



AULA 06 - Vídeo 2

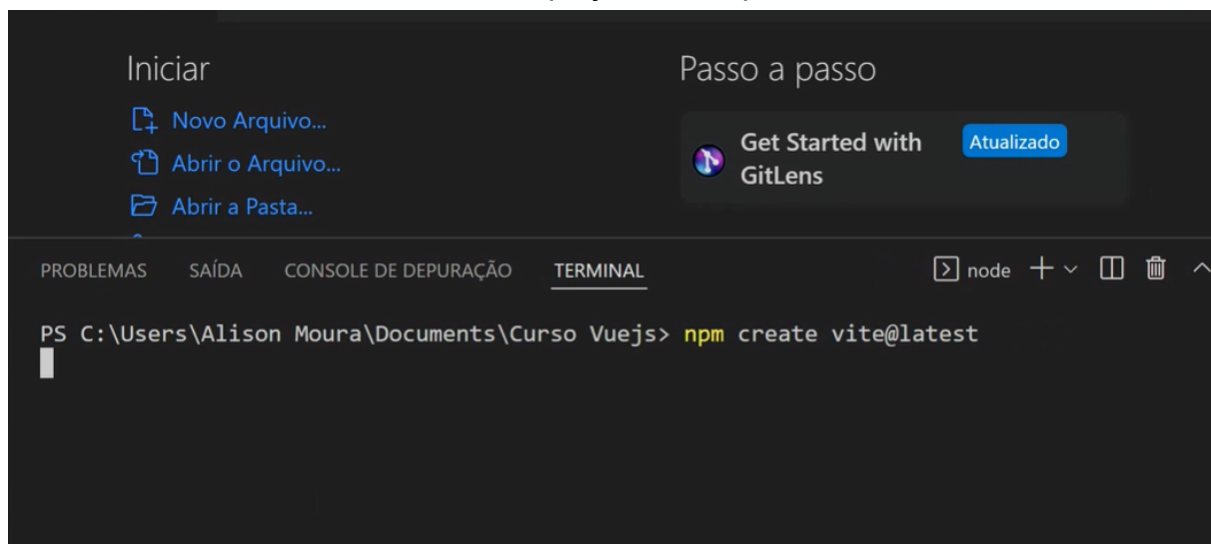
Introdução

Dando seguimento a aula anterior a qual já vimos o que é e para que serve o vite, que é um framework, uma biblioteca que funciona com você e outros frameworks, uma ferramenta para produção de projetos, também já tem um servidor http interno, e já monta um ambiente de desenvolvimento interno.

Seguindo a documentação que estamos vendo:

<https://vitejs.dev/guide/#trying-vite-online>

abriremos o Vscode e iniciaremos um projeto com npm



Seguindo a vídeo aula:

Ok to proceed? y (para seguir)

Em nome do projeto: perfil-profissional-web

Framework: vue

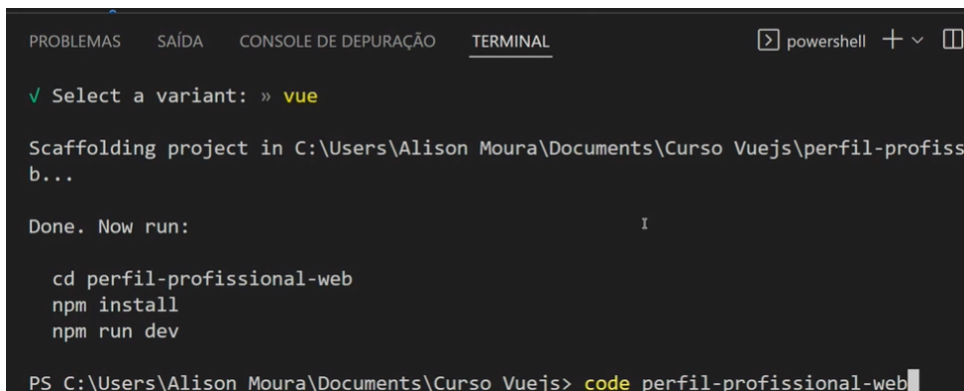
variant: vue

Aparecerá essas mensagens de passos a seguir:

```
Done. Now run:

cd perfil-profissional-web
npm install
npm run dev
```

Mas Abriremos o projeto no vscode, podendo ser no terminal: com o comando 'code' nativo ao instalar vscode.



The screenshot shows a VS Code terminal window with the following content:

```
PROBLEMAS  SAÍDA  CONSOLE DE DEPURAÇÃO  TERMINAL  powershell + -
```

```
✓ Select a variant: » vue

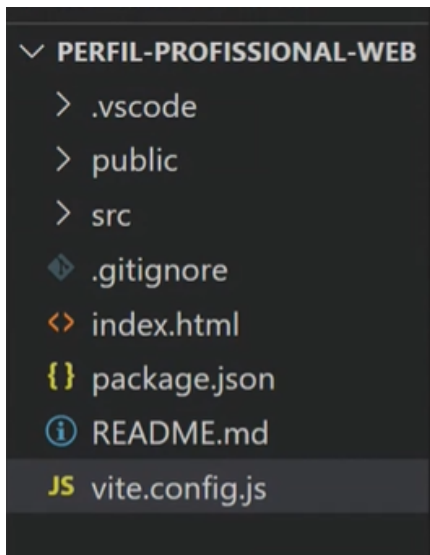
Scaffolding project in C:\Users\Alison Moura\Documents\Curso Vuejs\perfil-profiss
b...

Done. Now run:

cd perfil-profissional-web
npm install
npm run dev

PS C:\Users\Alison Moura\Documents\Curso Vuejs> code perfil-profissional-web
```

O 'scaffolding project', o template que ele nos dá após a instalação tem a seguinte estrutura



The screenshot shows a file explorer view of a project named 'PERFIL-PROFISSIONAL-WEB'. The structure is as follows:

- ▼ PERFIL-PROFISSIONAL-WEB
 - > .vscode
 - > public
 - > src
 - 📄 .gitignore
 - 📄 index.html
 - 📄 package.json
 - 📄 README.md
 - JS vite.config.js

A estrutura:

'vite.config.js'

possui maneiras de personalizar a configuração vite, alguns comportamentos, servidor local que inicia, etc..

'readme.md'

arquivo leitura , documentação sobre o próprio projeto 'linguagem markdown' anotações, converte em html que pode ser renderizado customizadamente.

'package.json'

Arquivo de dependências que o npm gerencia, suas versões, e os scripts que irá executar como o 'npm run dev', inicia servidor local http, funciona como live server, mas o vite tem o próprio servidor local, o 'npm run build' executa o vite também mas na versão build, que é versão de publicação do projeto, etc.

'index.html'

ja familiar nosso, tem um html simples, ja mudaremos as seguintes linhas

```
<html lang="pt-br">
```

```
<title>Perfil Profissional</title>
```

'gitignore'

basicamente um arquivo para você ignorar, certas pastas e arquivos para o git.

dizemos ao git esses arquivos e pastas você pode ignorar, mesmo se é alterado ou excluído o git vai ignorar

'vscodc'

pasta para você ter as configurações do vscodc, como configurações de formatação mas não vamos mexer muito nela

'public'

arquivos estáticos, como de imagem, que você não precisa que o processo de build do vite utilize, compacte, processe.

porque o build quando é executado a intenção é gerar um resultado um pouco mais minimizado mais compactado e mais leve para a aplicação e o usuário final

'src'

Código fonte do projeto que vai ser processado pela ferramenta de build no caso o vite

'assets'

Arquivos estáticos, parecido com público, mas os arquivos de assets vao no processo de build podendo ser processados, minimizados e compactados, arquivos de estilização como css podem ficar no assets também.

'components'

Série de Arquivos que vão ser componentes, conceito do vue mesmo

'App.vue'

'main.js'

Arquivos com extensão '.vue' conhecidos como single file components (SFC) arquivo único que recebe código Vue.

Todo arquivo '.vue' tem 3 áreas:

Área de script: Onde fica o Javascript.

Área de template: Html em si, o que fica visual.

Área de estilização: O css.

Não é uma regra de escrita, podemos ter arquivos separados e importa-los para utilizar, mas estilizações, funções específicas por exemplo só para o componente desejado usamos juntos às vezes.

Exemplo do 'App.vue' das divisões citadas:

```
<script setup>
// This starter template is using Vue 3 <script setup> SFCs
// Check out https://vuejs.org/api/sfc-script-setup.html#script-setup
import HelloWorld from './components/HelloWorld.vue'
</script>

<template>
  
  <HelloWorld msg="Hello Vue 3 + Vite" />
</template>

<style>
#app {
  font-family: Avenir, Helvetica, Arial, sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  text-align: center;
  color: #2c3e50;
  margin-top: 60px;
}
</style>
```