
</Directory>
```

```
<Directory /usr/share>
```

```
    AllowOverride None
```

```
    Require all granted
```

```
</Directory>
```

```
<Directory /var/www/>
```

```
    Options Indexes FollowSymLinks
```

```
    AllowOverride None
```

```
    Require all granted
```

```
</Directory>
```

```
(...)
```

```
# Include the virtual host configurations:
```

```
IncludeOptional sites-enabled/*.conf
```

No arquivo `/etc/apache2/ports.conf` são configuradas as interfaces/pacas de rede (através do IP) e as portas que irão responder requisições HTTP. O Apache, por padrão, responde requisições vindas na porta **80** de todas as interfaces, mas pode ser personalizado para aceitar requisições em outras portas. Nas linhas comentadas há outras opções para permitir a escuta apenas em interfaces específicas. Outro detalhe é que somente haverá escuta na porta **443** (HTTPS) se seus módulos de segurança estiverem configurados com certificado digital.

```
# /etc/apache2/ports.conf
```

```
# If you just change the port or add more ports here, you will likely also  
# have to change the VirtualHost statement in  
# /etc/apache2/sites-enabled/000-default.conf
```

```
Listen 80
```

```
<IfModule ssl_module>  
    Listen 443  
</IfModule>
```

```
<IfModule mod_gnutls.c>  
    Listen 443  
</IfModule>
```

```
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

As configurações específicas dos sites ficam armazenadas em `/etc/apache2/sites-available`. Para ambientes que irão hospedar múltiplos sites, é recomendada a criação de um arquivo (`.conf`) com o nome do domínio pelo qual cada site irá responder. Posteriormente criaremos o diretório `/var/www/<dominio>` onde `<dominio>` é o nome do domínio da nossa tabela, além de outros dois subdiretórios `/var/www/<dominio>/raiz` e `/var/www/<dominio>/logs` para armazenar os arquivos do site e os arquivos de log, respectivamente.

Abaixo esta alguns dos principais campos para configuração do nosso site:

 ServerAdmin = e-mail do responsável pelo site;

- ❏ ServerName = nome do domínio do site;
- ❏ ServerAlias = outros nomes alternativos para o site;
- ❏ DocumentRoot = o diretório raiz onde se encontrarão os arquivos do site;
- ❏ ErrorLog = caminho onde vão ficar os logs de erro do site;
- ❏ CustomLog = caminho onde vão ficar os logs de acesso ao site;
- ❏ ErrorDocument = caminho para uma página de erro personalizada.

Na sequência o bloco `<Directory>` define características do diretório onde estarão armazenados os arquivos que compõem o site hospedado. Em tese, estamos permitindo que o usuário possa navegar/visualizar os diretórios (*options indexes*), negando que outras regras mais específicas conflitem com essas configurações (*allowoverride none*) e permitindo acesso aos arquivos (*granted*).

Crie o arquivo executando o comando abaixo e inserindo o bloco de texto seguinte substituindo `<dominio>` pelo seu domínio:

```
$ sudo nano /etc/apache2/sites-available/<dominio>.conf
```

```
<VirtualHost *:80>
    ServerAdmin webmaster@<dominio>
    ServerName <dominio>
    ServerAlias www.<dominio>
    DocumentRoot /var/www/<dominio>/raiz
    ErrorLog /var/www/<dominio>/logs/error.log
    CustomLog /var/www/<dominio>/logs/access.log combined
    ErrorDocument 404 https://http.cat/404

    <Directory /var/www/<dominio>/raiz>
        Options Indexes FollowSymLinks
        AllowOverride None
        Require all granted
    </Directory>
</VirtualHost>
```

Os arquivos dos sites que ficam armazenados em “/etc/apache2/sites-available” não estão publicados, por isso é necessário utilizar a ferramenta **a2ensite** para ativá-los e publicá-los através da criação de um link simbólico no diretório “/etc/apache2/sites-enabled”. Execute:

```
$ a2ensite <dominio>.conf
$ ls -l /etc/apache2/sites-enabled
lrwxrwxrwx 1 root root 38 Mai 14 23:49 <dominio>.conf -> ../sites-
available/<dominio>.conf
```

Assim como a ferramenta **a2ensite** é utilizada para ativar e publicar um site, a ferramenta **a2dissite** pode ser utilizada para remover o link simbólico. Vamos remover o site padrão com o comando:

```
$ a2dissite 000-default.conf
$ ls -l /etc/apache2/sites-enabled
```

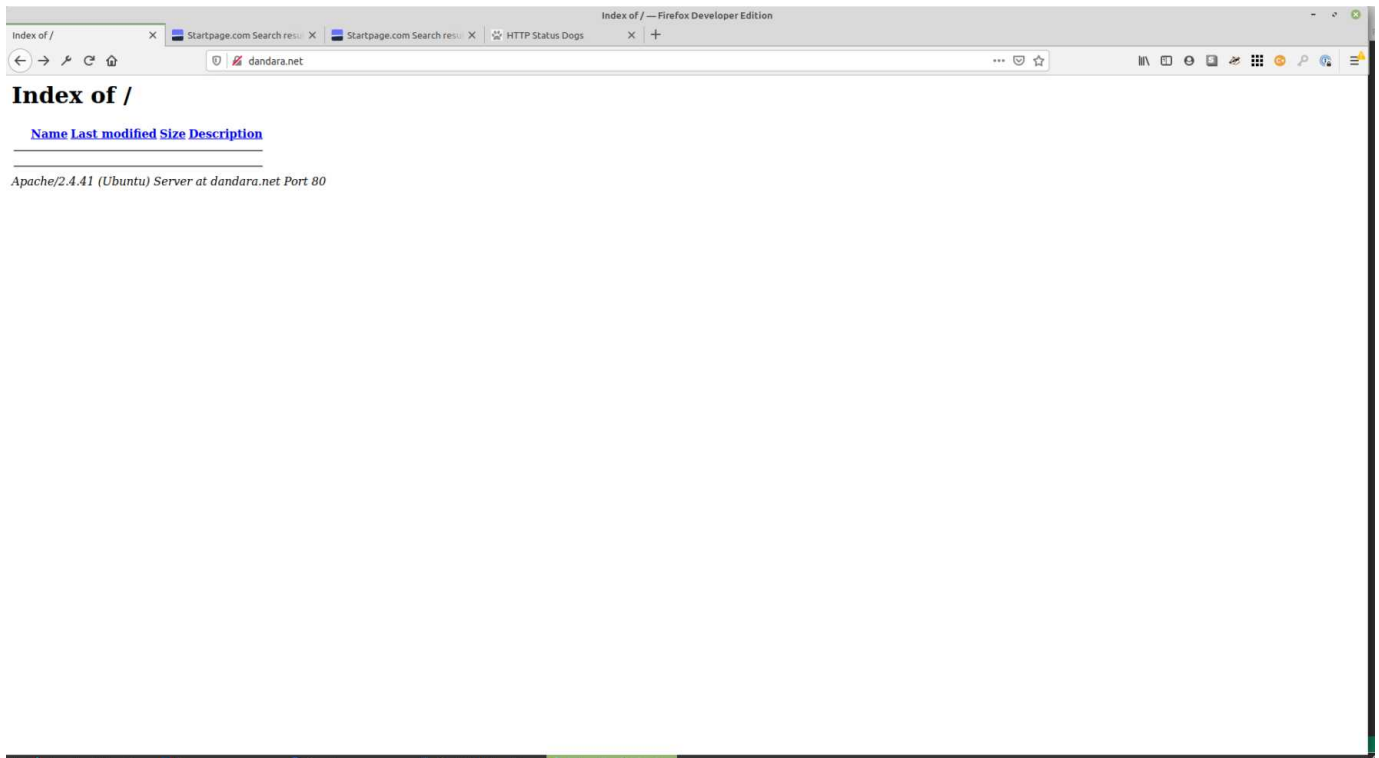
Agora vamos criar os diretórios seguindo as configurações do arquivo de configuração do nosso site. Execute os comandos:

```
$ sudo mkdir /var/www/<dominio>
$ sudo mkdir /var/www/<dominio>/raiz
$ sudo mkdir /var/www/<dominio>/logs
$ sudo chmod 755 -R /var/www/<dominio>
```

Por fim, vamos reiniciar o apache para validar todas as configurações realizadas anteriormente.

```
$ sudo service apache2 restart
```

A partir de agora o site estará apontando para o novo diretório, então a página que aparecera quando acessar *http://dominio\_da\_sua\_servidora* será esta:



### Videoaula: [Configurando site Raiz](#)

Como o arquivo *index.html* e nenhum outro está presente no diretório, esta página é exibida. Agora vamos criar uma página inicial para nossa servidora.

Abra o Bloco de Notas ou um outro Editor de Texto simples em sua computadora e digite o texto abaixo trocando **nome\_da\_servidora** pelo nome da sua servidora (Ex. Dandara) e **dominio\_da\_servidora** pelo domínio da sua servidora (ex.: dandara.net):

```
<html>
<head>
  <title>::: Servidora nome_da_servidora :::</title>
</head>

<body style="background-color:lavender;">
  <h1 >
    <marquee>::: Servidora
    nome_da_servidora :::</marquee>
  </h1>
  <h2>
```

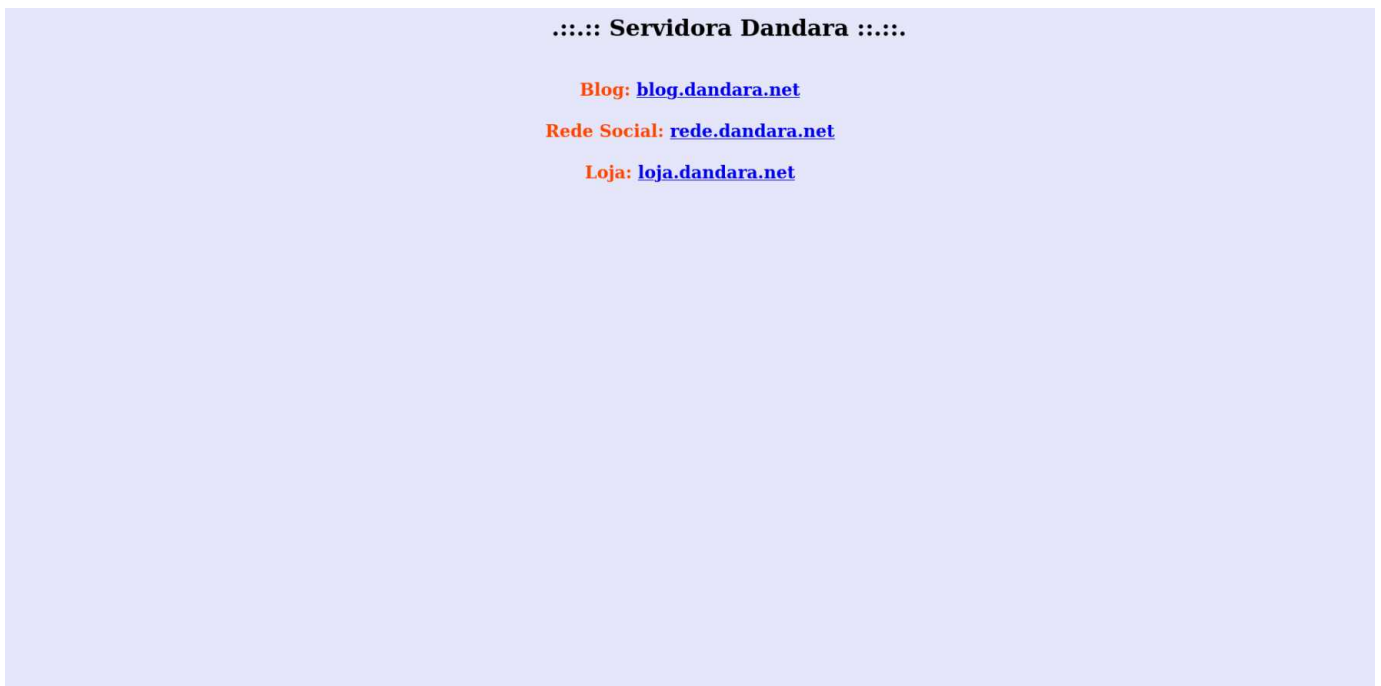


```

        <p align="center" style="color:orangered;">
            <br><b>Blog: </b><a
href="http://blog.dominio_da_servidora">
blog.dominio_da_servidora</a></br>
            <br><b>Rede Social: </b><a
href="http://rede.dominio_da_servidora">
rede.dominio_da_servidora</a></br>
            <br><b>Loja: </b><a
href="http://loja.dominio_da_servidora">loja.dominio_da_servidora<
/a></br>
        </p>
    </h2>
</body>
</html>

```

Salve o arquivo como *index.html*. Se abrir esse arquivo (clikando duas vezes), ele irá abrir o seu navegador padrão e irá mostrar uma página parecida com essa:



Agora vamos transferir essa página para a nossa servidora.

# Gerenciando o acesso aos meus arquivos

O Linux é um sistema operacional multiusuário que tem mecanismos para impedir que esses usuários não comprometam os dados de outros usuários ou do sistema. Nele, todo arquivo tem uma dona, e um conjunto de permissões que dizem quem pode fazer o que com ele.

Para um site, permissões de arquivos e pastas são um ponto importante para o seu funcionamento. As permissões possibilitam que o servidor web escreva nos seus arquivos ou edite-os. Mas por motivos de segurança, alguns arquivos precisam ser protegidos contra escrita ou edição. Imagine se uma pessoa mal intencionada consegue [invadir um site de notícias e cria uma fake news](#)? Não queremos isso.

Você pode alterar as permissões dos seus arquivos e pastas de várias formas: - Usando o Gerenciador de Arquivos - Através de um acesso FTP - Usando o comando `chmod` na linha de comando.

## Permissões de Arquivos

Na nossa servidora, todos os arquivos da sua home pertencem a usuária que criou na instalação, ou pelo menos deveriam. Para verificar isso, você pode executar:

```
$ cd ~  
$ ls -al
```

O “a” imprime arquivos ocultos, o “l” imprime informações extras (permissões inclusive).

Em tese, toda linha da saída deve ser mais ou menos como “**permissao um\_numero usuario grupo tamanho\_do\_arquivo data horario nome\_de\_arquivo**”:

```
-rw-rw-r-- 1 maria maria 179925 Nov 10 00:59 3.0.1.tar.gz
```



permissões independente das que há agora. E o “w” é escrita, poderia ser “x” ou “r”, ou mesmo “rw”, “xw”, etc. Para voltar às permissões originais você pode executar `$ chmod u+w arquivo` ou `$ chmod u=rw arquivo`.

Por trás destas letras (rwx) existe um algoritmo que atribui valores ao tipo de acesso que se quer outorgar a cada tipo de usuário.

4 = r = leitura

2 = w = escritura

1 = x = execução

As permissões se atribuem de acordo com a soma dos tipos já descritos. Por exemplo:

6 (4+2) = leitura e escritura

5 (4+1) = leitura e execução

3 (2+1) = escritura e execução

7 (4+2+1) = leitura, escritura e execução

A tabela completa:

000 = --- = 0 = nenhuma permissão

001 = --x = 1 = somente permissão de execução

010 = -w- = 2 = somente permissão de escrita

011 = -wx = 3 = somente permissões de escrita e execução

100 = r-- = 4 = somente permissão de leitura

101 = r-x = 5 = somente permissões de leitura e execução

110 = rw- = 6 = somente permissões de leitura e escrita

111 = rwx = 7 = permissões de leitura, escrita e execução (acesso total)

As combinações se dão na seguinte ordem: proprietário, grupo e usuários.

Por exemplo: 755, da permissão de leitura, escritura e execução ao proprietário, e ao grupo e outros as permissões de execução e leitura.

E o comando ficaria assim:

**`$ chmod 755 arquivo`**

No caso de permissões em diretórios, a lógica é um pouco diferente:

- ❏ “r” em um diretório é para poder listar os arquivos dele, com ls ou o que for.
- ❏ “w” em um diretório é para poder modificar seu conteúdo, ou seja, poder criar, remover arquivos, mover arquivos e mudar permissões.

- ❏ “x” em um diretório é para poder interagir com os arquivos e diretórios dentro dele, ou seja, entrar nele e listar o conteúdo de subdiretórios.

As permissões mais comuns aplicada em diretórios web são:

**Permissão 644 para arquivos: 644 ou [rw-r-r-]:** Owner com permissão de leitura e gravação, grupo com permissão somente de leitura, outros com permissão somente de leitura.

❏ **Permissão 755 para pastas: 755 ou [rwxr-xr-x]:** Owner com permissão total, grupo com permissão de leitura e execução, outros com permissão de leitura e execução.

❏ **Permissão 777 para arquivos/pastas com necessidade específica: 777 ou [rwxrwxrwx]:** Owner com permissão total, grupo com permissão total, outros com permissão total.

É perigoso dar aos tipos de usuária Grupo e Todos permissões de escrita. Isso permite que hackers mal intencionados editem os seus arquivos. Assim, os últimos dois dígitos do número que representa as permissões dos seus arquivos e pastas nunca devem ser 2, 3, 6 ou 7. A não ser que seja pra um propósito específico.

Mas e se quisermos mudar a dona ou o grupo de um arquivo/pasta? Com o comando `chown` você pode mudar o dono de arquivos, diretórios e links. Se uma usuária comum desejar realizar certas mudanças em um arquivo, uma superusuária pode usar o comando `chown` para alterar a dona do arquivo e permitir tal alteração. Anteriormente usamos o comando `sudo chown -R root:www-data /var/www <dominio>`, ele está dizendo que a dona da pasta e sub-pastas é a usuaria `root` e o grupo é `www-data`.

Para garantir que nossa usuaria tenha acesso aos arquivos do servidor web,

vamos adicioná-la ao grupo www-data que tem permissão ao apache executando estes comandos:

```
$ usermod -a -G www-data <usuario>
```

```
$ sudo chmod 775 -R /var/www/<dominio>
```

```
$ sudo chown -R root:www-data /var/www/<dominio>
```

Coloque os arquivos html, css, javascript e/ou php no diretório /var/www/html/ .

Após isso verifique o site no endereço: [http://endereço\\_IP\\_da\\_sua\\_servidora](http://endereço_IP_da_sua_servidora)

# Transferindo arquivos com o protocolo FTP/SFTP

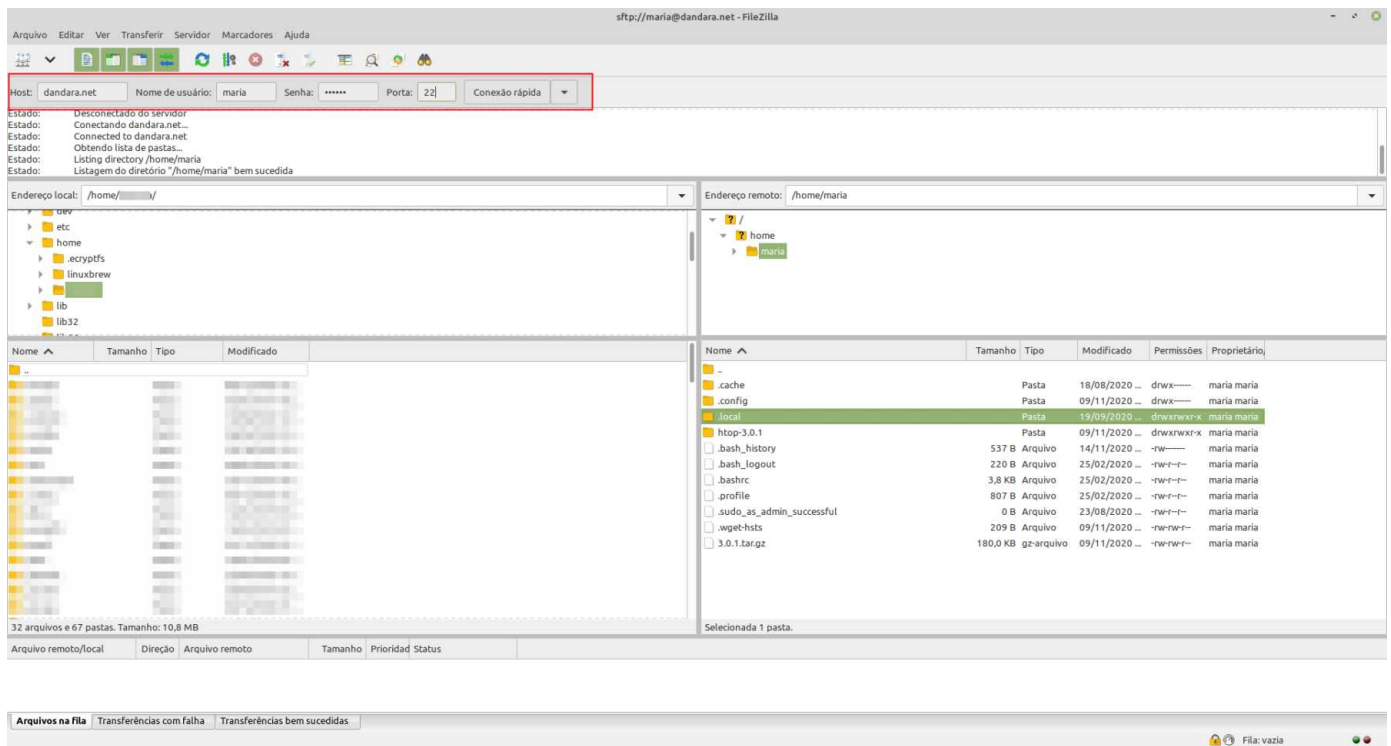
FTP é a sigla para *File Transfer Protocol*, um termo que, traduzido para o português, significa Protocolo de Transferência de Arquivos. Ele é basicamente um tipo de conexão que permite a troca de arquivos entre duas máquinas conectadas à uma rede acessível. Com isso, você pode enviar qualquer arquivo para uma outra máquina ou armazená-los em uma servidora de arquivos, ficando ela sempre disponível para o usuário acessar.

Quando trabalhamos com sites, normalmente fazemos esse intercâmbio de comunicação entre os arquivos de nossa computadora(cliente) para a servidora através de um software para transferir e gerenciar todos os dados com que queremos trabalhar. O primeiro passo é fazer o download do programa de FTP SFTP **Filezilla** um software de código aberto e um dos clientes FTP mais populares. Ele tem suporte a Windows, Linux, Mac OS X e muitos outros sistemas operacionais, incluindo suporte para vários idiomas (incluindo português) e recursos, como um Gerenciador de Sites, filas de transferência, favoritos e muito mais.

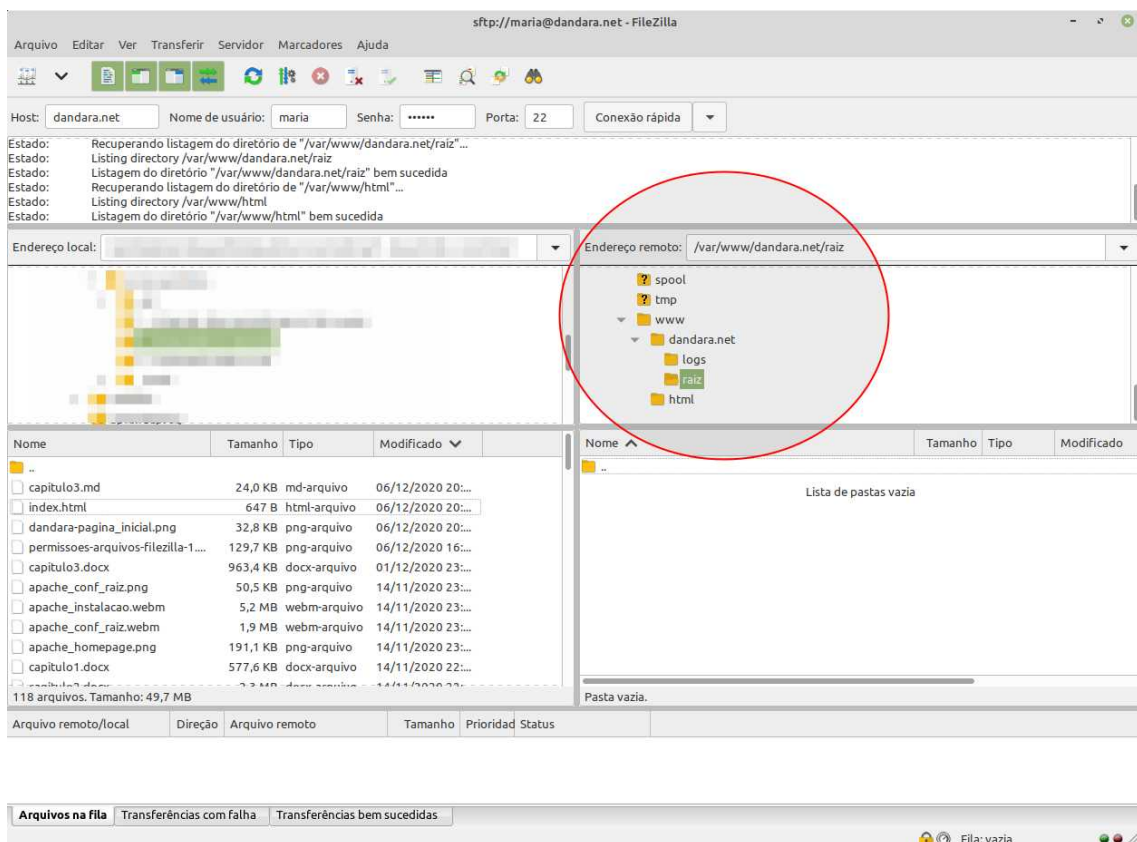
Baixe o conteúdo do nosso site de exemplo [aqui](#), descompacte em sua máquina e vamos transferir via sftp para a home do nosso site.

Faça o download em <https://filezilla-project.org> e quando terminar a instalação, abra o Filezilla.

Abrindo o Filezilla, informe os dados para conexão SFTP: IP do Host, usuário e senha. Estes dados são fornecidos no painel do seu plano de hospedagem de sites, mas nesse caso usaremos as informações de nossa Servidora que está na tabela. Em seguida, clique em **Conexão rápida** para se conectar com a servidores SFTP.

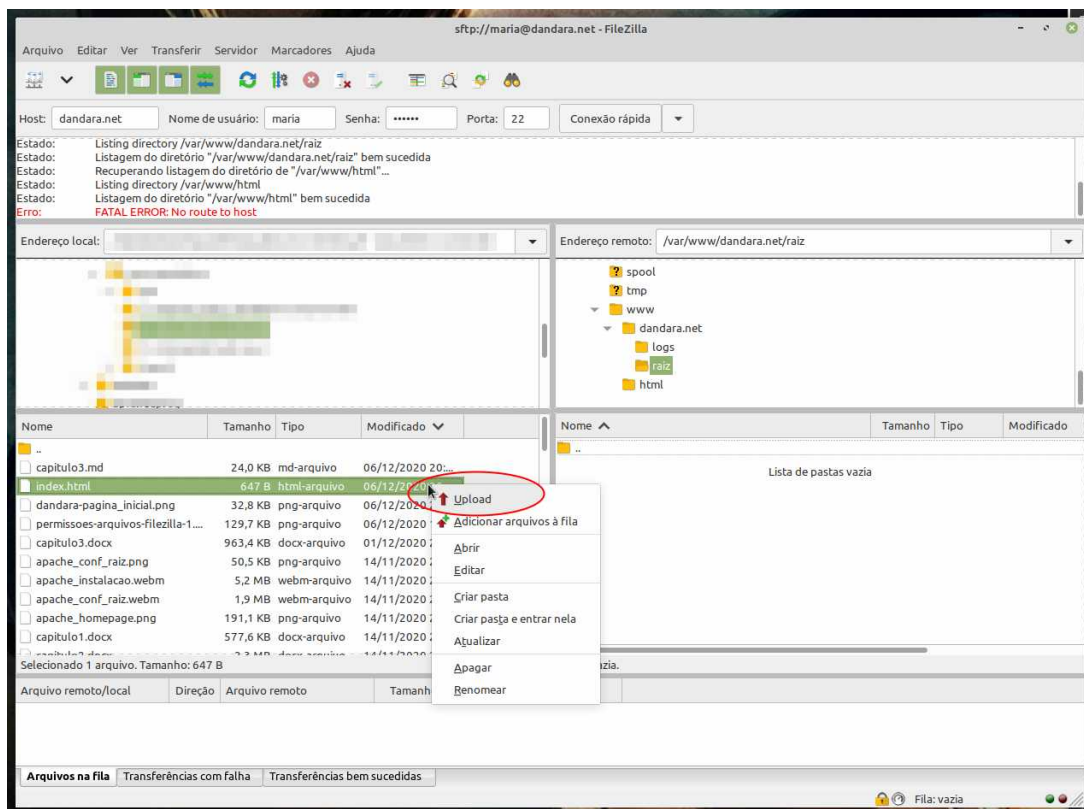


Uma vez conectado ao servidor FTP, localize na coluna a direita, **Endereço remoto**, a pasta `/var/www/<dominio>/raiz`, que é a raiz do nosso site no apache.

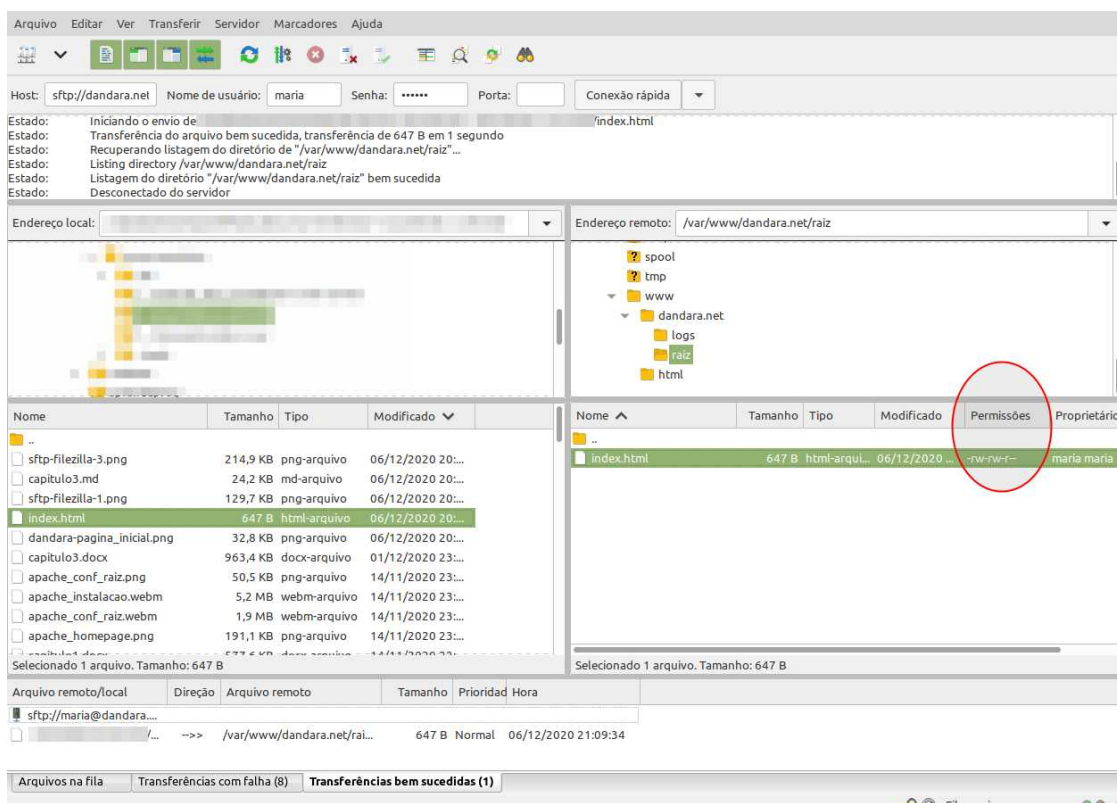




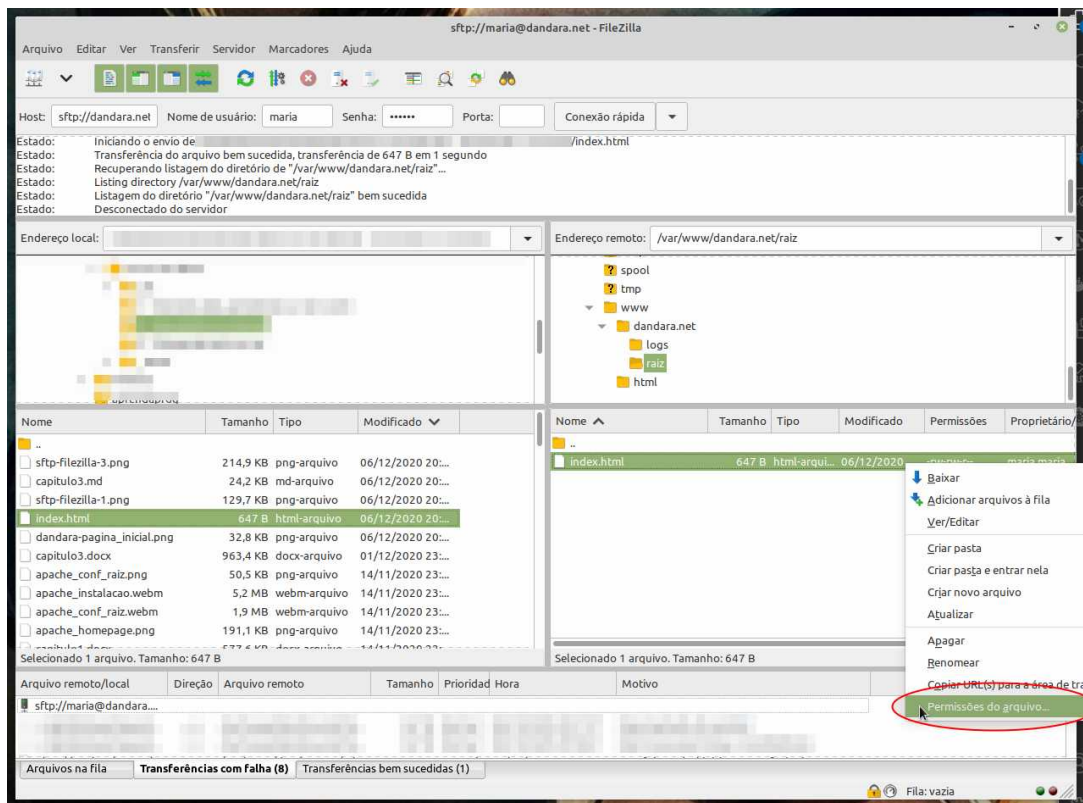
Na coluna do lado esquerdo, **Endereço Local**, navegue até a pasta onde se encontra o arquivo *index.html* que criamos e clique com o botão direito em cima do arquivo e selecione **Upload**.



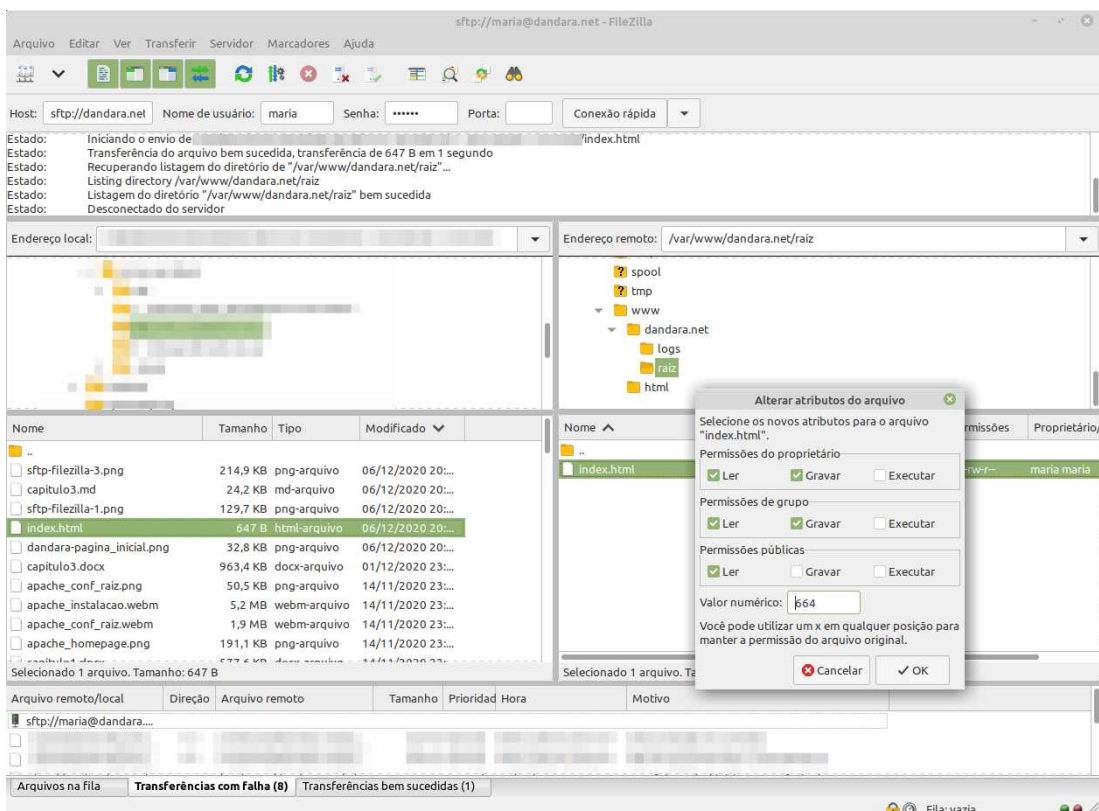
Nosso arquivo agora se encontra dentro da pasta, você pode observar a permissão atual dos arquivos e pastas na coluna Permissões.



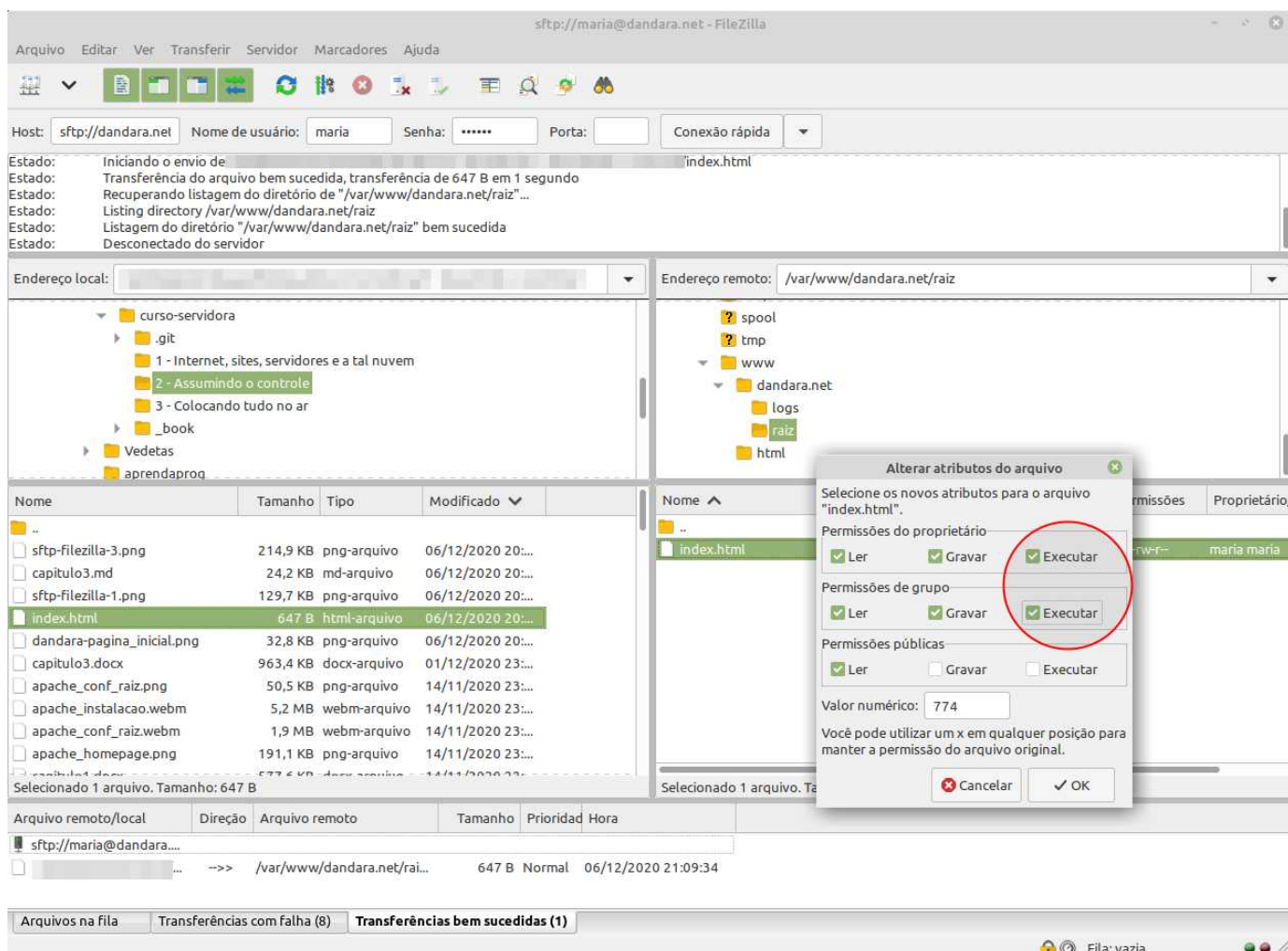
Para alterar as permissões do arquivo em questão, clique com o botão direito sobre o nome do arquivo ou pasta, e em seguida, clique em **Permissões** do arquivo.



Na tela a seguir, você poderá configurar as permissões desejadas. A permissão correta a ser delegada dependerá da aplicação a ser usada.



Para escolher a permissão, selecione as permissões clicando nas caixas correspondentes ou digite o valor numérico. Iremos dar permissão de *Executar* para o grupo e proprietária (774), clique em OK.



Logo em seguida, você poderá ver a permissão já alterada no servidor na coluna Permissões. Com isso, finalizamos a configuração e nossa página inicial está no ar! Verifique o site no endereço: [http://dominio\\_da\\_sua\\_servidora](http://dominio_da_sua_servidora).

Na próxima seção iremos subir os serviços de Blog, Rede Social e Loja Virtual para nossos links funcionarem.

# **3. Colocando todo no ar**

## **2.1 PREPARANDO O AMBIENTE**

# Ambiente LAMP - Linux + Apache + MySQL + PHP

LAMP é uma combinação de softwares livres e de código aberto. O acrônimo LAMP refere-se às primeiras letras de:

**L**inux (sistema operacional)

**A**pache (servidor web)

**M**ariaDB ou MySQL (software de banco de dados)

**P**HP (linguagens de programação) ou Python

Juntos, eles são os componentes principais para viabilizar o desenvolvimento de aplicações web de propósito geral, de alta disponibilidade e desempenho.

A combinação exata do software incluído em um pacote LAMP pode variar, especialmente com respeito ao software de script web, onde pode ser Perl ou Python no lugar do PHP. Apesar de os autores originais destes programas não os terem desenvolvido para trabalhar especificamente um com o outro, a arquitetura adotada e o conjunto de ferramentas de desenvolvimento são compartilhados e foram desenvolvidos com completa compatibilidade.

Essa combinação de software tornou-se popular devido a serem de código aberto, livres de custo, e assim de fácil adaptação. Quando usados juntos, formam servidores de aplicações para web.

# Instalando e Configurando o Interpretador PHP

O PHP é o componente do nosso ambiente que irá processar código para exibir o conteúdo dinâmico. Ele pode executar script, conectar às nossas bases de dados MySQL para obter informações, e entregar o conteúdo processado para o nosso servidor web exibir.

Vamos incluir alguns pacotes auxiliares também, de forma que o PHP possa executar sob o servidor Apache e conversar com nosso banco de dados MySQL:

```
$ sudo apt-get install php libapache2-mod-php php-mysql
```

Isto irá instalar o PHP sem problemas. Vamos testar isso em instantes.

Na maioria dos casos, vamos querer modificar a forma com a qual o Apache serve arquivos quando uma pasta é requisitada. Atualmente, se um usuário requisita uma pasta do servidor, o Apache irá olhar primeiramente para um arquivo chamado index.html. Queremos informar ao nosso servidor web para dar preferência aos arquivos PHP, então faremos o Apache olhar para um arquivo index.php primeiro.

Para fazer isto, digite este comando para abrir o arquivo dir.inf em um editor de texto com privilégios de root:

```
$ sudo nano /etc/apache2/mods-enabled/dir.conf
```

Ele terá esta aparência:

```
<IfModule mod_dir.c>  
    DirectoryIndex index.html index.cgi index.pl index.php  
    index.xhtml index.htm  
</IfModule>
```

Queremos mover o arquivo de índice PHP em destaque acima para a primeira posição depois da especificação DirectoryIndex , como segue:

```
<IfModule mod_dir.c>  
    DirectoryIndex index.php index.html index.cgi index.pl  
    index.xhtml index.htm  
</IfModule>
```

Quando você tiver concluído, salve e feche o arquivo pressionando “CTRL-X”. Você tem que confirmar a gravação digitando “Y” e em seguida pressione “ENTER” para confirmar a localização de salvamento do arquivo.

Após isso, precisamos reiniciar o servidor web Apache de forma que nossas alterações sejam reconhecidas. Você pode fazer isto digitando o seguinte:

```
$ sudo systemctl restart apache2
```

Podemos também verificar o status do serviço apache2 utilizando systemctl:

```
$ sudo systemctl status apache2
```

A fim de testar se nosso sistema está corretamente configurado para o PHP, podemos criar um script bem básico.

Vamos chamar este script de info.php. Para que o Apache possa encontrar o arquivo e servi-lo corretamente, ele deve ser salvo em um diretório muito específico, o qual é chamado de “web root”.

Este diretório está localizado em */var/www/<dominio>/raiz*. Podemos criar o arquivo neste local digitando:

```
$ sudo nano /var/www/<dominio>/raiz/info.php
```

Isto vai abrir um arquivo em branco. Queremos colocar o texto a seguir, que é um código PHP válido, dentro do arquivo:

```
<?php
phpinfo();
?>
```

Quando você tiver concluído, salve e feche o arquivo. Agora podemos testar se nosso servidor web pode exibir corretamente o conteúdo gerado por um script PHP. Para testar isso, temos apenas que visitar esta página em nosso navegador. Você vai precisar novamente do endereço IP público do seu servidor.

O endereço que você quer visitar será:

`http://endereço_IP_da_sua_servidora/info.php`

A página que você verá deve se parecer com isto:

PHP Version 7.4.3	
System	Linux dandara 5.4.0-53-generic #59-Ubuntu SMP Wed Oct 21 09:38:44 UTC 2020 x86_64
Build Date	Oct 6 2020 15:47:56
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.4/apache2
Loaded Configuration File	/etc/php/7.4/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.4/apache2/conf.d
Additional .ini files parsed	/etc/php/7.4/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.4/apache2/conf.d/10-opcache.ini, /etc/php/7.4/apache2/conf.d/10-pdo.ini, /etc/php/7.4/apache2/conf.d/20-calendar.ini, /etc/php/7.4/apache2/conf.d/20-ctype.ini, /etc/php/7.4/apache2/conf.d/20-exif.ini, /etc/php/7.4/apache2/conf.d/20-ffi.ini, /etc/php/7.4/apache2/conf.d/20-fileinfo.ini, /etc/php/7.4/apache2/conf.d/20-ftp.ini, /etc/php/7.4/apache2/conf.d/20-gettext.ini, /etc/php/7.4/apache2/conf.d/20-iconv.ini, /etc/php/7.4/apache2/conf.d/20-json.ini, /etc/php/7.4/apache2/conf.d/20-mysqli.ini, /etc/php/7.4/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.4/apache2/conf.d/20-phar.ini, /etc/php/7.4/apache2/conf.d/20-posix.ini, /etc/php/7.4/apache2/conf.d/20-readline.ini, /etc/php/7.4/apache2/conf.d/20-shmop.ini, /etc/php/7.4/apache2/conf.d/20-sockets.ini, /etc/php/7.4/apache2/conf.d/20-sysmsg.ini, /etc/php/7.4/apache2/conf.d/20-sysvsem.ini, /etc/php/7.4/apache2/conf.d/20-sysvshm.ini, /etc/php/7.4/apache2/conf.d/20-tokenizer.ini
PHP API	20190902
PHP Extension	20190902
Zend Extension	320190902
Zend Extension Build	API320190902.NTS
PHP Extension Build	API20190902.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, convert.iconv.*

This program makes use of the Zend Scripting Language Engine:  
Zend Engine v3.4.0, Copyright (c) Zend Technologies  
with Zend OPcache v7.4.3, Copyright (c), by Zend Technologies

### Configuration

Esta página basicamente fornece a você informações sobre seu servidor a partir da perspectiva do PHP. Ela é útil para depuração e para assegurar que suas configurações estão sendo corretamente aplicadas.

Se tiver êxito, então seu PHP está funcionando como esperado.

Você provavelmente vai querer remover este arquivo depois do teste, pois ele



realmente pode fornecer informações sobre seu servidor para usuários não autorizados. Para fazer isto, você pode digitar o seguinte:

```
$ sudo rm /var/www/<dominio>/raiz/info.php
```

Você sempre poderá recriar esta página se precisar acessar novamente as informações mais tarde.

Videoaula: [Instalando o PHP](#)

## Instalar módulos PHP






Para melhorar a funcionalidade do PHP, podemos opcionalmente instalar alguns módulos adicionais.

Para ver as opções disponíveis para os módulos e bibliotecas PHP, podemos direcionar os resultados do comando `apt-cache search php- | less` para o comando `less`, um paginador que lhe permite percorrer a saída de outros comandos:

```
$ sudo apt-cache search php- | less
```

Use as teclas de seta para rolar para cima e para baixo e `q` para sair.

O resultado são todos os componentes opcionais que você pode instalar. Ele lhe dará uma breve descrição de cada um:

-  libnet-libidn-perl - Perl bindings for GNU Libidn
-  php-all-dev - package depending on all supported PHP development packages
-  php-cgi - server-side, HTML-embedded scripting language (CGI binary) (default)
-  php-cli - command-line interpreter for the PHP scripting language (default)
-  php-common - Common files for PHP packages

- 📦 php-curl - CURL module for PHP [default]
- 📦 php-dev - Files for PHP module development (default)
- 📦 php-gd - GD module for PHP [default]
- 📦 php-gmp - GMP module for PHP [default]

Para obter mais informações sobre o que cada módulo faz, você pode buscar na Internet, ou olhar a descrição longa do pacote digitando:

```
$ sudo apt-cache show nome_do_pacote
```

Haverá uma grande quantidade de saída, com um campo chamado Description-en que terá uma explicação mais longa da funcionalidade que o módulo oferece.

Por exemplo, para encontrar o que o módulo php-cli faz, podemos digitar isto:

```
$ sudo apt-cache show php-cli
```

Juntamente com várias outras informações, você vai encontrar algo parecido com isto:

```
Description-en: command-line interpreter for the PHP scripting language (default)
```

```
    This package provides the /usr/bin/php command interpreter, useful for testing PHP scripts from a shell or performing general shell scripting tasks.
```

```
    PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used
```

```
    open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML.
```

```
    This package is a dependency package, which depends on Debian's default PHP version (currently 7.0).
```

Se, após pesquisar, você decidir que gostaria de instalar um pacote, você pode fazê-lo utilizando o comando `apt-get install` assim como fizemos para nossos outros softwares.

Se decidirmos que o `php-cli` é algo que precisamos, podemos digitar:

```
$ sudo apt-get install php-cli
```

Se você quiser instalar mais de um módulo, você pode fazer isso listando cada um, separado por um espaço, seguindo o comando `apt-get`, como abaixo:

```
$ sudo apt-get install package1 package2 ...
```

# Instalando e Configurando o Banco de Dados MySQL

O MySQL é um sistema de gerenciamento de bancos de dados. Basicamente, ele irá organizar e proporcionar acesso a bases de dados onde nosso site pode armazenar informação.

## Instalar módulos PHP

Vamos instalar o servidor MySQL e também alguns pacotes “auxiliares” que irão nos ajudar a obter nossos componentes para comunicarem uns com os outros:

```
$ sudo apt-get update
$ sudo apt-get install mysql-server php-mysql
```

Durante a instalação, seu servidor vai pedir para você selecionar e confirmar uma senha para o usuário “root” do MySQL. Esta é uma conta administrativa no MySQL que possui privilégios avançados. Pense nela como sendo similar à conta de root para o próprio servidor (no entanto, esta que você está configurando agora é uma conta específica do MySQL).

Quando a instalação estiver concluída, precisaremos executar alguns comandos adicionais para ter nosso ambiente MySQL configurado de forma segura.

Primeiro, iremos executar um script simples de segurança que vai remover alguns padrões perigosos e bloquear um pouco o acesso ao nosso sistema de banco de dados. Inicie o script interativo executando:

```
$ mysql_secure_installation
```

```
Would you like to setup VALIDATE PASSWORD component? [Y/n] Y
```

```
Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 0
```

```
New password: <senha>
```

```
Re-enter new password: <senha novamente>
```

```
Remove anonymous users? [Y/n] Y
```

Disallow root login remotely? [Y/n] Y  
Remove test database and access to it? [Y/n] Y  
Reload privilege tables now? [Y/n] Y

Você será solicitado a definir qual o nível de proteção da senha de usuário do MySQL, iremos colocar a Low (baixa), que validará se a senha tem mais de 8 caracteres. Digite 0 para escolher LOW e tecla ENTER, em seguida digite a senha que você definiu para a conta root do MySQL duas vezes seguidas de ENTER.

Para as demais perguntas, você deve simplesmente responder com y (Yes/Sim) e apertar a tecla "ENTER" em cada prompt. Isto irá remover alguns usuários e bases de exemplo, desabilitar logins remotos de root, e carregar estas novas regras para que o MySQL aplique imediatamente as alterações que fizemos.

Neste ponto, seu sistema de banco de dados está agora configurado e podemos avançar.

Dica: No site da SoftBlue (<http://www.softblue.com.br>) tem um curso completo e gratuito, com certificado, da linguagem SQL. Vale a pena fazer pra entender melhor como funciona um banco de dados.

## Dicas

No site da SoftBlue (<http://www.softblue.com.br>) tem um curso completo e gratuito, com certificado, da linguagem SQL. Vale a pena fazer pra entender melhor como funciona um banco de dados.

Videoaula: [Instalando o MySQL](#)

# Instalando Softwares em PHP

Quando vamos instalar um sistema em PHP na nossa servidora, a primeira coisa que devemos fazer é ler a documentação de como realizar a instalação, atualização e cópia de segurança do seu sistema.

Geralmente, uma boa documentação é dividida em:

- **Resumo da Instalação:** Resumo do que consistem os passos para a instalação do sistema.

- **Requisitos Mínimos:** Configurações mínimas de servidor (memória, processamento, disco, etc.), software (banco de dados, interpretadores, bibliotecas, etc.) e rede (regras de firewall, configurações de rede, etc.). Nestas configurações é necessário verificar se a versão dos softwares e da capacidade do servidor atendem o mínimo necessário para rodar tendo o suporte do desenvolvimento.

- **Requisitos Desejáveis:** Configurações desejáveis de servidor (memória, processamento, disco, etc.), software (banco de dados, interpretadores, bibliotecas, etc.) e rede (regras de firewall, configurações de rede, etc.). Nestas configurações é necessário verificar se as versões do software e da capacidade do servidor atendem o desejável para rodar com o desempenho e suporte esperado pelos desenvolvedores.

- **Guia de Instalação:** Passo a passo (simples ou detalhado) do processo de instalação. Nele vão estar as configurações de base de dados, configurações personalizadas de interpretadores, informações para download, comandos de instalação, etc.

- **Problemas Comuns:** Problemas que podem ocorrer no processo de instalação e as possíveis soluções.

📖 **Manual do Usuário:** Manual de uso do sistema para navegar e configurar o software na visão do administrador ou usuário.

📖 **Como Atualizar:** Passo a passo de como atualizar o software com segurança.

📖 **Como realizar uma cópia de segurança:** Fazer backups é mais que necessário, principalmente no caso de algum incidente onde seja necessário restaurar o sistema ou para caso uma atualização de algum erro.

Depois de ler toda a documentação, preparamos o ambiente com os requisitos necessários, instalamos o software e preparamos a rotina de backup.

Agora vamos ver um pouco mais detalhes destes pontos.

## Requisitos

Vamos a um exemplo de requisitos do PrestaShop (<http://doc.prestashop.com>):

System requirements:

PHP 5.4 or later.

Useful settings (in the php.ini file):

allow\_url\_fopen set to On,

register\_globals set to Off,

upload\_max\_filesize set to "16M" (or more).

Must-have PHP extensions (in the php.ini file): PDO\_MySQL, cURL, SimpleXML, mcrypt, GD, OpenSSL, DOM, SOAP, Zip, fileinfo.

Useful server tools: cron/crontab, Memcached.

MySQL 5.0 or later.

Better if:

Unix/Linux hosting.

Apache Web Server 2.0 or later or nginx Web Server.

Apache module settings:

mod\_rewrite enabled,  
mod\_security disabled,  
mod\_auth\_basic disabled.

At least 128 Mb of RAM dedicated to PHP. The more the better.

Access codes to your FTP server, your MySQL database

These should be provided by your web host if you are not doing a local installation.

Any text editor.

Any FTP client.

Any modern Web browser (if using Internet Explorer: at least IE9).

Nela está dizendo que é necessário que em seu servidor esteja instalado o PHP 5.4 ou mais novo, que as extensões PDO\_MySQL, cURL, SimpleXML, mcrypt, GD, OpenSSL, DOM, SOAP, Zip e fileinfo do PHP estejam habilitadas, as ferramentas cron/crontab, Memcached e que o PHP tem que ser configurado (no arquivo php.ini) com os parâmetros: - allow\_url = on - register\_globals = Off - upload\_max\_filesize = 16M

O banco de dados necessário é MySQL Server 5.0 ou o mais novo. E pra ter melhor desempenho é necessário que o servidor seja Unix/Linux, o Apache Server ser 2.0 (ou mais novo) com o módulo mod\_rewrite habilitado e os módulos mod\_security e mod\_auth\_basic desabilitados, ou Nginx Web Server. E pelo menos 128MB de RAM dedicados ao PHP.

Acesso as chaves de acesso do servidor FTP e MySQL, um editor de texto, um cliente FTP e um browser atualizado.

Caso uma dessas configurações não seja atendida pela sua servidora, é necessário que seja instalado ou atualizado com as necessidades descritas ou não é indicada a instalação do software.



No caso das extensões php, por exemplo, você pode buscar no Google (“instalar módulo php OpenSSL”) ou através apt-cache (“apt-cache search mcrypt”). Após instalar, reinicie o servidor web.

## Guia de Instalação

Nele estará descrito os procedimentos de instalação do software conforme esperado pelos mantenedores da ferramenta.

No Wordpress ([https://codex.wordpress.org/pt-br:Instalando\\_o\\_WordPress](https://codex.wordpress.org/pt-br:Instalando_o_WordPress)), por exemplo, tem todas as formas possíveis de instalação. Você encontra como instalar com ferramentas de serviços de hospedagem, instalação resumida, instalação detalhada, instalação de diferentes ambientes e múltiplos blogs. Cada uma delas explica como efetuar a instalação desde o download até o primeiro login. Na instalação através de ferramentas, tem links para guias ou vídeos do passo a passo pra usar o painel de controle da hospedagem e instalar. Na instalação resumida são explicados os passos necessários para instalar em poucas linhas, que para usuários experientes leva cerca de 5 minutos. Na instalação detalhada, mostra-se passo a passo com capturas de telhas o uso das ferramentas para realizar a instalação. Na instalação em diferentes ambientes, você acha a opção de uma instalação em ambiente local (máquina do usuário, desktop) com linux, mac e windows. E por último, na instalação de múltiplos blogs tem o manual para a configuração do ambiente para isso.

No caso do PrestaShop existem 3 formas de instalação: Instalação padrão em um servidor com FTP/SCP, a instalação local no seu computador de usuário (Windows, Mac e Linux) e a instalação por linha de comando da loja.

## Problemas Comuns

Nos problemas comuns estão reportadas as falhas e dúvidas comuns que podem ocorrer durante a instalação.

No OwnCloud (<https://doc.owncloud.org>), tem uma sessão chamada *Issues and Troubleshooting* onde tem vários problemas comuns reportados por usuários e as possíveis soluções para resolver ou contornar os problemas. Por exemplo, dos problemas que podem aparecer no log de erros temos o “*SQLSTATE[HY000] [1040] Too many connections*”, que quer dizer que tem muitas conexões ao banco e não é possível abrir novas. A solução sugerida na pagina é aumentar o limite de conexões da base de dados.

## Manual do Usuário

No Manual do Usuário ou do administrador estão todos os fluxos de navegação e configuração do software, para aprender mais sobre as suas funcionalidades e também repassar para os usuários finais.

No Prestashop, por exemplo, você tem o manual de usuário, o guia de desenvolvimento de temas, o guia de desenvolvimento de módulos, o guia de design, o guia de administrador de sistemas e o guia de mercado. Existe até um guia para SEO, para auxiliar na construção de estratégias de web marketing da loja virtual.

## Como Atualizar

A segunda coisa mais importante ao administrar uma servidora são as rotinas de atualização da mesma. Precisamos ver como executar atualizações manuais ou automáticas, os tipos de atualização (novas funcionalidades, segurança, correção de erros, etc.) em que horário executá-las e o procedimento para restaurar a versão anterior caso falhe a atualização (rollback).

Por exemplo, o Wordpress (<https://wordpress.org/news/>) lançou uma versão

4.7.4 que tinha 6 vulnerabilidades sérias de segurança. A recomendação era atualizar o mais rápido possível para a versão 4.7.5, senão o seu sistema seria passível de ataques de crackers. Então temos que ficar de olho nas mensagens de atualização de plugins, temas e do próprio wordpress. E, além disso, manter o sistema operacional, o php e o servidor web atualizado, pois os mesmos podem ter problemas de segurança, desempenho e até incompatibilidade com um software mais atual.

## **Cópias de Segurança**

A primeira coisa que queda qual devemos nos preocupar ao cuidar de uma servidora é com as rotinas de cópia de segurança (backup). Imagine que sua atualização não deu certo, seu site foi atacado, um registro do banco de dados apagado, ou uma pasta inteira deletada. O que fazer? Restaura-se o último backup das informações. Os dados de backup devem ser armazenados em alguma mídia fora do servidor de origem da cópia. No caso, pode ser em algum serviço de armazenamento em nuvem, em um HD (disco rígido) externo, pen-drives, DVD's, ou em um servidor projetado para backups.

As cópias de segurança devem obedecer alguns pontos, tais como: o tempo de execução, a periodicidade, a quantidade de exemplares das cópias armazenadas, o tempo que as cópias devem ser mantidas, a capacidade de armazenamento, a compactação (Ex.: zip, tar.gz) e criptografia dos dados. Assim, a velocidade de execução da cópia deve ser aumentada tanto quanto possível para que o grau de interferência desse procedimento nos serviços seja mínimo. A periodicidade deve ser analisada em função da quantidade de dados alterados na organização, no entanto se o volume de dados for elevado, as cópias devem ser diárias. Deve-se estabelecer um horário para realização da cópia, conforme a laboração da organização, devendo ser preferencialmente noturno ou em horário de pouco acesso. Para uma fácil localização, a cópia deve ser guardada por data e categoria, em local seguro (Ex.: backup\_20170719\_2330.tar.gz).

## Buscando guia de instalação que façam sentido

Instalar um software no servidor, principalmente em outra língua, pode ser uma tarefa complicada para a maioria das pessoas. Existem vários sites e blogs criados pela comunidade de tecnologia que ensinam passo a passo como fazer a instalação de um software. É uma boa opção quando não entendemos o manual.

Vamos a um exemplo:

Queremos instalar uma cloud privada, que tenha os mesmos serviços do Google Drive. Decidimos, usar o NextCloud. Ao digitar “como instalar o Nextcloud” na busca apareceu o seguinte site: <https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-nextcloud-on-ubuntu-20-04-pt>

Ele começa explicando o que é o NextCloud e quais são os requisitos para o mesmo ser instalado. Em seguida ele dá o passo a passo com todos os comandos que devem ser executados para colocar tudo no ar. Mas queremos ter certeza sobre o que vamos fazer, então que tal procurar no YouTube um vídeo ensinando? <https://www.youtube.com/watch?v=sqVEL8PVJmg>

Depois de assistir o vídeo, voltamos ao site do tutorial de instalação e criamos o server com segurança. Mas enfatizo que, seguir os guias oficiais é a melhor opção, principalmente por conta de alguns dos tutoriais que achamos na internet não são do site atualizado. Por isso sempre devemos verificar e efetuar os downloads do software nos sites oficiais e verificar se não teve mudança nas dependências.

Agora, nós vamos instalar o Prestashop (loja virtual), Wordpress (Blog), Humhub (Rede Social) nas próximas páginas. As instruções são baseadas na documentação oficial.

## **3.2 CRIANDO NOSSO BLOG COM WORDPRESS**

# Introdução ao Wordpress

O WordPress é um projeto de código aberto que você pode usar para criar sites, blogs ou aplicativos com ênfase em acessibilidade, desempenho, segurança e facilidade no uso. Usado por 63% de todos os sites que usam um sistema de gerenciamento de conteúdo (CMS), os sites com WordPress representam 36% de todos os sites que estão atualmente online. Eles acreditam que um ótimo programa deve funcionar com uma configuração mínima, para que você possa se concentrar em compartilhar sua história, produto ou serviços livremente. O programa básico do WordPress é simples e previsível, assim você pode começar rapidamente. Ele também oferece recursos poderosos para quem quer crescer.

O WordPress surgiu em 2003, quando Mike Little e Matt Mullenweg criaram um fork do b2/cafelog, isto é, uma versão alterada daquele programa. A necessidade de um sistema de publicação pessoal elegante e bem estruturado era clara mesmo naquela época. Hoje, o WordPress é baseado em PHP e MySQL, e licenciado sob a GPLv2. Também é a plataforma escolhida por mais de 38% de todos os sites da internet.

O WordPress, um projeto de código aberto, evoluiu ao longo do tempo com a ajuda de hábeis e dedicados desenvolvedores, designers, cientistas, autores de blogs e outros. O WordPress dá oportunidades para que qualquer um crie e compartilhe de receita de bolo até movimentos para mudar o mundo. Pessoas com experiência limitada em tecnologia podem usá-lo sem precisar de muitas configurações, enquanto que os mais experientes podem personalizá-lo de formas extraordinárias.

Ele é um programa feito e pensado para todos.

# Instalando o Wordpress

## Criando um banco de dados do MySQL e um usuário para o WordPress

O primeiro passo que vamos dar é um passo preparatório. O WordPress utiliza o MySQL para gerenciar e armazenar as informações do site e as do usuário. Já temos o MySQL instalado, mas precisamos criar um banco de dados e um usuário para o WordPress usar.

Para começar, faça login na conta root (administrativa) do MySQL emitindo este comando (note que este não é usuário root do seu servidor):

```
$ sudo mysql -u root -p
```

Será solicitada a senha que você configurou para a conta raiz do MySQL quando instalou o software.

Dentro banco de dados, podemos criar um banco de dados exclusivo para o WordPress controlar. Chame isso do que quiser, mas vamos usar o nome wordpress neste guia. Crie o banco de dados para o WordPress digitando:

```
mysql> CREATE DATABASE wordpress DEFAULT CHARACTER SET utf8 COLLATE  
utf8_unicode_ci;
```

*Nota: cada instrução do MySQL deve terminar em um ponto e vírgula (;). Verifique para garantir que a instrução foi seguida (ou seja, o ponto e vírgula foi usado), caso estiver enfrentando algum problema.*

Em seguida, vamos criar uma conta de usuário do MySQL separada que vamos usar exclusivamente para operar nosso novo banco de dados. Criar bancos de dados e contas específicas pode-nos ser útil do ponto de vista de gestão e segurança. Usaremos o nome wordpressuser neste guia, mas sinta-se livre para usar qualquer nome que seja relevante para você.

Vamos criar essa conta, definir uma senha e conceder o acesso ao banco de dados que criamos. Podemos fazer isso digitando o seguinte comando: Lembre-se de escolher aqui uma senha forte para seu banco de dados. Substitua-a em password:

```
mysql> CREATE USER 'wordpressuser'@'%' IDENTIFIED WITH  
mysql_native_password BY 'password';
```

Em seguida, informe ao banco de dados que nosso usuário wordpressuser deve ter acesso completo ao banco de dados que configuramos:

```
mysql> GRANT ALL ON wordpress.* TO 'wordpressuser'@'%';
```

Agora, você tem um banco de dados e uma conta de usuário, criados especificamente para o WordPress. Precisamos liberar os privilégios para que a instância atual do MySQL saiba sobre as alterações recentes que fizemos:

```
mysql> FLUSH PRIVILEGES;
```

Saia do MySQL digitando:

```
mysql> EXIT;
```

Anote no caderno o nome do banco, usuário e a senha que criamos. No próximo passo, vamos instaurar algumas bases para plug-ins do WordPress baixando extensões PHP para nosso servidor.

Videoaula: [Criando o Banco de Dados do Wordpress no MySQL](#)



## Como instalar as extensões adicionais do PHP

Ao configurar nossa pilha LAMP, precisamos apenas de um conjunto mínimo de extensões para fazer com que o PHP se comunique com o MySQL. O WordPress e muitos dos seus plug-ins potencializam extensões adicionais do PHP.

Podemos baixar e instalar algumas das extensões PHP mais populares para serem usadas com o WordPress digitando:

```
$ sudo apt update
$ sudo apt install php-curl php-gd php-mbstring php-xml php-xmlrpc
php-soap php-intl php-zip
```

Isso irá preparar o terreno para instalarmos plug-ins adicionais em nosso site do WordPress.

*Nota: cada plug-in do WordPress tem seus próprios requisitos. Alguns podem precisar de pacotes adicionais do PHP para ser instalados. Verifique a documentação do plug-in para descobrir seus requisitos PHP. Se estiverem disponíveis, eles podem ser instalados com o apt como mostrado acima.*

Precisaremos reiniciar o Apache para carregar essas novas extensões. Considerando que vamos fazer mais configurações no Apache na próxima seção, você pode esperar ou reiniciar agora para completar o processo das extensões do PHP.

```
sudo systemctl restart apache2
```

Videoaula: [Instalando módulos adicionais de PHP para o Wordpress](#)

## Ajustando a configuração do Apache para permitir que o .htaccess substitua e reescreva

A seguir, vamos fazer alguns pequenos ajustes na nossa configuração do Apache. Baseado nos tutoriais de pré-requisitos, você deve ter um arquivo de configuração para seu site no diretório `/etc/apache2/sites-available/`.

Utilizaremos o `/var/www/<dominio>/wordpress` como o diretório root de nossa instalação do WordPress. Vamos criar com o comando:

```
$ sudo mkdir /var/www/<dominio>/wordpress
```

Com nossos caminhos identificados, podemos seguir em frente e trabalhar com o .htaccess para que o Apache consiga lidar com as alterações de configuração para cada diretório. Abra o arquivo de configuração do Apache para seu site com um editor de texto como o nano.

```
sudo nano /etc/apache2/sites-available/blog.<dominio>.conf
```

Para permitir aos arquivos .htaccess, precisamos configurar a diretiva AllowOverride dentro de um bloco Directory apontando para nossa raiz de documentos. Adicione o seguinte bloco de texto dentro do bloco no seu arquivo de configuração, tendo certeza de usar o diretório root Web correto:

```
<VirtualHost *:80>
    ServerName blog.<dominio>
    ServerAdmin <seu_email>

    DocumentRoot /var/www/<dominio>/wordpress
    Alias /blog/ /var/www/<dominio>/wordpress/
    <Directory /var/www/<dominio>/wordpress/>
        AllowOverride All
    </Directory>
</VirtualHost>
```

Quando você terminar, salve e feche o arquivo. No nano, você pode fazer isso pressionando CTRL e X juntos, então Y, depois ENTER.

Em seguida, podemos habilitar o mod\_rewrite para que possamos usar o recurso de permalink (ou link permanente) do WordPress:

```
$ sudo a2enmod rewrite
```

O comando a2enmod chama um script que habilita o módulo especificado dentro da configuração do Apache.

Antes de implementarmos as alterações que fizemos, verifique se não cometemos nenhum erro de sintaxe executando o seguinte teste.

```
$ sudo apache2ctl configtest
```

Estando tudo certo vamos ativar o site executando:

```
$ a2ensite blog.<dominio>.config
```

Reinicie o Apache para implementar as alterações. Neste momento, certifique-se de fazer a reinicialização, mesmo que tenha reiniciado anteriormente neste tutorial.

```
$ sudo systemctl restart apache2
```

Em seguida, vamos baixar e configurar o WordPress propriamente dito.

Videoaula: [Configurando o servidor virtual para o Wordpress no Apache](#)

## Como baixar o WordPress

Agora que nosso software de servidor está configurado, podemos baixar e configurar o WordPress. Por motivos de segurança, é sempre recomendável obter a versão mais recente do WordPress do site deles.

Vá até um diretório gravável (recomendamos um temporário como o /tmp) e baixe o lançamento compactado.

```
$ cd ~  
$ curl -O https://wordpress.org/latest.tar.gz
```

Extraia o arquivo comprimido para criar a estrutura de diretórios do WordPress:

```
$ tar xzvf latest.tar.gz
```

Vamos mover esses arquivos temporariamente para nossa raiz de documentos. Antes de fazermos isso, adicionamos um arquivo de .htaccess fictício para que esteja disponível para o WordPress usar mais tarde.

Crie o arquivo digitando:

```
$ touch wordpress/.htaccess
```

Também vamos copiar o exemplo de arquivo de configuração no nome do arquivo que o WordPress lê:

```
$ cp wordpress/wp-config-sample.php wordpress/wp-config.php
```

Também podemos criar o diretório upgrade, para que o WordPress não se depare com problemas de permissão ao tentar fazer isso sozinho após uma atualização no software:

```
$ mkdir wordpress/wp-content/upgrade
```

Agora, podemos copiar o conteúdo todo do diretório para nossa raiz do wordpress. Estamos usando um ponto no final do nosso diretório fonte para indicar que tudo dentro do diretório deve ser copiado, incluindo arquivos escondidos (como o arquivo .htaccess que criamos):

```
$ sudo cp -a wordpress/. /var/www/<dominio>/wordpress
```

Certifique-se de substituir o diretório /var/www/<dominio>/wordpress pelo diretório que você configurou na sua servidora.

Antes de fazermos a configuração do WordPress baseada na Web, precisamos ajustar alguns itens em nosso diretório do WordPress. Um passo importante que precisamos realizar é configurar permissões e propriedade dos arquivos que sejam adequadas.

Vamos começar concedendo a propriedade de todos os arquivos para o usuário e grupo www-data. Este é o usuário sob o qual o servidor Web Apache executa. O Apache terá que ser capaz de ler e gravar arquivos do WordPress para atender o site e realizar atualizações automáticas.

Atualize a propriedade com o comando chown que lhe permite modificar as permissões dos arquivos e concederemos permissão para o www-data. Certifique-se de apontar para o diretório relevante do seu servidor.

```
$ sudo chown -R www-data:www-data /var/www/<dominio>/wordpress
```

Em seguida, iremos executar dois comandos find para definir as permissões para os diretórios e arquivos do WordPress:

```
$ sudo find /var/www/<dominio>/wordpress/ -type d -exec chmod 750 {}  
\;  
$ sudo find /var/www/<dominio>/wordpress/ -type f -exec chmod 640 {}  
\;
```

Essas permissões devem ser suficientes para que você trabalhe com o WordPress de maneira efetiva, mas note que alguns plug-ins e procedimentos podem exigir ajustes adicionais.

Videoaula: [Fazendo o download do Wordpress](#)

## Como configurar o arquivo de configuração do WordPress

Agora, precisamos fazer algumas alterações no arquivo de configuração principal do WordPress.

Ao abrirmos o arquivo, nossa prioridade será ajustar algumas chaves secretas para acrescentar alguma segurança a nossa instalação. O WordPress oferece um gerador seguro para esses valores para que você não tenha que tentar obter bons valores por conta própria. Estes valores são usados internamente, então a usabilidade não será prejudicada se houverem valores complexos e seguros aqui.

Para pegar valores seguros do gerador de chave secreta do WordPress, digite:

```
$ curl -s https://api.wordpress.org/secret-key/1.1/salt/
```

Você receberá de volta valores únicos que se assemelham ao resultado do bloco abaixo.

Aviso! É importante que você solicite sempre valores únicos. Não copie os valores abaixo!

Output

```
define('AUTH_KEY',  
'1jl/vqfs<XhdXoAPz9 DO NOT COPY THESE VALUES c_j{iwqD^<+c9.k<J@4H');  
  
define('SECURE_AUTH_KEY', 'E2N-h2]Dcvp+aS/p7X DO NOT COPY THESE VALUES {Ka(f;rv?  
Pxf})CgLi-3');  
  
define('LOGGED_IN_KEY',  
'W(50,{W^,OPB%PB<JF DO NOT COPY THESE VALUES 2;y&,2m%3]R6DUth[;88');  
  
define('NONCE_KEY',  
'll,4UC)7ua+8<!4VM+ DO NOT COPY THESE VALUES #`DXF+[$atzM7 o^-C7g');  
  
define('AUTH_SALT',  
'koMrurzOA+[L_IG}kf DO NOT COPY THESE VALUES 07VC*Lj*ID&?3w!BT#-');  
  
define('SECURE_AUTH_SALT', 'p32*p,]z%LZ+pAu:VY DO NOT COPY THESE VALUES C-?  
y+KODK_+F|0h{!_xY');  
  
define('LOGGED_IN_SALT',  
'i^/G2W7!-1H2OQ+t$3 DO NOT COPY THESE VALUES t6**bRVFSD[Hi])-qS`|');  
  
define('NONCE_SALT',  
'Q6]U:K?j4L%Z]}h^q7 DO NOT COPY THESE VALUES 1% ^qUswWgn+6&xqHN&%');
```

Essas são linhas de configuração que podemos colar diretamente no nosso arquivo de configuração para definir chaves seguras. Copie o resultado que acabou de receber.

Em seguida, abra o arquivo de configuração do WordPress:

```
$ sudo nano /var/www/<dominio>/wordpress/wp-config.php
```

Encontre a seção que contém os valores de exemplo para essas configurações.

...

```
define('AUTH_KEY', 'put your unique phrase here');
define('SECURE_AUTH_KEY', 'put your unique phrase here');
define('LOGGED_IN_KEY', 'put your unique phrase here');
define('NONCE_KEY', 'put your unique phrase here');
define('AUTH_SALT', 'put your unique phrase here');
define('SECURE_AUTH_SALT', 'put your unique phrase here');
define('LOGGED_IN_SALT', 'put your unique phrase here');
define('NONCE_SALT', 'put your unique phrase here');
```

...

Exclua essas linhas e cole os valores que copiou da linha de comando:

...

```
define('AUTH_KEY', 'VALUES COPIED FROM THE COMMAND LINE');
define('SECURE_AUTH_KEY', 'VALUES COPIED FROM THE COMMAND LINE');
define('LOGGED_IN_KEY', 'VALUES COPIED FROM THE COMMAND LINE');
define('NONCE_KEY', 'VALUES COPIED FROM THE COMMAND LINE');
define('AUTH_SALT', 'VALUES COPIED FROM THE COMMAND LINE');
define('SECURE_AUTH_SALT', 'VALUES COPIED FROM THE COMMAND LINE');
define('LOGGED_IN_SALT', 'VALUES COPIED FROM THE COMMAND LINE');
define('NONCE_SALT', 'VALUES COPIED FROM THE COMMAND LINE');
```

...

*Dica: Utilize o Filezilla para editar os arquivos com mais facilidade se desejar.*

A seguir, vamos modificar algumas das configurações de conexão do banco de dados no início do arquivo. Você precisa ajustar o nome do banco de dados, o usuário do banco de dados e a senha associada que você configurou no MySQL.

Esta configuração pode ser adicionada abaixo das configurações de conexão do banco de dados ou em qualquer outro lugar no arquivo com o comando:

```
$ sudo nano /var/www/<dominio>/wordpress/wp-config.php
```



```
...
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

/** MySQL database username */
define( 'DB_USER', 'wordpressuser' );

/** MySQL database password */
define( 'DB_PASSWORD', 'password' );

/** MySQL hostname */
define( 'DB_HOST', 'localhost' );

/** Database Charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The Database Collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );
```

Salve e feche o arquivo quando você terminar.

Videoaula: [Atualizando os arquivos de configuração do Wordpress](#)

## Configurando nosso blog

Agora vamos novamente alterar o nosso arquivo hosts em nosso computador, como vimos anteriormente. Dessa vez iremos apenas acrescentar o endereço `blog.<dominio_da_servidora>` na mesma linha do endereço do nosso domínio:

`<ip_da_servidora> <dominio_da_servidora> blog.<dominio_da_servidora>`

Exemplo:

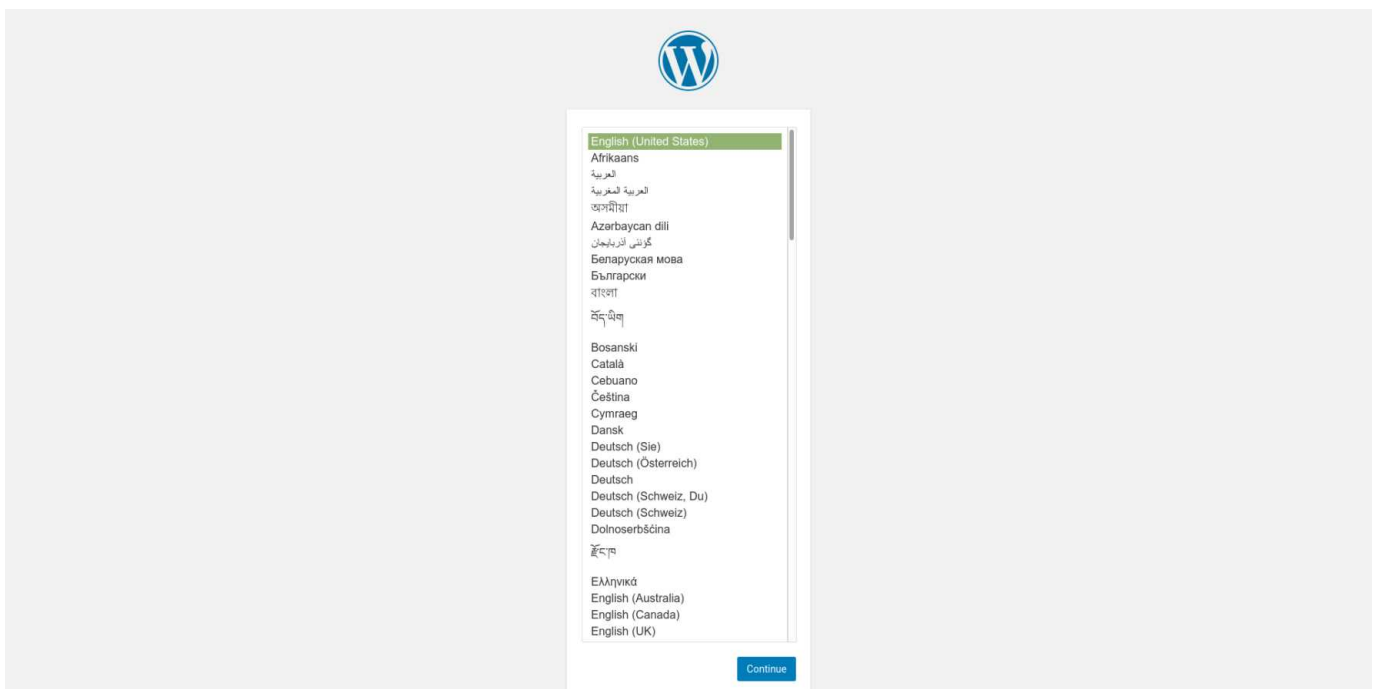
`127.0.0.1 localhost`

`192.168.15.27 dandara.net www.dandara.net blog.dandara.net`

Agora que a configuração do servidor está completa, podemos concluir a instalação através da interface com a Web. No seu navegador Web, navegue até o nome de domínio do seu

**servidor: `https://blog.<dominio>`**

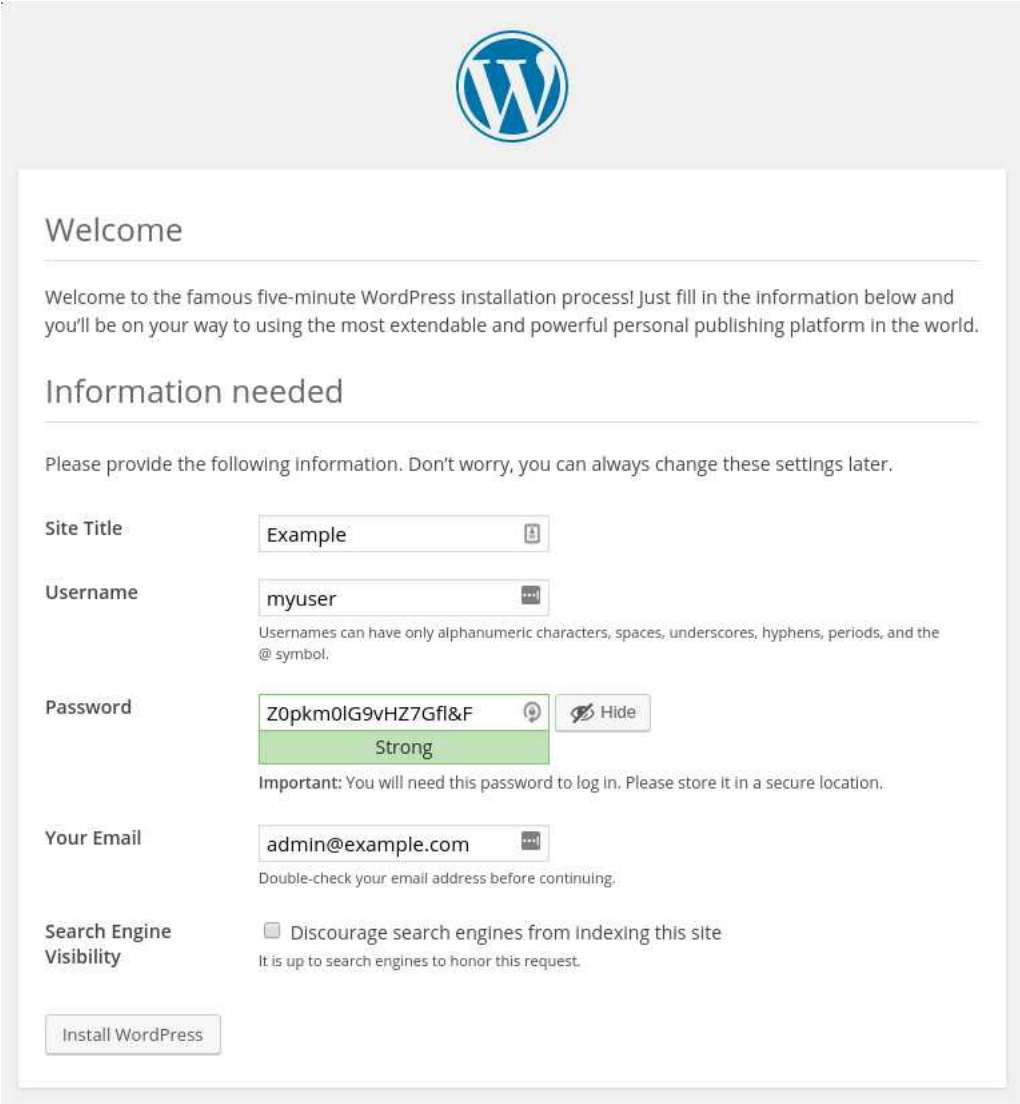
Selecione o idioma que você gostaria de usar:



Em seguida, você chegará à página de configuração principal.

Selecione um nome para seu site WordPress e escolha um nome de usuário. Recomenda-se escolher algo único e evitar nomes de usuário comuns como “admin” por fins de segurança. Uma senha forte será gerada automaticamente. Salve essa senha ou selecione uma senha forte alternativa.

Digite seu endereço de e-mail e selecione se deseja impedir que mecanismos de pesquisa façam a indexação do seu site:



WordPress logo

## Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

### Information needed

Please provide the following information. Don't worry, you can always change these settings later.

**Site Title**

**Username**   
Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

**Password**    
**Strong**  
Important: You will need this password to log in. Please store it in a secure location.

**Your Email**   
Double-check your email address before continuing.

**Search Engine Visibility**  Discourage search engines from indexing this site  
It is up to search engines to honor this request.

Quando clicar adiante, você será levado para uma página que solicita que você faça login:

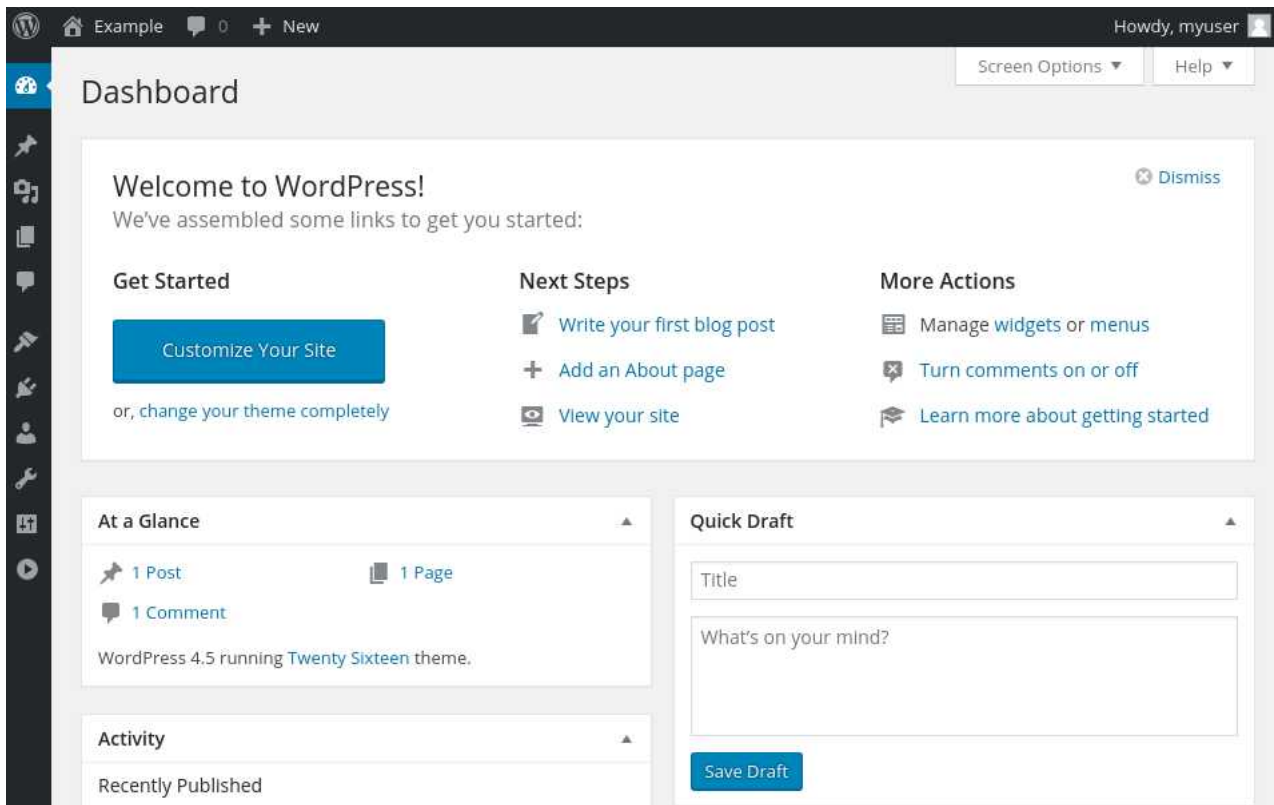
## Success!

WordPress has been installed. Thank you, and enjoy!

**Username** myuser

**Password** Your chosen password.

Assim que fizer login, será levado para o painel de administração do WordPress:



Neste momento, você pode começar a projetar seu site do WordPress! Se esta é a primeira vez que usa o WordPress, explore a interface um pouco para se familiarizar com seu novo CMS.

Para saber mais sobre como personalizar seu site usando o Wordpress leia a apostila [Empoderamento feminino através de ferramentas online \(HTML5, CSS3 e WORDPRESS\)](#). Nela você aprenderá o básico de HTML e CSS, que são linguagens que ajudarão a personalizar a aparência de seu site Wordpress.

## **3.3 CRIANDO NOSSA REDE SOCIAL COM HUMHUB**

## Introdução ao Humhub

HumHub é um software de rede social corporativa com uma estrutura construída para dar as ferramentas necessárias para tornar o trabalho em equipe fácil e bem sucedido, é uma excelente solução para quem pensa melhorar a comunicação interna do seu negócio ou para empreendedores que desenvolvam negócios sobre plataformas sociais.

O HumHub é todo em código aberto regida sob a licença GNU/GPLv3, com todos os sistemas de conexão entre usuários, mensagens e fotos que uma rede social completa precisa para ser eficiente. Há também módulos, como de calendário, enquetes e de notificações, que podem ser instalados separadamente, de acordo com a necessidade de cada projeto. O design da interface é bonito, simples e funcional.

Você pode experimentar a aplicação no próprio site da [Humhub](https://www.humhub.org/) antes de realizar a instalação.

## Instalando o Humhub

O Humhub também utiliza o MySQL para gerenciar e armazenar as informações do site e as do usuário. Então precisamos apenas criar um banco de dados e um usuário para o Humhub usar.

Para começar, faça login na conta root (administrativa) do MySQL emitindo este comando (note que este não é usuário root do seu servidor):

```
$ sudo mysql -u root -p
```

Será solicitada a senha que você configurou para a conta raiz do MySQL quando instalou o software.

Dentro banco de dados, vamos criar uma conta de usuário do MySQL separada que vamos usar exclusivamente para operar nosso novo banco de dados e em seguida criaremos um banco de dados exclusivo para o Humhub controlar.

```
mysql> CREATE DATABASE humhub DEFAULT CHARACTER SET utf8mb4 COLLATE  
utf8mb4_unicode_ci;  
mysql> CREATE USER 'humhubuser'@'localhost' IDENTIFIED WITH  
mysql_native_password BY 'password';  
mysql> GRANT ALL ON humhub.* TO 'humhubuser'@'localhost';  
mysql> FLUSH PRIVILEGES;
```

*Nota: cada instrução do MySQL deve terminar em um ponto e vírgula (;). Verifique para garantir que a instrução foi seguida (ou seja, o ponto e vírgula foi usado), caso estiver enfrentando algum problema.*

Saia do MySQL digitando:

```
mysql> EXIT;
```

Anote no caderno o nome do banco, usuário e a senha que criamos.

Agora vamos instalar algumas das extensões PHP que são requisitos para o Humhub executando:

```
$ sudo apt update
$ sudo apt install libapache2-mod-xsendfile php-fpm \
    php-imagick php-curl php-bz2 php-gd php-intl \
    php-mysql php-zip php-apcu-bc php-apcu php-xml php-ldap
```

Agora iremos substituir os seguintes parâmetros dentro do ficheiro de configuração PHP php.ini para os seguintes valores:

- ❏ max\_execution\_time = 30 -> max\_execution\_time = 600
- ❏ post\_max\_size = 8M -> post\_max\_size = 128M
- ❏ upload\_max\_filesize = 2M -> upload\_max\_filesize = 128M

```
$ sudo nano /etc/php/7.4/apache2/php.ini
```

```
...
max_execution_time = 600
...
post_max_size = 128M
...
upload_max_filesize = 128M
```

*Dica: Utilize a combinação **CRTL + R** no nano para localizar com facilidade os parâmetros.*

Precisaremos reiniciar o Apache para carregar essas novas extensões

```
$ sudo systemctl restart apache2
```

Videoaula: [Instalando módulos adicionais de PHP para o Humhub](#)



A seguir, utilizaremos o `/var/www/<dominio>/humhub` como o diretório root de nossa instalação do WordPress. Vamos criar com o comando:

```
$ sudo mkdir /var/www/<dominio>/humhub
```

Com nossos caminhos identificados, podemos seguir em frente e trabalhar com o `.htaccess` para que o Apache consiga lidar com as alterações de configuração para cada diretório. Abra o arquivo de configuração do Apache para seu site com um editor de texto como o nano.

```
sudo nano /etc/apache2/sites-available/rede.<dominio>.conf
```

Para permitir aos arquivos `.htaccess`, precisamos configurar a diretiva `AllowOverride` dentro de um bloco `Directory` apontando para nossa raiz de documentos. Adicione o seguinte bloco de texto dentro do bloco no seu arquivo de configuração, tendo certeza de usar o diretório root Web correto:

```
<VirtualHost *:80>
    ServerName rede.<dominio>
    ServerAdmin <seu_email>

    DocumentRoot /var/www/<dominio>/humhub

    <Directory /var/www/<dominio>/humhub/>
        Options -Indexes -FollowSymLinks
        AllowOverride All
    </Directory>

    <DirectoryMatch "/var/www/<dominio>/humhub/(\.|protected|
themes/\w+/views/uploads/file)">
        Order Deny,Allow
        Deny from all
    </DirectoryMatch>

    <FilesMatch "^\. ">
        Order Deny,Allow
```

```
        Deny from all
    </FilesMatch>

    <DirectoryMatch "/var/www/<dominio>/humhub/(static|uploads|
themes|assets)">
        Header set Cache-Control "max-age=172800, public"
    </DirectoryMatch>
</VirtualHost>
```

Quando você terminar, salve e feche o arquivo.

Antes de implementarmos as alterações que fizemos, verifique se não cometemos nenhum erro de sintaxe executando o seguinte teste.

```
$ sudo apache2ctl configtest
```

Ativar o host virtual, configurações adicionais do Apache2 e módulos necessários.

```
$ sudo a2enmod rewrite headers proxy_fcgi setenvif
$ sudo a2enconf php7.3-fpm
$ sudo a2ensite rede.<dominio>.conf
```

Reinicie o Apache para implementar as alterações. Neste momento, certifique-se de fazer a reinicialização, mesmo que tenha reiniciado anteriormente neste tutorial.

```
$ sudo systemctl restart apache2
```

Videoaula: [Configurando o servidor virtual para o Humhub no Apache](#)

Em seguida, vamos baixar e instalar o Humhub no servidor web. Substitua o <version> pela versão listada nessa página <https://www.humhub.com/en/download> (Na data da publicação desta apostila a versão disponível é **1.6.4**):

```
$ cd ~
$ curl -O https://www.humhub.com/download/package/humhub-
<version>.tar.gz
$ tar xvfz humhub-<version>.tar.gz
$ sudo cp -a humhub-<version>/ /var/www/<dominio>/humhub/
$ sudo chown -R www-data:www-data /var/www/<dominio>/humhub
```

Videoaula: [Fazendo o download do Humhub](#)

Agora vamos novamente alterar o nosso arquivo hosts em nosso computador, como fizemos para o blog . Dessa vez iremos apenas acrescentar o endereço rede.<dominio\_da\_servidora> na mesma linha do endereço do nosso domínio:

```
<ip_da_servidora> <dominio_da_servidora> blog.<dominio_da_servidora>
rede.<dominio_da_servidora>
```

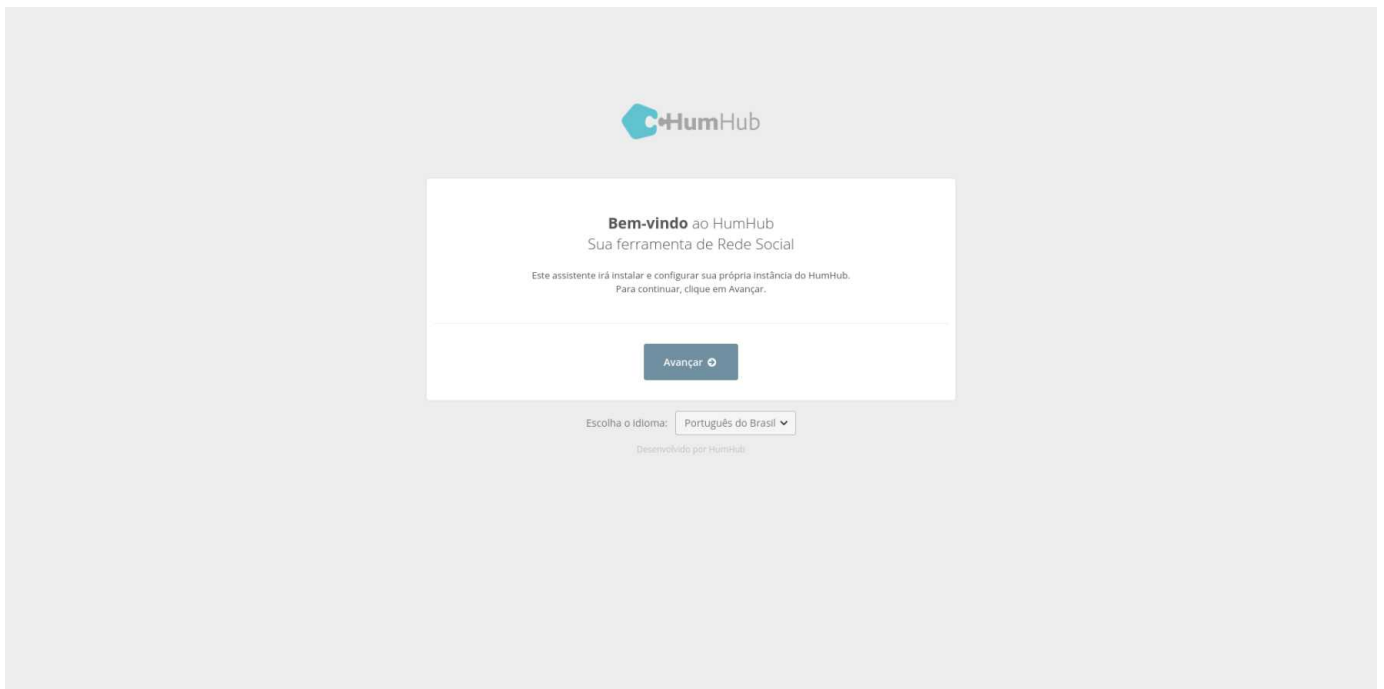
Exemplo:

```
127.0.0.1 localhost
```

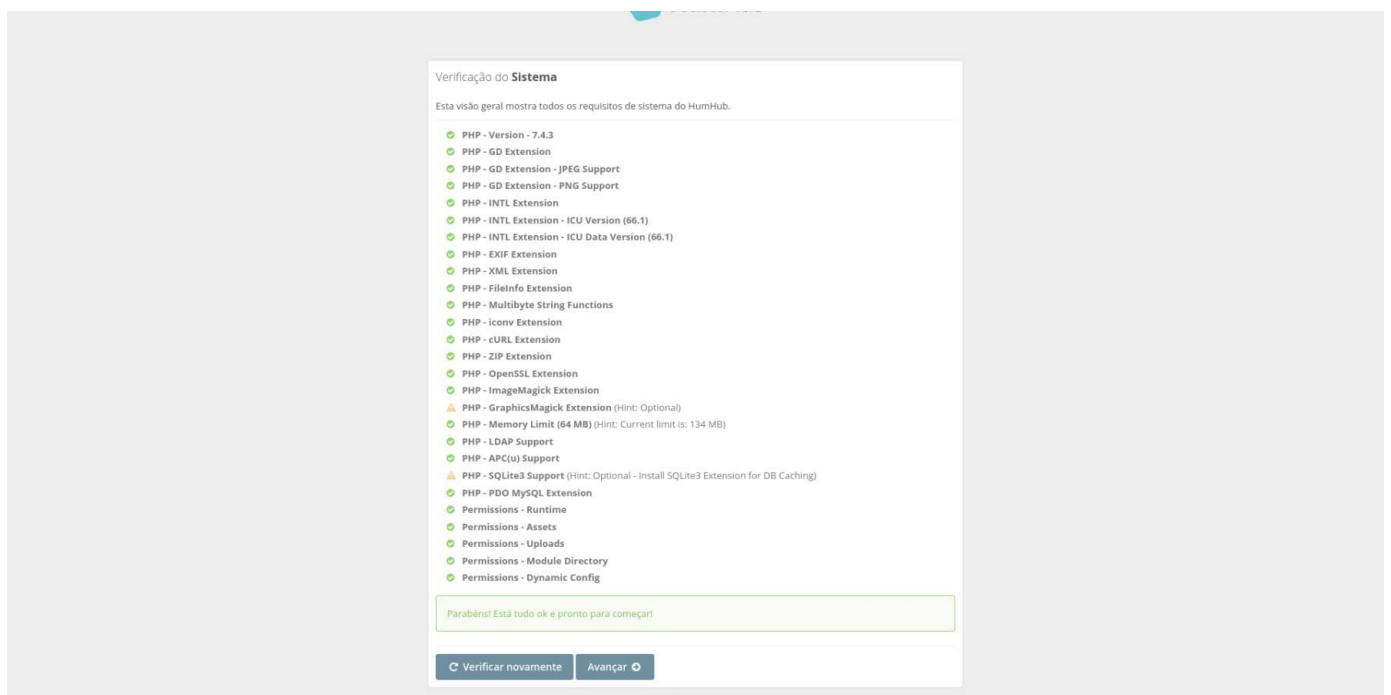
```
192.168.15.27 dandara.net www.dandara.net blog.dandara.net
rede.dandara.net
```

Acesse a url <http://rede.<dominio>> e vamos iniciar a instalação. Usaremos as configurações de banco de dados que anotamos no caderno.

Na tela inicial, escolha o idioma e clique em **Avançar**.



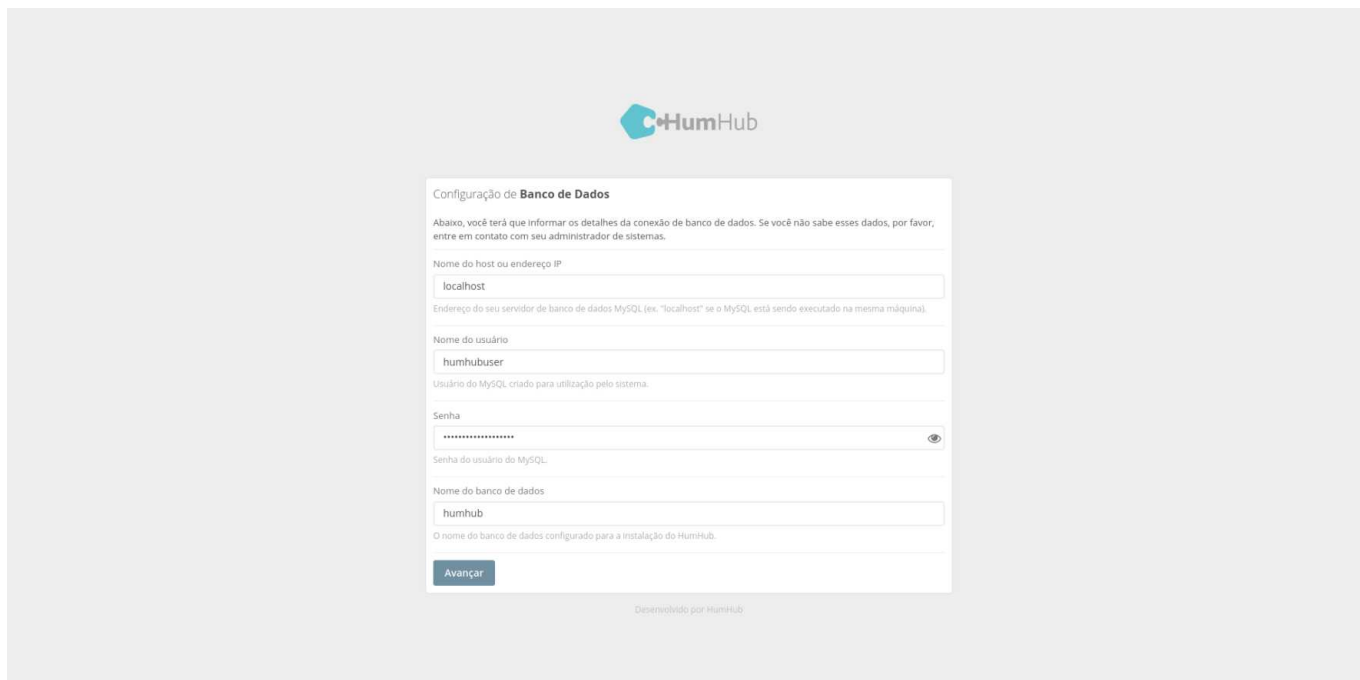
O HumHub irá executar um teste do sistema. Clique em **Avançar**.



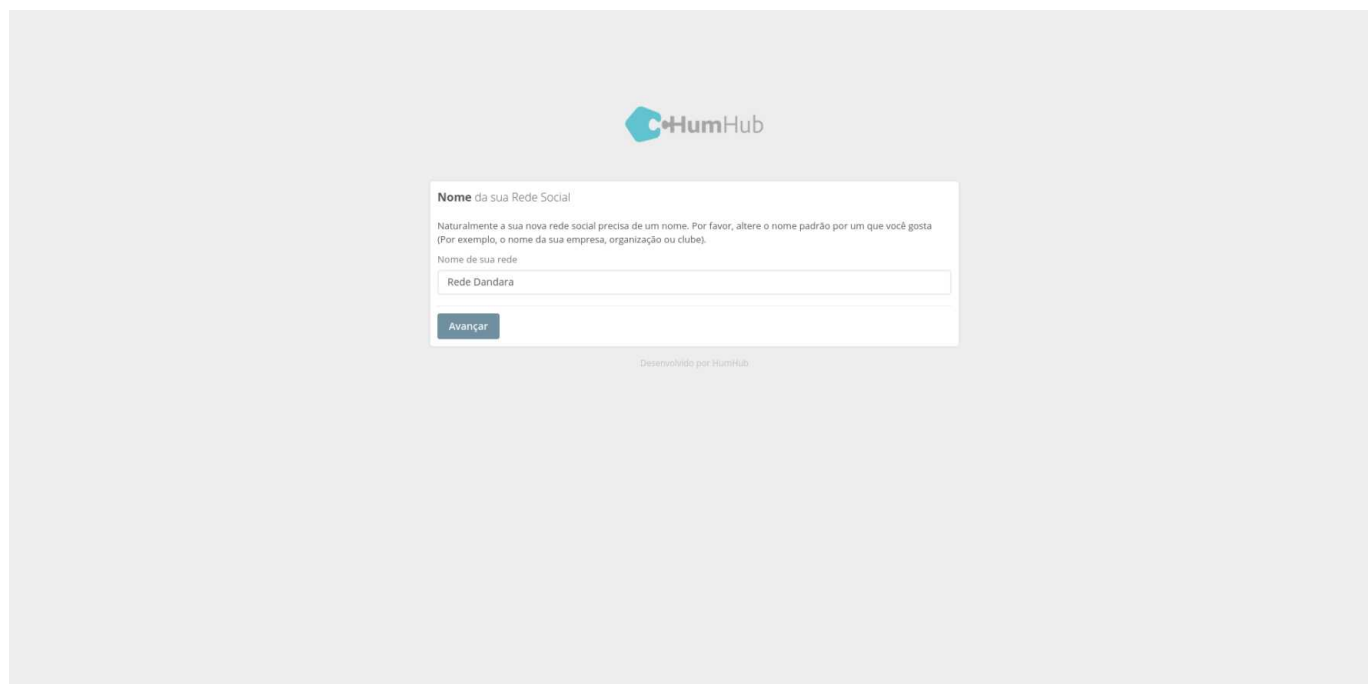
Na tela seguinte, entre com os dados do servidor:

- 🚩 Nome do servidor de banco: localhost
- 🚩 Nome do utilizador: humhubuser
- 🚩 Palavra-passe:
- 🚩 Name of Database: humhub

Clique em **Avançar**.



Agora vamos colocar o nome da nossa Rede Social, invento um nome e clique em **Avançar**.



Na tela que se abre, insira o nome de sua rede social e, nas telas seguintes, escolha um modelo pré-definido de configuração, defina a política de acesso de novos usuários, alguns módulos e introduza as credenciais do administrador. Lembre-se de anotar esses dados no caderno.



### Configuração

Para simplificar a configuração, temos configurações predefinidas para os casos de uso mais comuns, com diferentes opções de módulos e configurações. Você pode ajustá-los durante a próxima etapa.

#### Quero usar o HumHub para: \*

- Minha empresa (Intranet Social / Gerenciamento de Projetos)
- Minha instituição educacional (escola, universidade)
- Meu clube
- Minha comunidade
- Ignorar esta etapa, eu quero configurar tudo manualmente

Próximo

Desenvolvido por HumHub



### Configurações de Segurança

Aqui você pode decidir como novos usuários não registrados podem acessar o HumHub.

- O usuário externo pode se inscrever (O formulário de registro será exibido no Login)
- Usuários recém-registrados precisam ser ativados por um administrador primeiro
- Permitir o acesso de usuários não registrados ao conteúdo público (acesso de convidado)
- Os membros registrados podem convidar novos usuários por e-mail
- Permitir amizades entre membros

Próximo

Desenvolvido por HumHub



#### Módulos recomendados

HumHub é muito flexível e pode ser ajustado e/ou expandido para várias aplicações diferentes graças aos seus diferentes módulos. Os próximos módulos são apenas alguns exemplos e aqueles que consideramos mais importantes para o aplicativo escolhido.

Você pode sempre instalar ou remover módulos posteriormente. Você pode encontrar mais módulos disponíveis após a instalação na área de administração.

- Birthday Widget**  
Add a widget to the dashboard showing upcoming birthdays.
- Polls**  
Easy to use poll system.
- Mail**  
A private messaging system for direct communication.
- Most Active Users**  
Add a widget to the dashboard showing the most active users.

Próximo

Desenvolvido por HumHub



#### Conta de Administrador

Você está quase pronto. Nesta etapa, você deve preencher o formulário para criar uma conta de administrador. Com esta conta você pode gerenciar toda a rede.

Nome de usuário \*

E-mail

Nova senha \*

Confirmar nova senha \*

Geral

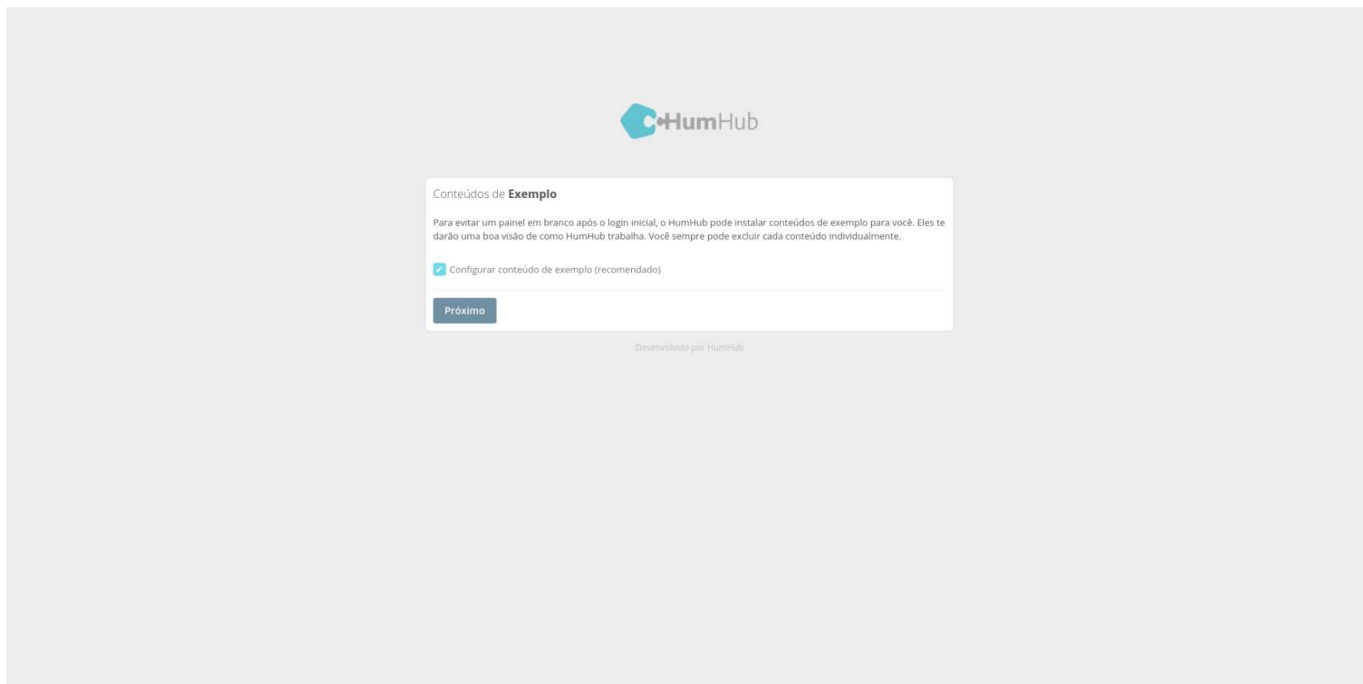
Primeiro nome \*

Último nome \*

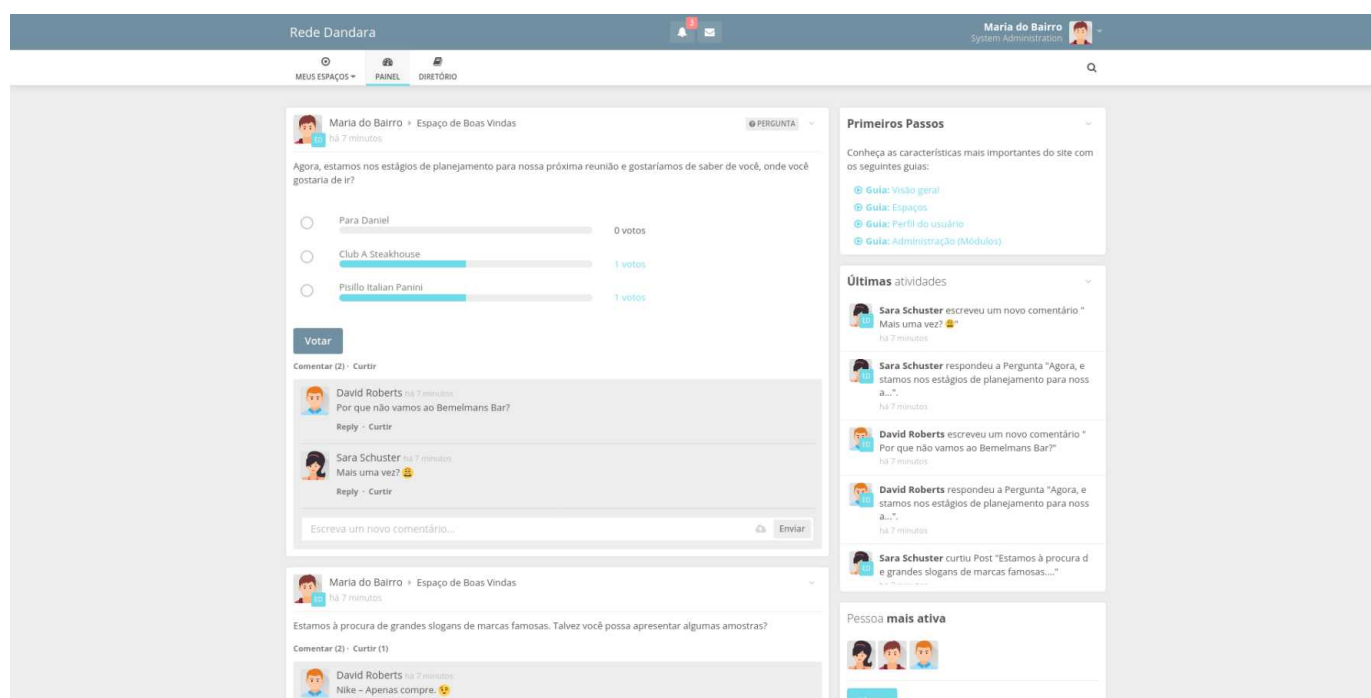
Criar conta de Administrador

Desenvolvido por HumHub

Por fim, escolha criar o conteúdo de Exemplo e clique em **Avançar**.



Finalizamos a criação da nossa rede! Clique no botão Entrar e faça Login, com o nome de usuário e senha que criou, clicando no botão Sign in / up que se encontra no topo direito da página.



Para configurarmos a nossa rede, é disponibilizado alguns guias no bloco *Primeiros Passos* que esta na coluna da direita. Esse guia vai demonstrar todas as funções básicas do Humhub para poder utilizar e também personalizar a nossa rede.



## **3.4 CRIANDO NOSSA LOJA COM O PRESTASHOP**

# Introdução ao PrestaShop

O PrestaShop é uma plataforma de gestão de e-commerce gratuita e repleta de funcionalidades essenciais a quem cuida de uma loja virtual. Essa solução funciona como um CMS totalmente focado em estruturar e administrar varejos online sem complicação e altamente personalizáveis.

A primeira e grande vantagem do Prestashop é o fato de ser uma ferramenta Opensource, ou seja, de código aberto. Hoje, mais de 150.000 lojas online estão suportadas por essa plataforma e a comunidade cresce a cada dia, proporcionando assim atualizações significativas no sistema de forma periódica e organizada. Outro fator positivo é o fato do Prestashop ter suporte à tradução do Português do Brasil e possui suporte a diversos módulos de pagamento como Pagseguro, Paypal, MOIP, entre outros.

Veja aqui a lista de funcionalidades disponíveis:

<https://www.prestashop.com/pt/funcionalidades>

# Instalando o PrestaShop

O Prestashop, assim como os outros, também funciona em ambiente LAMP. Então vamos configurar o banco de dados e o Virtual Host.

Para começar, faça login na conta root (administrativa) do MySQL emitindo este comando (note que este não é usuário root do seu servidor):

```
$ sudo mysql -u root -p
```

Será solicitada a senha que você configurou para a conta raiz do MySQL quando instalou o software.

Dentro banco de dados, vamos criar uma conta de usuário do MySQL separada que vamos usar exclusivamente para operar nosso novo banco de dados e em seguida criaremos um banco de dados exclusivo para o Humhub controlar.

```
mysql> CREATE USER 'prestashopuser'@'localhost' IDENTIFIED BY
'<SUA_SENHA>';
mysql> CREATE DATABASE prestashop;
mysql> GRANT ALL ON prestashop.* TO 'prestashopuser'@'localhost';
mysql> FLUSH PRIVILEGES;
mysql> EXIT;
```

Videoaula: [Criando o Banco de Dados do PrestaShop no MySQL](#)

Anote as credenciais e vamos instalar algumas das extensões PHP que são requisitos para o Humhub executando:

```
$ sudo apt update
$ sudo apt install php-common php-mbstring php-gd php-intl \
php-xml php-mysql php-zip php-curl php-xmlrpc
```

A seguir, utilizaremos o `/var/www/<dominio>/prestashop` como o diretório root de nossa instalação do WordPress. Vamos criar com o comando:

```
$ sudo mkdir /var/www/<dominio>/prestashop
```

Com nossos caminhos identificados, podemos seguir em frente e trabalhar com o `.htaccess` para que o Apache consiga lidar com as alterações de configuração para cada diretório.

Abra o arquivo de configuração do Apache para seu site com um editor de texto como o nano.

```
sudo nano /etc/apache2/sites-available/loja.<dominio>.conf
```

Para permitir aos arquivos `.htaccess`, precisamos configurar a diretiva `AllowOverride` dentro de um bloco `Directory` apontando para nossa raiz de documentos. Adicione o seguinte bloco de texto dentro do bloco no seu arquivo de configuração, tendo certeza de usar o diretório root Web correto:

```
<VirtualHost *:80>
    ServerName loja.<dominio>
    ServerAdmin <seu_email>

    DocumentRoot /var/www/<dominio>/prestashop
    Alias /loja/ /var/www/<dominio>/prestashop

    <Directory /var/www/<dominio>/prestashop/>
        AllowOverride All
```

```
</Directory>  
</VirtualHost>
```

Quando você terminar, salve e feche o arquivo

Antes de implementarmos as alterações que fizemos, verifique se não cometemos nenhum erro de sintaxe executando o seguinte teste.

```
$ sudo apache2ctl configtest
```

Ativar o host virtual, configurações adicionais do Apache2 e módulos necessários.

```
$ sudo a2enmod rewrite headers proxy_fcgi setenvif  
$ sudo a2enconf php7.3-fpm  
$ sudo a2ensite loja.<dominio>.conf  
$ sudo systemctl reload apache2
```

Reinicie o Apache para implementar as alterações. Neste momento, certifique-se de fazer a reinicialização, mesmo que tenha reiniciado anteriormente neste tutorial.

```
$ sudo systemctl restart apache2
```

Videoaula: [Instalando módulos adicionais do PHP e configurando o servidor virtual para o Humhub no Apache](#)

Em seguida, vamos baixar e instalar o Prestashop no servidor web. Substitua o <version> pela versão listada nessa página <https://www.prestashop.com/pt> baixar (Na data da publicação desta apostila a versão disponível é 1.7.6.8):

```
$ cd ~
$ curl -O
https://download.prestashop.com/download/releases/
prestashop_<version>
.zip
$ sudo apt-get install unzip
$ unzip prestashop_<version>.zip
$ sudo unzip prestashop.zip -d /var/www/<dominio>/prestashop
$ sudo chown -R www-data:www-data /var/www/<dominio>/prestashop
$ sudo find /var/www/<dominio>/prestashop/ -type d -exec chmod 755 {}
\;
$ sudo find /var/www/<dominio>/prestashop/ -type f -exec chmod 644 {}
\;
```

Videoaula: [Fazendo o download do PrestaShop](#)

Agora vamos alterar pela ultima vez o nosso arquivo hosts em nosso computador, como fizemos para o outros dois subdomínios. Dessa vez iremos apenas acrescentar o endereço loja.<dominio\_da\_servidora> na mesma linha do endereço do nosso domínio:

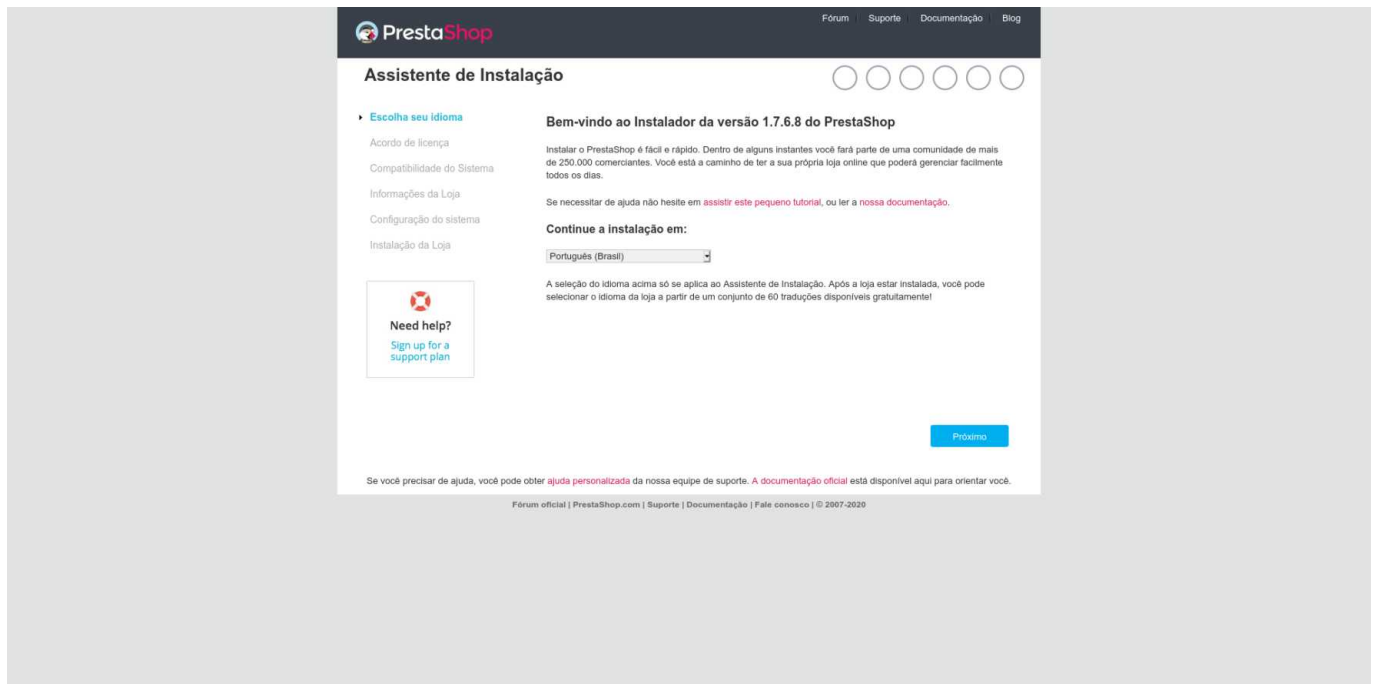
```
<ip_da_servidora> <dominio_da_servidora> blog.<dominio_da_servidora>
rede.<dominio_da_servidora> loja.<dominio_da_servidora>
```

Exemplo:

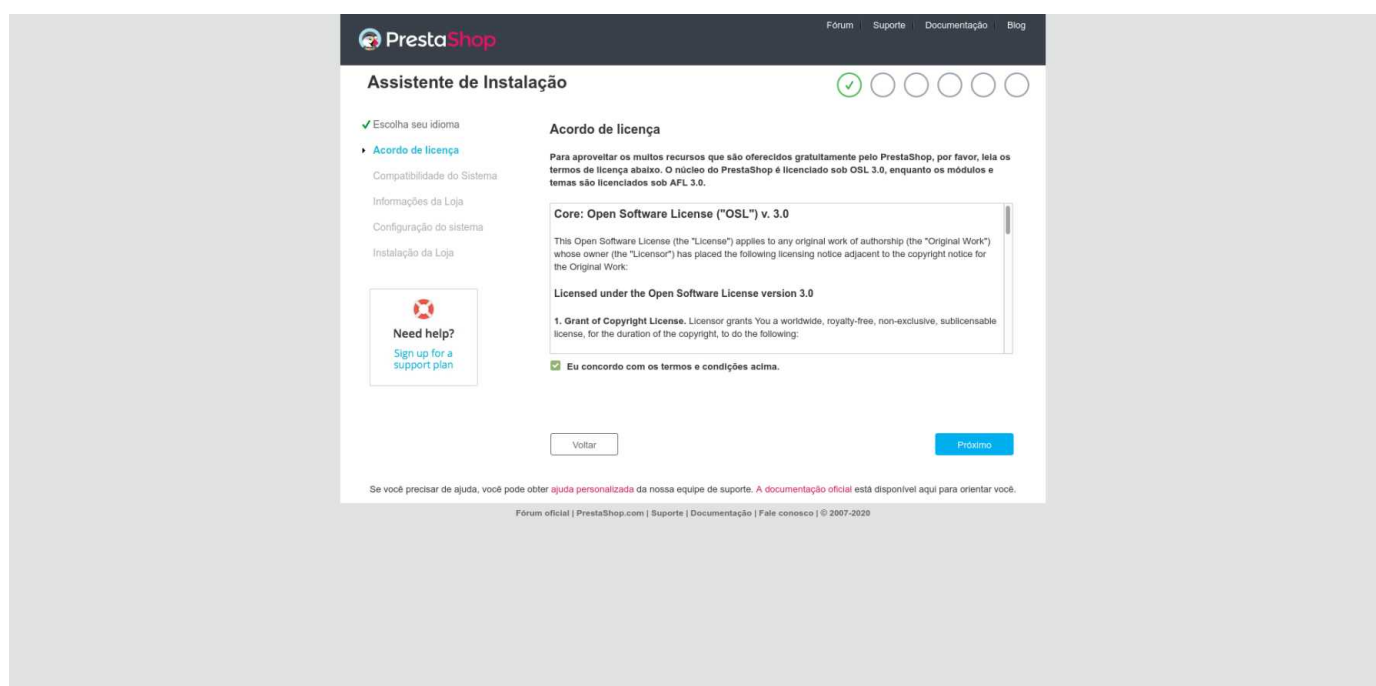
```
127.0.0.1 localhost
```

```
192.168.15.27 dandara.net www.dandara.net blog.dandara.net
rede.dandara.net loja.dandara.net
```

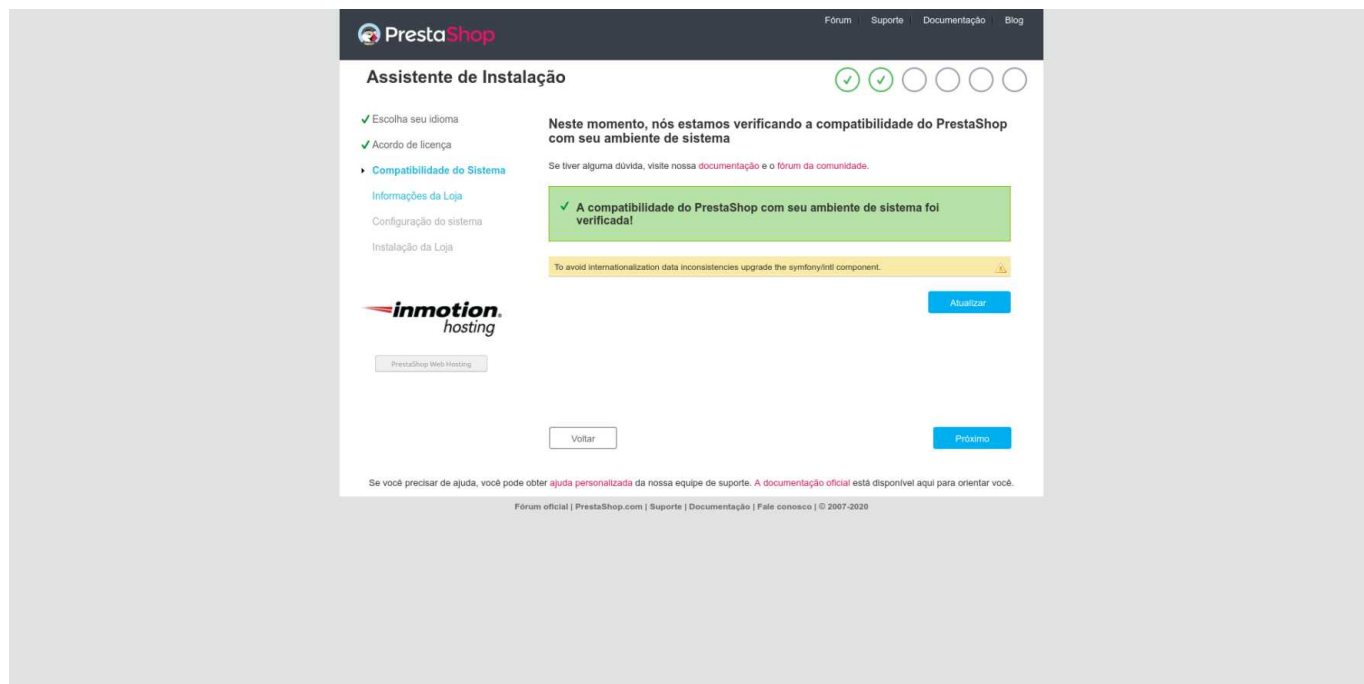
Acesse a url <http://loja.<dominio>> e vamos iniciar a instalação. Usaremos as configurações de banco de dados que anotamos no caderno. Você terá uma tela semelhante à que se encontra abaixo para alterar o idioma:



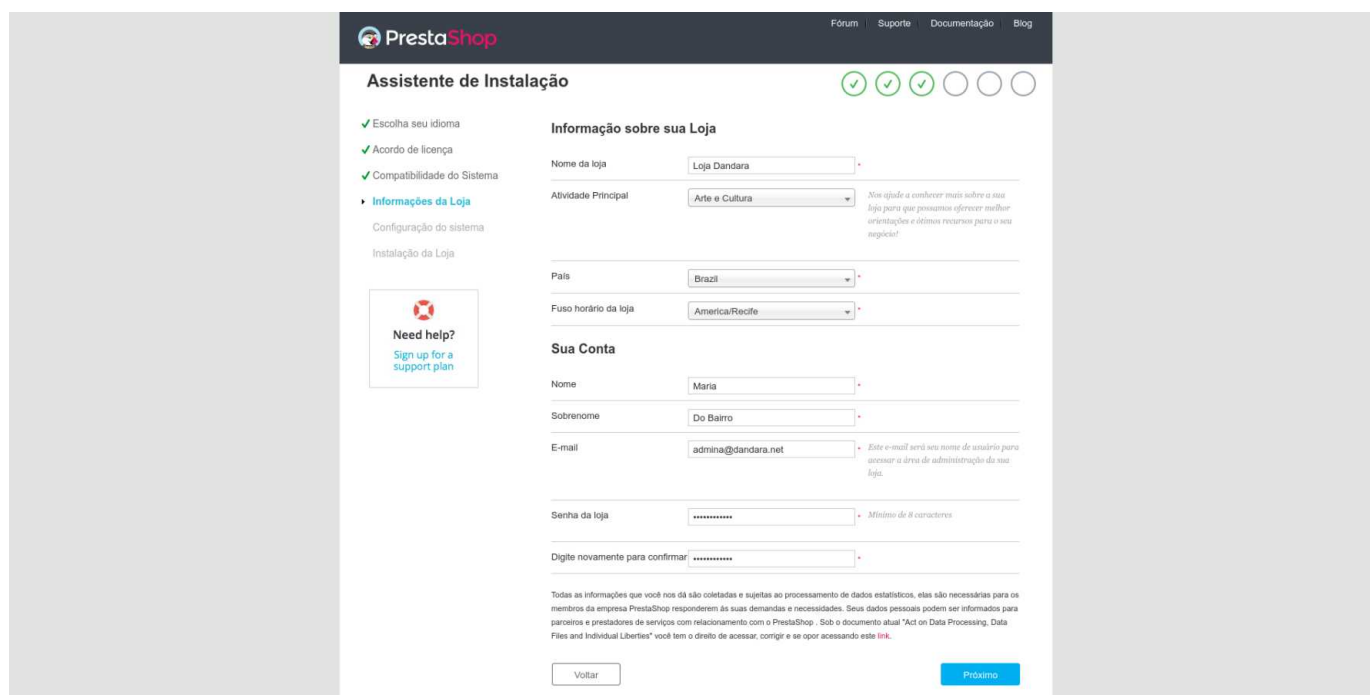
Clique em 'Próximo' para prosseguir no lado inferior direito da tela. Aceite os termos na próxima tela e clique em 'Próximo':



O script de instalação verificará se seu ambiente é compatível com o PrestaShop, como mostrado abaixo. Em seguida, clique em 'Próximo' para prosseguir.



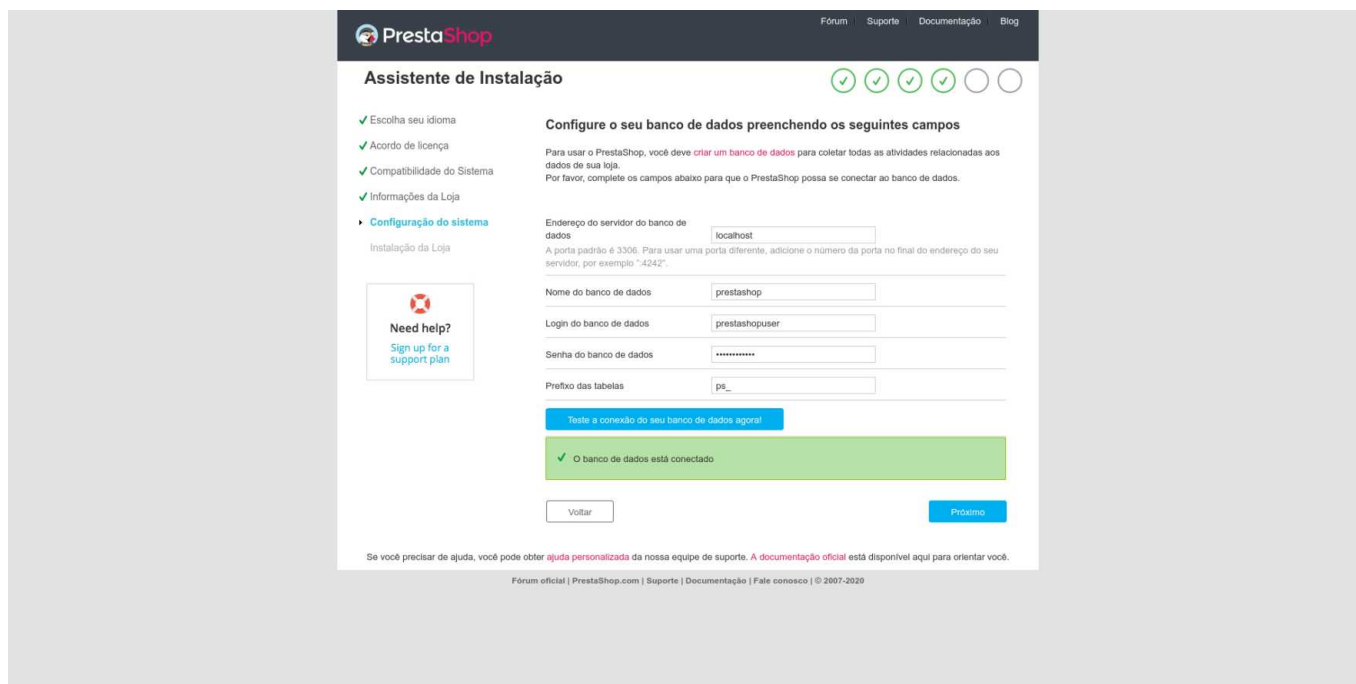
Em seguida, digite os detalhes de sua loja como mostrado abaixo e clique em 'Próximo':



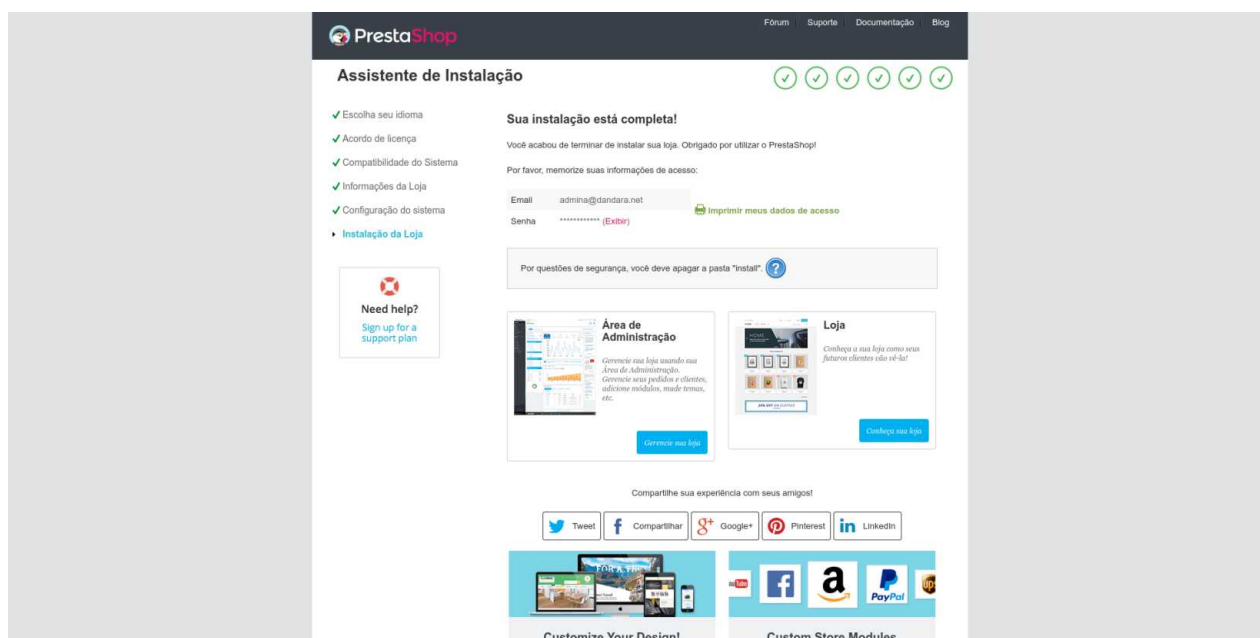


Finalmente, digite os detalhes do banco de dados que você criou acima na próxima tela e clique em 'Next' (Avançar):

- Nome do servidor de banco: localhost ou 127.0.0.1
- Name of Database: prestashop
- Nome do utilizador: prestashopuser
- Palavra-passe:



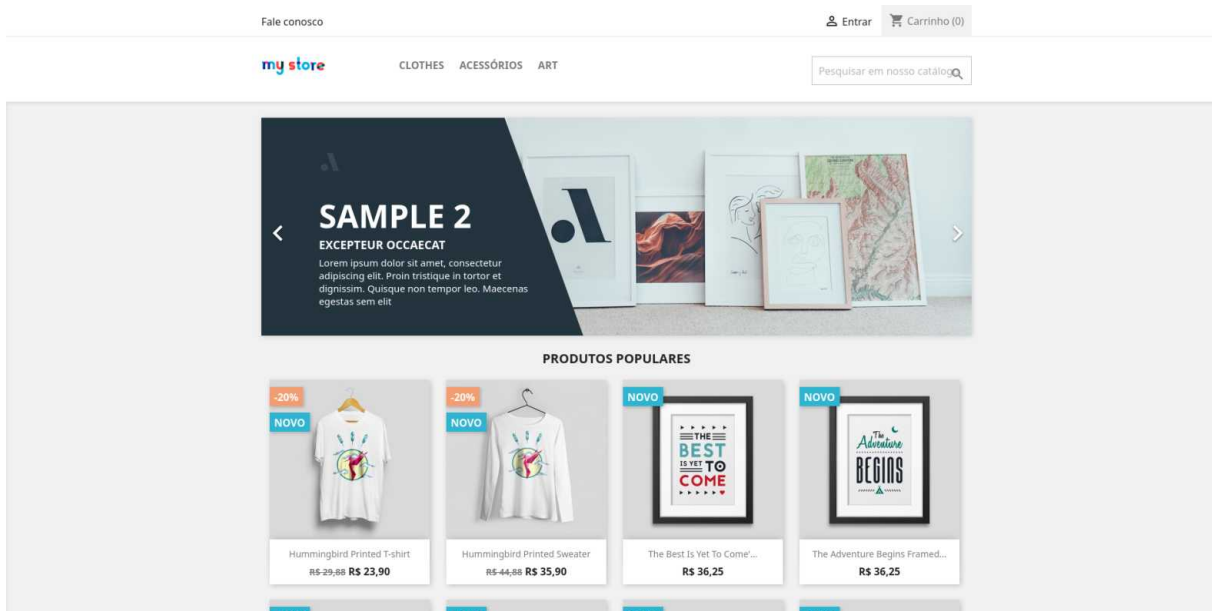
Se os detalhes estiverem corretos e não houver erros, sua instalação será concluída e você poderá agora visualizar ou logar na seção administrativa de sua loja PrestaShop.



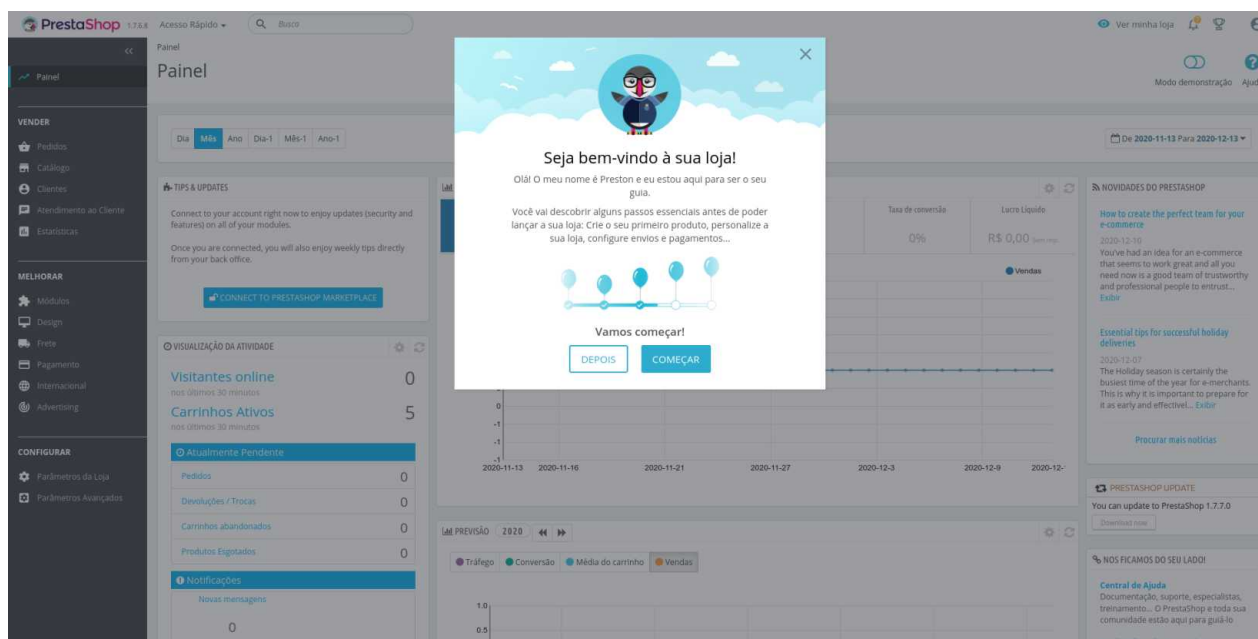
Por fim, apague a pasta com os scripts de instalação para destravarmos a Área de Administração:

```
$ sudo rm -r /var/www/dandara.net/prestashop/install/
```

Clique no botão Conheça Sua loja e veja ela funcionando:



Clique no botão Gerencie sua loja e digite seu email e senha para logar. Abrirá uma janela que irá te guiar para conhecer o painel de gerenciamento da loja.



No YouTube é possível encontrar vários tutoriais de como personalizar sua loja, configurar meios de pagamento e aulas completas de administração do Prestashop.

# Conclusão

Chegamos ao final de nossa guia. Até aqui aprendemos sobre Redes e Internet, Servidoras e o funcionamento disso tudo para nos prover serviços. Criamos uma servidora virtual e instalamos um sistema operacional dedicado a ela. Também aprendemos alguns comandos de Linux para melhor administrá-la. Passamos a nos conectar remotamente a nossa servidora virtual, criamos um domínio local e instalamos programas PHP em nossa servidora Web. Com tudo isso pudemos criar uma Blog, uma Rede Social e uma Lojinha Virtual.

Com tudo isso a gente conclui: Nossa servidora está no ar!

Demais não é? Ao entrar em `http://<dominio>` você verá que os links já estarão em funcionamento.

É possível seguir os mesmos passos da configuração do LAMP em uma máquina na cloud que venha a contratar. No caso de hospedagens compartilhadas as etapas de **download** e **configuração** permanecem, mas com a facilidade da transferência via FTP e ao painel de controle que nos permitirá criar o banco de dados, configurar os subdomínios e tudo mais de forma simples.

É importante que você siga todos os passos técnicos a risca, então caso algo não funcione retorne aos capítulos e verifique os comandos e as configurações. Em caso de dúvidas, dê uma olhada nos vídeos e imagens apresentados nos capítulos, lá você encontrará o passo a passo. Se a dúvida persistir, pesquise em outros sites que tenham uma linguagem que você compreenda melhor, existem muitos na internet ou entre em contato conosco **[contato@marialab.org](mailto:contato@marialab.org)**

Por fim, agradecemos muito por terem chegado até aqui! Esta Guia foi feita com muito carinho e queremos que ela seja apenas um ponto de partida para que você siga nos aprendizados sobre Infraestrutura Feminista. Links, publicações e imagens utilizadas nessa Guia foram selecionados a partir de critérios técnicos

e pedagógicos para um público iniciante no contato com servidoras, mas com conhecimentos básicos em informática.

Desejamos que esta Guia seja um suporte para que você monte a sua servidora em diversas plataformas, que exerça autonomia do armazenamento e distribuição de conteúdos nas redes e festeje demais com a comunidade feminista o surgimento de redes comunitárias, novos fóruns virtuais, redes, revistas digitais, portais de conteúdo, serviços gratuitos, wikis e tudo mais que você puderem inventar!

Conte-nos sobre seus avanços! Ficaremos felizes em celebrar cada novo projeto no ar!

# Construído por mãos feministas

**Texto:** Camilla "Lobinha" Gomes

**Coordenação pedagógica:** Luciana Ferreira

**Revisão:** Sueli Feliziani

**Diagramação:** Mariana Oliveira

**Apoio ao projeto:** Fernanda Chan, Daniela Araújo, Laila Almeida

**Realização**

**maria**  
[lab]

**Apoio**



Este manual está licenciado com uma [Licença Creative Commons -Atribuição Não Comercial Compartilha Igual 4.0 Internacional](#)

