

# Aprenda algoritmos com o Professor Marco Maddo



🎓 Bem-vindo(a) ao nosso projeto de estudos de **Algoritmos e Programação Multilíngue!**

Neste documento, você encontrará o passo a passo para instalar todas as ferramentas necessárias para compilar, testar e executar os algoritmos desenvolvidos em **Portugol, Java, Kotlin, Python, C++, JavaScript, TypeScript, Go Lang e Pascal**, além de configurar o **Git** para versionamento dos seus projetos. O conteúdo contempla os sistemas **Linux, macOS e Windows**. 🧠 🖥️

# Orientação Geral: Sem IDEs! Terminal é Poder! ⚡

Neste treinamento, **não utilizaremos IDEs pesadas como Visual Studio Code, IntelliJ, Eclipse ou PyCharm**. A proposta é que você aprenda a dominar a arte de programar com o terminal.

Por quê?

🧠 Porque queremos que você:

- Entenda **como o código é compilado**.
- Aprenda a **navegar com comandos**.
- Desenvolva a habilidade de **escrever e executar shell scripts**.
- Torne-se um(a) desenvolvedor(a) independente de ferramentas gráficas.

## ✅ Estrutura Recomendada de Pastas

```
algoritmos/  
├─ algoritmo01.md  
├─ algoritmo02.md  
├─ ...  
├─ java/  
├─ kotlin/  
├─ python/  
├─ pascal/  
├─ portugol/  
└─ ...
```

## Instalação do Git

### Linux (Debian/Ubuntu/Mint)

```
sudo apt update  
sudo apt install git -y
```

## macOS (via Homebrew)

```
brew install git
```

## Windows

Baixe o instalador em: <https://git-scm.com/download/win> e siga o assistente de instalação.

Verifique com:

```
git --version
```



## Instalação do Java (JDK)

## Linux

```
sudo apt install default-jdk -y
```

## macOS

```
brew install openjdk
```

Adicione ao PATH (macOS):

```
echo 'export PATH="/opt/homebrew/opt/openjdk/bin:$PATH"' >> ~/.zprofile
```

## Windows

Baixe o JDK em: <https://jdk.java.net/> ou <https://adoptium.net/>

Verifique com:

```
java -version
```

# Instalação do Kotlin

## Linux/macOS/Windows

Use o compilador de linha de comando:

```
brew install --cask kotlin-native # instalação com homebrews no macOS.  
sdk install kotlin                # se tiver SDKMAN (Linux/macOS)  
kotlinc                           # verifica se está instalado
```

Ou baixe de:

<https://kotlinlang.org/docs/command-line.html>

Para compilar:

```
kotlinc exemplo.kt -include-runtime -d exemplo.jar  
java -jar exemplo.jar
```

## Instalação do Python

### Linux

```
sudo apt install python3 python3-pip -y
```

### macOS

```
brew install python
```

### Windows

Baixe de: <https://www.python.org/downloads/>

Verifique com:

```
python3 --version
```

Execute com:

```
python3 exemplo.py
```

## Instalação do Pascal (Free Pascal Compiler - fpc)

### Linux

```
sudo apt install fpc -y
```

### macOS (com Homebrew)

```
brew install fpc
```

### Windows

Baixe o instalador em: <https://www.freepascal.org/>

Compile com:

```
fpc exemplo.pas
./exemplo          # Linux/macOS
exemplo.exe        # Windows
```

# Instalação do Compilador C (gcc)

## Linux

```
sudo apt update
sudo apt install build-essential -y
```

## macOS (com Xcode Command Line Tools)

```
xcode-select --install
```

Ou com Homebrew:

```
brew install gcc
```

## Windows

Recomendado instalar o pacote MSYS2:

<https://www.msys2.org/>

Depois de instalado, abra o terminal MSYS2 e execute:


```
pacman -Syu
pacman -S mingw-w64-x86_64-gcc
```

 Verifique:

```
gcc --version
```

 Compile e execute:

```
gcc exemplo.c -o exemplo
./exemplo
```

 Aprender C é como aprender a programar na raiz! Ideal para entender ponteiros, memória e performance.



## Recomendação de Editores de Texto Simples

Você pode usar qualquer editor de texto puro. Recomendamos os seguintes:

- 🧠 **VS Codium** (alternativa open source ao VS Code)
  - <https://vscodium.com/>
- 📝 **Notepad++** (Windows)
  - <https://notepad-plus-plus.org/>
- 🧑 **Geany** (leve e poderoso, multiplataforma)
  - Linux: `sudo apt install geany`
- 🛠️ **GNU nano** (terminal, simples e direto)
  - Linux/macOS: já vem instalado!

💡 Todos esses editores são leves, rápidos, e compatíveis com as linguagens deste curso.

## 🤖 Criando um Shell Script para Compilar em Java

Vamos criar um script chamado `compilar-java.sh` para compilar e rodar um programa `.java` no terminal.

## Conteúdo do arquivo `compilar-java.sh` :

```
#!/bin/bash

if [ -z "$1" ]; then
    echo "❌ Uso: ./compilar-java.sh NomeDoPrograma.java"
    exit 1
fi

arquivo=$1
classe=$(basename "$arquivo" .java)

echo "📦 Compilando $arquivo..."
javac "$arquivo"

if [ $? -eq 0 ]; then
    echo "🚀 Executando $classe..."
    java "$classe"
else
    echo "❌ Erro na compilação. Corrija o código."
fi
```

## Como usar:

1. Dê permissão de execução:

```
chmod +x compilar-java.sh
```

2. Execute:

```
./compilar-java.sh MeuPrograma.java
```

🔧 Este script ajuda você a compilar e rodar com um único comando. Excelente para automatizar tarefas repetitivas!



# Instalação do Portugol Studio (Visual para algoritmos)

Download: <https://portugol-webstudio.cubos.io/>



- Alternativamente, use o [Visualg](#) no Windows.

⚠️ Portugol WebStudio pode ser usado diretamente no navegador. Ideal para treinar algoritmos em pseudo-código!

## Redes do Professor Marco Maddo

- YouTube: [Professor Marco Maddo](#)
- LinkedIn: [Marco Maddo](#)
- GitHub: [@profmaddo](#)
- GitLab: [@profmaddo](#)
- Udemy: [Cursos Marco Maddo](#)

## Acesse também:

-  Site oficial do professor: [www.marcomaddo.com.br](http://www.marcomaddo.com.br)
-  Empresa TSSTI Tecnologia: [www.tssti.com.br](http://www.tssti.com.br)

Vamos juntos nessa jornada de aprendizado! ✨

# Instalação do Go (Golang)

## Linux

```
sudo apt update  
sudo apt install golang-go -y
```

## macOS

```
brew install go
```

## Windows

Baixe o instalador em: <https://go.dev/dl/> e siga o assistente.

 Verifique a instalação:

```
go version
```

 Para testar:

```
go run exemplo.go
```

✨ O Go é incrível para criar aplicações rápidas e concisas! Perfeito para quem quer aprender concorrência, APIs e muito mais.

## ⭐ Instalação do JavaScript (Node.js)

JavaScript é executado no navegador, mas para usar no terminal (backend ou scripts), instale o Node.js:

# Linux (Debian/Ubuntu/Mint)


```
sudo apt update  
sudo apt install nodejs npm -y
```

## macOS

```
brew install node
```

## Windows


Baixe de: <https://nodejs.org/>

 Verifique a instalação:

```
node -v  
npm -v
```

 Para rodar um arquivo:

```
node exemplo.js
```

 O Node.js transforma o JavaScript em uma super ferramenta de automação e criação de servidores!

## Instalação do TypeScript

TypeScript é uma versão turbinada do JavaScript com tipagem!

## Requer Node.js instalado!

Instale com o gerenciador npm :


```
npm install -g typescript
```

 Verifique:

```
tsc -v
```

 Para compilar:

```
tsc exemplo.ts  
node exemplo.js
```

 O TypeScript ajuda a evitar erros antes mesmo do código rodar. Ótimo para quem quer código limpo e seguro!

## Instalação do PHP

### Linux

```
sudo apt update  
sudo apt install php -y
```

### macOS


```
brew install php
```

### Windows


Baixe de: <https://www.php.net/downloads>

 Verifique a versão:

```
php -v
```

 Para rodar um script PHP:

```
php exemplo.php
```

 PHP é amplamente usado para criar sites e sistemas web. Simples, poderoso e muito usado no mercado!

# Instalação do Dart (para Terminal)

## Linux


```
sudo apt update
sudo apt install apt-transport-https
sudo sh -c 'wget -qO- https://dl-ssl.google.com/linux/linux_signing_key.pub | gpg --dearmor -o /usr/share/keyrings/dart.gpg'
sudo apt update
sudo apt install dart -y
```

## macOS

```
brew tap dart-lang/dart
brew install dart
```

## Windows


Baixe o SDK: <https://dart.dev/get-dart>

 Verifique a instalação:

```
dart --version
```

 Para rodar um arquivo:

```
dart exemplo.dart
```

 Dart é a linguagem por trás do Flutter! Mas você pode usá-lo sozinho para scripts e apps via terminal!