

# Git e Github

Github no Visual Studio



---

## Índice

Introdução	3
Abertura de um novo terminal	4
Clonando um repositório com o Visual Studio	7

---

# Introdução

O *Visual Studio Code* é um editor de código multiplataforma e suporta a sintaxe de um grande número de linguagens de programação.

A ferramenta fornece suporte e assistência a linguagens de diversos escopos: HTML, CSS, JavaScript, várias variantes de C, Java, SQL, PHP, Ruby, Visual Basic ou JSON entre muitas outras, suportando destaque, recuo automático, trechos e auto-completamento.

É um dos editores de código mais utilizados no mundo, com uma grande comunidade de apoio, uma infinidade de fóruns de ajuda e um enorme número de *plug-ins* disponíveis para facilitar a tarefa de programação, qualquer que seja a linguagem em que trabalhamos.

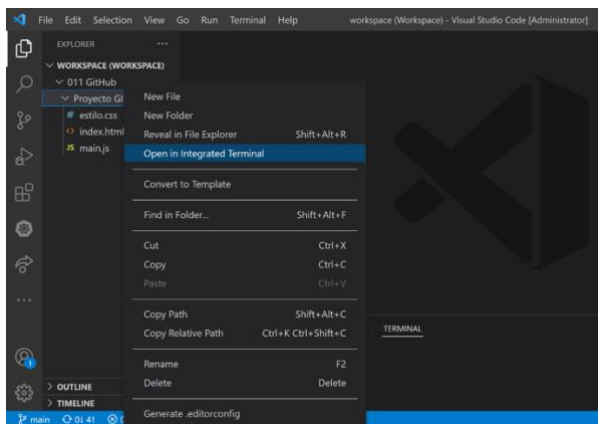
Além disso, o Visual Studio tem integração nativa com *Git*. Ou seja, ele vem pronto para trabalhar com *Git* sem a necessidade de instalar software adicional.

Para começar a trabalhar com *Git* no *Visual Studio*, tudo o que temos que fazer é ter um projeto criado, ou seja, um ou mais arquivos para começar a trabalhar, abrir o terminal em nosso diretório de trabalho e digitar:

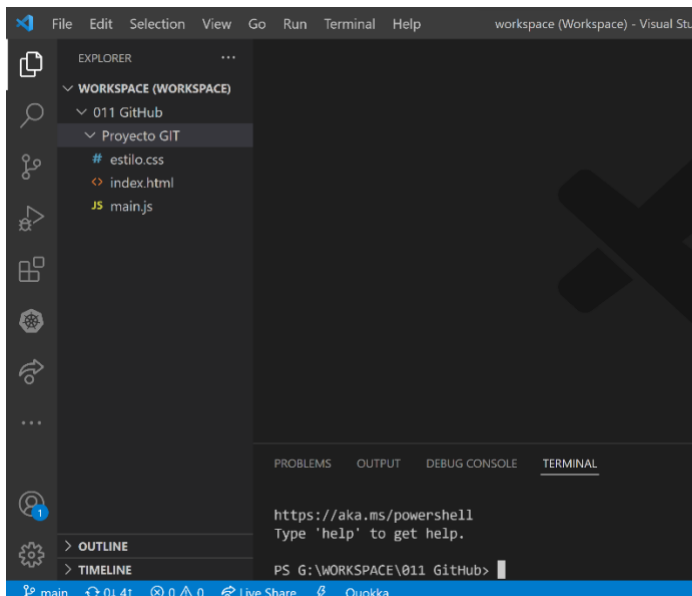
`git init`

## Abertura de um novo terminal

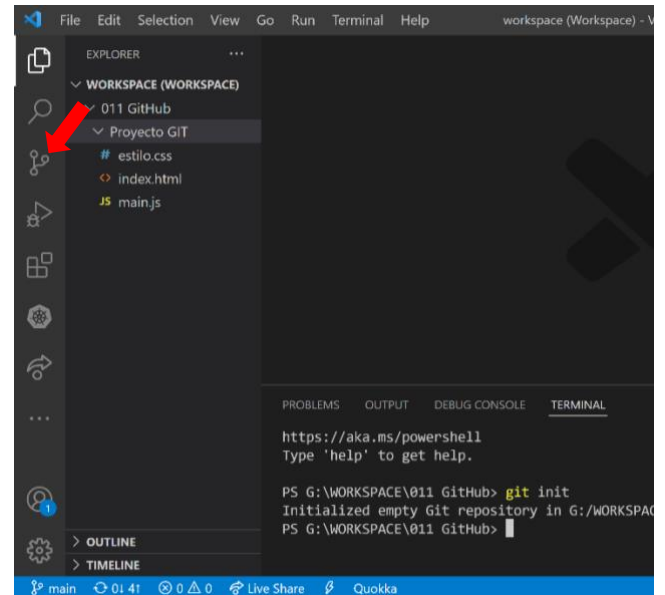
Para abrir um novo terminal, devemos nos colocar em nosso diretório de trabalho, pressionar o botão direito do mouse e clicar na opção "Open in integrated terminal":



Isto abrirá o terminal integrado do *Visual Studio*. Este terminal funciona de forma semelhante ao terminal *Git Bash* que temos usado até agora.



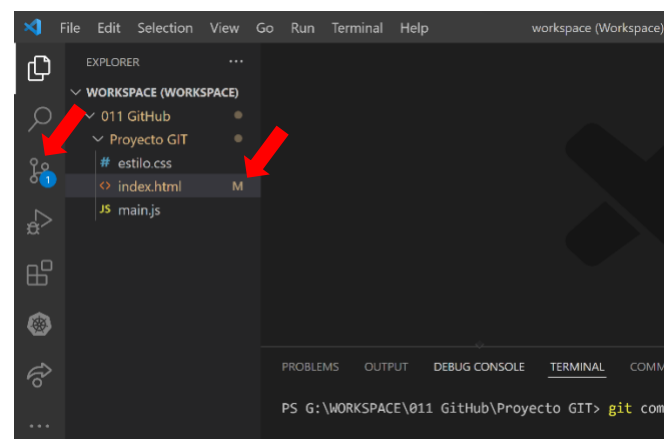
Uma vez que o terminal esteja aberto, tudo que temos que fazer é digitar `git init` e *Git* começará a monitorar nosso projeto.



O ícone destacado em vermelho é o que exibe o menu de opções *Git*. Ele nos informará sobre todas as mudanças pendentes em nossos arquivos para fazer *commit*.

Agora precisamos adicionar nossos arquivos à *staging área*.

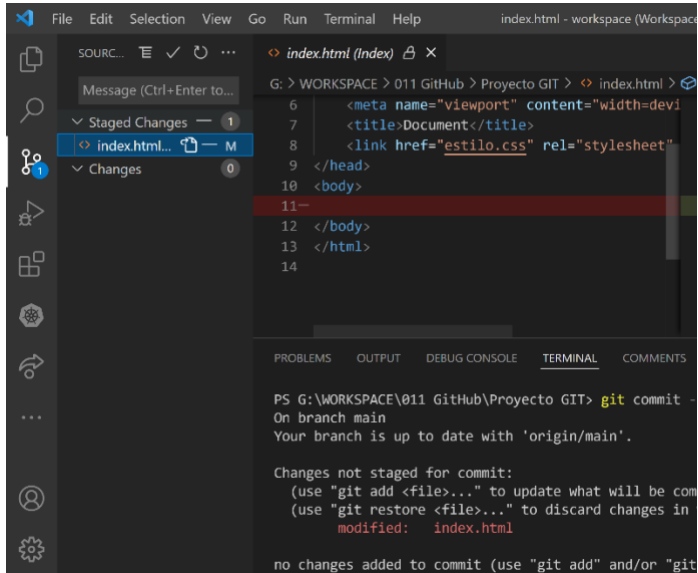
Lembre-se de que isto é feito com o `git commit -m "descripción"`.



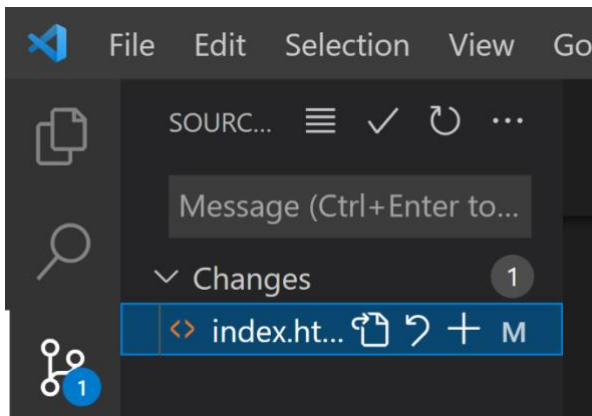
A letra **M** indica que há mudanças pendentes de sincronização nesse arquivo. E se virmos o ícone das filiais de *Git*, ele também indica essa mudança.

Todos os comandos *Git* que usamos até agora, `add`, `branch`, `commit`, `merge`...etc., são perfeitamente aplicáveis no terminal do Visual Studio.

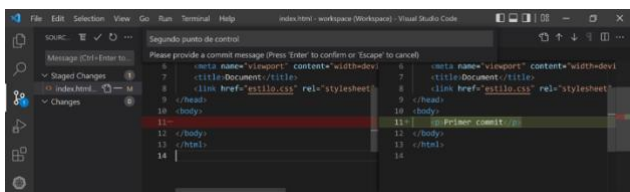
Se clicarmos em nosso arquivo, veremos como o *Visual Studio* abre uma tela dupla na qual nos mostra tanto a versão do nosso arquivo antigo quanto a nova versão com as mudanças que fizemos:



Mas também teremos a opção de fazer *commits* sem ter que passar pela *staging area*, bastando clicar no ícone marcado com a seta vermelha:

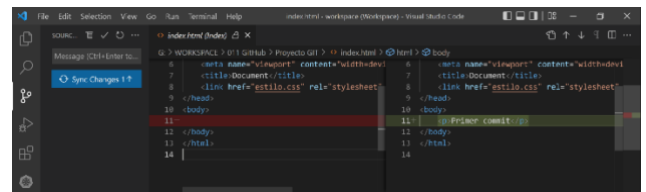


Clicando no ícone com a seta vermelha abrirá uma pequena janela onde teremos que escrever a descrição de nosso *commit*. Neste caso, vamos chamá-lo de "segundo ponto de verificação":

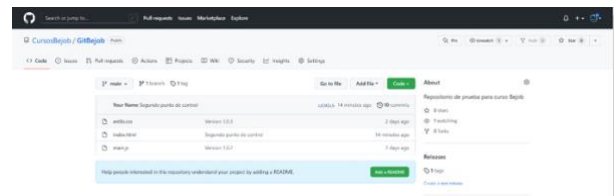


Digitar a descrição e pressionar ENTER criará nosso *commit*.

Neste momento, temos nossos arquivos atualizados, mas localmente. Para carregar as novas versões dos arquivos no repositório *GitHub*, teremos que clicar no botão "Sync Changes":



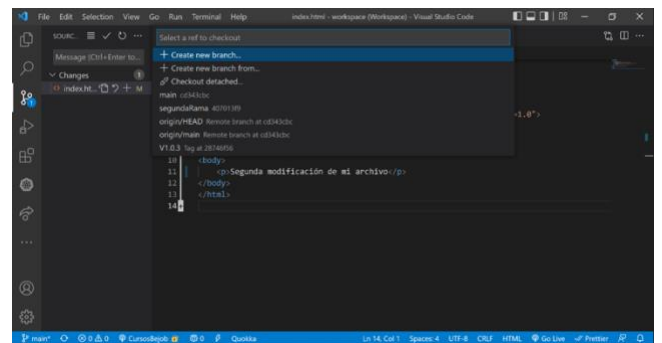
E teremos os arquivos atualizados e carregados para a nuvem de *GitHub*:



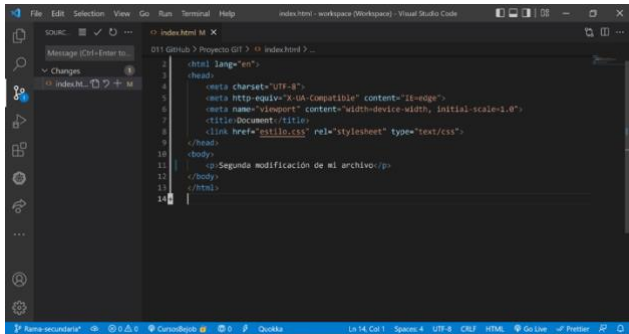
**Nota:** Para que esta sincronização ocorra, devemos estar registrados no *GitHub* e ter o repositório de nuvens criado.

Como podemos ver, o *Visual Studio* nos permite fazer o passo a *Staging area* e o *commit*, bastando clicar em um ícone.

Para criar novas filiais podemos fazê-lo usando os comandos que já conhecemos ou clicando na opção destacada na seguinte captura:

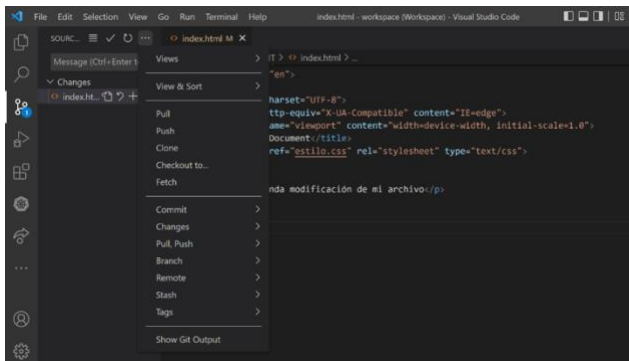


Clique sobre ele e um menu será aberto com diferentes opções, incluindo "Create new Branch", nomeamos nossa nova filial, por exemplo "Ramo secundário" e poderemos trabalhar a partir desta filial:



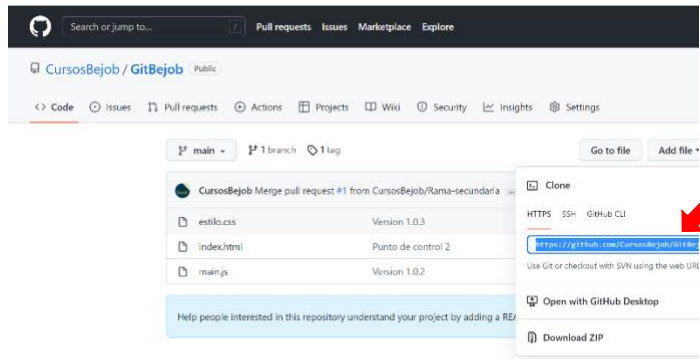
Além disso, a partir desta opção, podemos alternar entre os diferentes ramos que temos.

Esta operação e todas aquelas relacionadas ao *GitHub* também podem ser feitas a partir do menu suspenso que se abre quando você clica no ícone com os três pontos:



# Clonando um repositório com o Visual Studio

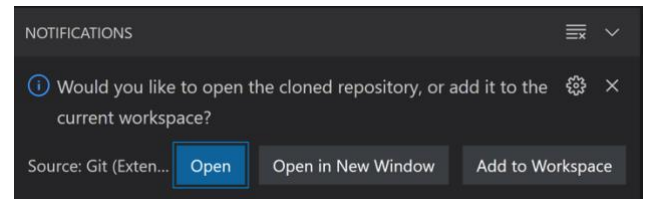
Se já temos um repositório no GitHub e queremos que nosso Visual Studio se sincronize com ele, devemos copiar a URL de nosso repositório:



E nos pedirá para dizer em qual diretório do nosso disco rígido queremos clonar o repositório.

Uma vez feito isso, todos os arquivos do repositório serão copiados para o diretório que escolhemos.

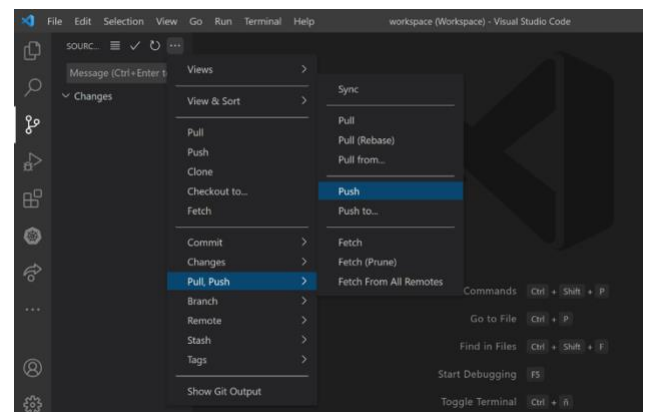
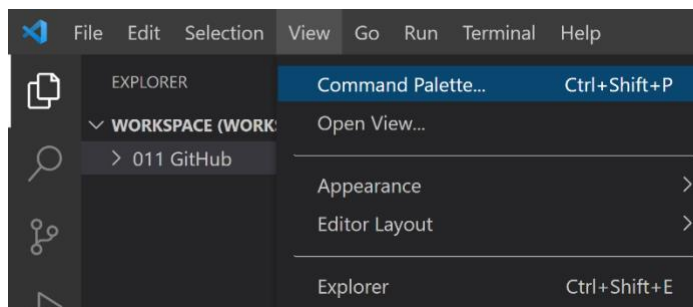
Receberemos uma pergunta rápida se queremos adicionar o repositório ao nosso ambiente de trabalho:



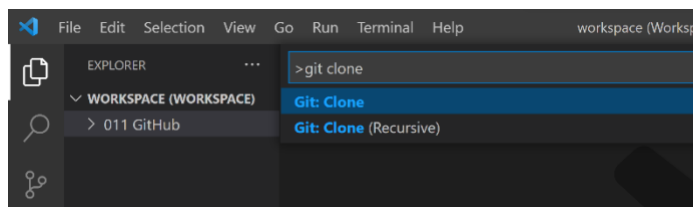
E se clicarmos em "open" podemos continuar trabalhando no repositório a partir do local.

Se fizermos alterações em qualquer um dos arquivos locais e quisermos carregá-los para a nuvem, teremos de executar o comando **push**:

Depois, vamos ao Visual Studio e abrimos o menu "view/Command Palette":



Digite o comando "git clone":



Ele nos pedirá a URL de nosso repositório:

