

```
1
2
3  Paradigmas_de_Programacao_do_VSC
4  {
5
6      BOOL fabio;
7      STRING gregory;
8      FLOAT iago;
9      INT william;
10
11  }
12
13
14
```

## OBJETIVO:

- Analisar um caso de estudo (módulo de terminal, de versionamento de código com GIT, carregamento de extensões e o de renomeação de arquivos, todos do VS Code) e identificar os paradigmas usados.

## PERGUNTAS NORTEADORAS:

- Qual ou quais paradigmas foram os mais vistos no módulo estudado?
- Quais são os motivos por trás da escolha desses paradigmas?
- Havia outras alternativas que pudessem ser melhores?


## Módulo de Terminal



Encontra-se no repositório [github.com/microsoft/vscode/tree/main](https://github.com/microsoft/vscode/tree/main), no diretório “vscode/src/vs/workbench/contrib/terminal”.





Compreende toda a lógica por trás dos terminais que ficam geralmente no canto inferior da interface, nos quais podemos:









- Compilar, depurar e executar códigos.
- Acessar servidores remotos.
- Executar comandos de acordo as variáveis de ambiente locais.
- Selecionar terminais fora do padrão do VSC, desde que sejam compatíveis.

# Metodologia de Pesquisa

vscode / src / vs / workbench / contrib / terminal / 

 **mavosoft** Fix PowerShell shell integration when strict mode is enabled. (#248625)   d34b681 · 3 hours ago  History

Name	Last commit message	Last commit date
 ..		
 browser	Only create store when it's used	5 days ago
 common	Fix PowerShell shell integration when strict mode is enabled. (#248625)	3 hours ago
 electron-sandbox	Track disposables for pty host direct proxy	3 months ago
 test	Expose shell's environment - pwsh (#237415)	4 months ago
 terminal.all.ts	Move most of history into terminalContrib	8 months ago
 terminalContribChatExports.ts	Split terminal contrib exports with impl into new file	8 months ago
 terminalContribExports.ts	Move history settings into contrib	8 months ago

Programming Language .end

# Metodologia de Pesquisa

- Uso do artigo de [DESOSA](#) para compreender a estrutura geral do VS Code e selecionar o escopo do estudo.
- Uso de IAs para identificar o diretório dos módulos e explicar cada arquivo estudado.
  - Microsoft Copilot (integrado ao Microsoft Edge);
  - GitHub Copilot.
- Analisar o uso de paradigmas de programação em todo o módulo, exceto pelos arquivos presentes no diretório *test*.

1  
2  
3 “ ”

4 .

- 5 → terminal.all.ts: Arquivo de importações de arquivos .js vindos
- 6 de " ../terminalContrib;
- 7 → terminalContribChatExports.ts: Arquivo adicional de
- 8 exportações de classes e configurações criado para não causar
- 9 dependências circulares.
- 10 → terminalContribExports.ts: Arquivo de exportações de classes e
- 11 configurações.

12 **PARADIGMA PRINCIPAL: Imperativo**

1 “./terminal.all.ts”

2

3 “./terminalContribExports.ts”

4

5

6

7

8

9

10

11

12

13

14

## “./browser”

- baseTerminalBackend.ts: Controla a comunicação entre o PTY e a interface de usuário.
- detachedTerminal.ts: Gerencia terminais que operam de forma independente do fluxo principal.
- environmentVariableInfo.ts: Gerencia as variáveis de ambiente dentro do terminal integrado.
- remotePty.ts: Lida com a comunicação entre terminais remotos e o ambiente local.
- remoteTerminalBackend.ts: Cria, controla e persiste processos de terminal em servidores remotos.

**PARADIGMA PRINCIPAL: Orientado a Objetos**

OUTROS PARADIGMAS: Imperativo e Funcional.



## “./browser”

- terminal.ts: Arquivo que centraliza toda a lógica do terminal.
- terminal.web.contribution.ts: Contém funções específicas que são utilizadas no contexto web.
- terminalActions.ts: Contém funções bastante utilizadas nas ações da interação humano-terminal.
- terminalCommands.ts: Define os comandos relacionados ao terminal do VSC.
- terminalConfigurationService.ts: Implementa a interface de serviço de configuração do terminal.
- terminalContextMenu.ts: Contém a lógica de menus de contexto no terminal.

**PARADIGMA PRINCIPAL: Imperativo**

OUTROS PARADIGMAS: Orientado a Objetos, Funcional e Lógico.

```
1 “./terminalContextMenu.ts”
2
3 “./terminal.web.contribution.ts”
4
5
6 “./terminal.ts”
7
8
9
10
11
12
13
14
```

## “./common”

- basePty.ts: Classe abstrata que gerencia e mantém conexão com um processo de terminal existente.
- environmentVariable.contribution.ts: Declara EnvironmentVariableService como a implementação única de IEnvironmentVariableService.
- environmentVariable.ts: Define a interface de gerenciamento de variáveis de ambiente.
- environmentVariableService.ts: Implementa os contratos de gerenciamento definidos em IEnvironmentVariableService.
- terminal.ts: Define configurações comuns utilizadas no terminal de comando.
- terminalColorRegistry.ts: Define configurações de cores, majoritariamente em valores constantes.

**PARADIGMA PRINCIPAL: Funcional**

OUTROS PARADIGMAS: Orientado a Objetos, Imperativo e Lógico.

```
1 “./basePty.ts”
2
3 “./environmentVariableService.ts”
4
5 “./terminalColorRegistry.ts”
6
7
8
9
10
11
12
13
14
```

Modulo analizado:main/src/vs/workbench/contrib/files/browser/fileActions.ts

# Paradigmas da renomeação de arquivo

- **Programação funcional**

# Programação funcional no código

```
export function validateFileName(pathService: IPathService, item: ExplorerItem, name: string, os: OperatingSystem): { content: string; severity: Severity } | null {  
    // Produce a well formed file name  
    name = getWellFormedFileName(name);  
  
    // Name not provided  
    if (!name || name.length === 0 || /^s+$/.test(name)) {  
        return {  
            content: nls.localize('emptyFileNameError', "A file or folder name must be provided."),  
            severity: Severity.Error  
        };  
    }  
}
```

Modulo analizado: main/src/vs/workbench/contrib/files/browser/fileActions.ts

## Foi bem utilizado?

Sim, a programação funcional foi bem utilizada pois para a função de renomeação de arquivos, pois existem várias validações se o nome do arquivo pode ser o definido pelo usuário

```
if (name !== item.name) {  
  // Do not allow to overwrite existing file  
  const child = parent.getChild(name);  
  if (child && child !== item) {  
    return {  
      content: nls.localize('fileNameExistsError', 'A file or folder **{0}** already exists at this location. Please choose a different name.'), name),  
      severity: Severity.Error  
    };  
  }  
}
```

```
// Relative paths only  
if (name[0] === '/' || name[0] === '\\') {  
  return {  
    content: nls.localize('fileNameStartsWithSlashError', 'A file or folder name cannot start with a slash.'),  
    severity: Severity.Error  
  };  
}
```

# Referências

- [https://www.researchgate.net/profile/Nara-Bigolin-2/publication/350482515\\_Paradigmas\\_de\\_programacao\\_Uma\\_introducao/links/632a69220a708521500b4aed/Paradigmas-de-programacao-Uma-introducao.pdf](https://www.researchgate.net/profile/Nara-Bigolin-2/publication/350482515_Paradigmas_de_programacao_Uma_introducao/links/632a69220a708521500b4aed/Paradigmas-de-programacao-Uma-introducao.pdf)
- [https://d1wqtxts1xzle7.cloudfront.net/63643201/Artigo\\_Paradigmas\\_de\\_Programacao20200616-43235-1rw6ubt-libre.pdf?1592304776=&response-content-disposition=inline%3B+filename%3DParadigmas\\_de\\_Programacao.pdf&Expires=1744680299&Signature=PC8nte1CySxol3rkbeAU3lmjnm4GxjrGiop4sTG8npWXaqqaxYNoS99MchG6tnBm9nyRA6jtAQHCQ-wVXTJrur4gmdxpuW2HiWoAKR-97bwU7InCjSx8lmNEgGYFgBVDji7Z6ZuToOS1ZQfxRFdICPhBn56S7zn2dh1PtHFzYaNhcl4MPzJ6ulTilKweAaPZRauyLFvfRCh~j6KoxzXCPlmGZH~eDoJS1R5IP287aK6eD8QPQI-QcIS9e~YoKu9q4F6KsHCiDnym8Cw8M7~OobM2f844lklazo~xfpa5jU4BbBjoRaeAWJ1k-rhWjHh7Diof6m32BN9HN2Xk7useA\\_\\_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA](https://d1wqtxts1xzle7.cloudfront.net/63643201/Artigo_Paradigmas_de_Programacao20200616-43235-1rw6ubt-libre.pdf?1592304776=&response-content-disposition=inline%3B+filename%3DParadigmas_de_Programacao.pdf&Expires=1744680299&Signature=PC8nte1CySxol3rkbeAU3lmjnm4GxjrGiop4sTG8npWXaqqaxYNoS99MchG6tnBm9nyRA6jtAQHCQ-wVXTJrur4gmdxpuW2HiWoAKR-97bwU7InCjSx8lmNEgGYFgBVDji7Z6ZuToOS1ZQfxRFdICPhBn56S7zn2dh1PtHFzYaNhcl4MPzJ6ulTilKweAaPZRauyLFvfRCh~j6KoxzXCPlmGZH~eDoJS1R5IP287aK6eD8QPQI-QcIS9e~YoKu9q4F6KsHCiDnym8Cw8M7~OobM2f844lklazo~xfpa5jU4BbBjoRaeAWJ1k-rhWjHh7Diof6m32BN9HN2Xk7useA__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA)
- DESOSA (28/04)
- GitHub - microsoft/vscode: Visual Studio Code



## Anotações

Ideia de estudo de caso: VS Code, módulo de terminal; fontes propostas:

1. DESOSA Com anotações (Texto da arquitetura do vscode com as seções que indicam/relacionam com paradigmas sublinhadas)
2. "The Architecture of VS Code" – por Erich Gamma (Microsoft)
3. Análise Crítica do código pelo ChatGPT

Doc de apoio:

[https://docs.google.com/document/d/1l10ejcFvL1DJ-K1KinTLGzFo05jRYJh1v\\_EHa5ov0I/edit?tab=t.0#heading=h.3n0c8zljzlti](https://docs.google.com/document/d/1l10ejcFvL1DJ-K1KinTLGzFo05jRYJh1v_EHa5ov0I/edit?tab=t.0#heading=h.3n0c8zljzlti)

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14

. END