

# VS Code: Produtividade Infinita

---



# Sumário

- VS Code: Produtividade Infinita ..... 1**
  - Produtividade 200% - Iniciando no VS Code da melhor maneira ..... 5**
    - O básico do VS Code..... 6**
    - Status Bar ..... 10**
    - Activity Bar..... 16**
  - Produtividade 500% - Aprimorando nas tarefas do dia a dia ..... 22**
    - Atalhos ..... 23**
    - Operações com a Paleta de Comandos ..... 27**
  - Produtividade 1000% - Se tornando um mestre do VS Code..... 30**
    - Customizando o VS Code..... 31**
    - Depuração de Código (Debug)..... 36**
  - Produtividade +8000% - Evoluindo ainda mais nas tarefas ..... 41**
    - KeyMaps ..... 42**
    - Emmets..... 44**
    - Snippets ..... 45**
  - Produtividade 250000% - Indo rumo ao infinito ..... 47**
    - Controle de Versões com GIT ..... 48**
    - Lints..... 50**
    - Recomendações ..... 52**
    - Formatadores (Beautification) ..... 53**
  - Produtividade 1000000% - Tópicos Avançados, apenas um passo para o infinito..... 54**
    - Whitespaces ..... 55**
    - Auto Save..... 55**
    - Customizar barra de título ..... 56**

<b>Selecionar apenas uma área específica .....</b>	<b>57</b>
<b>Fira Code com Font Ligatures.....</b>	<b>58</b>
<b>Linhas Vazias.....</b>	<b>59</b>
<b>Breadcrumbs (Auxiliares de navegação).....</b>	<b>59</b>
<b>Delimitar Seleção.....</b>	<b>59</b>
<b>Testes Unitários.....</b>	<b>60</b>
<b>Produtividade ∞ - Alcançando o Infinito .....</b>	<b>61</b>
<b>Extensões - Gerais.....</b>	<b>62</b>
<b>Extensões - Markdown.....</b>	<b>75</b>
<b>Extensões - JSON.....</b>	<b>77</b>
<b>Extensões - Python.....</b>	<b>77</b>
<b>Extensões - Java.....</b>	<b>78</b>
<b>Extensões - C# e DotNetCore.....</b>	<b>78</b>
<b>Extensões - C/C++.....</b>	<b>79</b>
<b>Extensões - Banco de Dados.....</b>	<b>80</b>
<b>Extensões - HTML.....</b>	<b>81</b>
<b>Extensões - TypeScript.....</b>	<b>83</b>
<b>Extensões - JavaScript.....</b>	<b>84</b>
<b>Extensões - Angular.....</b>	<b>85</b>
<b>Extensões - React.....</b>	<b>86</b>
<b>Extensões - Node.JS.....</b>	<b>87</b>
<b>Extensões - Vue.....</b>	<b>88</b>
<b>Extensões - PHP.....</b>	<b>89</b>
<b>Extensões - BASH.....</b>	<b>90</b>
<b>Extensões - PowerShell.....</b>	<b>91</b>
<b>Extensões - Ruby.....</b>	<b>92</b>

<b>Extensões - Go .....</b>	<b>93</b>
<b>Extensões - Haskell.....</b>	<b>94</b>
<b>Produtividade <math>\infty</math> - Chegamos ao infinito .....</b>	<b>95</b>
<b>Produtividade <math>\infty^2</math> - Os Extras.....</b>	<b>96</b>
<b>Extra: Materiais de Ensino .....</b>	<b>97</b>
<b>Extra: Lista de Diversos Outros Atalhos .....</b>	<b>97</b>
<b>Extra: Personalização .....</b>	<b>98</b>
<b>Bônus.....</b>	<b>103</b>
<b>Bônus: Cursos Gratuitos de VS Code .....</b>	<b>103</b>
<b>Autores.....</b>	<b>104</b>
<b>Felipe Cabrera Ribeiro dos Santos.....</b>	<b>104</b>
<b>ByLearn Cursos Online.....</b>	<b>104</b>

---

# **Produtividade 200% - Iniciando no VS Code da melhor maneira**

---

Aprenda o básico do VS Code de uma forma bem produtiva e eficaz, para saber como tirar vantagens até mesmo das opções, ferramentas e menus mais básicos e fáceis do VS Code!

---

## O básico do VS Code

Aprenda o básico a se saber do VS Code, com isso você já terá o mínimo para poder utilizá-lo em qualquer um de seus projetos.

### O que é o VS Code

O Visual Studio Code é um editor de código-fonte desenvolvido pela Microsoft para Windows, Linux e macOS. Ele inclui suporte para depuração, controle Git incorporado, realce de sintaxe, complementação inteligente de código, snippets e refatoração de código. Ele também é customizável, fazendo com que os usuários possam mudar o tema do editor, teclas de atalho e preferências.

Ele também é um software livre e de código aberto!



## A interface de usuário



- **A:** Activity Bar é a barrinha lateral onde você pode mudar as 'Views' da SideBar, como Explorer, Pesquisas, Git, Debug, Extensões (veremos mais a fundo abaixo);
- **B:** Side bar é onde teremos as opções específicas da 'View' selecionada na Activity Bar;
- **C:** Editor Groups são os grupos dos editores, a parte principal do programa, onde escrevemos nosso código, ele é formado por Abas (tabs), que são os arquivos abertos no momento;
- **D:** Aqui temos os painéis do VS Code, nele temos nosso Output (a saída do programa), o Console de Debug, Terminais e Problemas (Erros e Warnings, que veremos adiante);
- **E:** Por fim, a status bar, a barrinha no canto inferior, serve para facilitar nossa vida com alguns atalhos rápidos e bem úteis que falaremos mais no decorrer desse livro.

## Trabalhando com Workspace

Workspace serve como uma 'pasta' especial do VS Code onde ficarão todos os arquivos do nosso projeto, todos os arquivos que iremos trabalhar na janela atual do VS Code. O melhor de tudo é que cada Workspace pode armazenar configurações próprias e salvar os estados atuais de cada arquivo/projeto.

## Múltiplos arquivos (editores)

Por que editar um arquivo por vez se podemos editar mais? Um bom programador sempre edita no mínimo 5 arquivos por vez! Brincadeiras a parte, visualizar vários arquivos por vez é algo muito bom, você consegue compará-los, analisá-los e até alterá-los se necessário.

Para isso, basta apertar *Ctrl + J* e você irá dividir a tela do VS Code.

## Editando várias linhas por vez

Um recurso bem interessante do VS Code é a opção de múltiplos cursores ao mesmo tempo. Para tal função, precisamos apenas segurar a tecla *ALT* e clicar nas posições onde queremos que os cursores sejam inseridos.

## Renomeando de forma fácil

Esqueça ficar renomeando variáveis uma a uma! Basta selecionar a variável, método ou classe desejada e apertar *F2*. Após renomear a selecionada, todas as ocorrências no projeto também mudarão para o que você escolheu.



## **Usando o terminal embutido**

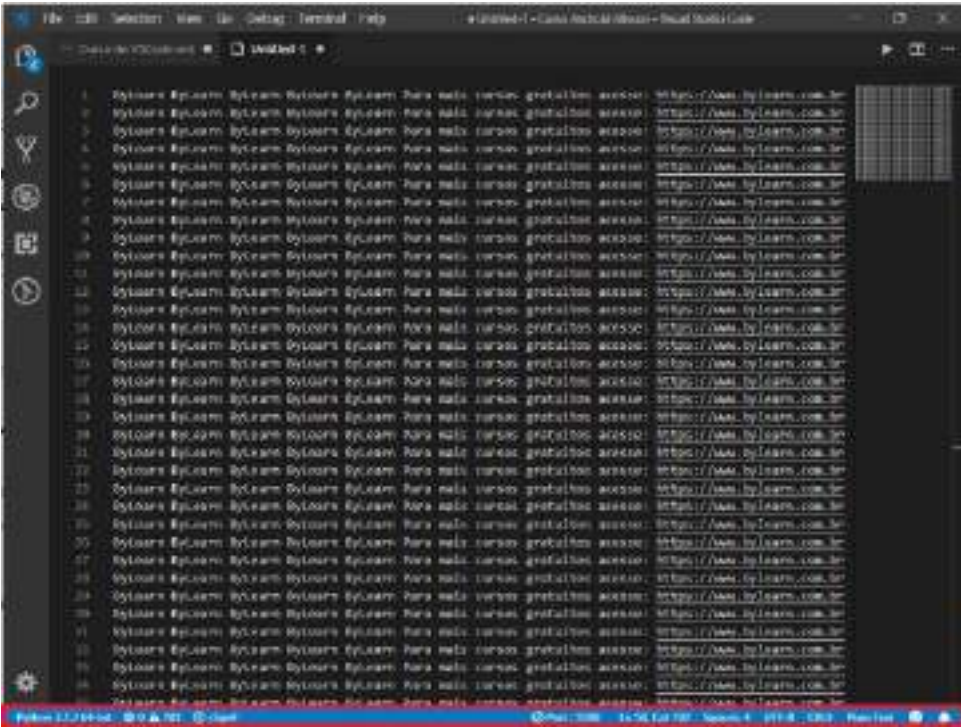
Você sabia que o VS Code tem um terminal embutido? Basta apertar *Ctrl + Shift + `* e será criado um novo terminal. Lembre-se sempre de usá-lo quando for preciso algum comando bash, por exemplo. Abrir o terminal do nativo do sistema é apenas um gasto de tempo.

## **Pesquise em vários lugares de uma vez só**

Usando a função Search da Activity Bar (logo abaixo explicaremos melhor) você pode procurar algo em Todos os arquivos do projeto, te poupando e muito o tempo.

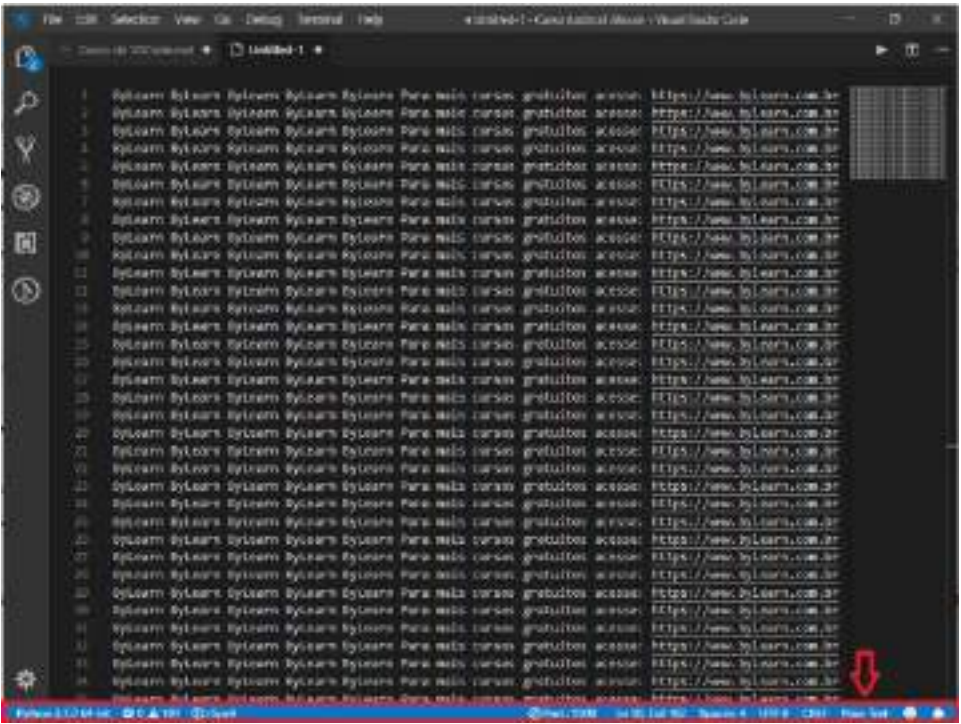
# Status Bar

A Status bar é a barrinha que fica na parte inferior do VS Code, possuindo originalmente uma coloração azul clara, possuindo várias opções padrões que irão acelerar nossa produtividade.



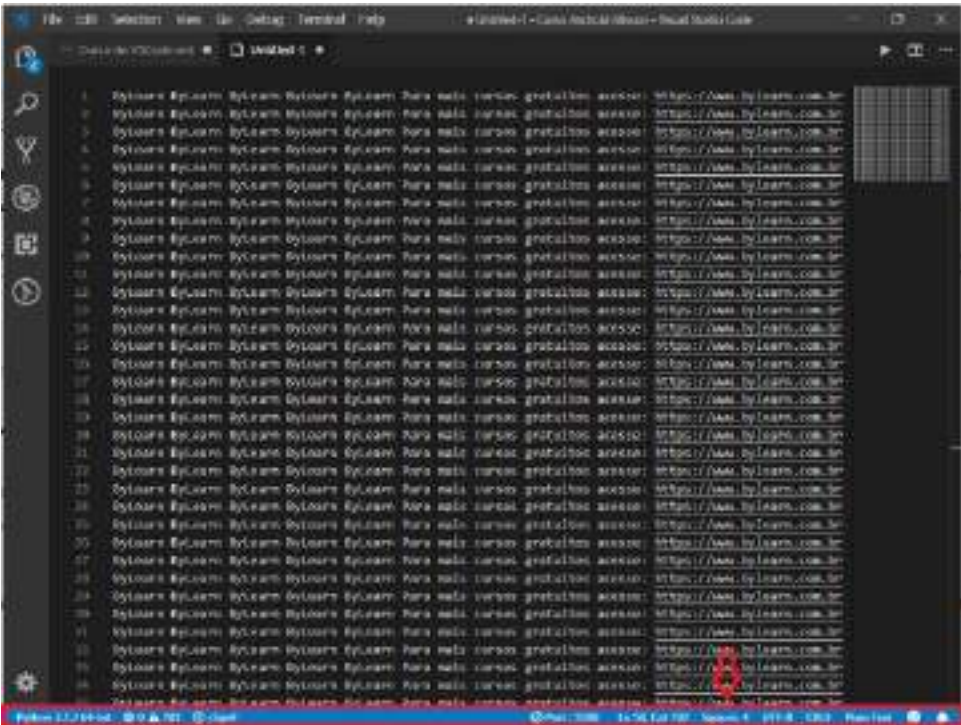
# Mudando o tipo de arquivo

Na Status Bar podemos alterar o tipo de arquivo antes de salvar ou caso por algum motivo o VS Code tenha errado na hora de detectar. Para isso, basta apertar na opção ao lado esquerdo do rosto sorridente (no canto direito), onde provavelmente estará "Plain Text" caso não tenha escolhido nenhum tipo ainda.



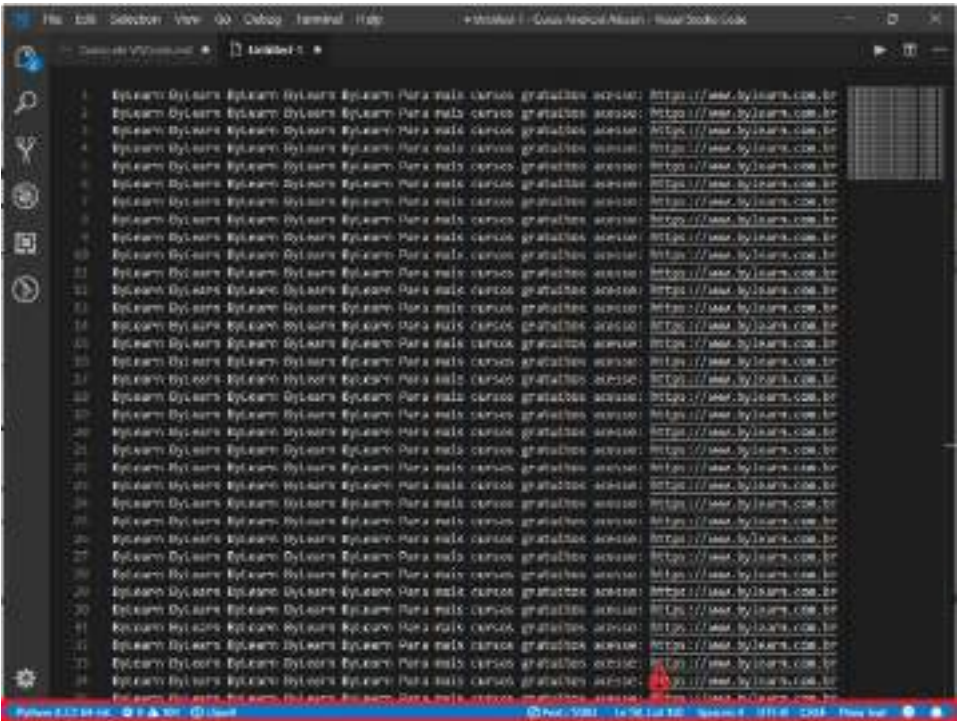
# Mudando o formato da indentação

Na parte direita da Status Bar também temos "Spaces: 4", lá podemos mudar o formato da indentação (Espaço ou Tab).



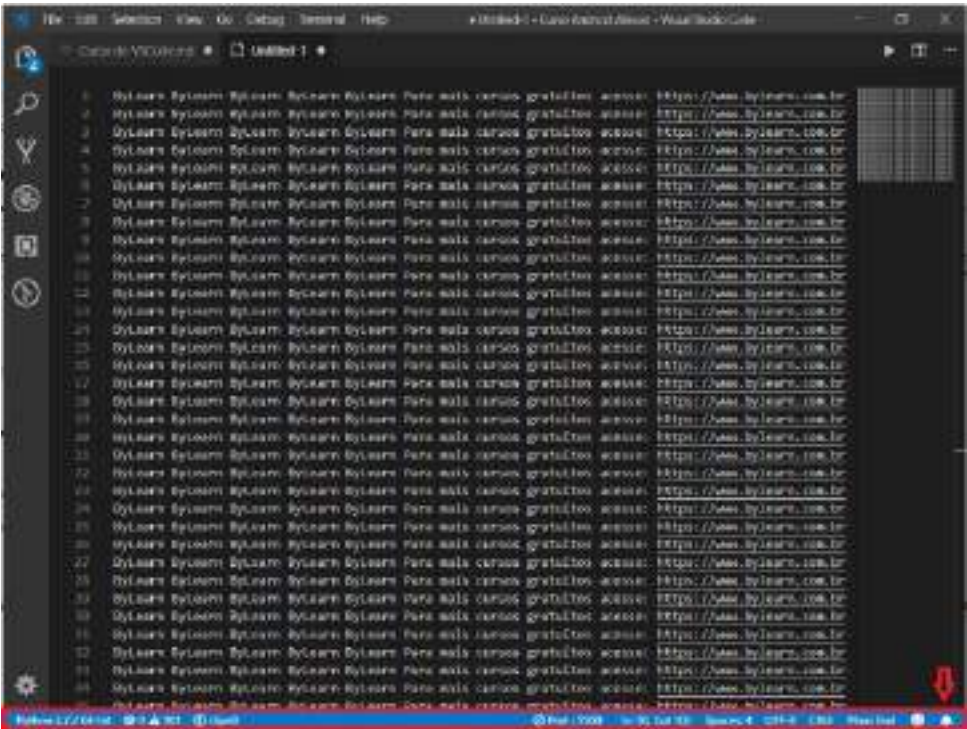
# Navegando entre linhas

Um arquivo grande é sempre horrível, ainda mais quando precisamos ir para uma determinada linha. Porém, com o VS Code nós podemos selecionar a linha desejada e o cursor é "Teletransportados" para lá ;) Para isso basta apertar na opção "Ln , Col" na parte direita da Status Bar.



# Notificações

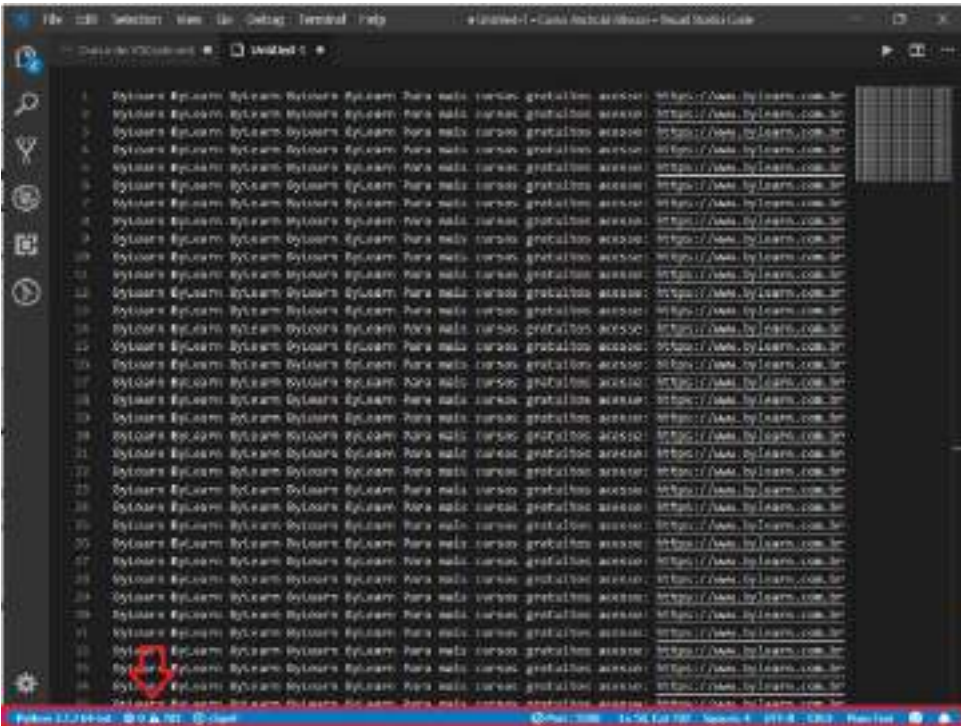
Você também pode ter acesso as ultimas notificações do VS Code, sejam elas de erro, instalação de extensões, atualizações, pedidos de configuração e etc... Para isso basta clicar no sininho no canto direito da Status Bar.





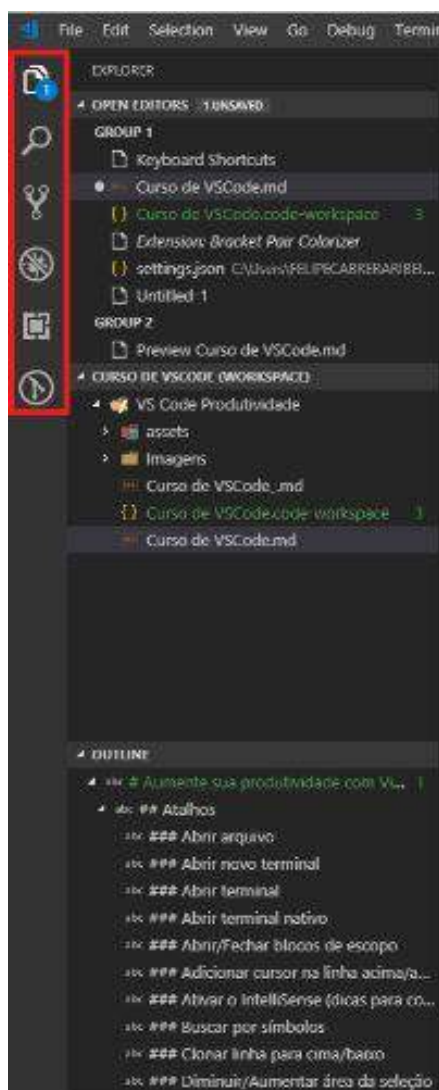
# Warnings e Erros

Para quem já usou o Visual Studio, a importância dessa função é bem clara! Nela podemos saber todos os erros no nosso código e alguns avisos, seja de melhorias, possíveis problemas ou dica para seguir algum guideline. Basta apertar o botão do 'X' ou triângulo na parte esquerda da Status Bar.



## Activity Bar

É a famosa barra lateral que fica no canto esquerdo do VS Code. PS: A parte entre a Activity Bar e a parte onde escrevemos é chamada de Side Bar :).





## Explorer

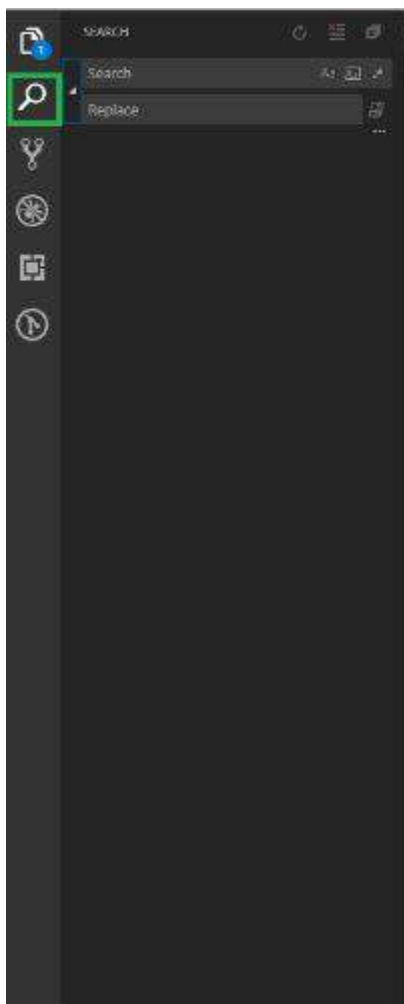
Primeira opção da Activity Bar, é nele que estão todos os arquivos do nosso Workspaces e todos os outros abertos na sessão atual do VS Code.

1. Open Editors: Arquivos abertos e seus respectivos grupos;
2. Workspace: Nosso Workspace atual e todos os arquivos e pastas contidos nele;
3. Outline, indica os 'títulos' e 'sessões' do nosso arquivo atual se for um texto, ou variáveis, métodos e classes se estivermos com algum projeto aberto. Bem útil para servir de atalho para navegação.



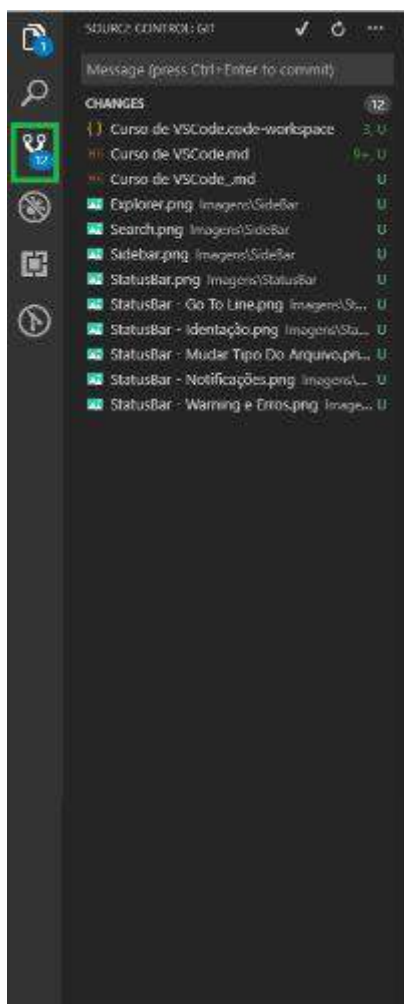
## Search

Área responsável pelas pesquisas, tanto no arquivo atual quanto no workspace e em todos os arquivos abertos no VS Code (pesquisando em todos de uma vez só). Também é possível usar a função Replace para substituir todas as ocorrências da busca.



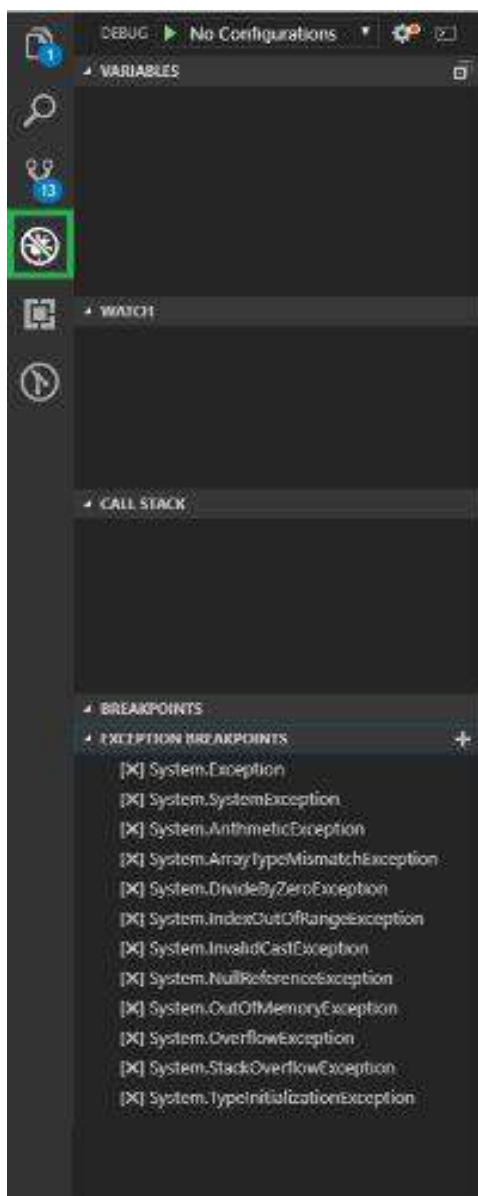
## Git

Para os amantes do Git, o VS Code já vem com uma opção nativa bem potente (que veremos mais ao final), que nos permite controlar totalmente o repositório git do projeto, enviando commits, fazendo pushes e pulls, merges e etc...



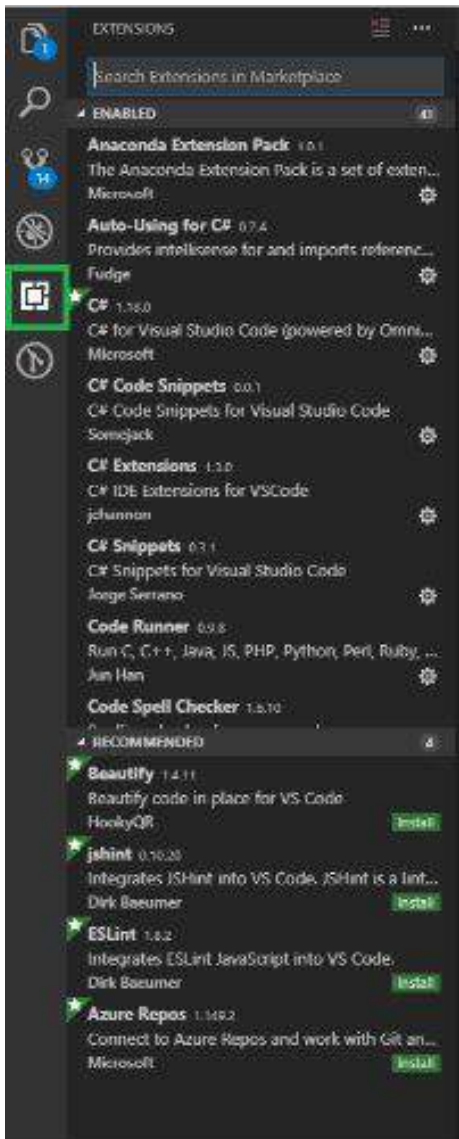
## Debug

Uma das áreas mais importante da Activity Bar ao mexermos com algum projeto, é aqui que podemos debugar nosso código e analisar passo a passo o que está acontecendo no projeto.



## Extensões

Outra área bem importante e interessante é a parte de extensões, aqui podemos instalar várias e várias extensões que vão nos ajudar em 1000% a aprimorar nossa produtividade (você verá várias delas no final do curso).



---

## **Produtividade 500% - Aprimorando nas tarefas do dia a dia**

---

Eleve o seu poder de programador a outro nível, torne tudo sua rotina cada vez mais fácil com os melhores atalhos e comandos rápidos do VS Code.

---

## Atalhos

Conheça os principais e mais úteis atalhos do VS que vão aumentar e muito a sua produtividade.

### Atalhos - VS Code Workspace

#### Ocultar a Activity Bar (barra esquerda)

Ctrl + B

#### Paleta de Comandos

Método 1: Ctrl + Shift + P Método 2: F1

#### Buscar por símbolos

Ctrl + Shift + O

#### Abrir terminal

Ctrl + '`

#### Abrir novo terminal

Ctrl + Shift + '`

#### Abrir terminal nativo

Ctrl + Shift + C

#### Mostrar Erros

Ctrl + Shift + M

## **Navegar entre erros**

F8 e Shift + F8 para avançar e retroceder nos erros.

## **Atalhos - Arquivo**

### **Abrir um arquivo**

Ctrl + P

### **Formatar o arquivo**

Método 1: Ctrl + K , F Método 2: Shift + Alt + F

### **Mover arquivo para o outro lado**

Ctrl + Alt + Seta Esquerda/Direita

### **Diminuir/Aumentar área da seleção**

Shift + Alt + Seta Esquerda/Direita

### **Visualizar Preview**

Ctrl + K, V

### **Abrir/Fechar blocos de escopo**

Fechar: Ctrl + Shift + ´ Abrir: Ctrl + Shift + [

### **Ativar o IntelliSense (dicas para completar a digitação)**

Ctrl + Espaço



**Peek (Mostrar a Definição (F12) sem sair do arquivo atual)**

Alt + F12

## **Atalhos - Linhas**

**Selecionar toda a linha**

Ctrl + L

**Recortar toda a linha**

Ctrl + X (Não tendo nada selecionado).

**Adicionar cursor na linha acima/abaixo**

Ctrl + Alt + Seta Cima/Baixo

**Mover linha para cima/baixo**

Alt + Seta Cima/Baixo

**Clonar linha para cima/baixo**

Shift + Alt + Seta Cima/Baixo

**Ir para o Fim/Início da linha**

Início: Home Fim: End

**Selecionar até o Fim/Início da linha**

Até o Início: Shift + Home Até o fim: Shift + End

## Selecionar toda a palavra

Deixe o cursor em qualquer parte da palavra escolhida e aperte:

*Ctrl + D*

## Selecionar todas as palavras iguais

Um dos atalhos mais legais (e propositalmente deixado por último) é o de selecionar todas as palavras iguais a sua. Para isso, selecione uma palavra como no atalho anterior (*Ctrl + D*) e aperte *D* novamente para cada palavra igual a essa que deseja selecionar.

Exemplo:

```
public int funcao(int x)
{
    x++;
    if(x % 2 != 0)
        x++;

    return x/2;}

```

Deixar uma variável com nome 'X' é assassinar as boas práticas da programação, é impossível saber o que ela está fazendo e qual a necessidade dela no código apenas batendo o olho! Precisamos mudar o nome para algo mais intuitivo, como `filmeID`. Para isso, selecionamos qualquer um dos 'x' e apertamos *Ctrl + D* até que todos estejam selecionados. Depois disso basta escrevermos '`filmeID`' que todos passarão a ter esse nome.

---

## Operações com a Paleta de Comandos

### Permitir acesso via Terminal (Visual Code CLI)

Procure por 'Shell' e selecione "Allow Workspace Shell Configuration"

Agora você pode ir até o terminal (como o CMD, no caso do Windows) e abrir um arquivo no code via: 'code nome\_do\_arquivo.ext'.

### Operações com linha

Procure por 'Line' e você encontrará todos os comandos envolvendo linha, tais como:

- Sort Lines Ascending: Ordenar as linhas de forma alfabética;
- Join Lines: Junta todas as linhas selecionadas em uma só;
- Go to Line...: Vai até a linha selecionada.

### Operações de Desenvolvedor

Para saber as operações e comandos para desenvolvedor basta pesquisar por 'Developer'.

### Comandos do Emmet

Procure por 'Emmet' para saber as principais opções e comandos para uso do Emmet no VS Code.

## **Operações com Arquivos**

Procure por 'File' para saber as principais opções e comandos para os arquivos do VS Code.

## **Operações de 'Teletransporte'**

Ok... Não é tão legal quanto um teletransporte, mas pelo menos é útil. Procurando por 'Go' você terá vários comandos para levar você até certo lugar, como um arquivo específico, linha, símbolo e etc...

## **Operações com Extensões**

Procure por 'Extensions' para saber os principais comandos com extensões, como por exemplo, atualizar as extensões instaladas.

## **Operações para acabar com suas dúvidas**

Para resolver suas dúvidas e entender melhor do VS Code é só procurar por 'Help'.

## **Operações com as Preferências**

Procure por 'Preferences' para poder configurar as opções de preferência, como temas, opções de usuário e atalhos.

## **Operações com o Terminal**

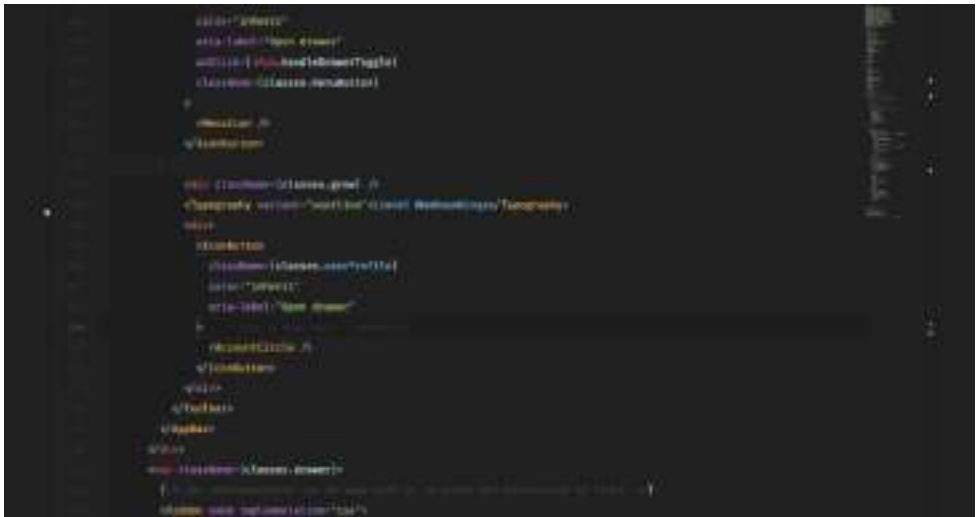
Procure por 'Terminal' e você encontrará os principais comandos para realizar com o terminal embutido do VS Code.

## Operações para Modificar sua visualização

Procure por 'View' e você terá inúmeras opções para editar a visualização padrão do VS Code, como fechar/abrir um editor, mostrar a aba de extensões, aba de busca e etc...

### Modo Zen

Um modo bem legal do VS Code é o Modo Zen, que oculta tudo que pode te atrapalhar a escrever e deixa o editor em tela cheia mostrando apenas a parte onde escrevemos o texto. Para isso basta procurar por 'Zen' na paleta de comandos e selecionar 'View: Toggle Zen Mode'.



---

## **Produtividade 1000% - Se tornando um mestre do VS Code**

---

Vamos levar a produtividade para um outro nível! Neste capítulo aprenderemos a deixar o VS Code a nossa cara e a como depurarmos nosso código de forma fácil e eficaz!

---

## Customizando o VS Code

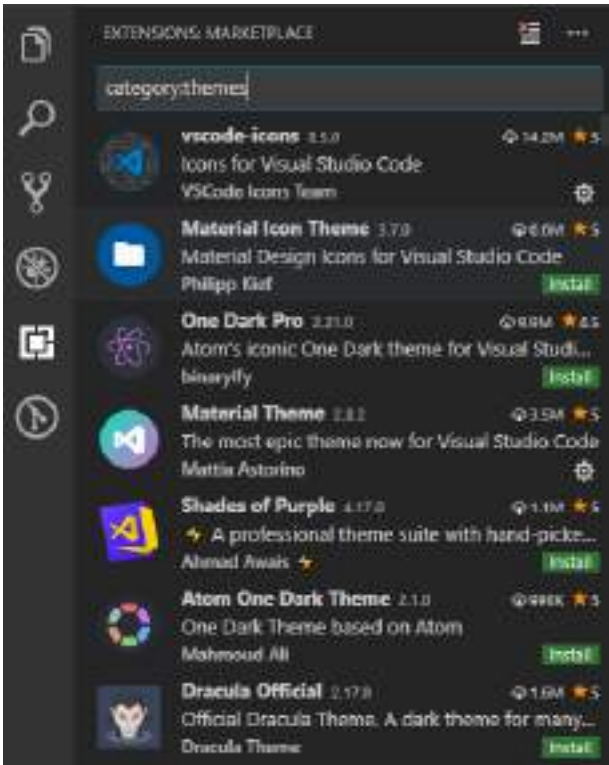
Agora chegou a hora de customizarmos o VS Code, para deixá-lo ainda mais a nossa cara e podermos conseguir uma maior produtividade através da familiaridade com a interface.

De presente, deixaremos no final deste livro algumas dicas de customização, na sessão 'Extras', vale a pena conferir algumas dicas selecionadas a dedos (elas são todas gratuitas) :)

# Temas

Para customizar temas é muito fácil, basta abrir a paleta de comandos e procurar por: "Theme" e selecionar "Preferences: Color Theme" e selecionar qual o tema desejado.

Caso não encontre lá e queira achar algum outro, não tem problema! Neste menu de seleção de temas nós temos uma opção "Install Additional Color Themes" basta apertar nela que somos direcionados a área de extensões filtrando apenas por temas ("category:themes"). Agora só precisamos achar um que nos agrada e instalá-lo!



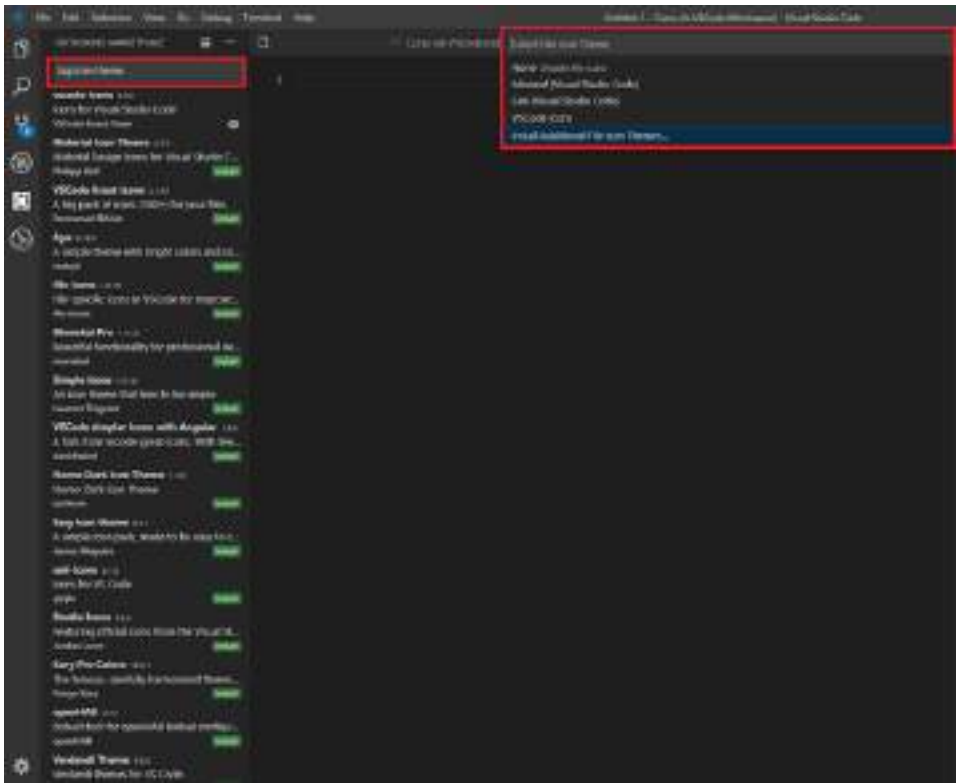




# Ícones

Outra coisa bem legal de se alterar são os ícones. Eles servem, principalmente, para indicar o tipo do nosso arquivo atual, tanto na Side Bar quanto na pesquisa na hora de mudar o tipo do arquivo (como visto na sessão básica do livro).

Para editá-los, assim como fizemos com os Temas, vamos apertar F1 (ou Ctrl + Shift + P) para abrir a Paleta de Comandos e procuramos por "Icon Theme" e assim como com os Temas, caso não encontremos um estilo de ícone do nosso gosto, basta selecionar a opção "Install Additional File Icon Themes" e o VS Code irá nos redirecionar para as extensões filtrando apenas as categorias onde ficam os temas de ícones.

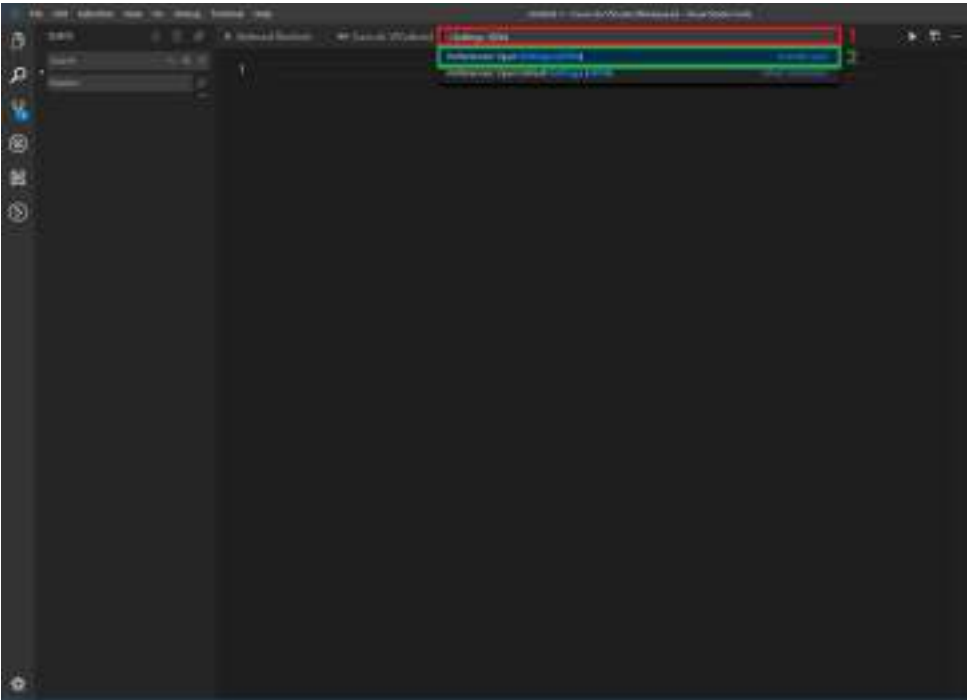


## Customizando via JSON

Por fim, para configurar via JSON nós podemos procurar na paleta de comandos (F1) por "Settings JSON" e selecionar "Preferences: Open Settings (JSON)".

Pronto, agora irá abrir um JSON que nos permite configurar o que quisermos no VS Code :). Todas as configurações feitas previamente (como tema, ícones, fontes e afins) podem ser editados diretamente por lá.

No final deste livro, na sessão Extras também deixaremos um exemplo de configuração via JSON usados por nós da ByLearn para melhorar nossa eficiência e produtividade.



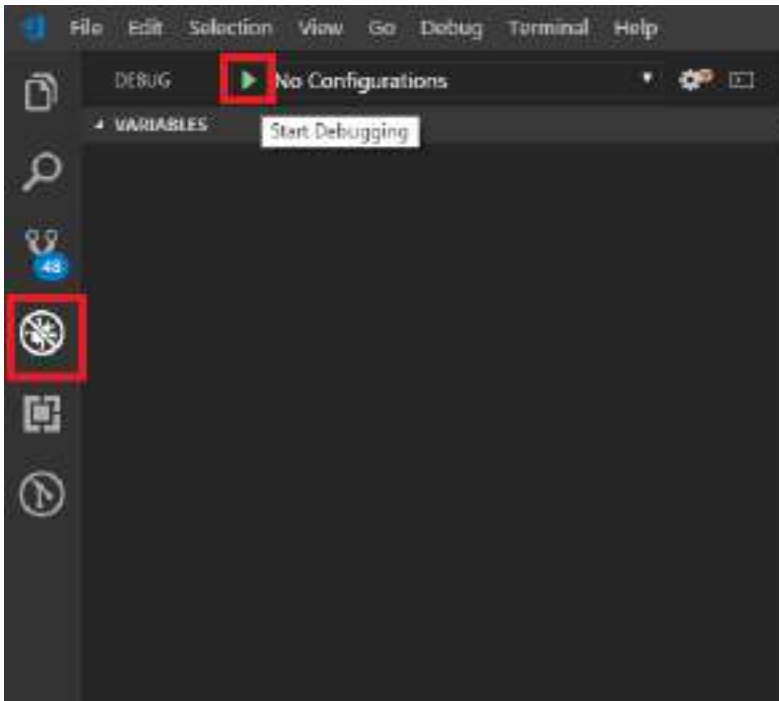
---

## Depuração de Código (Debug)

### Depurando com o VS Code

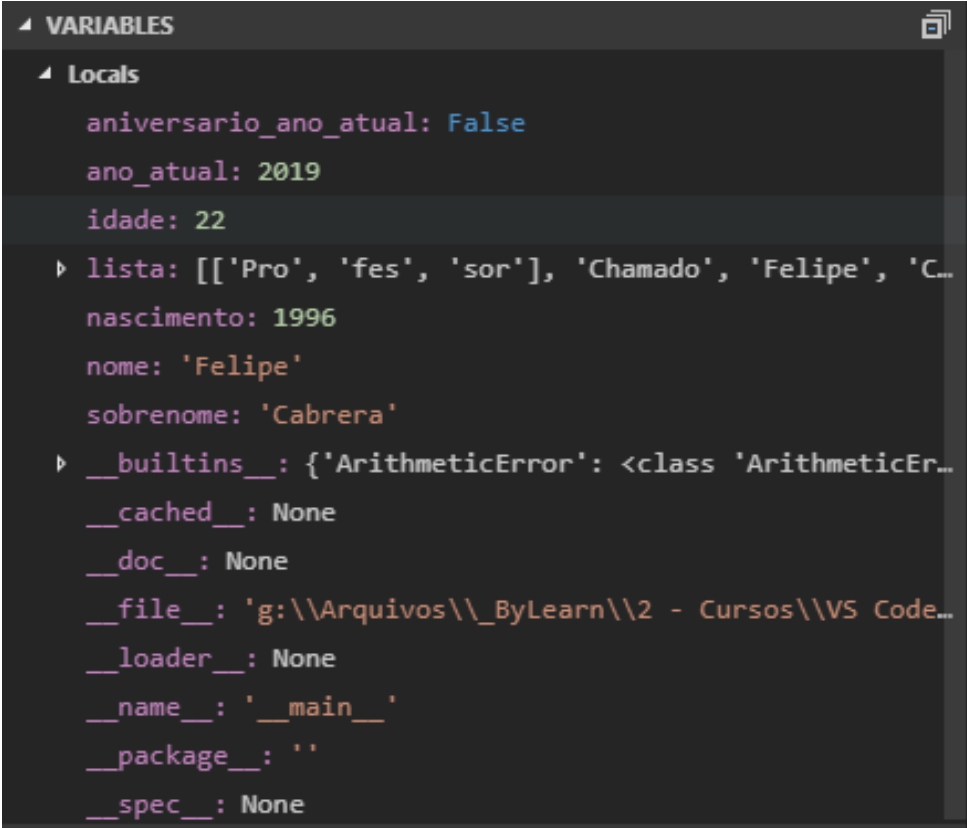
Podemos depurar nosso código no VS Code de forma bem simples, basta na Title Bar (barra de título, na parte de cima) selecionarmos o menu Debug e escolher a opção 'Start Debugging', ou simplesmente apertar o botão F5.

Após isso o debug irá iniciar automaticamente.



## Variables

Indica as variáveis que estão em uso no atual momento no debug, como por exemplo, ao atingir um Breakpoint, mostrará as variáveis em uso e quais seus valores atuais.

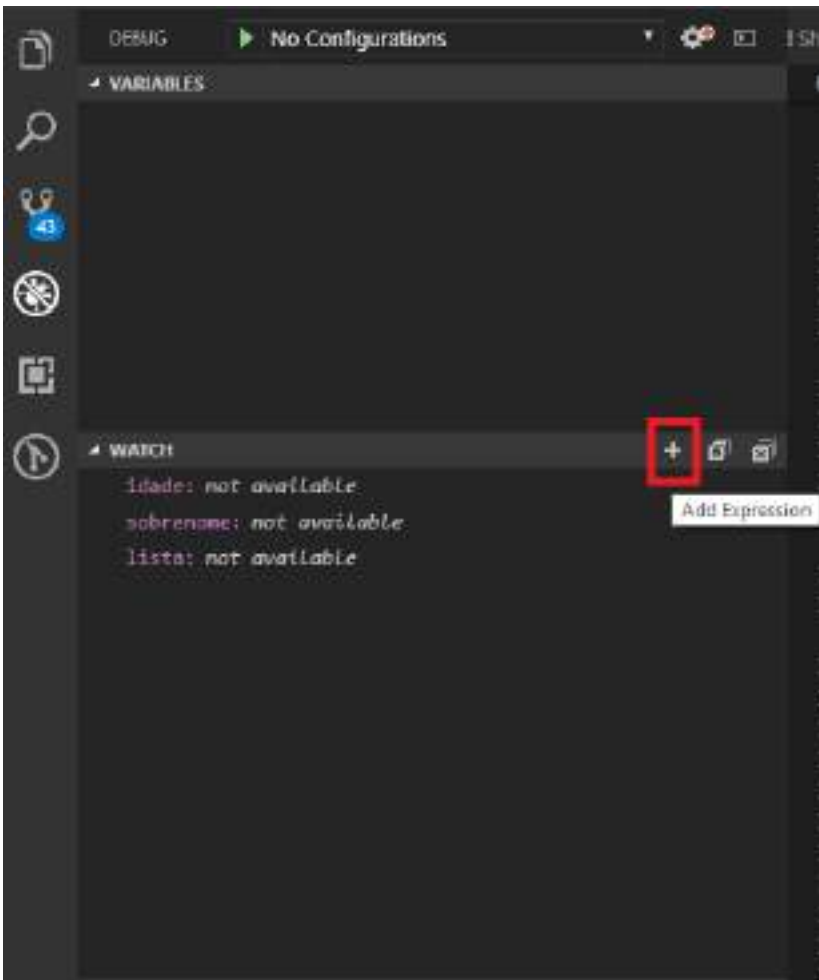
A screenshot of the 'VARIABLES' window in a debugger. The window has a dark background with light-colored text. At the top, it says 'VARIABLES' with a small icon to the right. Below it, there is a section titled 'Locals' with a small arrow icon. The variables listed are: 'aniversario\_ano\_atual: False', 'ano\_atual: 2019', 'idade: 22', 'lista: [['Pro', 'fes', 'sor'], 'Chamado', 'Felipe', 'C...]', 'nascimento: 1996', 'nome: 'Felipe'', 'sobrenome: 'Cabrera'', '\_\_builtins\_\_: {'ArithmeticError': <class 'ArithmeticEr...}', '\_\_cached\_\_: None', '\_\_doc\_\_: None', '\_\_file\_\_: 'g:\\Arquivos\\\_ByLearn\\2 - Cursos\\VS Code...', '\_\_loader\_\_: None', '\_\_name\_\_: '\_\_main\_\_'', '\_\_package\_\_: ''', and '\_\_spec\_\_: None'. The text is color-coded: strings are in quotes, numbers are in green, and class names are in orange.

```
▲ VARIABLES
  ▲ Locals
    aniversario_ano_atual: False
    ano_atual: 2019
    idade: 22
    ▶ lista: [['Pro', 'fes', 'sor'], 'Chamado', 'Felipe', 'C...
      nascimento: 1996
      nome: 'Felipe'
      sobrenome: 'Cabrera'
    ▶ __builtins__: {'ArithmeticError': <class 'ArithmeticEr...
      __cached__: None
      __doc__: None
      __file__: 'g:\\Arquivos\\_ByLearn\\2 - Cursos\\VS Code...
      __loader__: None
      __name__: '__main__'
      __package__: ''
      __spec__: None
```

## Watch

Para ativar uma variável no Watch você precisa apenas clicar no botão com símbolo de mais (+) ao lado do nome Watch ou durante o debug selecionar alguma variável, abertar botão direito e selecionar 'Debug: Add to Watch'.

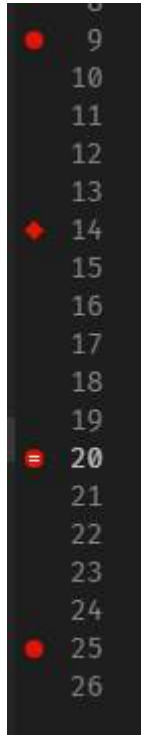
A variável cujo nome foi inserido no Watch irá ficar sendo observado durante todo o código e apresentará informações específicas.



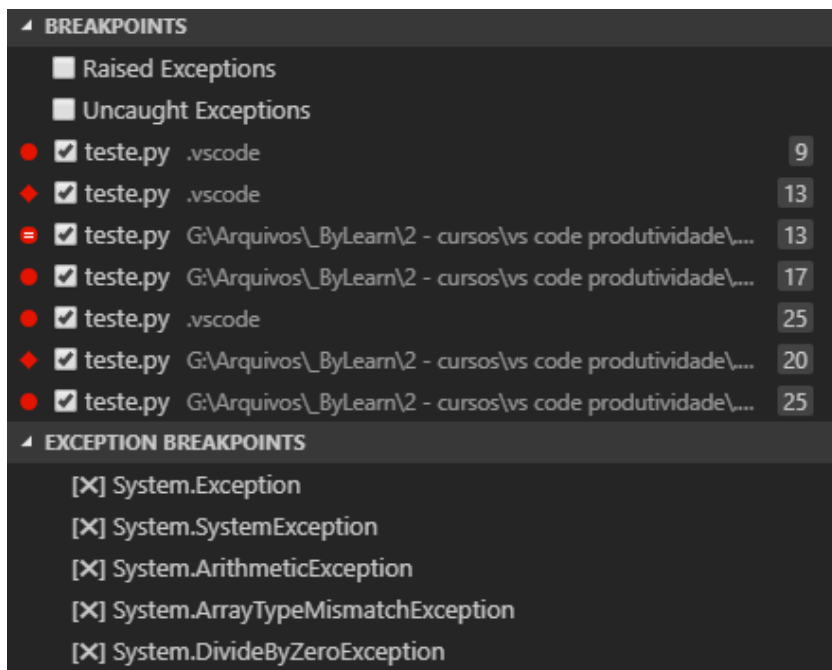
## Breakpoints

Os breakpoints são as condições de parada da depuração. Caso um código atinja um breakpoint, ele irá 'travar' até que o programador escolha continuar o código (F5).

Para adicionar um Breakpoint, basta apertar ao lado esquerdo da numeração de linhas do editor, como mostra a imagem abaixo:



Também temos como ver quais são os breakpoints que colocamos no nosso código e quais Exceptions (erros) também irão efetuar um breakpoint.



## Tipos de Breakpoint

O VS Code tem mais de um tipo de Breakpoint e para ativá-los, basta clicar com o botão Direito do mouse na parte à esquerda na numeração de linhas no editor

1. Default: O normal, é só o código chegar lá que ele irá executar o Breakpoint;
  - i. Representado por um círculo vermelho.
2. Condicional: Precisa de uma condição lógica para ativar o Breakpoint, algo como 'x == 2';
  - i. Representado por um círculo vermelho com um símbolo de igualdade (=) dentro dele.
3. Log Message: Bem diferente dos outros, ele não pausa o código, apenas mostra uma mensagem no painel Debug Console. Ótimo para substituir um 'print()'.
  - i. Representado por um losângulo vermelho



---

## **Produtividade +8000% - Evoluindo ainda mais nas tarefas**

---

Evolua e torne-se um programador supremo, faça em um dia o que você faria na semana. Escreva um 10% do código original e tenha o mesmo resultado com o Emmet e Snippets.

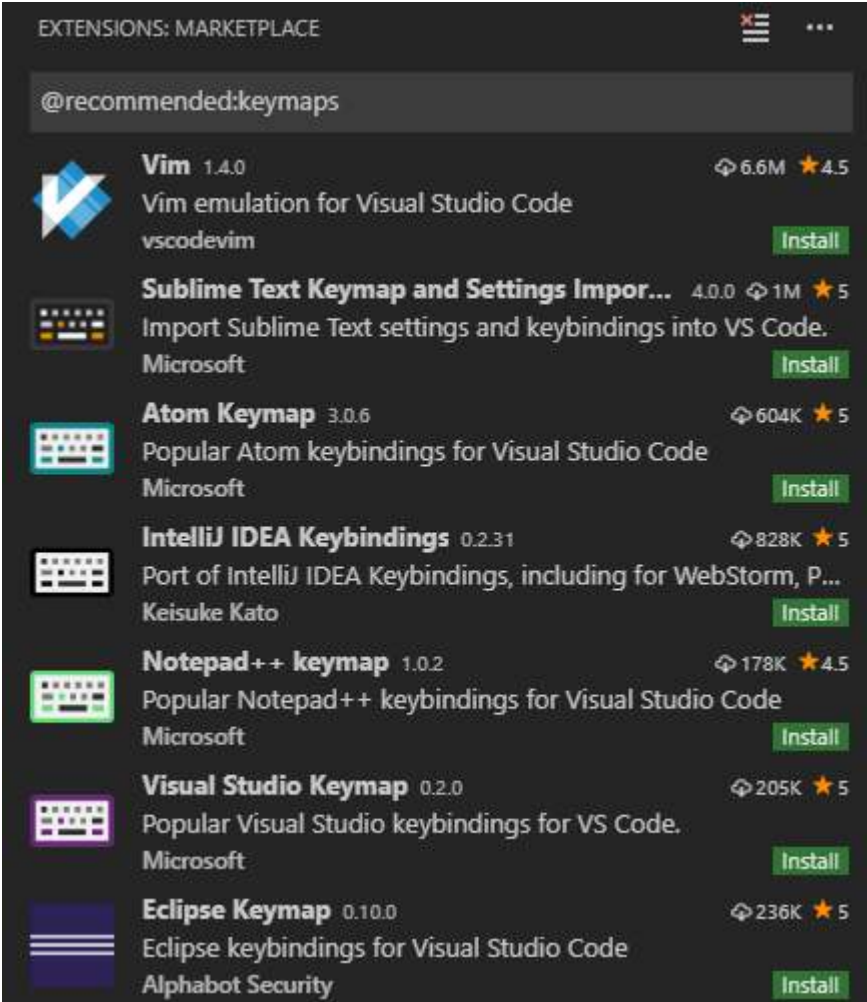
# KeyMaps

Antes de começarmos a falar dos 'atalhos' mais complexos do Emmet e dos Snippets, vamos ter **CERTEZA** que você está familiarizado com todos os atalhos do VS Code!

Você pode tanto dar uma nova lida nos atalhos mostrados anteriormente quanto também mudar os atalhos atuais para outros que você prefira e ache mais cômodo. Para isso, basta irmos em '*File > Preferences > Keyboard Shortcuts*' e mudamos os atalhos que quisermos.



Porém, se você veio de algum outro editor de texto ou IDE (como o Visual Studio), você pode se aproveitar do Keymap deles, fazendo os atalhos ficarem iguais aos que você está acostumado :D. Basta irmos em '*File > Preferences > Keymap Extensions*' e veremos uma série de extensões que já possuem os atalhos configurados de acordo com o Keymap de outros editores.



---

## Emmets

Agora que os atalhos estão certos, vamos aprender os "Atalhos 2.0 - Ultra Turbo", os Emmets! Os Emmets possuem a seguinte definição:

Emmet é um conjunto de plug-ins para editores de texto que permitem a codificação e edição em HTML, XML, XSL e outros formatos de código estruturado de alta velocidade via assistente de conteúdo.

Em outras palavras, eles permitem escrevermos um bloco de código de forma resumida, como por exemplo:

*Comando:* **ul>li\*5**

*Código:*

```
<ul>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
</ul>
```

*Comando:* **ul>li.item\$\*5**

*Código:*

```
<ul>
  <li class="item1"></li>
  <li class="item2"></li>
  <li class="item3"></li>
  <li class="item4"></li>
  <li class="item5"></li>
</ul>
```

*Comando:* **nav>ul>li**

*Código:*

```
<nav>
  <ul>
    <li></li>
  </ul>
</nav>
```

Acha que isso é tudo? O Emmet tem mais umas centenas de atalhos que fariam esse livro pesar mais de 1gb. Portanto, vamos deixar abaixo uma lista com a Cheat-Sheet oficial do Emmet para você consultar sempre que for preciso <3.

[Emmet Cheat Sheet](#)

---

## Snippets


Agora vamos falar dos "Atalhos 3.0 - Ultra Mega Turbo Blaster", os snippets. Basicamente eles são acronomos/sigla/atalhos para um certo bloco de código específico e o melhor de tudo: **eles são configuráveis e você pode fazer os seus :D**.

Quem programa C# conhece o famoso atalho CW que nos gera *Console.WriteLine()* e os amantes de java nunca deixam o *sysout* de lado, para gerar o *System.out.println()*, tais atalhos são os chamados Snippets!

O VS Code por padrão já possui alguns, mas há **vários** outros que podemos instalar e configurar. Primeiro temos que ir até o menu de Extensões e em seguida procurar por "Snippets" e pronto: Temos uma imensa lista de Snippets prontos para usarmos, das mais diversas linguagens.

EXTENSIONS MARKETPLACE

snippets



**Angular v7 Snippets** 7.0.1

Angular v7 snippets by John Papa

John Papa

4.8M ★ 5

[Install](#)



**Angular 8 Snippets - TypeScript, HTML, A...** 8.0.1

240 Angular Snippets (TypeScript, HTML, Angular Material, FL...

Mikael Morland

9.2M ★ 5

[Install](#)



**JavaScript (ES6) code snippets** 1.7.2

Code snippets for JavaScript in ES6 syntax

charalampos karypidis

3.8M ★ 4.5

[Install](#)



**HTML Snippets** 0.2.1

Full HTML tags including HTML5 Snippets

Mohamed Abusaid

2.6M ★ 4.5

[Install](#)



**ES7 React/Redux/GraphQL/React-Native...** 2.2.1

Simple extensions for React, Redux and GraphQL in JS/TS with...

dionajdier

2.3M ★ 5

[Install](#)



**Vue 2 Snippets** 0.2.11

A Vue.js 2 Extension

hollowtree

1.5M ★ 5

[Install](#)



**Reactjs code snippets** 2.3.0

Code snippets for Reactjs development in ES6 syntax

charalampos karypidis

895K ★ 6

[Install](#)



**Laravel Blade Snippets** 1.18.0

Laravel blade snippets and syntax highlight support

Winnie Lin

806K ★ 4.5

[Install](#)



**React-Native/React/Redux snippets for e...** 2.0.3

Code snippets for React-Native/React/Redux es6/es7 and FL...

ElQuimper

777K ★ 5

[Install](#)



**Bootstrap 4, Font awesome 4, Font Awes...** 0.1.0

Bootstrap 4 snippets based on documentation + Font awes...

Ashok Koyl

1M ★ 4.5

[Install](#)



**Bootstrap 3 Snippets** 0.1.0

Bootstrap 3 Snippets for VS Code

William Whitehead

528K ★ 3.5

[Install](#)



**jQuery Code Snippets** 0.0.1

Over 130 jQuery Code Snippets

Don Jayamanne

454K ★ 4

[Install](#)



**Vue VSCode Snippets** 1.0.0

Snippets that will supercharge your Vue workflow

samir.draouer

277K ★ 5

[Install](#)



**Angular 7 and TypeScript/HTML VS Co...** 1.0.21

VS Code snippets for Angular and TypeScript/HTML

Dan Wahlin

559K ★ 5

[Install](#)


**Simple React Snippets** 1.2.2

Dead simple React snippets you will actually use

313K ★ 5

---

## **Produtividade 250000% - Indo rumo ao infinito**

---

Já estamos perto do infinito! Vamos aprender agora como controlar nossas versões através do GIT, receber recomendações do próprio VS Code e a maravilha dos Lints.

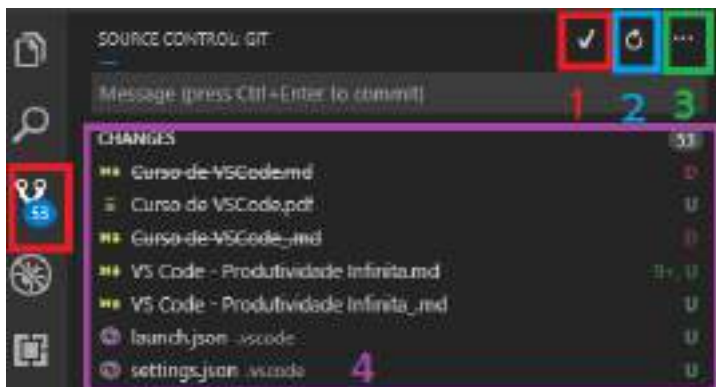
---

## Controle de Versões com GIT

Todo bom programador já ouviu falar de Git ou algum outro controlador de versões. Segundo a definição, Git é:

Git é um sistema de controle de versões distribuído, usado principalmente no desenvolvimento de software, mas pode ser usado para registrar o histórico de edições de qualquer tipo de arquivo.

Usar o Git no VS Code é muito simples, basta selecionar a opção Git na Activity Bar como já ensinamos posteriormente. Temos então a seguinte imagem:

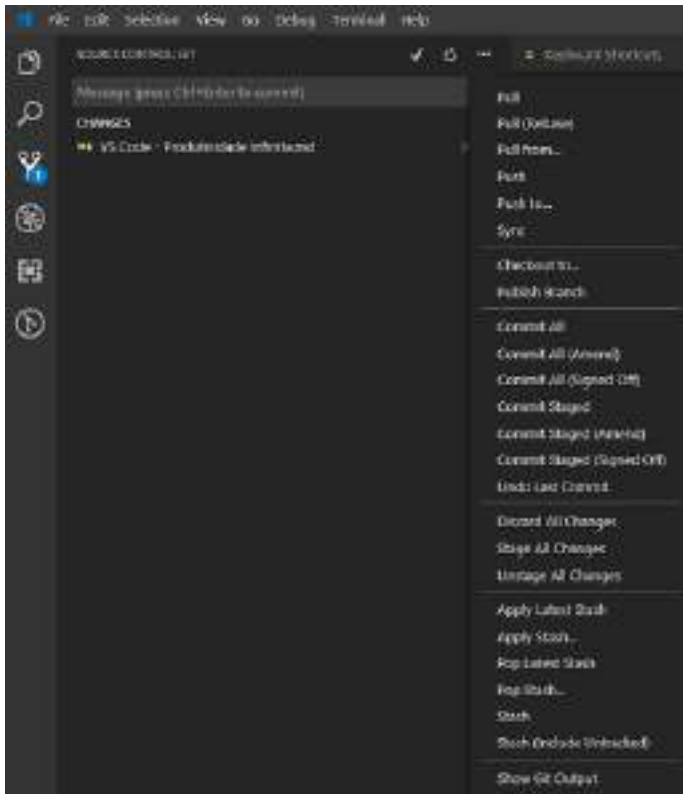


1. Commit: Preparamos nossas mudanças para serem enviadas ao repositório (salvando completamente o estado atual do projeto);
  - i. PS: Até executar o envio (PUSH) as alterações são apenas locais.
2. Recarregar: Força atualizar tudo o foi mudado no projeto;



3. Mais Ações: Ações extras do git, como por exemplo, enviar os Commits ao repositório e receber as ultimas mudanças dos seus colegas de equipe;
4. Mudanças: Indica tudo que mudou no nosso projeto desde o ultimo Commit.

Por sua vez, a aba de '*Mais Ações*' possui as demais funções do git que também são muito úteis, como por exemplo:



Pull: Recebe as ultimas alterações do repositório

1. Push: Envia suas ultimas alterações para o repositório
  2. Checkout To: Permite mudar a branch para qual vai enviar os commits
    - i. Branch: Para qual das '*divisões*' do repositório serão enviadas as alterações, como por exemplo: versão Release, Beta, Alpha, Dev
  3. Stage All: Armazenas todas suas alterações para serem enviadas no próximo Commit
  4. Show Git Output: Mostra o Output (saída) do git, mostrando via CLI (linha de comando) tudo o que está acontecendo no git, como os commits enviados, os pulls recebidos e etc...
- 

## Lints

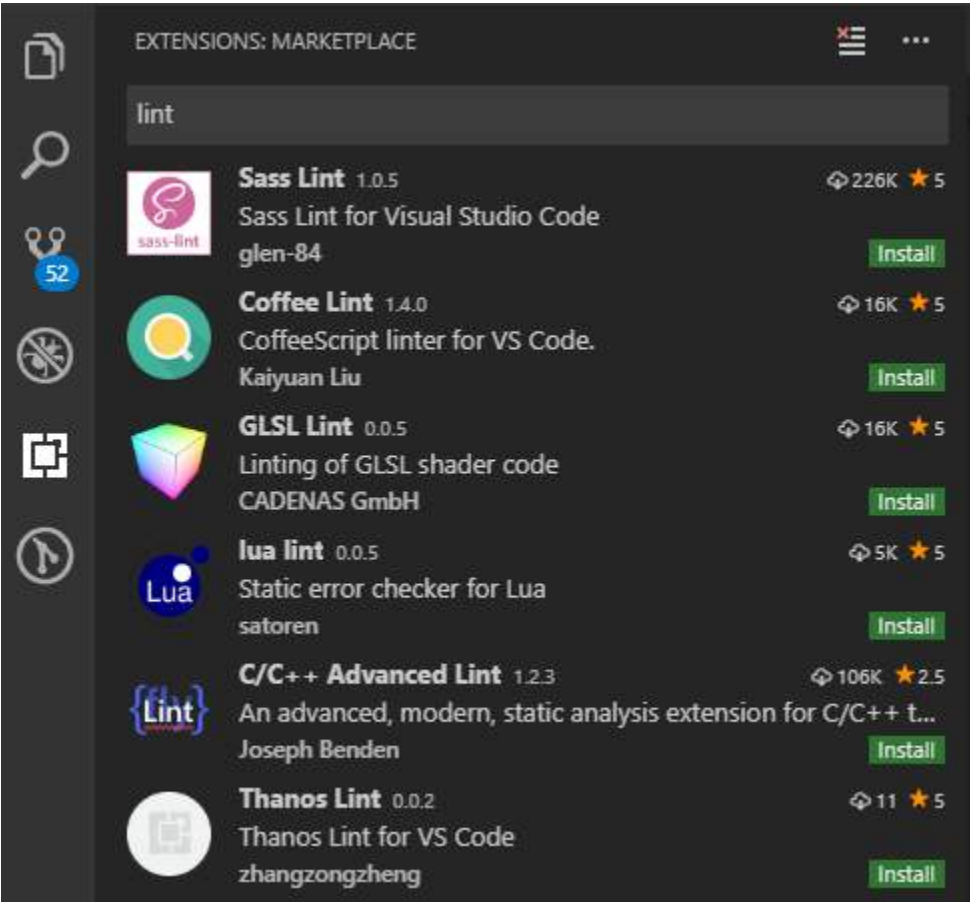
Lint segundo sua definição pode ser entendido como:

lint era o nome originalmente dado a um programa em particular que sinalizava algumas construções suspeitas e não portáteis (provavelmente bugs) no código-fonte da linguagem C. O termo agora é aplicado genericamente a ferramentas que sinalizam uso suspeito em software escrito em qualquer linguagem de computador.

O que em outras palavras pode ser entendido como:

O Lint facilita a sua codificação uma vez que ele sinaliza partes erradas ou que ferem algum padrão de guideline da linguagem. Com isso, você consegue saber qual a parte que precisa modificar para corrigir. Os Lints também podem colorir o seu código a fim de exaltar melhor o que é cada parte e sua função (onde é uma variavel, onde é uma função e etc...)

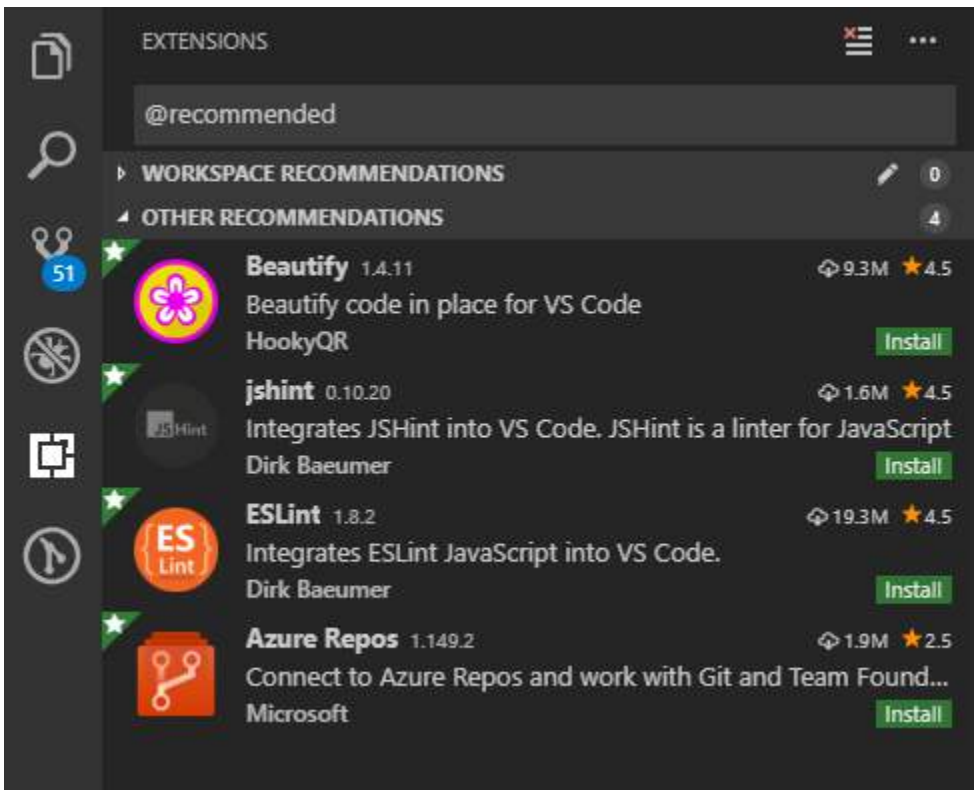
Para conseguir baixar vários Lints prontos para as mais diversas linguagens, basta ir em Extensões e procurar por *Lint*.



## Recomendações

O VS Code também possui uma série de recomendações para os usuários, que aumentam conforme você vai utilizando o programa, ou seja, ele vai aprendendo seus gostos e o que você mais usa e vai te dando recomendações.

Para acessar é muito fácil basta ir até a aba de extensões e digitar "@recommended". Em seguida você receberá recomendações tanto para o seu Workspace atual quanto para o seu uso rotineiro.



---

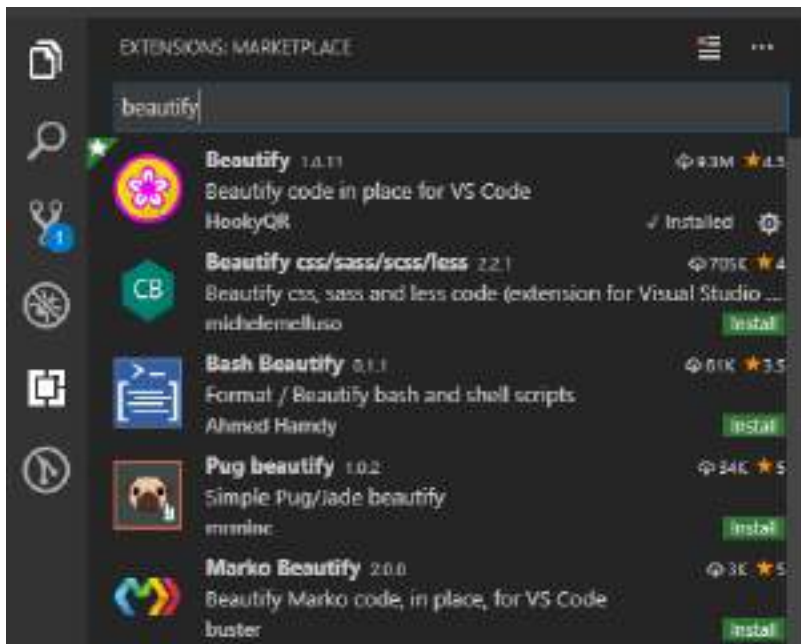
## Formatadores (Beautification)

Os formatadores mais completos executam uma ação conhecida como *Beautify*, que baseia-se em pegar um código humanamente difícil de se ler e transforma em uma versão bem mais bonita e fácil de ler e entender.

Um exemplo disso são os arquivos *Minificados*, que executam técnicas para reduzir o peso dos arquivos, deixando tudo em apenas uma linha. Tal prática é bem popular com códigos JS. As extensões formatadoras são boas para exatamente isso, elas pegam aquele código totalmente difícil de ler e os tornam muito mais fácil de ler.

Além dos códigos em JS, também são bem úteis para JSON, CSS e HTML

Para encontrá-las basta procurar por *Beautify* nas extensões.



---

# **Produtividade 1000000% - Tópicos Avançados, apenas um passo para o infinito**

---

Estamos chegando! A produtividade infinita está logo a frente!  
Vamos apenas aprender mais alguns conceitos e dicas :)

---

## Whitespaces

Algo bem legal no VS, ainda mais para linguagens como Python que os blocos são delimitados pela indentação, é saber o quanto de espaços iniciais você já deu em uma determinada linha, além de saber se deu dois espaços seguidos sem querer.

Para ativar isso podemos ir nas opções (*File > Preferences > Settings*) e procurar por *Render Whitespace*. Em seguida temos duas opções:

1. All: Todos os espaços;
2. Boundary: Só os espaços nas bordas (início e fim) das linhas e espaços duplos.

## Auto Save

Também nas opções (*File > Preferences > Settings*) podemos ativar o Auto Save para salvar automaticamente nosso documento. Para isso basta procurar lá por *AutoSave*, escolher a opção desejada *afterDelay*, *onFocusChange* ou *onWindowChange* e caso escolha a primeira, definir o tempo de Delay logo abaixo.

Recomendação: Ative com a opção *afterDelay* e escolha um tempo de delay de 100.

## Copiar e colar múltiplas linhas

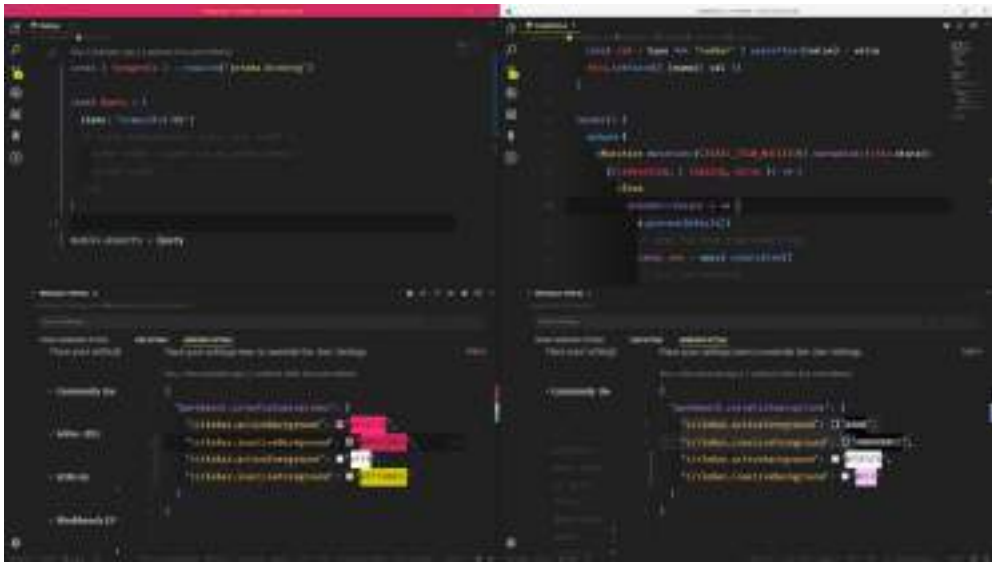
Ao clonar o cursor para várias linhas (Segurando o Alt, como já ensinamos), você também pode copiar o conteúdo de cada linha de uma vez só e colar quando for necessário.

## Customizar barra de título

Na paleta de comandos procure por 'Open Settings (JSON)' No JSON que será aberto adicione:

```
"window.titleBarStyle": "custom",
  "workbench.colorCustomizations": {
    "titleBar.activeBackground": "#FFFFFF",
    "titleBar.activeForeground": "#FFFFFF",
  }
```

Tal técnica é bem útil para diferenciar várias janelas do VS Code e saber do que cada. uma se trata.





## Selecionar apenas uma área específica

Basta apertar o botão do meio (apertar a bolinha) e arrastar mouse na área desejada.

What is Lorem Ipsum?  
Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

What is Lorem Ipsum?  
Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

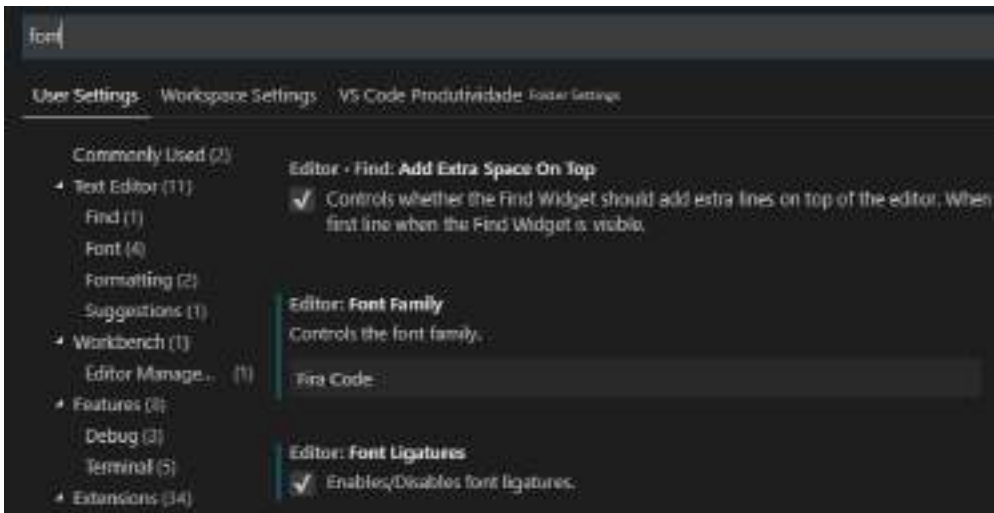
What is Lorem Ipsum?  
Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised

# Fira Code com Font Ligatures

Falando sobre fontes, uma das mais utilizadas no VS Code é a Fira Code, uma fonte própria para desenvolvedores, contando com alguns recursos como ligaduras tipográficas (que permitem a junção de dois ou mais caracteres em um só) e símbolos feitos pensando no dia a dia de um programador, além do design de fontes ser feito exatamente para não confundir o desenvolvedor como nos casos de '0 e O', 'il IL'.

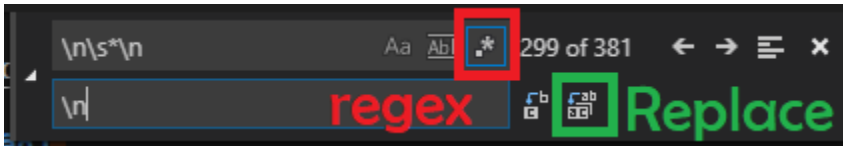
Para utilizar a Fira Code basta baixar do repositório oficial [aqui](#) e instalar no seu sistema operacional.

Em seguida, vamos em Opções no VS Code (*File > Preferences > Settings*) e procuramos por *Font*, configurando as propriedades *Font Family* e *Font Ligatures* como na imagem abaixo:



## Linhas Vazias

Para excluir todas as linhas em branco é muito simples, basta apertar *Ctrl+H* e você verá a opção de Search & Replace (Buscar e Substituir), ativar a buscar por Regex (*Alt + R* no campo de busca) e então basta buscar por `"\n\s*\n"` e pedir para substituir por apenas `"\n"`, em seguida apertando no botão Substituir Todos (*Ctrl + Alt + Enter*).



## Breadcrumbs (Auxiliares de navegação)

Através dos Breadcrumbs fica bem fácil saber onde você está na workspace, em qual classe, método ou até mesmo variável está trabalhando, tudo isso pelo caminho de navegação que fica na parte superior do VS Code.

## Delimitar Seleção

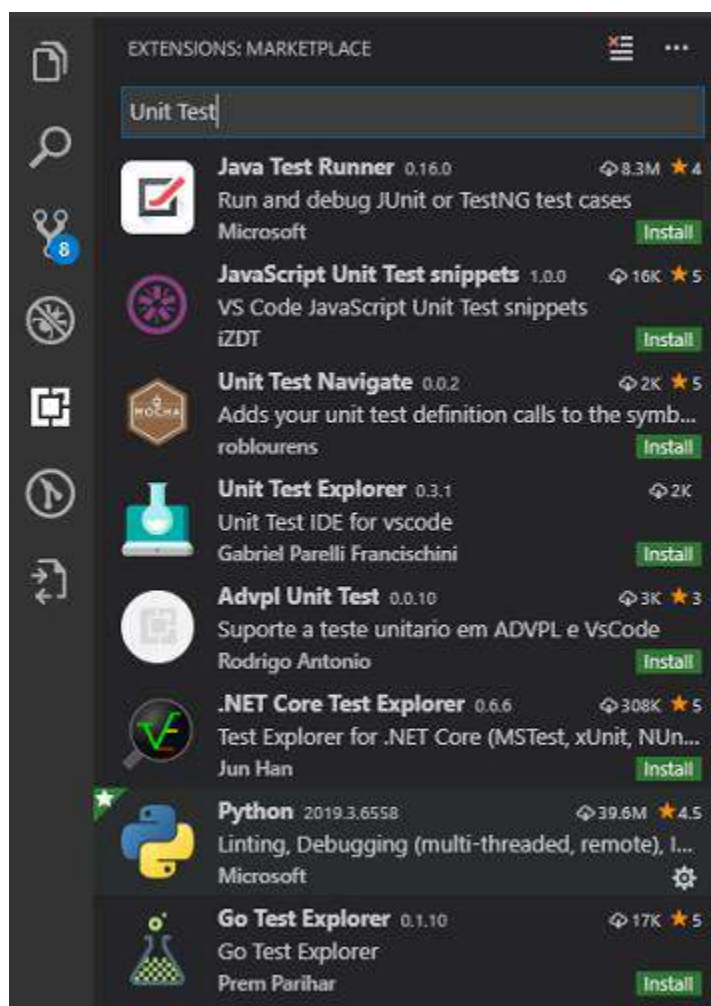
Eu aposto que várias vezes você já tentou delimitar alguma seleção com aspas, parenteses, barras e afins, mas sempre precisou inserir uma antes e uma depois, para não substituir o texto, certo?

Com o VS Code basta você selecionar as palavras e apertar o delimitador de inicio, como `[`, `(`, `'`, `"` e automaticamente o VS Code já irá colocar o respectivo delimitador no inicio e fim da seleção.

## Testes Unitários

Os testes unitários ajudam muito na hora de testar seu código e ver se não restou nenhum tipo de erro ou problema que passou nos testes rotineiros (compilar, tentar usar e ver se pegou). Por sorte, o VS Code tem muitas extensões para testes unitários, em várias linguagens.

Para instalar essas extensões basta ir na sessão de Extensões na Activity Bar e procurar por *Unit Test*.



---

## **Produtividade $\infty$ - Alcançando o Infinito**

---

Pronto! Basta aprendermos a usar as extensões de forma eficaz e iremos alcançar a sonhada produtividade infinita!

---

## Extensões - Gerais

### Code Runner

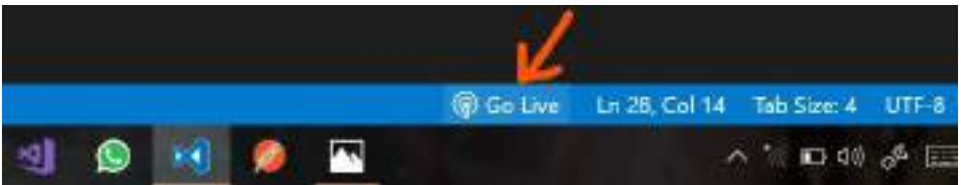
Permite rodar códigos direto no VS Code nas seguintes linguagens:

C, C++, Java, JavaScript, PHP, Python, Perl, Perl 6, Ruby, Go, Lua, Groovy, PowerShell, BAT/CMD, BASH/SH, F# Script, F# (.NET Core), C# Script, C# (.NET Core), VBScript, TypeScript, CoffeeScript, Scala, Swift, Julia, Crystal, OCaml Script, R, AppleScript, Elixir, Visual Basic .NET, Clojure, Haxe, Objective-C, Rust, Racket, AutoHotkey, Autolt, Kotlin, Dart, Free Pascal, Haskell, Nim, D, Lisp, Kit, e comandos customizados

*Comando:* Aperte **Ctrl + Alt + N** para iniciar a aplicação

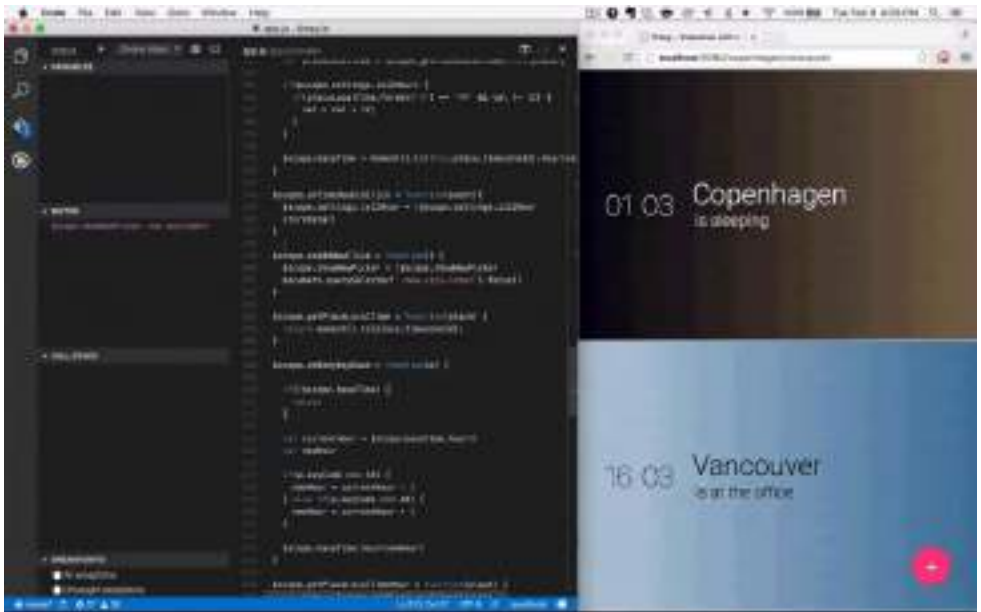
### Live Server

Abre um servidor local para desenvolvimento web, fazendo seu código abrir no navegador pelo próprio VS Code.



## Debugger for chrome

Permite executar a depuração de código direto no código, ótimo para debugar aplicações Web e códigos JS



## Visual Studio IntelliCode

Ativa o novo Visual Studio IntelliCode (Criado para o VS 2019) para diversas linguagens, como C#, C++, XAML, Pytho, JavaScript e TypeScript.

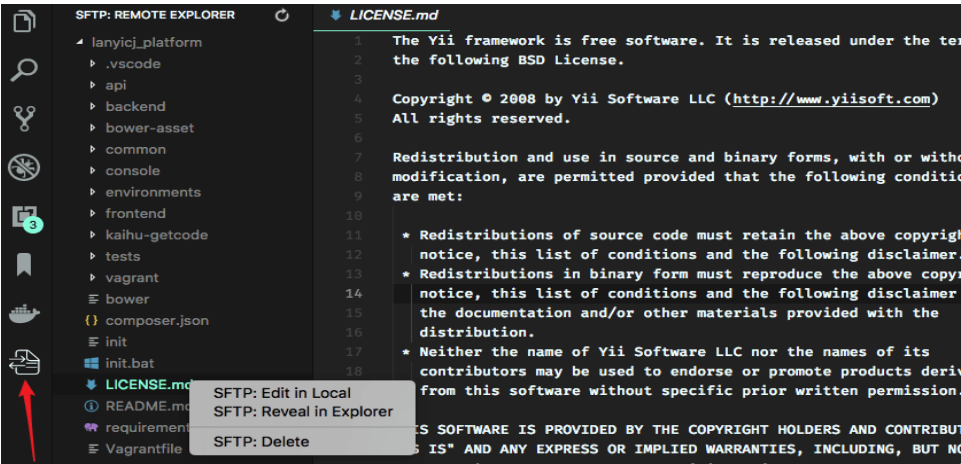
## Beautify

Como já foi dito anteriormente, funções com tag/nome *Beautify* servem para deixar seu código mais bonito, estruturado e organizado.

No VS Code a principal extensão para isso chama exatamente *Beautify* e funciona com JSON, HTML, SASS e CSS. Basta apertar *F1* e digitar "Beautify file".

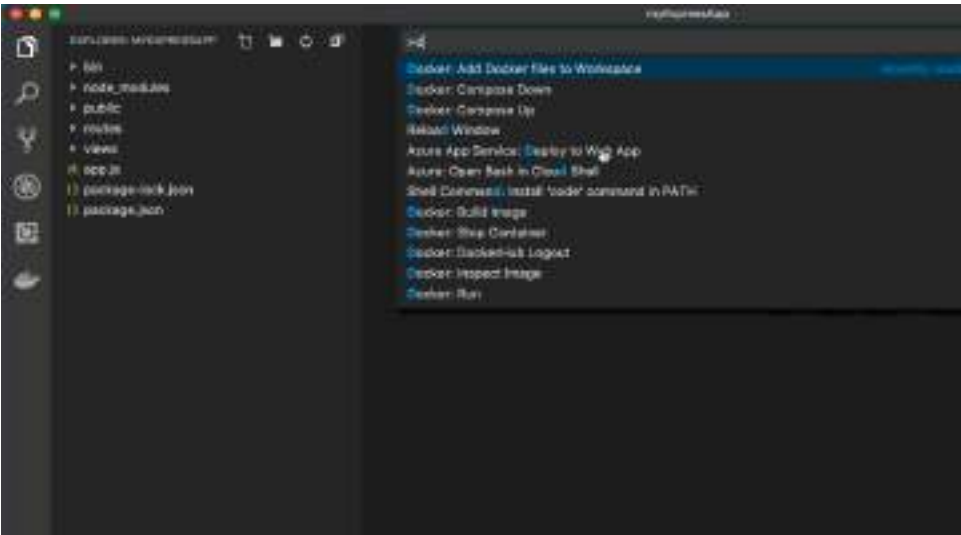
## SFTP

Extensão super poderosa para SFTP e FTP, permitindo editar arquivos direto o servidor, editar localmente e subir para o servidor, fazer download, backup, sincronizar, checar as diferenças e muito mais.



## Docker

Extensão que permite o suporte ao Docker no VS Code.

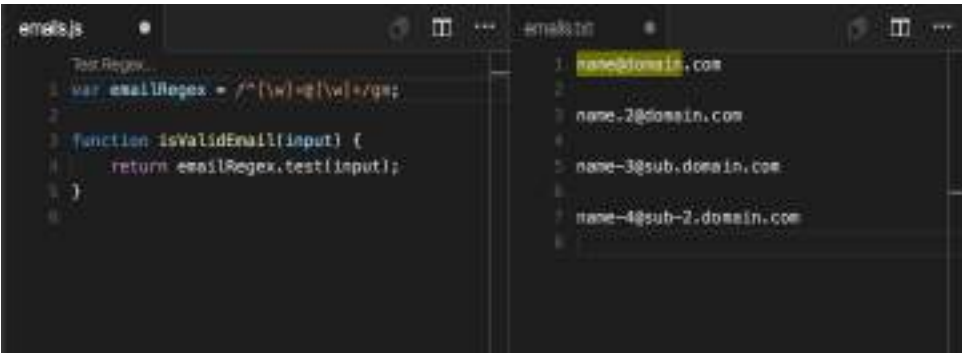




## Regex Previewer

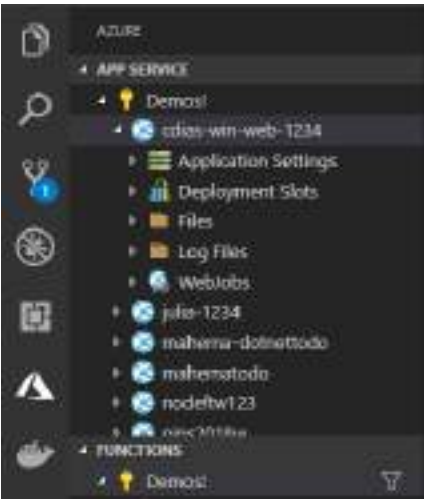
Permite verificar se o nosso Regex está pegando ao tentar encontrar padrões em um arquivo com regex com outro na tela ao lado contendo o texto a ser testado.

Comando: Aperte **Ctrl+Alt+M** para iniciar o Preview!



## Azure Tools

Extensão que dá suporte as ferramentas Azure, podendo trabalhar com tudo o que é oferecido pela Azure.



## Spell Checker

Checa a gramática do código, dizendo ao programador quando ele errou na escrita (possui suporte para português).

## GitLens

Extensão para o Git que ajuda muito, mostrando muitas outras informações, como quem editou cada arquivo/método/variável/linha por último e quando foi isso. Também permite executar um fluxo completo do git (pull, stage, commit, push, sync, checkout e etc...).

## Indent Rainbow

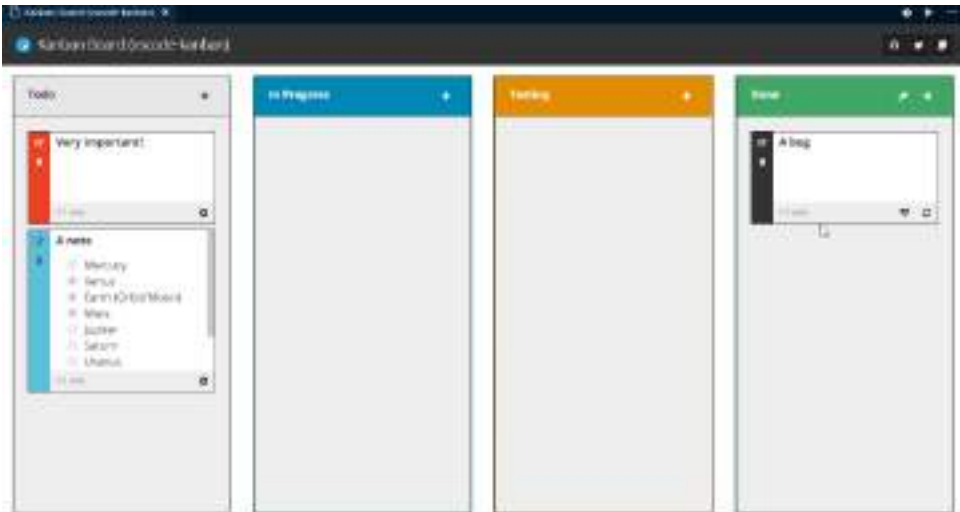
Mostra as identações (espaços) com cores diferentes em um gradiente, para mostrar o quão indentado está.

```
nimbench_ex2.nim src/misc
1 import nimbench
2
3 bench(str, m):
4   for n in 0..<m:
5     var dest = "apple"
6     while(dest.len<100):
7       var
8         offs=dest.len-5
9         i=offs
10        while i<offs+10:
11          dest.add(dest[i])
12          i.inc
13        doNotOptimizeAway(dest)
14
15 runBenchmarks()
16 |
```

## Kanban

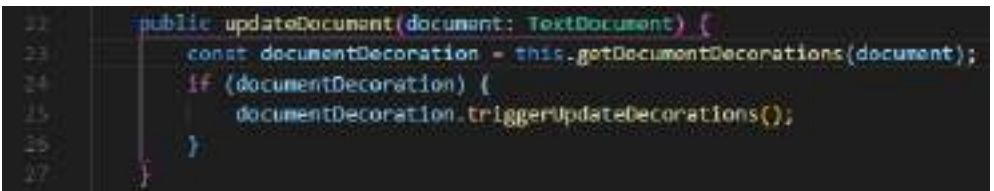
Uma extensão ótima que irá aprimorar e muito sua produtividade no VS Code através dos quadros de Kanban.

*Comando:* Aperta **F1** e execute **Kanban: Open Board** para abrir o quadro Kanban.



## Bracket Pair Colorizer

Pinta parênteses correspondentes da mesma cor e os difere de outros, para você sempre saber onde começa e termina cada um dos blocos delimitados por parênteses.



## Todo Tree

Permite marcar e destacar comentários com tags como TODO, FIXME, e afins.



## Auto Close Tag

Extensão tão popular que a partir do VS Code 1.6 ela foi implementada de forma nativa para HTML, porém, ainda é muito útil para XML, PHP, Vue, JavaScript, TypeScript, JSX, TSX e muito mais.

## Minify

Minificador automático para códigos JS, CSS e HTML.

## Better Comments

Adiciona uma série de tags para comentários, como alertas, TODO, destaques, parametros e etc...

```
2  export class MyClass {
3
4      /**
5       * MyMethod
6       * * Important information is highlighted
7       * ! Deprecated method, do not use
8       * ? Should this method be exposed in the public API?
9       * TODO: refactor this method so that it conforms to the API
10      * @param myParam The parameter for this method
11      */
12     public MyMethod(myParam: any): void {
13         let myVar: number = 123;
14
15         /** This is highlighted
16         if (myVar > 0) {
17             throw new TypeError(); //! This is an alert
18         }
19
20         /**? This is a query
21         let x = 1;
22
23         /// this.lineOfCode() == commentedOut;
24
25         //TODO: Create some test cases
26     }
27 }
```

## Settings Sync

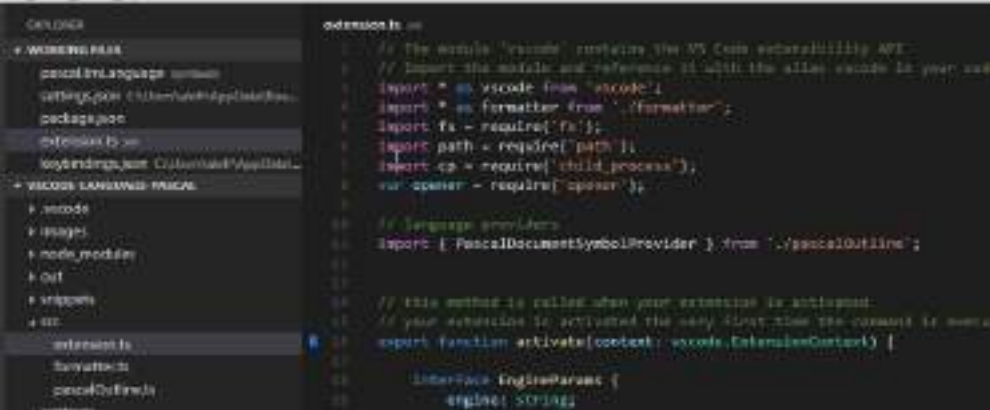
Permite sincronizar suas configurações com outros computadores através de um repositório GitHub.

## Gi

Permite gerara arquivos .gitignore de forma simples e fácil.

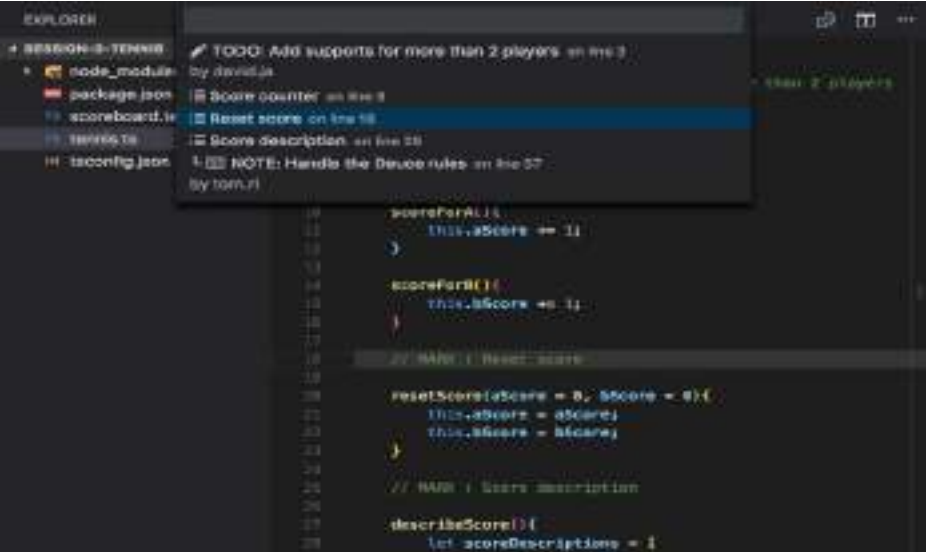
## Bookmark

Permite salvar linhas como favoritas e pular para elas facilmente.



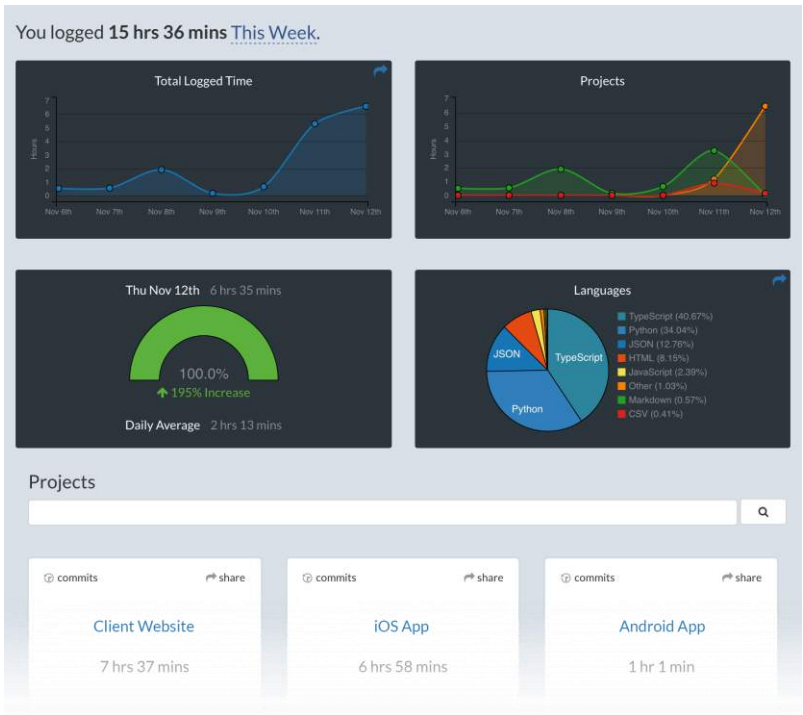
## Mark Jump

Alternativa ao BookMark, onde o programador pode, através de um comentário, marcar uma linha para ser rapidamente visitada novamente no futuro.



## WakaTime

Extensão muito útil para medir o tempo em que você passou codificando algum projeto, tanto para medir melhor o empenho gasto para cobrar pelo freelance quanto para avaliar sua melhora em produtividade (que espero já estar **infinita**).

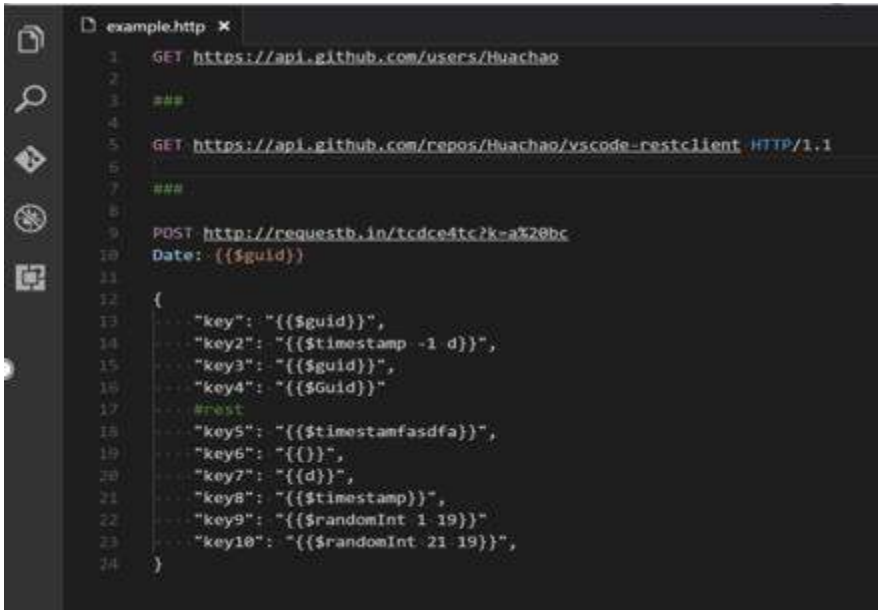


## Path Intellisense

Permite gerar um autocompletar para nome de arquivos e diretórios de acordo com o arquivo atual em relação a hierarquia de diretórios do seu projeto.

## REST Client

Permite executar requisições HTTP (GET/POST) diretamente do VS Code.



```
example.http x
1 GET https://api.github.com/users/Huachao
2
3 ###
4
5 GET https://api.github.com/repos/Huachao/vscode-restclient HTTP/1.1
6
7 ###
8
9 POST http://requestb.in/tcdce$tc?k-a%20bc
10 Date: {{$guid}}
11
12 {
13   "key": "{{$guid}}",
14   "key2": "{{$timestamp -1 d}}",
15   "key3": "{{$guid}}",
16   "key4": "{{$Guid}}"
17   #rest
18   "key5": "{{$timestampfasdfa}}",
19   "key6": "{}",
20   "key7": "{(d)}",
21   "key8": "{{$timestamp}}",
22   "key9": "{{$randomInt 1 19}}",
23   "key10": "{{$randomInt 21 19}}",
24 }
```

## Partial Diff

Permite checar a similaridade de duas seleções, mostrando onde são iguais e onde são diferentes.



# TinyPNG

Extensão super útil na hora de comprimir imagens e otimizar o peso do seu projeto. Tudo isso de forma simples e prática.



## Polacode

Permite tirar printscreens elegantes em Polaroid do seu código.

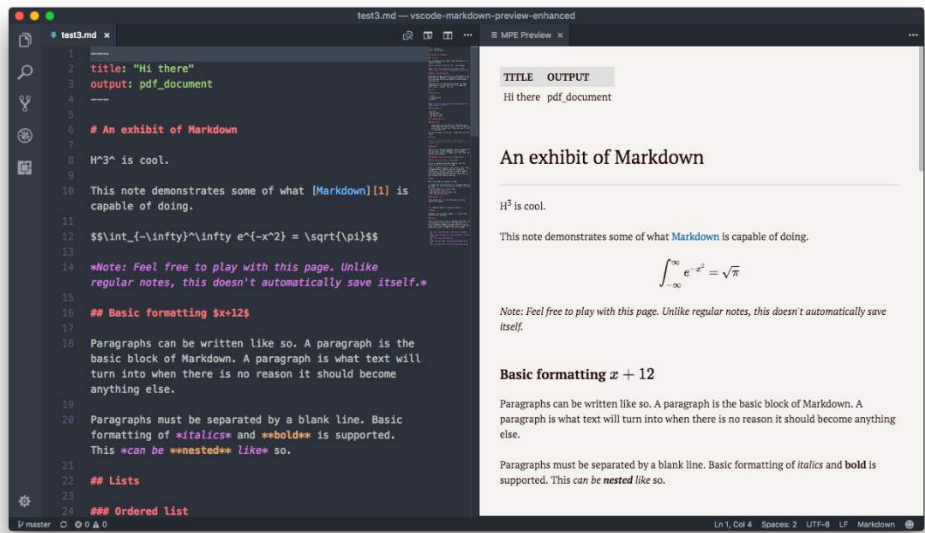


# Extensões - Markdown

## Markdown Preview Enhanced

Gera uma tela de Preview bem melhor do que a existente por padrão no VS Code, cheia de funções como até mesmo gerar um e-book através de um Markdown.

PS: Essa extensão foi usada para a criação deste ebook.



# Markdown All in One

Uma extensão completa para se trabalhar com Markdown, cheia de atalhos e funcionalidades muito úteis, como a criação de sumários.

abcabcabc

normal **bold** wordUnderCursor **jumpOut**

- [ ] todo

# Section 1

## Subsection 1.1 <!-- omit in toc -->

# Section 2

## Subsection 2.1

Table of Contents (up to date)

- [Section 1](#section-1)
- [Section 2](#section-2)
  - [Subsection 2.1](#subsection-21)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed ut perspiciatis unde omnis iste natus error sit voluptas. Architecto quae soluta maxime p

---

## Extensões - JSON

### [Paste JSON as Code](#)

Permite transcrever um código em JSON para várias outras linguagens, tais como: TypeScript, Python, Go, Ruby, C#, Java, Swift, Rust, Kotlin, C++, Flow, Objective-C, JavaScript e Elm.

### [Prettify JSON](#)

Permite visualizar códigos em JSON de uma forma mais organizada, estruturada e bonita.

---

## Extensões - Python

### [Python](#)

Extensão *tudo em um* para Python, contendo Linting, Depuração, Intellisense, Formatação de Código, Refatoração, Snippets, Jupyter Notebook e muito mais...

### [Python Preview](#)

Permite visualizar uma prévia do código em Python codificado por você de forma bem detalhada, muito eficaz para depuração!

### [Anaconda Extension Pack](#)

Extensão que permite o funcionamento da Anaconda no VS Code.

---

## Extensões - Java

### [Language Support for Java\(TM\) by Red Hat](#)

Habilita o suporte ao Java no VS Code com muitas funções como Linting, Intellisense, Formatação, Refatoração, Maven/Gradle e muito mais.

### [Debugger for Java](#)

Um depurador de código para Java bem completo e eficaz.

---

## Extensões - C# e DotNetCore

### [C#](#)

Extensão oficial com tudo o que C# precisa para o VS Code.

### [C# Extensions](#)

É um conjunto de pacotes para a extensão C#, com várias funções.

### [C# XML Documentation Comments](#)

Permite criar documentações XML a partir da digitação de ///, como acontece no Visual Studio.

### [Awesome DotNetCore Pack](#)

Um pacote para C# e DotNetCore contendo várias outras extensões menores, se tornando uma extensão *tudo em um*.

---

## Extensões - C/C++

### [C/C++](#)

Extensão oficial da Microsoft para habilitar suporte ao C++ no VS Code.

### [C/C++ Extension Pack](#)

Pacote repleto de extensões para o C/C++, como melhorias ao Clang, Autocompletar e Snippets.

---

## Extensões - Banco de Dados

### [SQL Server \(Mssql\)](#)

Desenvolva com Microsoft SQL Server, Azure SQL Database e SQL Data Warehouse direto do seu VS Code.

### [SQLTools](#)

Extensão para SQL cheio de ferramentas, árvore de visualização da tabela, possibilidade de rodar as *queries* de forma simples, intellisense, histórico, exportar como JSON e muito mais.

### [SQLite](#)

Extensão que adiciona suporte ao SQLite ao VS Code.

### [Mongo Runner](#)

Por fim, uma extensão aos que preferem NoSQL, o Mongo Runner traz ao VS Code tudo que é preciso para se trabalhar com o banco de dados MongoDB.



---

## Extensões - HTML

### [HTML Snippets](#)

Uma das extensões mais úteis para se trabalhar com HTML, ela gera vários snippets para os mais diversos casos.

### [HTML Boilerplate](#)

Essa extensão provê o suporte ao template HTML Boilerplate através de snippets.

### [Sorting HTML and Jade attributes](#)

Ordene seus atributos da forma correta em HTML.

### [Auto Rename Tag](#)

Renomeie todas suas tags de forma simples, rápida e fácil.

### [Dummy Text Generator](#)

Gere textos aleatórios (tais como o Lorem Ipsum) de forma fácil e prática para testar seus código.

---

## Extensões - CSS/SASS

### [CSS Peek](#)

Adiciona a função Peek (Ctrl + Shift + F12) e Go To Definition (F12) para classes CSS dentro de HTML. Ou seja, você consegue saber exatamente o que as classes do CSS estão fazendo com seu elemento HTML.

### [IntelliSense for CSS class names in HTML](#)

Implementa a função de auto completar classes CSS em um arquivo HTML.

### [CSS Preview](#)

Adiciona uma prévia ao seu CSS.

*Comando:* Para ligar a prévia aperte **Ctrl + Shift + P**

### [CSS Compressor](#)

Compacta e Minifica seu código CSS.

*Comando:* Para formatar aperte **Shift + Alt + F**

### [Beautify css/sass/scss/less](#)

Deixa seu código CSS, SASS, SCSS e LESS mais bonito e visualmente fácil de entender.

*Comando:* Para formatar aperte **F1** e digite "beautify.format"

---

## Extensões - TypeScript

### [TypeScript Hero](#)

Ferramentas adicionais para TypeScript, tais como organizar e ordenar imports.

### [Move TS](#)

Permite mover e renomear um arquivo TS e automaticamente arrumar todas as ocorrências e referências.

### [TypeScript Import](#)

Automatiza seus imports para as classes usadas em TypeScript.

---

## Extensões - JavaScript

### [Turbo Console.Log\(\)](#)

Permite uma alternativa fácil e rápida ao comum Console.Log().

*Comandos:* Alt + Shift + u e Alt + Shift + C

### [JavaScript \(ES6\) code snippets](#)

Extensão contendo uma imensa variedade de Snippets de JavaScript.

### [JavaScript Booster](#)

Vários comandos de refatoração e organização de código para aprimorar ainda mais nossa produtividade em JavaScript.

### [ESLint](#)

Trás o poderoso ESLint para o VS Code.

---

## Extensões - Angular

### [Angular 8 Snippets](#)

Extensão com mais de 200 snippets para o Angular, sempre atualizando para a ultima versão.

### [Angular Extension Pack](#)

Um super pacote com as extensões para Angular mais famosas e produtivas.

---

## Extensões - React

### [React Native Tools](#)

Uma série de ferramentas para poder desenvolver em React Native, como por exemplo, a possibilidade de depurar seu código.

### [ES7 React/Redux/GraphQL/React-Native snippets](#)

Os principais Snippets para React, Redux, GraphQL e React-Native.

### [React Extension Pack](#)

Pacote com várias extensões muito úteis para React.

---

## Extensões - Node.JS

### NPM

Suporte ao NPM (Node Package Manager ) pelo VS Code.

### NPM IntelliSense

Auto completar modulos do NPM.

### Node Debug

Ferramentas de depuração de código para Node.JS.

### Node Exec

Execute os seus arquivos Node.JS.

### Node Snippets

Principais Snippets para Node.JS e suas libs.

### Node Essentials

Conjunto de ferramentas e extensões para se tornar produtivo ao máximo no Node.JS

---

## Extensões - Vue

### [Vetur](#)

A principal extensão para VUE, com mais de 12 milhões de Downloads.

Possui ferramentas como Snippets, comandos Emmet, Linting, Formatações, Auto Completar, Depuração de Código e muito mais...

### [Vue.js Extension Pack](#)

Pacote de extensões para Vue.js que reúne várias ferramentas bem úteis para programação em Vue.



---

## Extensões - PHP

### [PHP IntelliSense](#)

Habilita o uso do IntelliSense para o PHP.

### [PHP Debug](#)

Feramentas de Depuração de código para PHP.

---

## Extensões - BASH

### [Bash Debug](#)

Permite a depuração de códigos BASH.

### [Bash Beautify](#)

Uma extensão *Beautify* feita para BASH, tornando seus códigos muito mais bonitos e organizados.

### [Bash IDE](#)

Uma série de ferramentas para BASH, como ir para a declaração, procurar referências, ativar o Code Outline, autocompletar e ferramentas de marcação e destaque.

---

## Extensões - PowerShell

### PowerShell

Uma série das principais ferramentas para desenvolvimento em PowerShell, como destaques, snippets, IntelliSense e afins.

### PowerShell Pro Tools

Outra extensão que é um pacote para PowerShell. O principal diferencial é a possibilidade de converter para C# e criação de Windows Forms.

---

## Extensões - Ruby

### Ruby

Extensão completa para desenvolvimento Ruby no VS Code, contendo:

1. Depuração de código
2. Lintings
3. Formatadores
4. Auto completar
5. IntelliSense

### Ruby on Rails

Pacote com mais de 10000 snippets e templates para Ruby on Rails.

---

## Extensões - Go

### Go

O principal (e oficial) pacote para programar em Go no VS Code. Possui funções como IntelliSense, Code Navigation, edição de código, diagnóstico de código, depuração, teste e muito mais.

---

## Extensões - Haskell

### [Haskelly](#)

Ferramenta completa para suporte ao desenvolvimento Haskell, com destaques, snippets, hovers, auto completar e mais.

---

## Produtividade $\infty$ - Chegamos ao infinito

---

Conseguimos! Chegamos ao infinito! Agora somos como deuses da produtividade! Só quero deixar aqui meu muito obrigado a todos vocês que acompanharam o livro/curso até aqui e de brinde deixar para vocês alguns extras bem legais, como configurações já prontas, materiais de estudo e dicas de personalização.



---

## Produtividade $\infty^2$ - Os Extras

---

Coloque sua produtividade (já infinita) ao quadrado com essas dicas excepcionais de configurações extras :).

Devido a nossa limitação em enviar links por ebook e livros, deixaremos o título da postagem e onde encontrá-las.



---

## Extra: Materiais de Ensino

1. [Github Awesome VS Code](#) (*repositório no GitHub*)  
a. <https://bit.ly/2v04ahj>
2. [Microsoft Tips and Tricks](#) (*site oficial VS Code*)  
a. <https://bit.ly/2UBrK3h>
3. [VS Code Can Do That?](#) (*site*)  
a. <https://vscodecandothat.com/>

## Extra: Lista de Diversos Outros Atalhos

1. [Microsoft Keybinding](#) (*site oficial VS Code*)  
a. <https://bit.ly/2KIRbGz>
2. Clique na engrenagem no canto inferior esquerdo e selecione Keyboard Shortcuts.

---

## Extra: Personalização

Temas, Fontes e Ícones são como cores de camisa, cada um tem a sua preferência e todas são válidas. Também deixamos aqui algumas configurações extras para a edição via JSON :D

### Extra: Dica de Ícones

Assim como fontes e temas, também vamos deixar algumas sugestões para os ícones, sendo elas:

1. [\*VSCode-Icons\*](#)
2. [\*Material Icon Theme\*](#)
3. [\*VSCode Great Icons\*](#)
4. [\*Simple Icons\*](#)
5. [\*Monokai Pro\*](#)

### Extra: Dicas de Fontes

Algumas dicas de fontes para programação podem ser encontradas nos links abaixo, com prévias das fontes em uso, para encontrar a sua ideal.

1. [\*Programming Fonts—A List\*](#) (blog no Medium)
  - a. <https://bit.ly/2OXOaFE>
2. [\*11 Best Programming Fonts\*](#) (blog no Medium)
  - a. <https://bit.ly/2s14lHr>
3. [ProgrammingFonts.org](https://app.programmingfonts.org/) (Teste Suas Fontes) (site)
  - a. <https://app.programmingfonts.org/>

## Extra: Dica de Temas

Escolher um tema sem dúvidas é a mais difícil das customizações, muitos preferem o estilo Dark, para evitar dor nos olhos, outros preferem cores claras, mas ambos passam mais de horas escolhendo as cores certas para cada estilo. Para evitar tanta dor de cabeça, escolhemos alguns links onde outros profissionais já tiveram essa árdua tarefa de passar horas escolhendo um tema, tudo com uma prévia feita para você decidir ainda mais rápido qual seu tema favorito :)

1. *Best VS Code themes* (blog no Medium)
  1. <https://bit.ly/2X93GS8>
2. *6 Hottest VS Code Themes To Use in 2019* (blog no Medium)
  1. <https://bit.ly/2G8Brw2>
3. *Top 10 Visual Studio Code themes* (blog no Medium)
  1. <https://bit.ly/2P6V7UT>
4. *VS Code Themes* (site)
  1. <https://bit.ly/2UA7miY>

## Extra: Configurações Produtividade 10000%

```
"editor.formatOnSave": true, // Formata o arquivo ao salvar
"editor.formatOnPaste": true, // Formata o arquivo ao colar algo
"javascript.format.enable": true, // Ativar a formatação em arquivo JS
"files.trimTrailingWhitespace": true, // Remove os espaços no final da linha
"editor.detectIndentation": true, // Detecta a indentação automaticamente
"editor.quickSuggestionsDelay": 0, // Delay para dar sugestões
"files.autoSave": "afterDelay", // Habilitar autosave
"files.autoSaveDelay": 100, // Delay do autosave
"breadcrumbs.enabled": true, // Habilitar Breadcrumbs
"workbench.editor.highlightModifiedTabs": true, // Habilita mostrar quais abas foram editadas e não salvas ainda
"explorer.sortOrder": "type", // Ordena os arquivos por tipo primeiro e depois por ordem alfabética
"files.trimFinalNewlines": true, // Retira as linhas em branco no final do arquivo
"editor.mouseWheelZoom": true, // Habilita poder dar zoom com Ctrl + Scroll do mouse
"editor.renderWhitespace": "boundary", // Mostra 'pontinhos' no início da linha indicando espaços
"git.autofetch": true, // Faz um auto-fetch no git de tempo em tempo
"[python]": { // Tabulação seguindo pep8
  "editor.insertSpaces": true,
  "editor.tabSize": 4
},
```

```

"emmet.includeLanguages": { // Linguagens com emmet
  "html": "html",
  "erb": "erb",
  "javascript": "javascriptreact",
  "vue": "html",
  "ejs": "html"
},
"files.exclude": { // Arquivos para ignorar no VS Code
  "**/.git": true,
  "**/.svn": true,
  "**/.hg": true,
  "**/CVS": true,
  "**/.DS_Store": true,
  "**/_site": true,
  "**/tmp": true,
  "**/*.pyc": true,
  "**/*.asset-cache": true,
  "**/*.cache": true,
  "**/*.jekyll-cache": true,
  "**/*.tweet-cache": true,
  "**/__pycache__": true,
  "**/.pytest_cache": true,
  "**/*.egg-info": true,
  "**/*.coverage": true,
  "**/celerybeat-schedule": true,
  "**/node_modules": true,
  "**/bower_components": true
},
"files.eol": "\n", // Declara o tipo de fim de linha
"[markdown]": { // Não remove espaços no fim da linha,
  pois muitos programadores usam espaços duplos para
  indicar quebra de linha. Remova caso não seja seu caso.
    "files.trimTrailingWhitespace": false
  }
}

```

## Extra: Configurações Customização 10000%

```
"workbench.colorCustomizations": { // Troque #FFFFFF pela sua cor favorita :)
  "activityBar.background": "#FFFFFF", // Cor da ActivityBar
  "statusBar.background": "#FFFFFF", // Cor da status bar
  "statusBar.foreground": "#FFFFFF", // Cor do texto da status bar
  "editorLineNumber.foreground": "#FFFFFF", // Cor do texto do número indicador de linhas
  "editorOverviewRuler.errorForeground": "#FFFFFF", // Cor do texto dos erros
  "titleBar.activeBackground": "#FFFFFF", // Cor da barra de título
  "titleBar.activeForeground": "#FFFFFF", // Cor do texto da barra de título
},
"editor.fontFamily": "Fira Code", // Fonte a ser utilizada
"editor.fontLigatures": true, // Ativar ligadura tipográfica (unir 2 ou mais símbolos em um só)
"editor.fontSize": 15, // Tamanho da fonte
"terminal.integrated.fontSize": 15, // Tamanho da fonte no terminal
"editor.minimap.enabled": false // Desabilita o minimapa lateral ('preview' do lado direito)
```

---

# Bônus

---

## Bônus: Cursos Gratuitos de VS Code

Além deste livro, também preparamos vários cursos em vídeo para vocês!

Os cursos são totalmente gratuitos e abordam de forma super prática e didática tudo o que apresentamos neste livro. Eles estão bem divididos, abordando apenas o necessário para você, como o que um desenvolvedor Front-End Web precisa saber, outro para o que um programador C# precisa conhecer, outro com o necessário para o programador Java, outro para Python e assim vai...

Tenho certeza que você irá gostar desse incrível bônus!

Para conferir todos os cursos acesse:

1. [Nossos Cursos - \(http://bit.ly/ByLearnVSCode\)](http://bit.ly/ByLearnVSCode)
2. [Nosso site - \(http://bit.ly/ByLearn\)](http://bit.ly/ByLearn)

---

## **Autores**

---

**Felipe Cabrera Ribeiro dos Santos**

**ByLearn Cursos Online**