

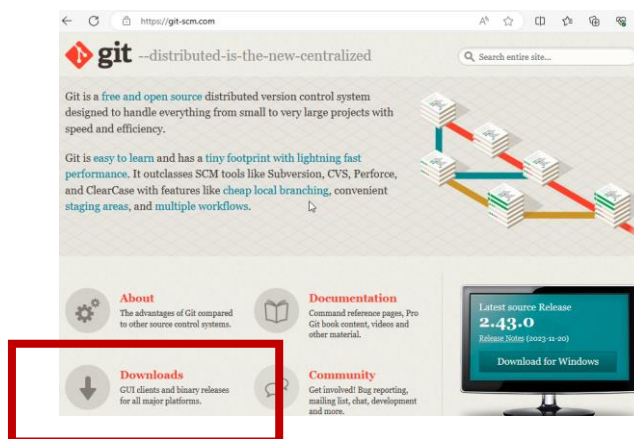
MODALIDADE:	Educação e Formação de Adultos (EFA)	EFA NS (Profissional)	
CURSO:	Programador/a Informático/a		
UFCD:	Programação em C/C++ - avançado	CÓDIGO UFCD:	0810
FORMADOR/A:	Bruno Silva	DATA:	

## OBJETIVOS

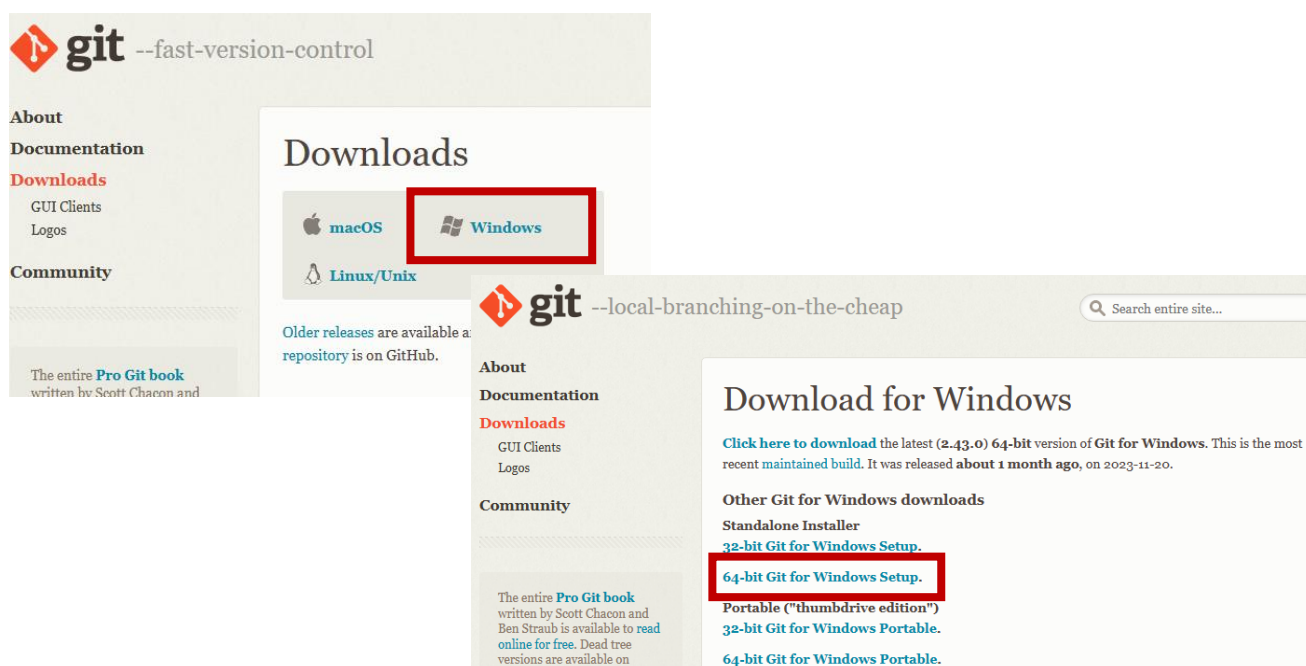
- Tutorial de como instalar GIT

## Processo de instalação GIT (versão Windows)

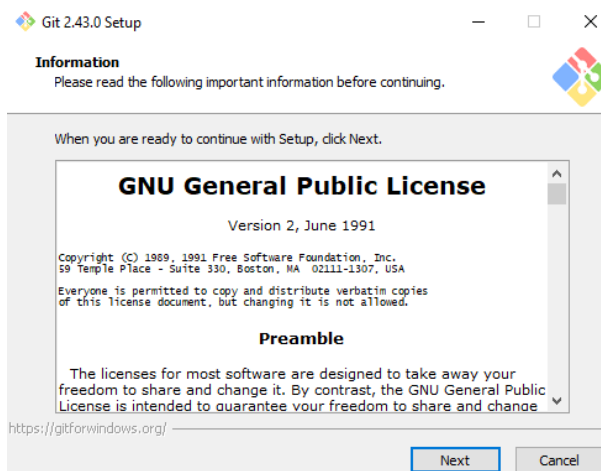
**Passo 1** – Abrir página oficial do Git (<https://git-scm.com/>) e clicar na opção downloads;



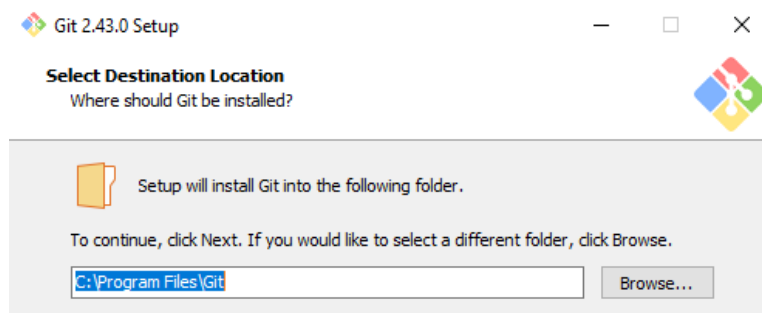
**Passo 2** – Selecionar a versão Windows versão 64 bits:



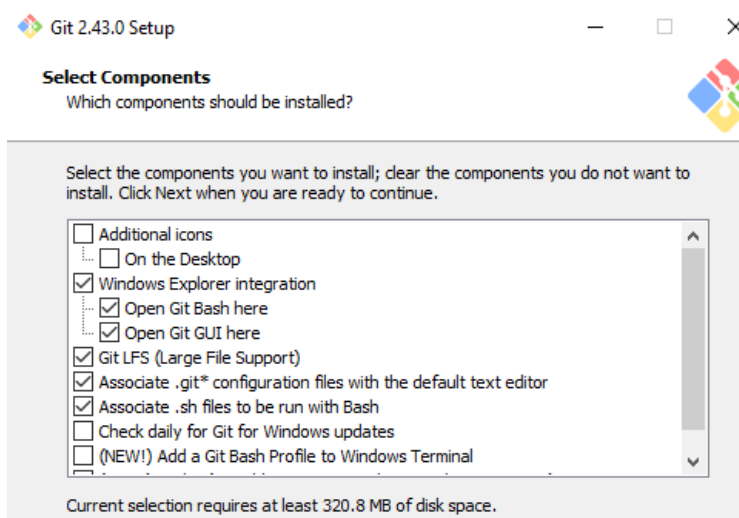
## Passo 3 – Aceitar termos e condições:



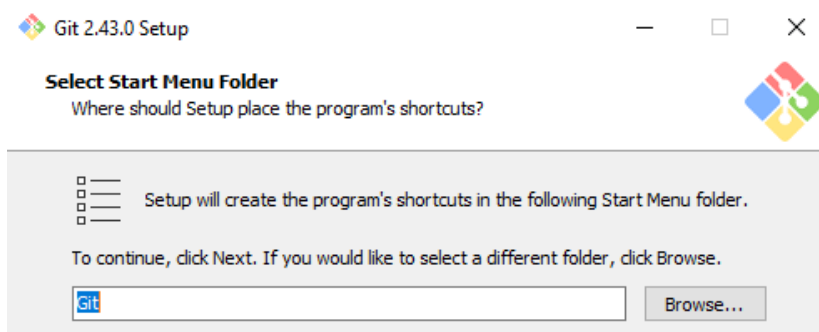
## Passo 4 – Indicar o local da instalação no computador (podemos deixar o recomendado):



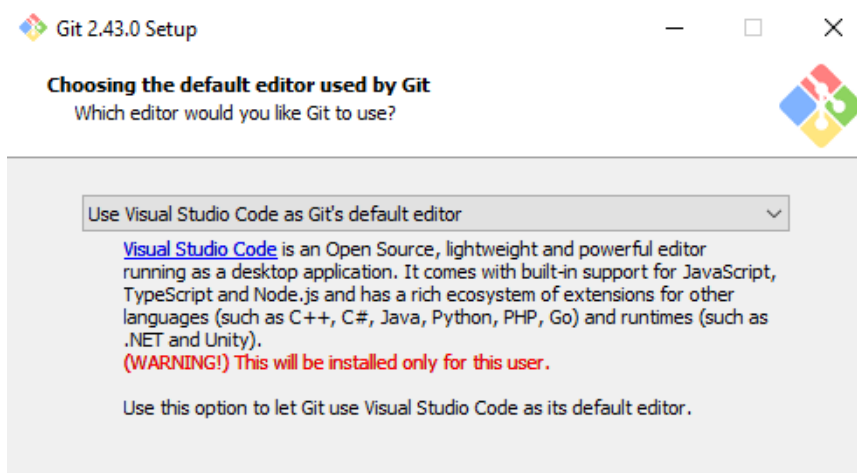
## Passo 5 – Selecionar os componentes da instalação:



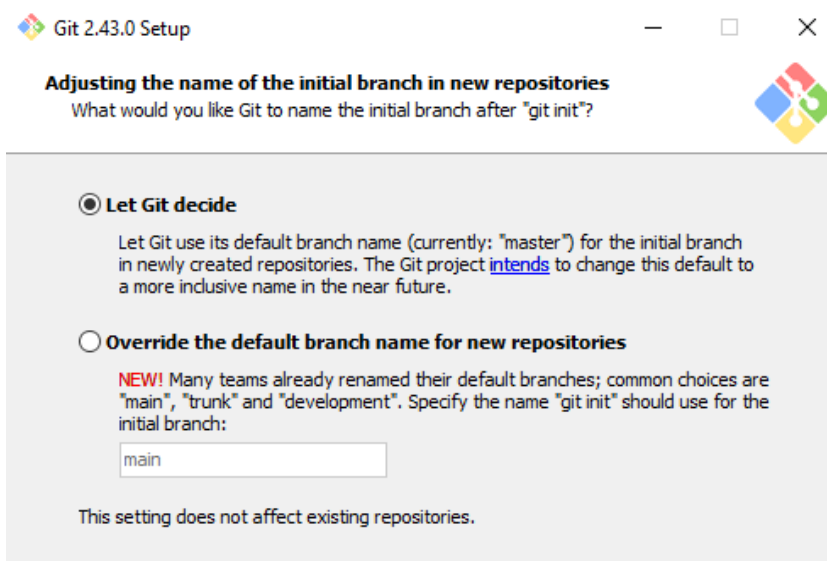
**Passo 6** – Definir o nome dos ícones da aplicação GIT no sistema operativo:



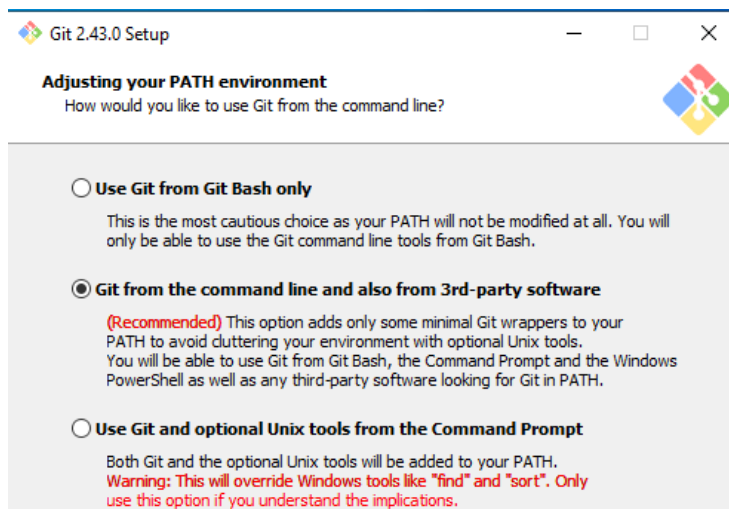
**Passo 7** – Identificar o editor de código a ser usado (selecionar a opção “Use Visual Studio Code ...”):



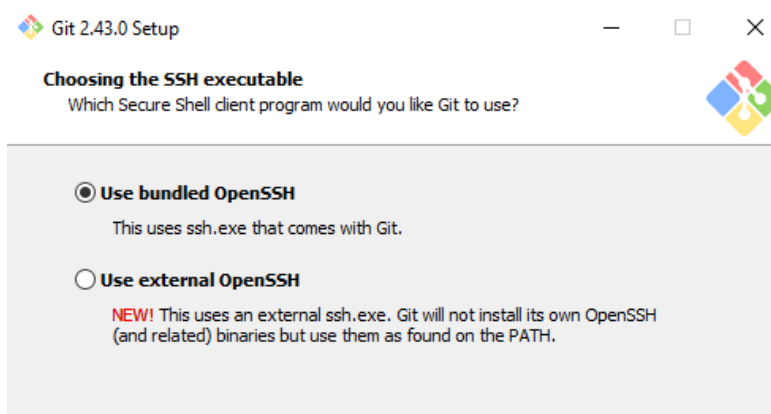
**Passo 8** – Definir como é gerido os nomes dos “branches”, ou seja, ramificações do código principal (deixar o recomendado):



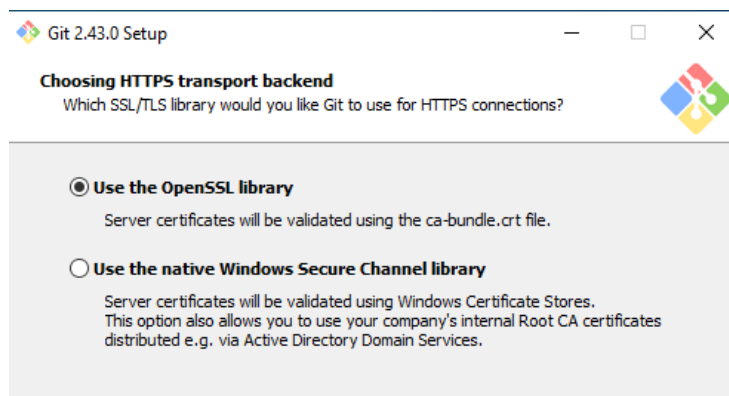
**Passo 9** – Definir o caminho do sistema para carregar os dados do sistema (deixar o recomendado):



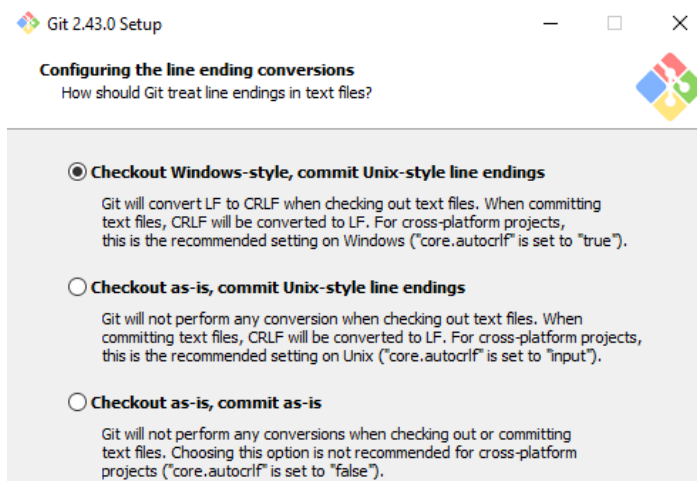
**Passo 10** – Como deve gerar os dados, caso seja preciso usar chaves públicas/privadas (deixar o recomendado):



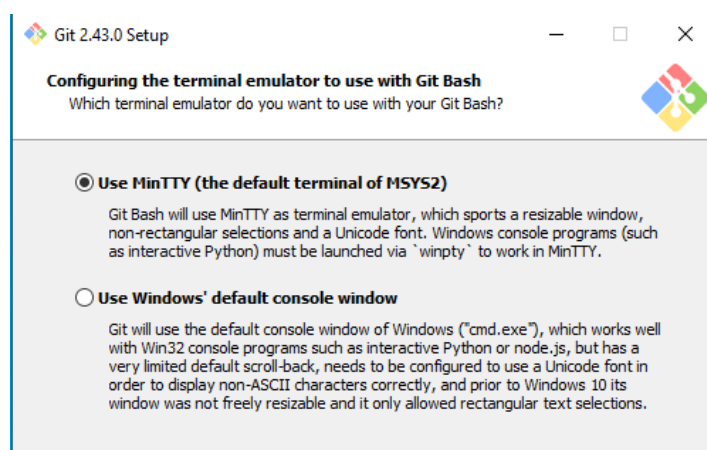
**Passo 11** – Continuação do passo anterior e indicar a 1ª opção “OpenSSL” (deixar o recomendado):



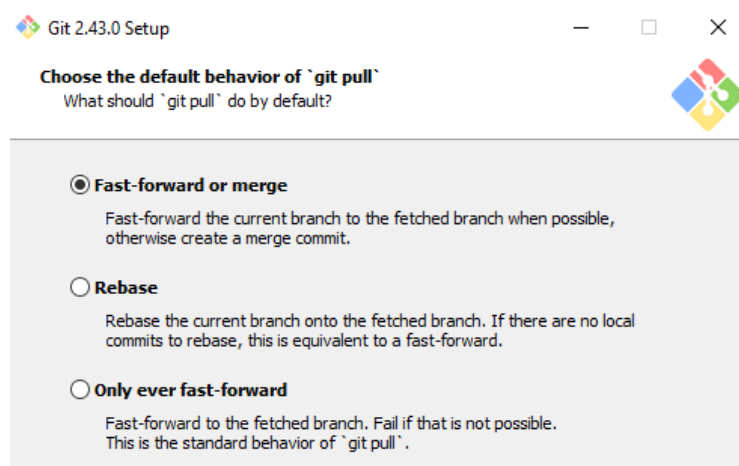
**Passo 12** – Configurar o modo de como aparece a interface de código no ecrã (deixar o recomendado para o ambiente windows):



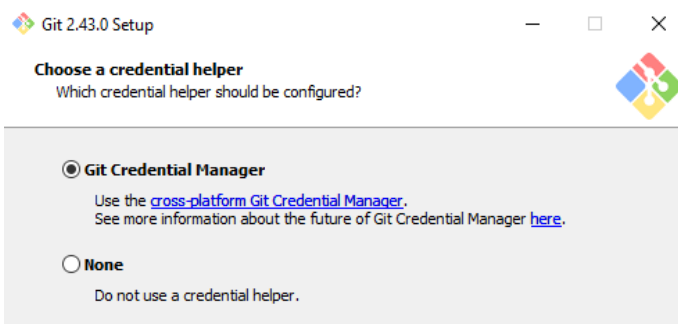
**Passo 13** – Continuação do passo anterior, mas neste caos, indicar se pretendemos usar o editor personalizado do MinTTY (deixar o recomendado):



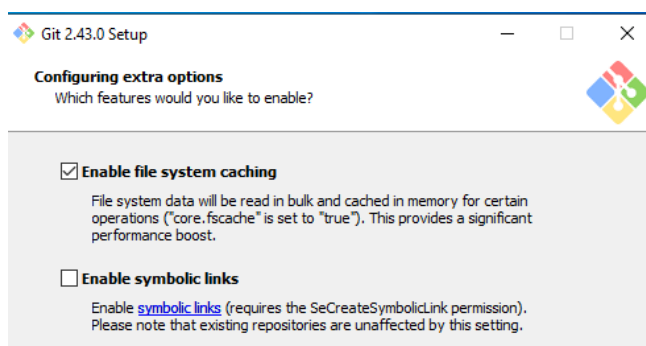
**Passo 14** – Definir o comportamento da operação “git pull” (deixar o recomendado):



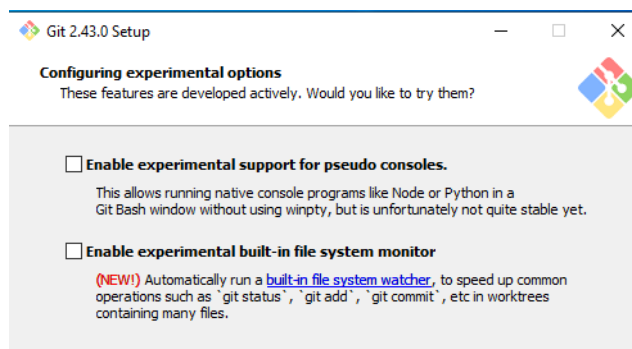
**Passo 15** – Definir a gestão de credenciais (deixar o recomendado):



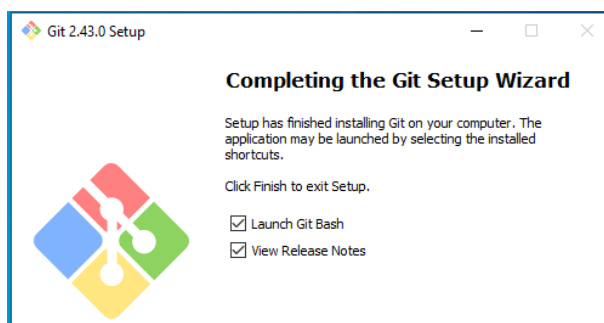
**Passo 16** – Definir opções extra de armazenamento dos dados (deixar o recomendado):



**Passo 17** – Ativar opções experimentais (não deixar nenhuma opção ativa):



**Passo 18** – Deixar instalar o programa e no final confirmar a conclusão da instalação:



## Comandos GIT

<b>Nome/Comando</b>	<b>Descrição</b>
<b>Inicializar</b> \$ git init	Inicializa um novo repositório Git, dentro de uma pasta.
<b>Clonar</b> \$ git clone	Obtém uma cópia de um repositório existente.
<b>Adicionar</b> \$ git add <ficheiro>	Usado para fazer track & stage de ficheiros ou apenas stage de um ficheiro modificado.
<b>Commit</b> \$ git commit -m "texto"	Usado para guardar um snapshot no repositório
<b>Status</b> \$ git status	Usado para ver o estado em que estão os ficheiros no diretório
<b>Log</b> \$ git log	Usado para ver o histórico de commits do repositório.
<b>Remover</b> \$ git rm <ficheiro>	Usado para remover um ficheiro do diretório e atualizar corretamente o repositório.
<b>Pull (descarregar)</b> \$ git pull <repositorio_remoto>	Obter as atualizações de um repositório (esta operação pode causar conflitos que deverá resolver manualmente)
<b>Push (enviar)</b> \$ git push <repositorio_remoto> <nome_branch>	Enviar as atualizações para um repositório / brach
<b>Criar Branch</b> \$ git branch <nome_branch>	Usado para criar uma nova branch, a partir da atual.
<b>Mover branch</b> \$ git checkout <nome_branch>	Usado para alternar entre branches
<b>Merge (juntar branch)</b> \$ git merge <nome_branch_a_juntar>	Usado para "juntar/fundir" histórico bifurcado. Por exemplo, permite juntar o trabalho numa branch ao trabalho na master. Erros de conflito devem ser cuidadosamente analisados