

Instalação dos requisitos para desenvolvimento em Windows 10/11

1. Preparando o ambiente de trabalho

Clique na barra de pesquisa (Pesquisar) ao lado do botão iniciar e digite: **powershell**

Clique na opção "Executar como administrador"

Informe o usuário com privilégio administrativo e respectiva senha

Uma janela com título: "Administrador: Windows PowerShell" estará disponível.

É nessa janela que executaremos os demais comandos

1.1. Instalando o gerenciador de pacotes chocolatey

Copie e cole o conteúdo a seguir no terminal powershell com privilégio de administrador e pressione ENTER

```
Set-ExecutionPolicy Bypass -Scope Process -Force;  
[System.Net.ServicePointManager]::SecurityProtocol =  
[System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex  
(New-Object  
System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1')
```

Após alguns instantes a instalação será concluída. Nota: A versão será igual ou mais recente.

Verifique o sucesso desta etapa com:

```
> choco --version  
2.5.0
```

2. Instalando os programas necessários

2.1. Instalando o VSCode (editor de texto com destaque para sintaxe de códigos)

Acesse: <https://code.visualstudio.com/download>

Escolha "System Installer" e clique em **x64**

2.2. Instalando o Git (para utilizar com GitHub)

```
> choco install git.install -y
```

2.3. Instalando o Node.JS

```
> choco install nodejs-lts -y
```

2.4. Instalando o java 11

```
> choco install openjdk11 -y
```

2.5. Gerenciador de versões do NodeJS

```
> choco install nvm -y
```

2.6. Gerenciador de pacotes alternativo ao NPM (padrão)

```
> choco install yarn -y
```

2.7. Visualizador do celular na tela do computador

```
> choco install scrpy -y
```

2.8. **[ATENÇÃO!! Instalação OPCIONAL!] Android Studio v2024.2.2.14** (Requer 1.15GB)

Para execução e testes do projeto desenvolvido é possível usar o Android Studio (emulador requer ao menos 16GB RAM)

```
> choco install androidstudio -y
```

2.9. Executando a aplicação no celular

Para testes utilizando computadores mais simples, precisamos de ao menos 4GB de RAM no PC para utilizar os programas indicados neste documento (React Native + EXPO) e o aplicativo **Expo-GO** no celular.

ATENÇÃO! O computador e o celular devem estar na mesma rede!

3. Verificar as instalações

A reinicialização do computador é necessária. Desligue e ligue seu equipamento novamente antes de executar os passos a seguir para verificar as instalações que acabou de realizar.

Abra um powershell (não é necessário privilégio administrativo)

3.1. Listar todos os pacotes instalados com chocolatey

```
> choco list
```

E os demais programas:

```
> git --version
```

```
> node --version
```

```
> java --version
```

```
> nvm --version
```

```
> scrcpy -version
```

ATENÇÃO! O screenCopy, quando executado para espelhamento de tela do celular no computador, vai solicitar autorização no dispositivo. Caso o dispositivo não tenha o "Modo desenvolvedor" ativado, é necessário ativar antes de usar este programa.

No Android o processo pode variar um pouco conforme a versão do sistema.

Verifique sua versão do Android e pesquise sobre este procedimento

Exemplo de texto para pesquisa: **como ativar o modo desenvolvedor no Samsung A15**

Siga as instruções para desbloqueio dessa funcionalidade

4. Pesquisar outros pacotes para instalar com chocolatey

Buscando informações sobre o navegador Google Chrome, por exemplo

```
> choco search google chrome
```

Instalando o Google Chrome mais recente

```
> choco install googlechrome -y
```

5. Configurando o VSCode (instalação de extensões)

****Geral****

- GitLens by GitKraken
- Code Spell Checher by Street Side Software
- Preview by Haixin Chen
- Live Preview by Microsoft
- Thunder client by Thunder Client

****Para Android****

- Prettier by Prettier.io
- React Native Tools by Microsoft
- ESLint by Microsoft
- Color Highlight by Sergii N

6. Acessando o github no prompt com segurança no Windows

Etapas:

1. Verificar se já existem chaves SSH em seu computador
2. Criar um par de chaves ssh
3. Adicionar sua chave publica SSH na sua conta no github
4. Instalar o GitHub cli (GH)
5. Configurar o GH
6. Outras Referências

6.1. Verifique se existem chaves SSH em seu computador

Abra um prompt (CMD) e veja se o diretório ".ssh" existe e se possui arquivos.

```
C:\Users\usuario> dir
```

Caso não exista um diretório ".ssh", crie com o seguinte comando:

```
C:\Users\usuario> mkdir .ssh
```

Se existir, talvez já tenha alguma chave presente, tais como:

```
C:\Users\usuario> cd .ssh  
C:\Users\usuario\.ssh> dir
```

- chave (chave privada)
- chave.pub (chave publica)
- id_rsa.pub
- id_ecdsa.pub
- id_ed25519.pub

6.2. Crie um par privado/público de chaves SSH para utilizar

Para criar chaves que utilizaremos com o GitHub, devemos usar o mesmo e-mail que foi utilizado na criação da conta do GitHub.

Opção 1:

```
C:\Users\usuario\.ssh> ssh-keygen -t ed25519 -C "usuario@provedor.com"
```

Caso este comando não funcione, tente a Opção 2:

```
C:\Users\usuario\.ssh> ssh-keygen -t rsa -b 4096 -C "seu_email@provedor.com"
```

O exemplo a seguir é o resultado do uso da opção 1

```
C:\Users\usuario\.ssh>ssh-keygen -t ed25519 -C "usuario@provedor.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (C:\Users\usuario\.ssh/id_ed25519):
Aperte ENTER
Enter passphrase (empty for no passphrase): Aperte ENTER
Enter same passphrase again: Aperte ENTER
Your identification has been saved in C:\Users\usuario\.ssh/id_ed25519
Your public key has been saved in C:\Users\usuario\.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:1Z12oDtQQtb6Ib3La2W+sCQ97/0BsvpfDmLS/+fnwJ8 usuario@provedor.com
The key's randomart image is:
+--[ED25519 256]--+
|      .+..  . |
|      . +.o o |
|      ooo + . |
|      .o.o. . |
|      S  +o=  |
|      .*o*   |
|      .+=Oo+  |
|      .o*==+* |
|      ..o+++E@|
+-----[SHA256]-----+
```

Explicação sobre a execução deste comando:

Um novo par de chaves SSH é criado, usando a `string` do e-mail fornecido como uma etiqueta. É solicitado um nome para o arquivo da chave. Ao pressionar ENTER o local padrão e o nome sugerido são utilizados. Caso você tenha criado chaves SSH anteriormente, `ssh-keygen` pode pedir que você reescrever a chave com mesmo nome.

Nesse caso, recomenda-se criar uma chave SSH personalizada, onde o usuário pode substituir `id_ALGORITHM` pelo nome da chave personalizada. Quando for solicitada uma frase secreta segura (Ex: BatatinhaQuandoNasceEspalhaRamaPeloChao) essa informação não será preenchida para facilitar o uso das chaves, pois a frase pode ser solicitada posteriormente, dificultando seu uso.

Verifique as chaves criadas:

```
C:\Users\usuario\.ssh> dir
...
06/03/2025  18:16                411 id_ed25519
06/03/2025  18:16                103 id_ed25519.pub
```

Para visualizar o conteúdo da chave pública:

```
C:\Users\usuario\.ssh> type id_ed25519.pub
SHA256:1Z12oDtQQtb6Ib3La2W+sCQ97/0BsvpfDmlS/+fnwJ8 usuario@provedor.com

C:\Users\usuario\.ssh>
```

6.3. Adicionar sua chave pública SSH na sua conta no GitHub

- Acesse: <https://www.github.com/usuario>
- Faça o "Sign in" com usuário e senha
- Clique no ícone do usuário no canto superior direito.
- Escolha no menu a opção Configurações/Settings.
- No menu existente no lado esquerdo, na seção "Acesso", clique em "Chaves SSH e GPG".
- Na seção Chaves SSH, clique no botão "Nova chave SSH"
- No campo Título, adicione uma descrição como "Laptop pessoal"
- No campo Tipo de chave escolha "Chave de autenticação"
- No campo Chave, cole a chave copiada para a área de transferência (SHA256:1712...)
- Clique no botão "Adicionar chave SSH".
- Pronto!

6.4. Instalando o GitHub cli (command line interface, também conhecido como GH)

Acesse o website: <https://cli.github.com>

Faça o download para windows:

https://github.com/cli/cli/releases/download/v2.68.1/gh_2.68.1_windows_amd64.msi

Instale o "gh" com privilégio administrativo

6.5. Configurando o GH

Depois de instalar o programa, abra um terminal com "GitHub CMD" ou "GitHub Bash"

Faça a autenticação no GitHub:

```
> gh auth login

? What account do you want to log into? GitHub.com
? What is your preferred protocol for Git operations on this host? SSH
? Upload your SSH public key to your GitHub account?
  Use SKIP se fez o processo do item 3
```

-ou-

Permita o uso da chave indicada

? How would you like to authenticate GitHub CLI? Login with a web browser

! First copy your one-time code: ABCD-1234

Press Enter to open github.com in your browser...

No navegador o código ABCD-1234 será solicitado

✓ Authentication complete.

- gh config set -h github.com git_protocol ssh

✓ Configured git protocol

✓ Logged in as SEU_USUARIO

! You were already logged in to this account

Pronto!

Por padrão, o editor será o "nano", mas é possível utilizar o bloco de notas.

\$ gh config set editor nome_do_editor

Antes de começar a usar em seus projetos, ajuste o nome e e-mail do usuário:

```
C:\Users\usuario> git config --global user.name "Your Name"
```

```
C:\Users\usuario> git config --global user.email  
"youremail@yourdomain.com"
```

6.6. Criando um novo repositório com GH

```
C:\Users\usuario\projetos\android> gh repo create
```