

Guia Ágil

TechGuide - Alura

DevOps

Nível 1

☐ Linux - Fundamentos:

- Linux é um termo popularmente empregado para se referir a sistemas operacionais que utilizam o Kernel Linux. As distribuições incluem o Kernel Linux, além de softwares de sistema e bibliotecas.
- Conhecer o sistema de diretórios do Linux
- Compactar e descompactar arquivos
- Editar arquivos no console com o VI
- Gerenciar os processos rodando na máquina
- Conhecer as variáveis de ambiente e o PATH
- Gerenciar pacotes
- Realizar comunicação remota com o SSH e SCP

☐ Git e GitHub - Fundamentos:

- Git é um sistema de controle de versão distribuído gratuito e de código aberto projetado para lidar com tudo, desde projetos pequenos a muito grandes com velocidade e eficiência.
- GitHub é um serviço de hospedagem para desenvolvimento de software e controle de versão usando Git.

- Criar um repositório
- Clonar um repositório
- Fazer commit, push e pull de e para o repositório
- Reverter um commit
- Criar branches e pull requests
- Lidar com merge e conflitos

☐ **HTTP - Fundamentos:**

- HTTP significa Hyper Text Transfer Protocol. A comunicação entre computadores cliente e servidores web é feita enviando solicitações HTTP e recebendo respostas HTTP.
- Entender a diferença dos verbos HTTP
- Testar os requests e ver os status codes no navegador
- Saber fazer uma requisição HTTP na linha de comando com WGET
- Baixar uma imagem com WGET
- Fazer um post

☐ **Linha de comando - Fundamentos:**

- CLI é um programa de linha de comando que aceita entradas de texto para executar funções do sistema operacional.
- Conhecer os principais comandos

☐ **Cloud - Fundamentos:**

- Cloud, ou computação em nuvem é a distribuição de serviços de computação pela Internet usando um modelo de preço pago conforme o uso. Uma nuvem é composta de vários recursos de computação, que abrangem desde os próprios computadores (ou instâncias, na terminologia de nuvem) até redes, armazenamento, bancos de dados e o que estiver em torno deles. Ou seja, tudo o que normalmente é necessário para montar o equivalente a uma sala de servidores, ou mesmo um data center completo, estará pronto para ser utilizado, configurado e executado.

- Conhecer a diferença entre IaaS, PaaS e SaaS
- Conhecer os maiores provedores de cloud
- Especializar-se em algum provedor

☐ **Shell Scripting:**

- Automatizar tarefas repetitivas utilizando as linguagens disponibilizadas pelo terminal.
- Criação de tarefas com execução automática
- Automatizar a configuração e instalação de aplicações em novos sistemas
- Usar o PowerShell

☐ **Redes de Computadores - Fundamentos:**

- Rede de computadores é uma malha que interliga milhares de sistemas computacionais para a transmissão de dados. Também conhecidos como nós, esses dispositivos interconectados enviam, recebem e trocam tráfego de dados, voz e vídeo, graças ao hardware e software que compõe o ambiente.
- Configurações de redes são essenciais para que seja possível acessar uma aplicação, principalmente se ela estiver na nuvem.
- Entenda melhor os componentes de rede e a suas atribuições
- Diferenciar os serviços disponibilizados a nível de rede
- Saber as diferenças entre as camadas de rede
- Saber o que é servidor web, proxy reverso e load balancer

Nível 2

☐ **Contêineres:**

- Os contêineres são pacotes de software que contêm todos os elementos necessários para serem executados em qualquer ambiente. Gerenciamento de contêineres é uma área crucial na computação em nuvem e DevOps, que envolve o uso de tecnologias para automatizar o processo de criação, implantação, escalonamento e monitoramento de contêineres. Contêineres

são unidades de software padronizadas que permitem aos desenvolvedores empacotar todas as dependências de um aplicativo (código, bibliotecas, configurações, etc.) em um único pacote. Isso permite que o aplicativo seja executado de forma consistente em qualquer ambiente de infraestrutura.

- A tecnologia de contêineres, como exemplificada pelo Docker, fornece um ambiente consistente e portátil para desenvolvimento, teste e implantação de aplicativos, o que é vital para o trabalho eficiente de engenharia de dados. Além disso, o Kubernetes, um sistema de orquestração de contêineres, permite o gerenciamento, a automação e a escalabilidade de aplicações baseadas em contêineres em ambientes de produção. Dominar esses conceitos e tecnologias possibilita a engenheiros de dados construir e manter pipelines de dados eficientes e confiáveis.
- O Kubernetes (também conhecido como k8s ou kube) é uma plataforma de orquestração de containers open source que automatiza grande parte dos processos manuais necessários para implantar, gerenciar e escalar aplicações em containers.
- Isolar seu software para funcionar independentemente
- Implantar software em clusters
- Modularizar seu sistema em pacotes menores
- Conhecer a plataforma Docker
- Conhecer Kubernetes

☐ **Infraestrutura como código (IaC):**

- A infraestrutura como código permite com que seja criada e configurada toda a infraestrutura de uma aplicação de maneira automática.
- Existem muitas ferramentas que têm a função de preparar uma infraestrutura através de código, porém algumas se destacam cada vez mais, principalmente o Terraform e o Ansible
- Descrever a infraestrutura atual e criar a sua documentação
- Atualizar a infraestrutura de maneira automática
- Configurar máquinas e serviços sem utilização de painéis proprietários dos sistemas de cloud

- Conhecer o Ansible
- Conhecer o Terraform

☐ **Entrega e integração contínuas (CI/CD):**

- CI/CD é a abreviação de Continuous Integration/Continuous Delivery, traduzindo para o português "entrega e integração contínuas". Trata-se de uma prática de desenvolvimento de software que visa tornar a integração de código mais eficiente por meio de builds e testes automatizados.
- Automatizar a integração de código entre varias partes da equipe se tornou cada vez mais importante, ja que assim é possível acelerar o desenvolvimento e diminuir o tempo de entrega de software.
- Executar testes automatizados da aplicação para verificar seu funcionamento.
- Realizar a entrega de atualizações de forma automática e com segurança.
- Realizar testes de conexão e testes de carga para evitar que a aplicação apresente problemas ao ser atualizada.

☐ **SQL - Fundamentos:**

- SQL (Structured Query Language, traduzindo, Linguagem de Consulta Estruturada) é uma linguagem de programação padronizada que é usada para gerenciar bancos de dados relacionais e realizar várias operações sobre os dados neles contidos.
- Conhecer os comandos mais comuns do SQL
- Usar SELECT para consultar uma tabela
- Usar INSERT para inserir dados em uma tabela
- Usar UPDATE para atualizar uma tabela
- Usar DELETE para remover dados de uma tabela
- Usar JOIN para conectar os dados de múltiplas tabelas
- Conhecer as cláusulas (FROM, ORDER BY, etc)

Nível 3

☐ Engenharia de Confiabilidade de Sites (SRE):

- A Engenharia de Confiabilidade de Sites (SRE) é a prática de usar ferramentas de software para automatizar tarefas de infraestrutura de TI, como gerenciamento de sistemas e monitoramento de aplicativos.
- Coletar métricas sobre a saúde da aplicação e da máquina em que ela se encontra
- Montar dashboards para analisar todas as informações coletadas

☐ Arquitetura de Microsserviços:

- Microsserviços são uma abordagem de arquitetura na qual o software consiste de pequenos serviços independentes que se comunicam entre si e são organizados de acordo com seus domínios de negócio.
- Aprender o conceito de arquitetura planejada para microsserviços
- Realizar a comunicação usando APIs
- Melhorar a escalabilidade de um sistema

☐ Firewalls, IDS e IPS:

- Definições e diferenças entre dispositivos de segurança de rede, firewalls, sistemas de prevenção de intrusão (IPS) e sistemas de detecção de intrusão (IDS).
- Um firewall permite o tráfego dependendo de um conjunto de regras que foram configuradas. Ele é baseado nos endereços de origem, destino e porta. Um firewall pode negar qualquer tráfego que não satisfaça os critérios especificados.
- IDS são dispositivos de sistema de monitoramento passivo que monitoram o tráfego de rede à medida que viajam pela rede, comparam padrões de assinatura, e acionar um alarme se for detectada atividade suspeita ou ameaça de segurança conhecida.
- O IPS é um dispositivo ativo que impede ataques, bloqueando-os.

☐ Internet das Coisas (IoT) - Fundamentos:

- Quase todos os objetos agora podem ser conectados à Internet, de máquinas de lavar a componentes sofisticados de motor de avião, a itens orgânicos, como plantações e vacas. A capacidade de conectar coisas e capturar dados úteis está transformando as empresas de todos os setores e abrindo portas para novas especializações de carreira. Descubra a base desse novo campo com atividades práticas para conectar os sensores com segurança aos serviços em nuvem nas redes IP e coletar dados em um sistema de IoT (Internet das Coisas) de ponta a ponta.
- Como rodar seu primeiro sistema em um Arduino

☐ **DevSecOps:**

- DevSecOps significa desenvolvimento, segurança e operações. É uma abordagem à cultura, automação e design de plataforma que integra a segurança como uma responsabilidade compartilhada durante todo o ciclo de vida da TI.

Habilidade Auxiliar: Back-end

☐ **Conceitos de Orientação a Objetos:**

- A Programação Orientada a Objetos é um paradigma de programação de software baseado na composição e interação entre diversas unidades chamadas de 'objetos' e as classes, que contêm uma identidade, propriedades e métodos). Ela é baseada em quatro componentes da programação: abstração digital, encapsulamento, herança e polimorfismo.
- Como funcionam objetos
- Criar e utilizar construtores
- O que são classes
- Criar e utilizar métodos
- Como funciona encapsulamento
- O que é herança
- O que é polimorfismo
- Como funcionam interfaces

- O que são abstrações

☐ **JSON:**

- JSON significa JavaScript Object Notation (notação de objeto JavaScript). É um formato de texto para armazenar e transmitir dados.
- Criar um objeto
- Transformar um objeto em uma string
- Transformar uma string em objeto
- Manipular um objeto

☐ **Design Patterns:**

- Na engenharia de software, um "padrão de projeto" (Design Pattern em inglês) é uma solução geral e reutilizável para um problema que ocorre normalmente dentro de um determinado contexto de projeto de software.
- Conhecer e aplicar os principais Design Patterns

☐ **Python - Fundamentos:**

- Python é uma linguagem de programação de alto nível, de uso geral, amplamente utilizada em aplicações web, desenvolvimento de software, ciência de dados e Machine Learning. Sua filosofia de projeto enfatiza a legibilidade do código com o uso de indentação significativa. Python é dinamicamente tipada e tem um garbage collector.
- Conhecer os tipos primitivos
- Declarar variáveis, considerando os diferentes tipos
- Usar estruturas condicionais ('if', 'else')
- Conhecer os operadores de atribuição e comparação
- Usar estruturas de repetição e laços ('while', 'for')
- Usar funções, passando parâmetros e argumentos
- Manipular métodos
- Manipular arrays e listas
- Obter dados de uma API

- Criar construtores
- Funções anônimas

☐ **Go (GoLang) - Fundamentos:**

- Go é uma linguagem de programação de alto nível compilada e estaticamente tipada, desenvolvida no Google. Ela é sintaticamente semelhante ao C, mas também tem memory safety, coletor de lixo, tipagem estrutural e concorrência.
- Entender a sua sintaxe
- Conhecer os tipos primitivos
- Declarar e usar variáveis e constantes
- Usar estruturas condicionais (if, else)
- Usar estruturas de repetição e laços (while, for)
- Usar funções, passando parâmetros e argumentos
- Implementando métodos e reutilizando eles
- Tratamento de erros
- Convenções de código
- Manipular Coleções, arrays e listas

☐ **Java - Fundamentos:**

- Java é uma linguagem de programação amplamente usada para codificar aplicações Web. Java é uma linguagem multiplataforma, orientada a objetos e centrada em rede que pode ser usada como uma plataforma em si. É uma linguagem de programação rápida, segura e confiável para codificar tudo, desde aplicações móveis e software empresarial até aplicações de big data e tecnologias do servidor.
- Conhecer os tipos primitivos
- Declarar variáveis, considerando os diferentes tipos
- Usar estruturas condicionais ('if', 'else')
- Conhecer os operadores de atribuição e comparação

- Usar estruturas de repetição e laços ('while', 'for')
- Usar funções, passando parâmetros e argumentos
- Manipular métodos
- Manipular arrays e listas
- Obter dados de uma API
- Criar construtores

☐ **JavaScript - Fundamentos:**

- JavaScript é a linguagem de programação mais popular do mundo e é uma das principais tecnologias da World Wide Web, juntamente com HTML e CSS. Ela possui tipagem dinâmica, orientação a objetos baseada em protótipos e funções de primeira classe. Ela é multi-paradigma e suporta estilos de programação orientados a eventos, funcionais e imperativos.
- Conhecer os tipos primitivos
- Declarar variáveis, considerando a diferença entre 'var', 'let' e 'const'
- Usar estruturas condicionais ('if', 'else')
- Conhecer os operadores de atribuição e comparação ('=', '==', '===')
- Usar estruturas de repetição e laços ('while', 'for')
- Usar funções, passando parâmetros e argumentos
- Manipular arrays e listas
- Aprender o conceito de Orientação a Objetos
- Fazer um CRUD
- Obter dados de uma API
- Fazer chamadas assíncronas usando 'Async/Await', 'Promise', etc

☐ **PHP - Fundamentos:**

- PHP é uma linguagem de programação que permite aos desenvolvedores web criar conteúdo dinâmico que interage com bancos de dados. O PHP é basicamente usado para desenvolver aplicações de software baseadas na web.

- Conhecer os tipos primitivos
- Declarar variáveis
- Usar estruturas condicionais ('if', 'else')
- Conhecer os operadores de atribuição e comparação ('=', '==', '===')
- Usar estruturas de repetição e laços ('while', 'for')
- Usar funções, passando parâmetros e argumentos
- Manipular arrays e listas

☐ **C# - Fundamentos:**

- C# é uma linguagem de programação, multiparadigma, de tipagem forte, desenvolvida pela Microsoft como parte da plataforma .NET. O código fonte é compilado para Common Intermediate Language (CIL) que é interpretado pela máquina virtual Common Language Runtime (CLR). É projetada para funcionar na Common Language Infrastructure da plataforma .NET Framework.
- Conhecer os tipos primitivos
- Declarar variáveis, considerando os diferentes tipos
- Usar estruturas condicionais ('if', 'else')
- Conhecer os operadores de atribuição e comparação
- Usar estruturas de repetição e laços ('while', 'for')
- Usar funções, passando parâmetros e argumentos
- Manipular métodos
- Manipular arrays e listas
- Obter dados de uma API
- Fazer chamadas assíncronas, etc
- Criar construtores

Habilidade Auxiliar: Front-end

☐ **HTML - Fundamentos:**

- HTML é uma linguagem de marcação que define a estrutura do seu conteúdo. HTML consiste em uma série de elementos que você usa para mostrar algo de uma determinada maneira ou agir de uma certo modo. As tags podem criar um hiperlink de uma palavra ou imagem para outro lugar, podem colocar palavras em itálico, podem aumentar ou diminuir a fonte e assim por diante.
- Aprender quais tags são necessárias para um HTML básico
- Criar um parágrafo de texto
- Exibir uma imagem
- Conhecer a diferença entre 'h1', 'h2', 'h3', etc
- Criar um texto com hyperlink
- Criar um formulário com campos relevantes
- Criar uma lista de itens ordenada ou não ordenada
- Criar uma lista de itens dentro de uma lista suspensa (dropdown list)
- Conectar com um arquivo de CSS
- Criar uma tabela
- Adicionar IDs e classes

☐ CSS - Fundamentos:

- Cascading Style Sheets (CSS) é uma linguagem usada para descrever a apresentação de um documento escrito em uma linguagem de marcação como HTML ou XML. CSS pode ser usado para estilos de texto de documentos muito básicos — por exemplo, para alterar a cor e o tamanho de títulos e links. Ele pode ser usado para criar um layout — por exemplo, transformar uma única coluna de texto em um layout com uma área de conteúdo principal e uma barra lateral para informações relacionadas. Pode até ser usado para efeitos como animações.
- Aprender a estrutura visual de uma página, com 'margin' e 'padding'
- Estabelecer o tamanho com 'width' e 'height'
- Aprender sobre a posição de um elemento ('static', 'relative' ou 'absolute')

- Aprender sobre o 'display' de exibição de um elemento ('block', 'inline', 'inline-block')
- Aprender a posicionar imagens em relação ao texto
- Aprender sobre alinhamento
- Aprender sobre estilo de fontes
- Aprender as diferenças e vantagens de usar as diferentes unidades de medida em CSS (% , relativas, etc)
- Conectar com os elementos (IDs, classes) de um arquivo HTML
- Alterar características de um elemento quando o mouse passar por cima dele ('hover')
- Aprender box-sizing
- Aprender Flexbox
- Aprender Grid

☐ **DOM - Fundamentos:**

- O Document Object Model (DOM) é uma interface de programação para documentos web. Ele representa a página para que os programas possam alterar a estrutura, o estilo e o conteúdo do documento. O DOM representa o documento como nós e objetos; dessa forma, linguagens de programação podem interagir com a página.
- Entender como funciona a árvore do DOM
- Acessar e manipular elementos do HTML e CSS
- Acessar os pais e filhos de um elemento
- Inserir um novo elemento na árvore
- Remover um elemento da árvore
- Esperar por um evento em certo elemento da página usando 'addEventListener()'

☐ **Estratégias de SEO:**

- SEO significa otimização para motores de busca e que diz respeito as estratégias usadas para ranquear um site dentro de mecanismos de busca

como o Yahoo, o Bing e, claro, o mais famoso de todos, o Google.

- Escolher palavras-chave
- Entender como o Google classifica páginas
- Conhecer os fatores de ranqueamento
- Fazer Link Building

TechGuide - Alura

Alura, PM3 e FIAP

O Techguide.sh é um projeto open source