



# Level Up: Missão Programador!

Suba de Nível no Mundo da Tecnologia





# Suba de Nível no Mundo da Tecnologia

## Bem-vindo ao Level Up: Missão Programador!

Este eBook é para quem está começando em TI. Aprenda dicas práticas, entenda as linguagens e a diferença entre frontend e backend.

Prepare-se para dar os primeiros passos e subir de nível na programação!



# Como Entrar na Área de TI

Começar em TI exige foco. Siga estes passos para iniciar sua jornada com sucesso:



## Aprenda Lógica

Comece dominando a lógica de programação, a base de tudo.



## Escolha uma Linguagem

Python é ótima para iniciantes. Escolha uma para começar.



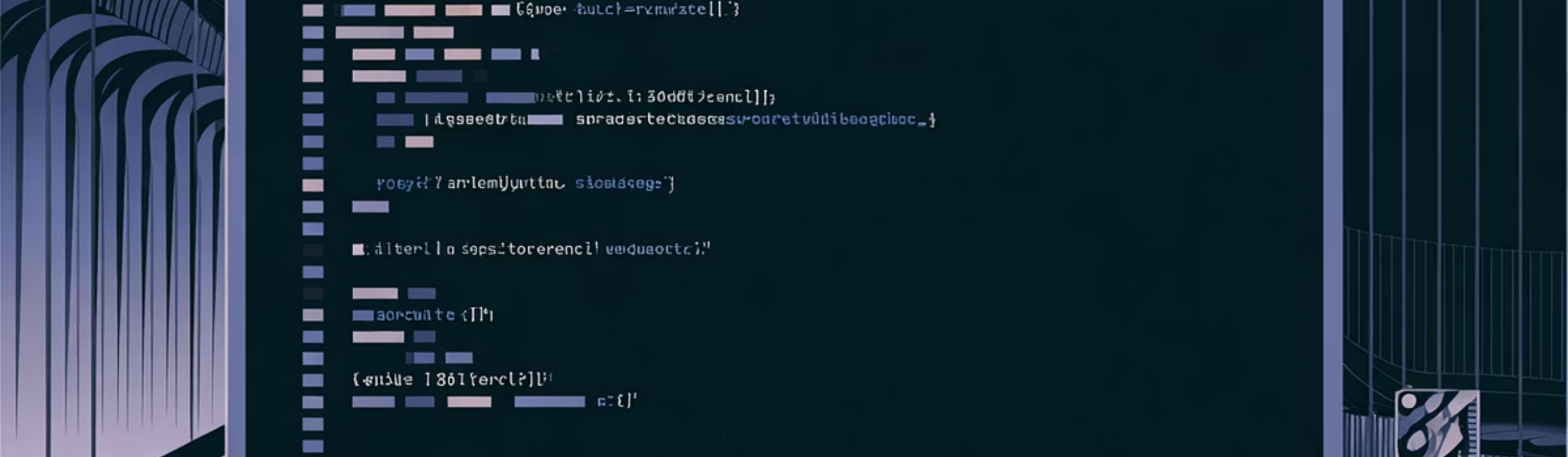
## Crie Projetos

Desenvolva projetos simples (calculadoras, sites) para praticar.



## Construa seu Portfólio

Mostre seus projetos. Pequenos projetos valem mais que certificados!



# Diferenças Entre Linguagens de Programação

Cada linguagem tem um propósito específico. Conheça as principais:



## Python

Automação, dados, IA. Ex: `print('Ola, mundo!')`



## JavaScript

Web, frontend e backend. Ex: `console.log('Ola, mundo!')`



## Java

Apps corporativos e Android. Ex: `System.out.println('Ola, mundo!')`



## SQL

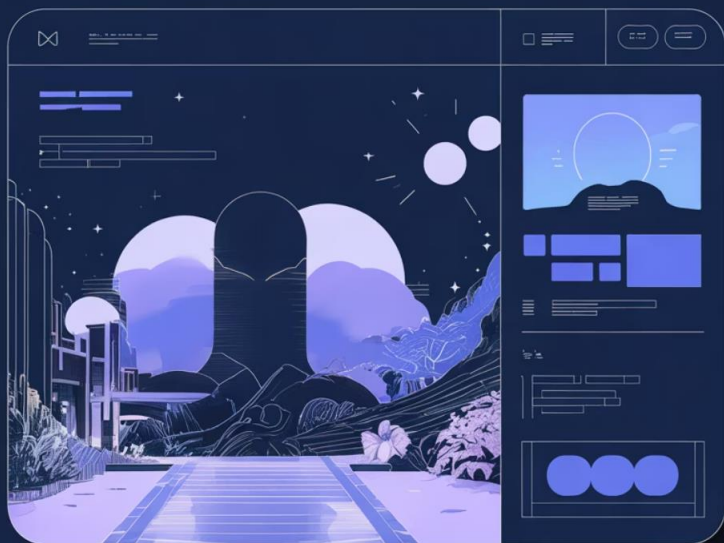
Banco de dados. Ex: `SELECT * FROM usuarios;`

# Frontend x Backend

## Frontend (Aparência)

Tudo que o usuário vê e interage. É a interface.

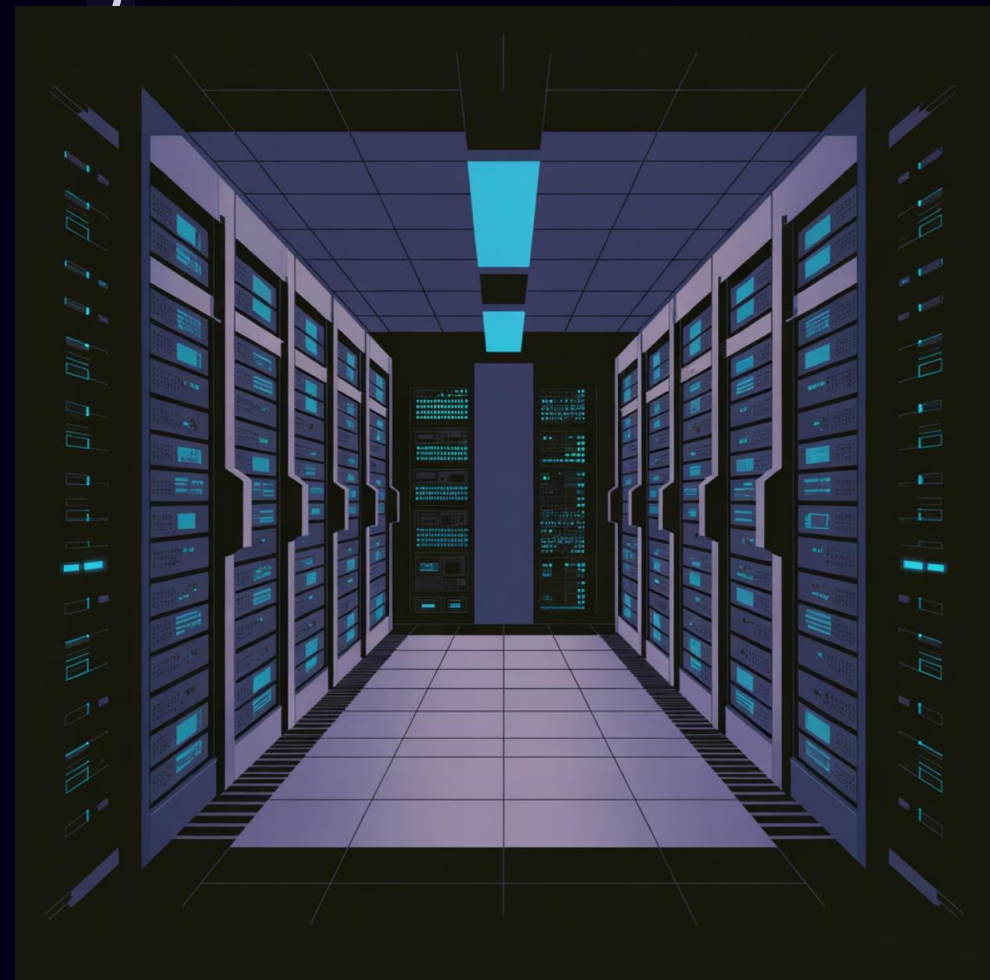
- HTML, CSS, JavaScript
- Exemplo: Um botão de "Enviar"



## Backend (Lógica)

Lógica que roda no servidor, processando dados e regras.

- Python, Java, Node.js
- Exemplo: Função `cadastrar_usuario(nome`  
`)`



Frontend = aparência; Backend = lógica por trás.

# Ferramentas e Projetos Para Iniciantes

01  
10

## VS Code

Editor de código essencial para escrever e depurar.



## Git e GitHub

Versionamento de código e plataforma para seu portfólio.



## Calculadora Simples

Projeto inicial para praticar lógica em Python ou JS.



## Página Pessoal

Crie seu próprio site ou portfólio para mostrar habilidades.

Dica: Comece com uma linguagem, um editor e Git. Não tente aprender tudo de uma vez!

# Variáveis, Condicionais e Loops

## Variáveis

Caixas que guardam informações (ex: nome, idade). Usadas para guardar dados de usuários.

```
nome = 'Andrielly'
```

## Condicionais (If/Else)

Decidem o caminho do código (ex: verificar permissão de idade).

```
if idade >= 18: print('Pode criar conta')
```

## Loops (For/While)

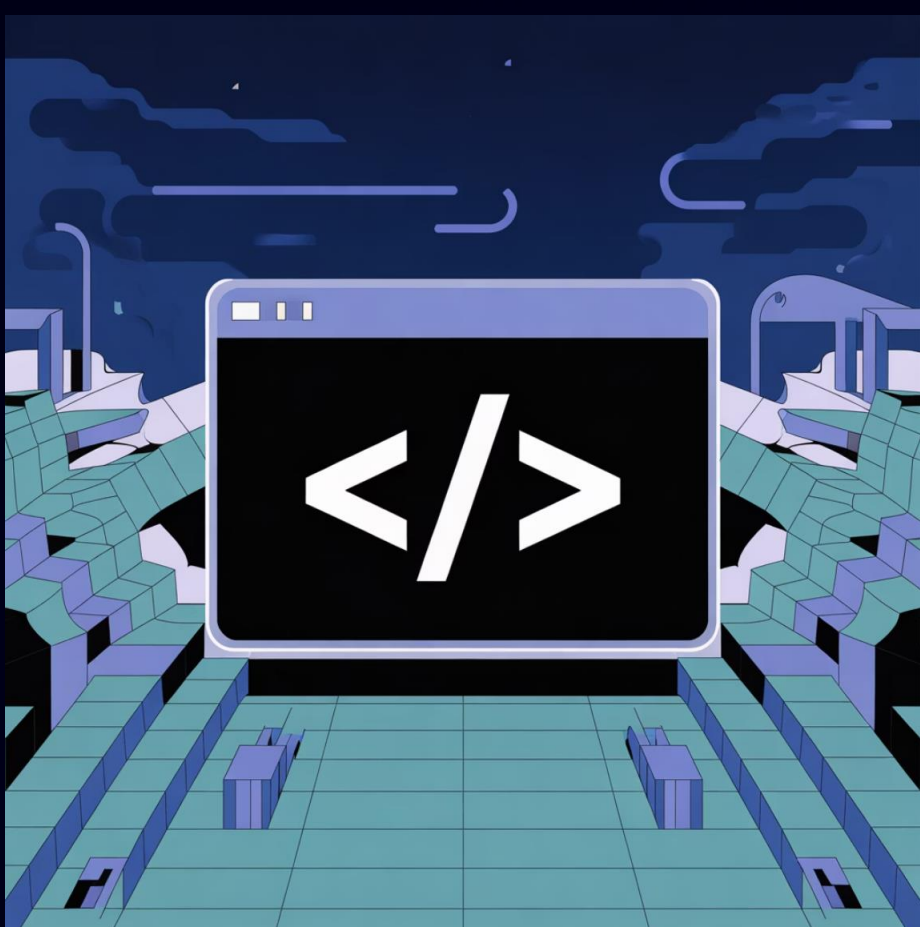
Repetem ações automaticamente (ex: enviar mensagens a uma lista).

```
for nome in nomes: print(f'Bem-vindo, {nome}!')
```



# Funções: Blocos Reutilizáveis

Funções organizam o código, tornando-o mais limpo e fácil de manter.



Funções são essenciais para reutilizar código, como calcular descontos ou enviar notificações em sistemas.

## Exemplo Python:

```
def saudacao(nome):  
    return f'Ola,  
{nome}!' print(saudaca  
o('Andrielly'))
```



# Coleções de Dados: Listas e Dicionários

Coleções permitem armazenar múltiplos valores de forma estruturada.

## Listas (Arrays)

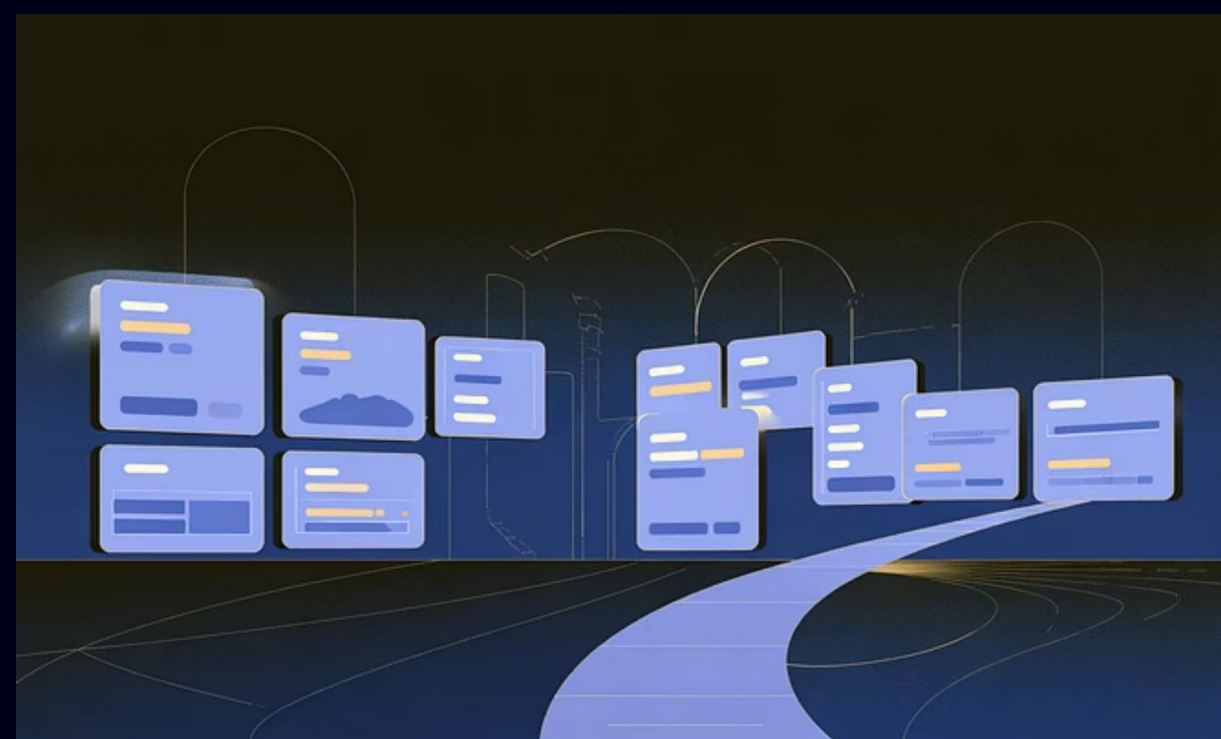
Guardam itens em ordem. Útil para armazenar produtos ou nomes.

```
frutas = ['maca', 'banana', 'laranja']
```

## Dicionários (Objetos)

Guardam dados em pares chave-valor. Ideal para perfis de usuários.

```
usuario = {'nome': 'Andrielly', 'idade': 20}
```



# Tratamento de Erros (Try/Except)

Garanta que seu programa não "quebre" ao lidar com entradas incorretas do usuário ou dados externos.

## Bloco TRY

Onde o código potencialmente problemático é executado (ex: pedir input numérico).

## Bloco EXCEPT

Executado se um erro (ex: `ValueError`) ocorrer no bloco TRY.

## Prevenção

Essencial para formulários de cadastro e processamento de dados.

```
try: numero = int(input('Digite um numero: '))except  
ValueError: print('Isso nao e um numero valido!')
```



# Próximos Passos: Continue Subindo de Nível!

Parabéns! 🎉

Você agora conhece os conceitos básicos de programação, a diferença entre linguagens, frontend e backend, e viu exemplos práticos de código.

Lembre-se: a chave para crescer em TI é **praticar todos os dias** e nunca ter medo de errar.

Agora é sua vez: **coloque a mão na massa, explore e continue subindo de nível!**

Compartilhe este post com um futuro programador!