

# Guia de Estudos – React JavaScript

- [Introdução](#)
- [Dominando o SCRUM](#)
- [Front-end x Back-end x Full-Stack](#)
- [Git e GitHub - Conceitos Básicos](#)
- [Fluxograma](#)
- [HTML - HyperText Markup Language](#)
- [CSS - Cascading Style Sheets](#)
- [Media Queries no CSS](#)
- [JavaScript \(JS\)](#)
- [JavaScript \(JS\) Assíncrono](#)
- [Node JavaScript \(JS\)](#)
- [React JS \(JavaScript\)](#)
- [Git + GitHub - Na Prática](#)
- [Banco de Dados - Postgres](#)
- [Cursor IA](#)
- [Linux](#)
- [Guias rápidos - Diversos](#)

# Introdução

---

## ? Objetivo da Trilha

Ao final dessa trilha, você deve **conseguir explicar com suas próprias palavras** os conceitos abaixo e ter clareza de quando e por que utilizá-los no dia a dia. A trilha é dividida em vários módulos para ficar mais organizado para sua jornada de aprendizado.

---

## ? Estrutura da Trilha

- Esta trilha foi pensada para nivelar conhecimentos técnicos e alinhar práticas utilizadas no dia a dia da empresa.
- Cada módulo representa um tema específico e possui conteúdos recomendados e perguntas-guia que indicam o que você precisa sair sabendo.

## ? Fundamentos

- Front-end x Back-end x Full-Stack
- Git e GitHub
- HTML / CSS / JS

## ? React e Ecossistema

- React JS - Introdução
- Criando projetos
- Hooks e padrões

## ? Complementos Importantes

- Metodologias ágeis
  - Banco de dados
  - Ferramentas auxiliares
- 

## ? Sobre os materiais de estudo

- Os vídeos e links indicados são **recomendações**
- Você pode utilizar **outros materiais** (artigos, vídeos, cursos, docs)
- O mais importante é:
  - Compreender os conceitos

- Conseguir explicá-los com suas próprias palavras
  - Saber quando e por que utilizá-los
- 

## ? Entrega de Projetos

### ??? Seu Projeto

- Criar repositório no GitHub
- Nome do repositório:

### ? Checklist:

- README.md resumindo o projeto ou por que do projeto.
- Commits organizados.

### ? Link de repositórios (modelos):

- [gaabrenno • Tasks Day - Gerenciador de Tarefas](#)

# Dominando o SCRUM

---

## ? Vídeo(s) Recomendado(s)

- Curso no Hotmart: [Dilmar • Dominando o SCRUM](#)
    - Esse é bom fazer em paralelo com os outros, pois ele é meio longo.
    - No final vai receber o seu diploma.
- 

## ? Material de Apoio

- Referência rápida para uso prático.
- SCRUM + Agilidade:
  - Link: [Manifesto para Desenvolvimento Ágil de Software](#)
  - Link: [Scrum Guide](#)

# Front-end x Back-end x Full-Stack

---

## ? Vídeo(s) Recomendado(s)

- 📺 Vídeo: [Rafaella Ballerini • O QUE É FRONT-END E BACK-END???](#)
    - 🕒 Duração: 6:22
- 

## ?? Pontos-chave (Perguntas-guia)

- Se você consegue responder isso, você entendeu o módulo.
- Essas perguntas poderão ser usadas futuramente para validação.

## ? Perguntas:

- O que é front-end?
- O que é back-end?
- O que é full-stack?
- Como que o trabalho de front-end, back-end, e full-stack trabalham juntos?

# Git e GitHub - Conceitos Básicos

---

## ? Vídeo(s) Recomendado(s)

- 📺 Vídeo: [Rafaella Ballerini • O QUE É GIT E GITHUB? - definição e conceitos importantes](#)  
[1/2](#)
    - 🕒 Duração: 15:42
  - **Obs.:** Não precisa ver parte 2/2, na parte **intermediária** da trilha vamos ver mais a fundo e na prática.
- 

## ?? Pontos-chave (Perguntas-guia)

- Se você consegue responder isso, você entendeu o módulo.
- Essas perguntas poderão ser usadas futuramente para validação.



## ? Perguntas:

- O que é Git?
- O que é GitHub?
- Qual a diferença entre Git e GitHub?
- Qual o benefício do uso do Git e GitHub para as empresas?

# Fluxograma

---

## ? Vídeo(s) Recomendado(s)

-  Vídeo: [Blog Abri Minha Empresa • FLUXOGRAMA O QUE É E COMO FAZER UM \(em 06 Passos Bem Práticos\)](#)
    -  Duração: 8:41
  - **Obs.:** Só precisa entender os conceitos básicos do que é e como funciona um Fluxograma.
- 

## ?? Pontos-chave (Perguntas-guia)

- Se você consegue responder isso, você entendeu o módulo.
- Essas perguntas poderão ser usadas futuramente para validação.

## ? Perguntas:

- O que é um **fluxograma** e para que ele é usado?
  - Quais são os **símbolos básicos** de um fluxograma (início/fim, processo, decisão)?
  - Para que serve um **retângulo** em um fluxograma?
  - Para que serve um **losango** em um fluxograma?
  - O que significa seguir o fluxo com **setas conectando símbolos**?
  - Como você representa um **loop (repetição)** num fluxograma?
  - Qual o papel do símbolo de **entrada/saída**?
  - Por que é importante ter apenas um **início e um fim** no fluxograma?
  - Como fluxogramas ajudam na **comunicação de processos** e entendimento de lógica?
- 

## ? Material de Apoio

Referência rápida para uso prático.

- [Checklist Fácil • Tipos de Fluxograma](#)
- [Breezetreer • Fluxograma Cheat Sheet](#)
- [Canva](#)



# HTML - HyperText Markup Language

---

## ? Vídeo(s) Recomendado(s)

- 📺 Vídeo: [Felipe Rocha • Full Stack Club • Curso de HTML Para Completos Iniciantes](#)
    - 🕒 Duração: 42:30
- 

## ?? Pontos-chave (Perguntas-guia)

- Se você consegue responder isso, você entendeu o módulo.
- Essas perguntas poderão ser usadas futuramente para validação.

## ? Perguntas:

- O que é HTML e para que ele serve na web?
  - HTML é uma linguagem de programação? Por quê?
  - Para que servem `<head>` e `<body>` ?
  - O que é uma lista ordenada e não ordenada?
  - Qual a diferença entre **tag**, **elemento** e **atributo**?
  - Qual é a **estrutura mínima** de um documento HTML?
- 

## ? Material de Apoio

Referência rápida para uso prático.

- [HTML Cheat Sheets](#)
- [Devhints Cheat Sheets • HTML](#)
- [Devhints Cheat Sheets • HTML Meta Tags](#)
- [Devhints Cheat Sheets • Input Tag](#)
- [Devhints Cheat Sheets • Applinks](#)
- [Devhints Cheat Sheets • Internet Explorer](#)
- [Devhints Cheat Sheets • HTML Emails](#)
- [W3Schools • HTML Tutorial](#)

- [MDN Web Docs • HTML: HyperText Markup Language](#)

# CSS - Cascading Style Sheets

---

## ? Vídeo(s) Recomendado(s)

- 📺 Vídeo: [Felipe Rocha • Full Stack Club • Curso de CSS Para Completos Iniciantes](#)
    - 🕒 Duração: 1:13:37
- 

## ?? Pontos-chave (Perguntas-guia)

- Se você consegue responder isso, você entendeu o módulo.
- Essas perguntas poderão ser usadas futuramente para validação.

## ? Perguntas:

- O que é CSS e qual o seu papel na web?
  - Qual a diferença entre HTML e CSS?
  - O que é um seletor CSS?
  - O que são propriedades e valores?
  - Qual a diferença entre `class` e `id` ?
- 

## ? Material de Apoio

Referência rápida para uso prático.

- [CSS-Tricks • CSS Almanac](#)
- [CSS-Tricks • CSS Flexbox Layout Guide](#)
- [W3 Schools • CSS Tutorial](#)
- [MDN Web Docs • CSS: Cascading Style Sheets](#)
- [Devhints Cheat Sheets • CSS](#)
- [Devhints Cheat Sheets • CSS Grid](#)
- [Devhints Cheat Sheets • CSS Tricks](#)
- [Devhints Cheat Sheets • CSS Flexbox](#)
- [Devhints Cheat Sheets • CSS System Fonts](#)
- [Devhints Cheat Sheets • CSS Webfonts](#)

- [Devhints Cheat Sheets • CSS Unicode Symbols](#)
- [Devhints Cheat Sheets • CSS Bootstrap](#)

# Media Queries no CSS

---

## ? Vídeo(s) Recomendado(s)

- 📺 Vídeo: [Felipe Rocha • Full Stack Club • Aprenda MEDIA QUERIES no CSS em 10 MINUTOS](#)
    - ⌚ Duração: 10:32
- 

## ?? Pontos-chave (Perguntas-guia)

- Se você consegue responder isso, você entendeu o módulo.
- Essas perguntas poderão ser usadas futuramente para validação.

### ? Perguntas:

- O que é uma media query?
  - Para que servem media queries no CSS?
  - Qual a diferença entre `min-width` e `max-width` ?
  - O que significa design responsivo?
  - Quando usar media queries em um projeto?
  - O que acontece se não usar media queries em layouts responsivos?
  - Como aplicar estilos diferentes para mobile e desktop?
- 

## ? Material de Apoio

Referência rápida para uso prático.

- [W3 Schools • CSS Media Queries](#)
- [W3 Schools • Responsive Web Design - Media Queries](#)

# JavaScript (JS)

---

## ? Vídeo(s) Recomendado(s)

- 📺 Vídeo: [Rocketseat • TUDO que você deve estudar de JavaScript antes do React](#)
    - ⏱ Duração: 1:25:23
  - 📺 Playlist: [Matheus Battisti - Hora de Codar • Curso de JavaScript 2024](#)
    - ⏱ Duração da playlist completa: 2:33:39
- 

## ?? Pontos-chave (Perguntas-guia)

- Se você consegue responder isso, você entendeu o módulo.
- Essas perguntas poderão ser usadas futuramente para validação.

### ? Perguntas:

- O que é JavaScript e qual seu papel no front-end?
  - Qual a diferença entre JavaScript, HTML e CSS?
  - O que são variáveis e para que servem?
  - Qual a diferença entre `var`, `let`, e `const` ?
  - O que são tipos de dados em JavaScript?
  - O que é uma função?
  - O que é um array?
  - O que é um objeto?
  - O que são condicionais `if` e `else` ?
  - O que são loops `for` e `while` ?
- 

## ? Material de Apoio

Referência rápida para uso prático.

- [W3 Schools • JavaScript Tutorial](#)
- [GitHub • Modern JavaScript Cheat Sheet](#)
- [MDN Web Docs • JavaScript](#)
- [Over API • JavaScript Cheat Sheet](#)
- [Devhints Cheat Sheets • Canvas](#)

- [Devhints Cheat Sheets • Dom Selection](#)
- [Devhints Cheat Sheets • JavaScript Date](#)
- [Devhints Cheat Sheets • JavaScript lazy shortcuts](#)
- [Devhints Cheat Sheets • Promises](#)
- [Devhints Cheat Sheets • ES2015+](#)
- [Devhints Cheat Sheets • JavaScript Arrays](#)
- [Devhints Cheat Sheets • NPM](#)
- [Devhints Cheat Sheets • strftime format](#)

# JavaScript (JS) Assíncrono

---

## ? Vídeo(s) Recomendado(s)

- 📺 Vídeo: [Felipe Rocha • Full Stack Club • Curso de JavaScript Assíncrono - Callbacks, Promises & Async/Await](#)
    - 🕒 Duração: 35:04
- 

## ?? Pontos-chave (Perguntas-guia)

- Se você consegue responder isso, você entendeu o módulo.
- Essas perguntas poderão ser usadas futuramente para validação.

## ? Perguntas:

- O que significa código assíncrono em JavaScript?
  - Qual a diferença entre código síncrono e assíncrono?
  - O que é a Call Stack?
  - O que é uma Promise?
  - O que é `async` / `await` ?
  - Qual a diferença entre usar `.then()` e `async/await` ?
  - Em que situações código assíncrono é usado no dia a dia?
- 

## ? Material de Apoio

Referência rápida para uso prático.

- [W3 Schools • JavaScript Tutorial](#)
- [GitHub • Modern JavaScript Cheat Sheet](#)
- [MDN Web Docs • JavaScript](#)
- [MDN Web Docs • Asynchronous JavaScript](#)
- [MDN Web Docs • async function](#)
- [Over API • JavaScript Cheat Sheet](#)
- [Devhints Cheat Sheets • Canvas](#)
- [Devhints Cheat Sheets • Dom Selection](#)

- [Devhints Cheat Sheets • JavaScript Date](#)
- [Devhints Cheat Sheets • JavaScript lazy shortcuts](#)
- [Devhints Cheat Sheets • Promises](#)
- [Devhints Cheat Sheets • ES2015+](#)
- [Devhints Cheat Sheets • JavaScript Arrays](#)
- [Devhints Cheat Sheets • NPM](#)

# Node JavaScript (JS)

---

## ? Vídeo(s) Recomendado(s)

- 📺 Vídeo: [Felipe Rocha • Full Stack Club • Curso de Node.js Para Completos Iniciantes](#)
    - 🕒 Duração: 1:47:41
- 

## ?? Pontos-chave (Perguntas-guia)

- Se você consegue responder isso, você entendeu o módulo.
- Essas perguntas poderão ser usadas futuramente para validação.

### ? Perguntas:

- O que é Node.js?
  - Node.js é uma linguagem ou um runtime?
  - Qual a diferença entre JavaScript no browser e no Node.js?
  - Para que tipo de aplicação o Node.js é mais usado?
  - O que é o npm?
  - O que é um `package.json` ?
  - Qual a diferença entre dependências e devDependencies?
  - O que são módulos no Node.js?
  - Qual a diferença entre `require` e `import` ?
  - O que é o Event Loop no Node.js?
  - O que é um servidor HTTP em Node.js?
  - Como Node.js lida com requisições assíncronas?
- 

## ? Material de Apoio

Referência rápida para uso prático.

- [Devhints Cheat Sheets • Assert](#)
- [Devhints Cheat Sheets • fs](#)
- [Devhints Cheat Sheets • Node JS API](#)
- [Devhints Cheat Sheets • Node JS Path API](#)
- [Devhints Cheat Sheets • NPM](#)

- [Devhints Cheat Sheets • Processes](#)
- [Devhints Cheat Sheets • Package json](#)
- [Devhints Cheat Sheets • Node JS Stream](#)

# React JS (JavaScript)

---

## ? Vídeo(s) Recomendado(s)

- 📺 Vídeo: [Programação Web • Curso de React Para Iniciantes](#)
  - 🕒 Duração: 57:13
- 📺 Vídeo: [Programação Web • Projeto Prático](#)
  - 🕒 Duração: 1:06:55
  - **Obs.:** Continuação do vídeo anterior
- 📺 Vídeo: [Felipe Rocha • Full Stack Club • Curso de React para Completos Iniciantes \[2026\]](#)
  - 🕒 Duração: 2:25:22

---

## ?? Pontos-chave (Perguntas-guia)

- Se você consegue responder isso, você entendeu o módulo.
- Essas perguntas poderão ser usadas futuramente para validação.

## ? Perguntas:

- O que é React e qual problema ele resolve?
- O que é um componente em React?
- O que é JSX e por que o React usa JSX?
- O que são props?
- Qual a diferença entre props e state?
- O que é `useState` ?
- Quando o componente re-renderiza?
- Como funciona o fluxo de dados no React?
- O que é `useEffect` e quando usar?
- O que é renderização condicional?
- Em que partes do produto a empresa usa React?

---

## ? Material de Apoio

## Referência rápida para uso prático.

- [React • Quick Start](#)
- [Devhints Cheat Sheets • React JS](#)
- [Devhints Cheat Sheets • React JS \(v0.14\)](#)
- [Devhints Cheat Sheets • React-router](#)

# Git + GitHub - Na Prática

---

## ? Vídeo(s) Recomendado(s)

- 📺 Vídeo: [Felipe Rocha • Full Stack Club • Curso de Git & GitHub Para Completos Iniciantes](#)
    - 🕒 Duração: 45:53
    - **Obs.:** Recomendado depois você fazer o setup da sua chave SSH para sua máquina.
  - 📺 Vídeo Setup do SSH no Windows: [hcode • Como Configurar Chave SSH no GitHub: Guia Passo a Passo.](#)
    - 🕒 Duração: 5:58
  - 📺 Vídeo Setup do SSH no Ubuntu: [Winner Dev • Criando chave SSH para o GitHub em 2022 | Simples e Rápido](#)
    - 🕒 Duração: 3:23
- 

## ?? Pontos-chave (Perguntas-guia)

- Se você consegue responder isso, você entendeu o módulo.
- Essas perguntas poderão ser usadas futuramente para validação.

## ? Perguntas:

- Como criar um repositório no GitHub?
  - Como clonar um repositório para a máquina local?
  - Qual a diferença entre HTTPS e SSH?
  - Por que usar chave SSH no GitHub?
  - Como verificar se a conexão SSH está funcionando?
  - Como fazer alterações locais e versionar (`add`, `commit`) ?
  - Como enviar alterações para o GitHub (`push`) ?
  - O que é `pull` e quando usar?
  - O que fazer quando ocorre conflito de merge?
- 

## ? Material de Apoio

Referência rápida para uso prático.

- [Devhints Cheat Sheets • Git Branches](#)

- [Devhints Cheat Sheets • Git Tricks](#)
- [GitHub Docs • Fluxo de GitHub](#)
- [GitHub • Git Cheat Sheet](#)
- [GitHub Docs • Conectando-se ao GitHub com o SSH](#)
- [Devhints Cheat Sheets • Linux](#)

# Banco de Dados – Postgres

## ? Vídeo(s) Recomendado(s)

- 📺 Playlist: [Fabricio Vallim • Curso Básico de SQL](#)
    - 🕒 Duração: 2:45:15
- 

## ?? Pontos-chave (Perguntas-guia)

- Se você consegue responder isso, você entendeu o módulo.
- Essas perguntas poderão ser usadas futuramente para validação.

## ? Perguntas:

- O que é um banco de dados relacional?
  - O que é SQL e PostgreSQL?
  - Qual a diferença entre banco de dados e tabela?
  - O que é uma chave primária?
  - O que é uma chave estrangeira?
  - O que é normalização de dados?
  - Para que serve `SELECT`?
  - Qual a diferença entre `WHERE` e `HAVING` ?
  - O que é `JOIN` e quais tipos existem?
  - Qual a diferença entre `INNER JOIN`, `RIGHT JOIN`, e `LEFT JOIN` ?
  - Para que servem `INSERT`, `UPDATE` e `DELETE` ?
  - O que é `ORDER BY` ?
  - O que é `GROUP BY` ?
  - O que é um **CRUD**?
  - O que são índices e por que impactam performance?
- 

## ? Material de Apoio

Referência rápida para uso prático.

- [Ferramenta • Visualizador dos Joins do SQL](#)
- [Neon • PostgreSQL Cheat Sheet](#)
- [SQL Bolt • Introduction to SQL](#)
- [Devhints Cheat Sheet • PostgreSQL JSON](#)



# Cursor IA

---

## ? Vídeo(s) Recomendado(s)

- 📺 Vídeo: [Matheus | IA Coding • A Nova Forma de Programar com IA usando Cursor \(E Como Você Pode Dominar Isso!\)](#)
    - ⌚ Duração: 31:27
- 

## ?? Pontos-chave (Perguntas-guia)

- Se você consegue responder isso, você entendeu o módulo.
- Essas perguntas poderão ser usadas futuramente para validação.

### ? Perguntas:

- O cursor é um `fork` do VS Code (Visual Studio Code). O que isso significa?
  - O que é o Cursor e qual problema ele resolve?
  - Qual a diferença entre o Cursor e um editor tradicional?
  - Em que situações o Cursor acelera o desenvolvimento?
  - Como usar prompts de forma clara e objetiva?
  - O que significa “dar contexto” para a IA?
  - Quando **não** confiar cegamente na resposta da IA?
  - Como usar o Cursor para:
    - Entender código existente?
    - Refatorar código?
    - Gerar testes?
  - Quais cuidados tomar ao usar IA com código sensível?
  - Como revisar e validar código gerado por IA?
  - Qual é o papel do desenvolvedor quando usa IA no fluxo de trabalho?
- 

## ? Material de Apoio

Referência rápida para uso prático.

- [Cursor • Site Oficial](#)
- [Prompting Guide • Guia de Engenharia Prompt](#)
- [Devhints Cheat Sheets • Claude Code](#)



# Linux

---

## ? Vídeo(s) Recomendado(s)

- 📺 Vídeo: [Prof. Juliano Ramos - Linux do Zero ao Hacker! • Guia definitivo do Linux para iniciantes 2025!](#)
    - 🕒 Duração: 1:31:45
  - 📺 Vídeo Setup do SSH no Ubuntu: [Winner Dev • Criando chave SSH para o GitHub em 2022 | Simples e Rápido](#)
    - 🕒 Duração: 3:23
- 

## ?? Pontos-chave (Perguntas-guia)

- Se você consegue responder isso, você entendeu o módulo.
- Essas perguntas poderão ser usadas futuramente para validação.

## ? Perguntas:

- O que é Linux?
- Qual a diferença entre Linux e Windows?
- O que é uma distribuição Linux (distro)?
- O que é o terminal no Linux?
- Qual a diferença entre interface gráfica (GUI) e linha de comando (CLI)?
- Por que aprender linha de comando é importante no Linux?
- Para que serve o comando `ls` ?
- Para que serve o comando `cd` ?
- Para que serve o comando `pwd` ?
- Como criar pastas e arquivos pelo terminal?
- Como remover arquivos e diretórios?
- Como visualizar o conteúdo de arquivos pelo terminal?
- O que são permissões no Linux?
- Qual a diferença entre usuário comum e root?
- O que faz o comando `sudo` ?
- Como instalar programas no Linux?
- O que é um gerenciador de pacotes?
- Qual a diferença entre `apt`, `snap`, e `flatpak` ?
- Como atualizar o sistema pelo terminal?
- O que é SSH e para que serve?

- Como gerar uma chave SSH no Ubuntu?
  - Como adicionar a chave SSH no GitHub?
  - Como verificar informações básicas do sistema Linux pelo terminal?
  - Por que o Linux é muito utilizado em servidores?
- 

## ? Material de Apoio

Referência rápida para uso prático.

- [Devhints Cheat Sheets • Linux](#)
- [GitHub Docs • Conectando-se ao GitHub com o SSH](#)
- [GitHub • Ubuntu Cheat Sheet](#)

# Guias rápidos – Diversos

---

## ? Material de Apoio Extra

Referência rápida para uso prático.

- [Devhints Cheat Sheets • Linux](#)
- [Devhints Cheat Sheets • Claude Code](#)
- [Devhints Cheat Sheets • RESTful API](#)
- [Devhints Cheat Sheets • HTTP Status](#)
- [Devhints Cheat Sheets • Command Line Stuff](#)
- [Devhints Cheat Sheets • Markdown](#)
- [Devhints Cheat Sheets • Visual Studio Code](#)