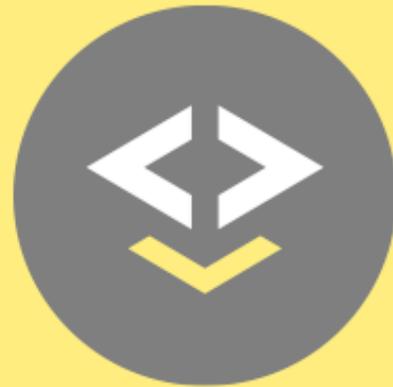


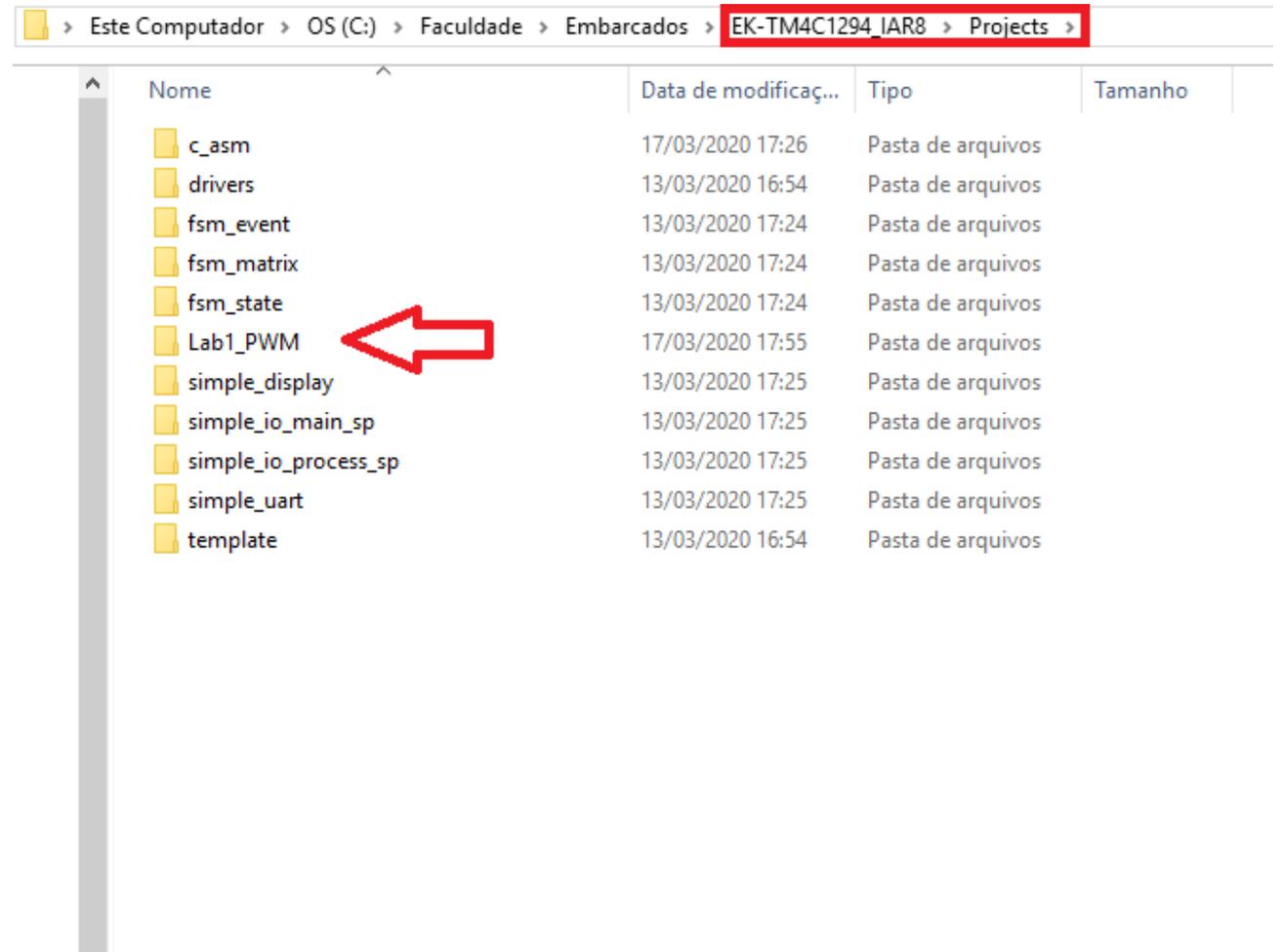
Montagem base para Projetos no IAR EWARM



IAR Embedded
Workbench

Criando o Projeto

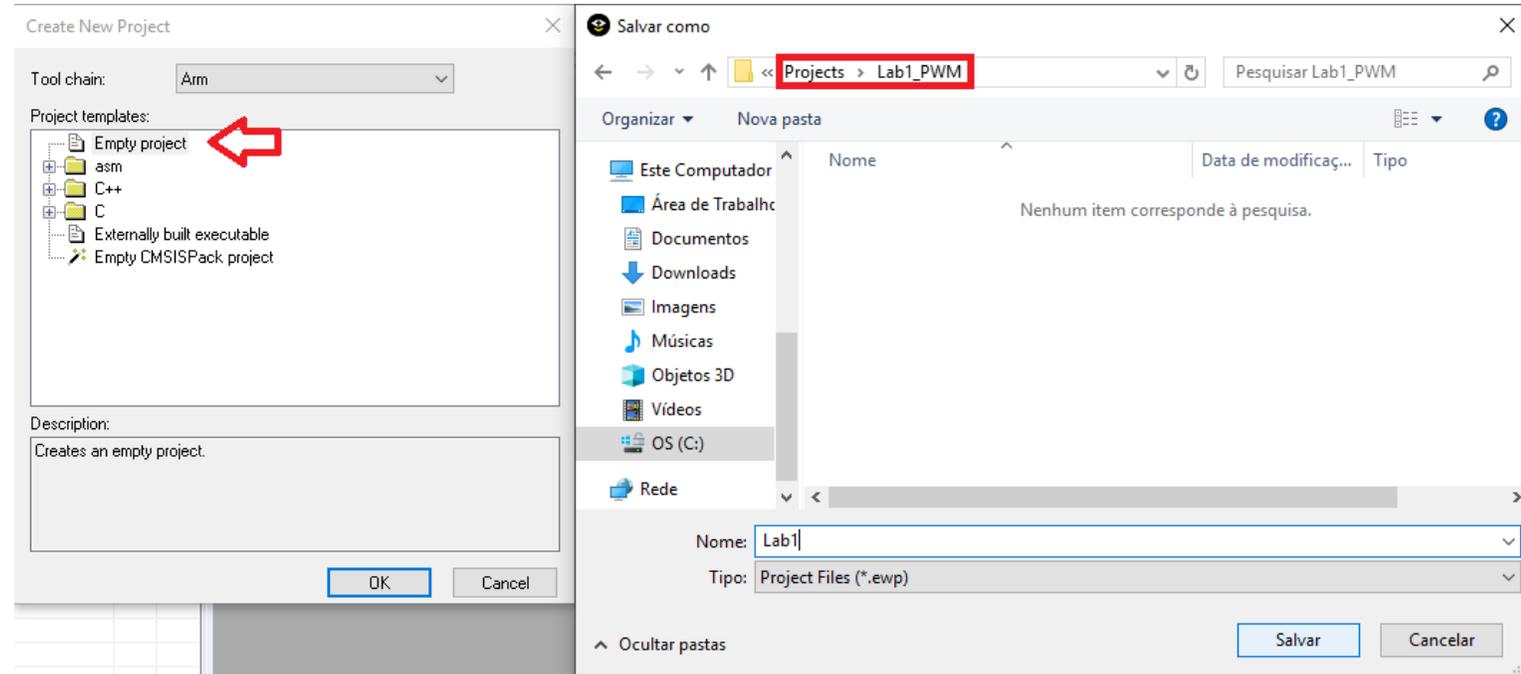
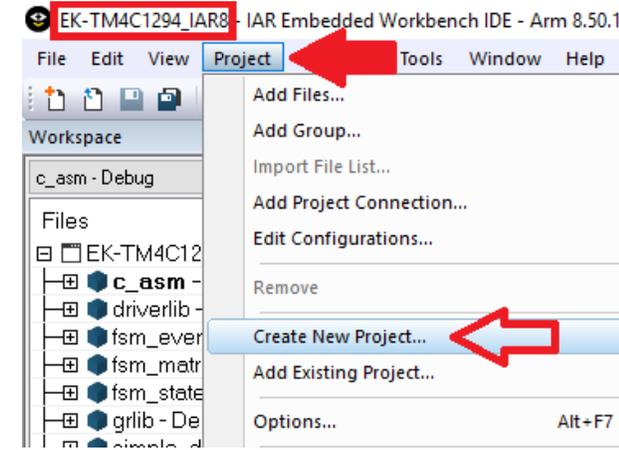
- Criar um subdiretório com o nome do projeto dentro do subdiretório “**Projects**”
 - Este se encontra no exemplo baixado no site do professor
- Nosso exemplo aqui é criada a pasta **Lab1_PWM**



Nome	Data de modificaç...	Tipo	Tamanho
c_asm	17/03/2020 17:26	Pasta de arquivos	
drivers	13/03/2020 16:54	Pasta de arquivos	
fsm_event	13/03/2020 17:24	Pasta de arquivos	
fsm_matrix	13/03/2020 17:24	Pasta de arquivos	
fsm_state	13/03/2020 17:24	Pasta de arquivos	
Lab1_PWM	17/03/2020 17:55	Pasta de arquivos	
simple_display	13/03/2020 17:25	Pasta de arquivos	
simple_io_main_sp	13/03/2020 17:25	Pasta de arquivos	
simple_io_process_sp	13/03/2020 17:25	Pasta de arquivos	
simple_uart	13/03/2020 17:25	Pasta de arquivos	
template	13/03/2020 16:54	Pasta de arquivos	

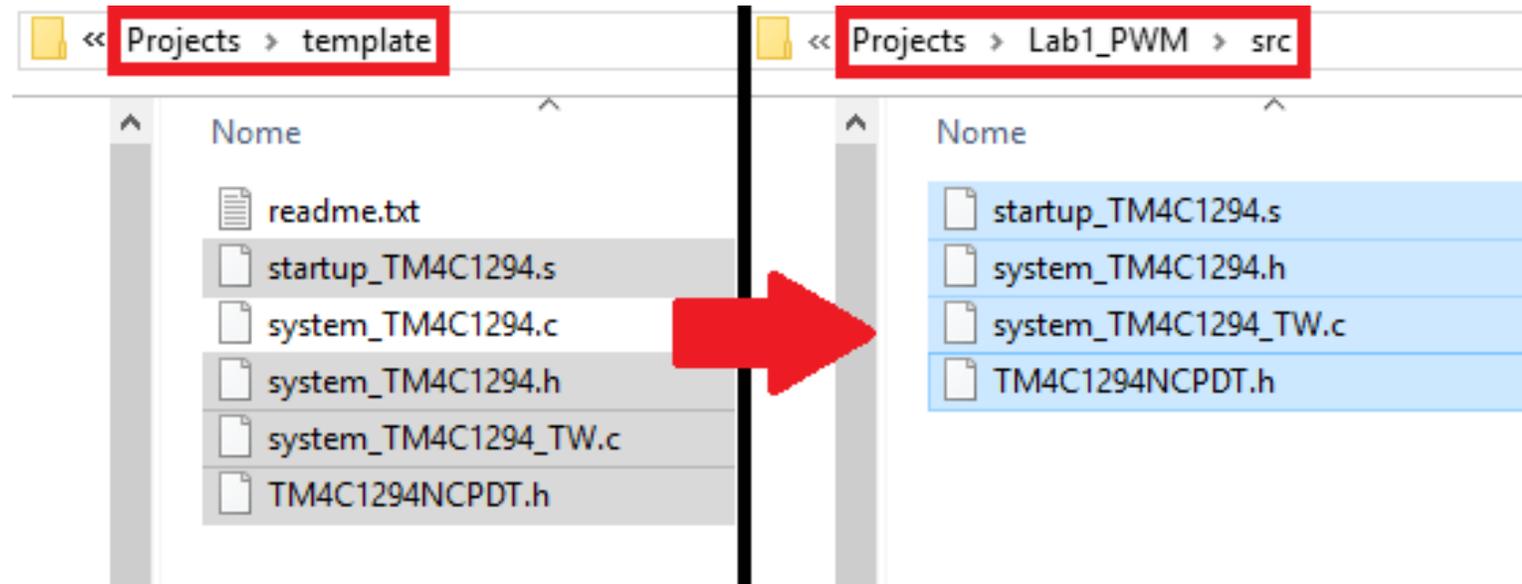
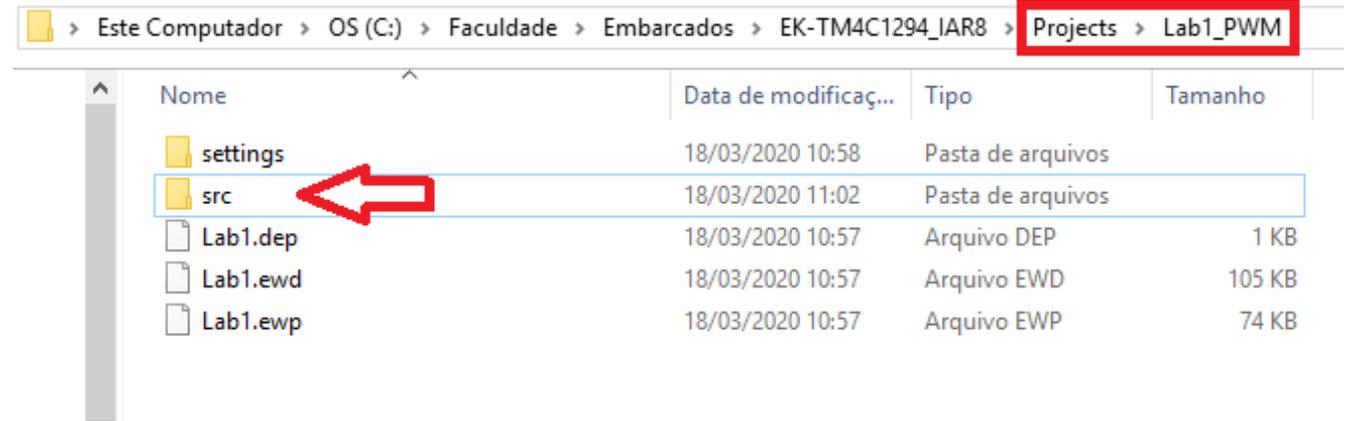
Criando o Projeto

- No IAR, criar um novo projeto dentro do “EK-TM4C1294_IAR8.eww”, workspace baixado junto com o exemplo
 - Selecionar Menu “Project” -> “Creat new Project”
 - Selecionar “Empty Project”
- Salvar o arquivo de projeto no subdiretório recém-criado (Lab1_PWM)



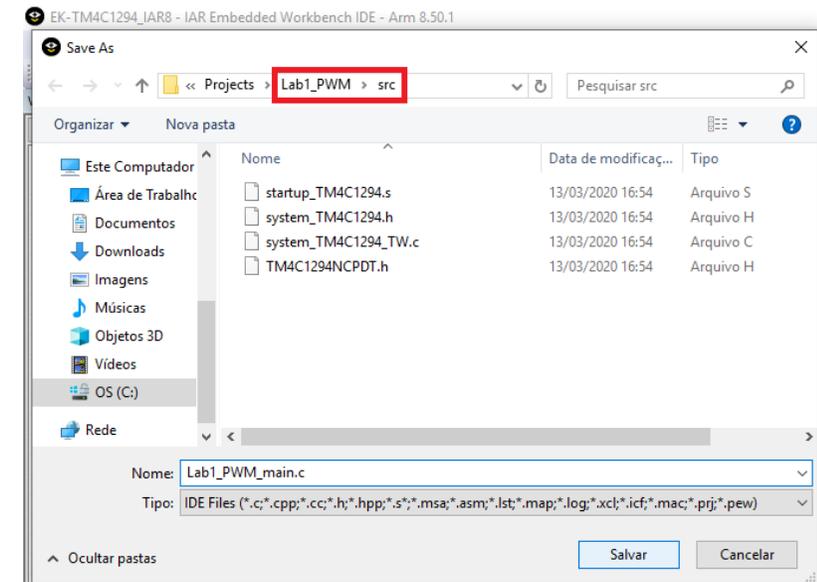
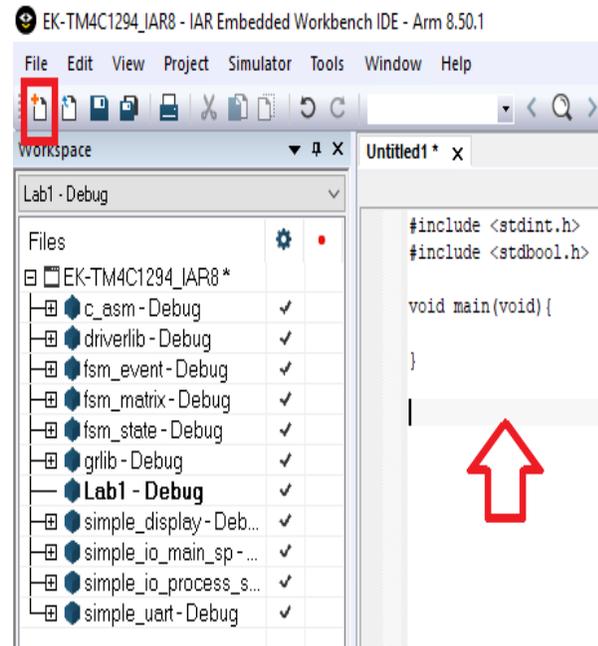
Criando o Projeto

- Criar um subdiretório “**src**” para armazenar os arquivos do código-fonte do novo projeto
 - Todos códigos-fonte que serão criados ficarão nesta pasta. Por exemplo o **main.c** do projeto como faremos mais para frente
- Copiar os arquivos do subdiretório “**template**” para o subdiretório “**src**” recém-criado
 - Copiar **ou** o arquivo de sistema **system_TM4C1294.c** (se a biblioteca driverlib não for utilizada) **ou** o arquivo de sistema **system_TM4C1294_TW.c** (se a biblioteca driverlib for utilizada)
 - No exemplo utilizaremos a biblioteca driverlib



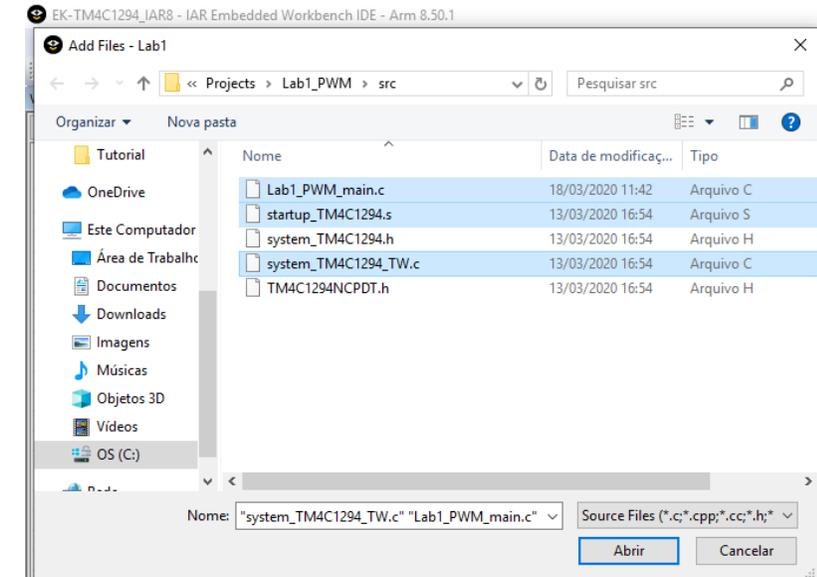
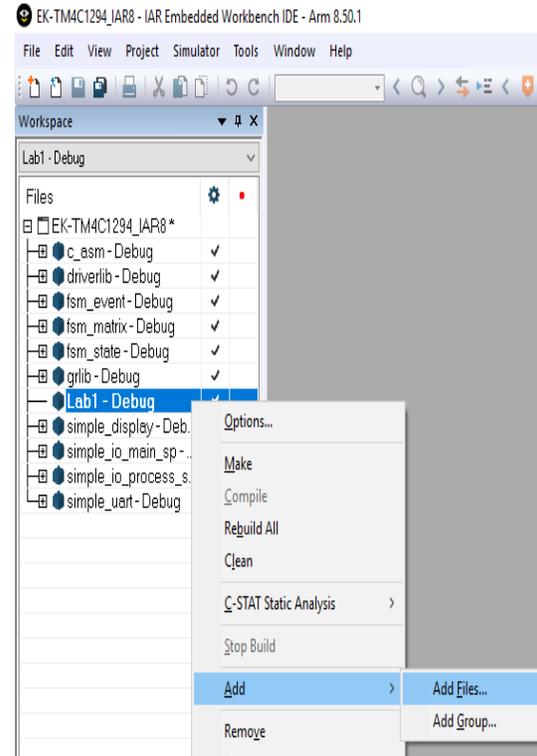
Arquivos do projeto

- No IAR, criar um novo arquivo para ser o “main” do projeto
 - Apertar no botão para criar um novo projeto
 - Escrevi um programa seco somente com bibliotecas básicas do c e a função “main” para primeira compilação
- Salvar o main na pasta “src” do projeto



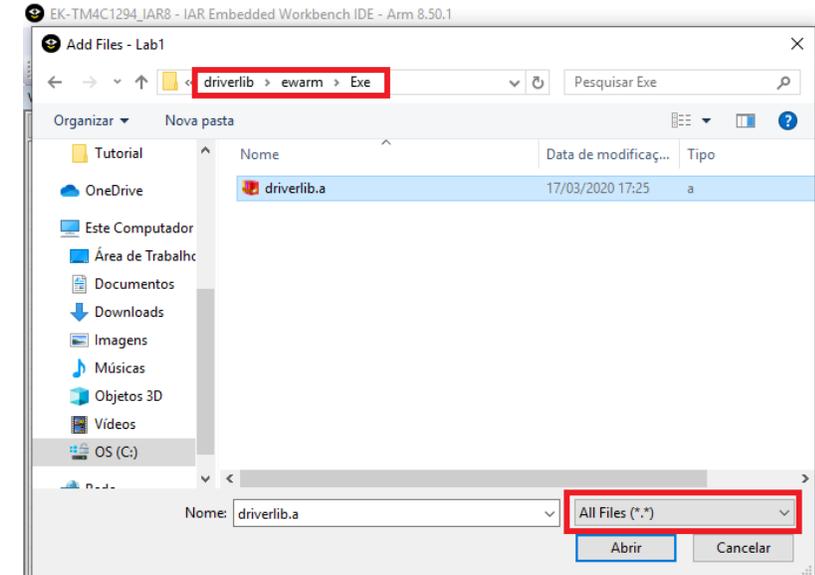
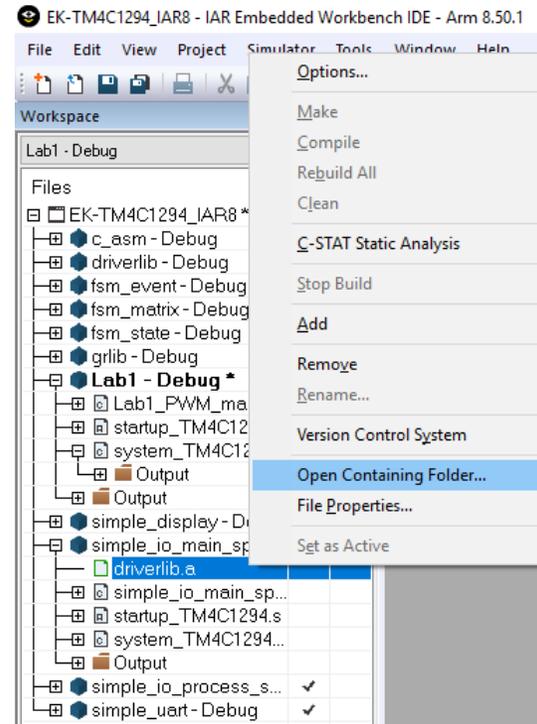
Arquivos do projeto

- No IAR, Clicar com o botão direito do mouse sobre o projeto recém criado na área de trabalho e adicionar os arquivos:
 - Arquivo de inicialização **startup_TM4C1294.s**
 - Arquivo de sistema **OU system_TM4C1294.c** (se a biblioteca driverlib não for utilizada) **ou** o arquivo de sistema **system_TM4C1294_TW.c** (se a biblioteca driverlib for utilizada)
 - Seus próprios arquivos de código-fonte para a aplicação do projeto (ASM, C ou C++) -> o **main** por exemplo



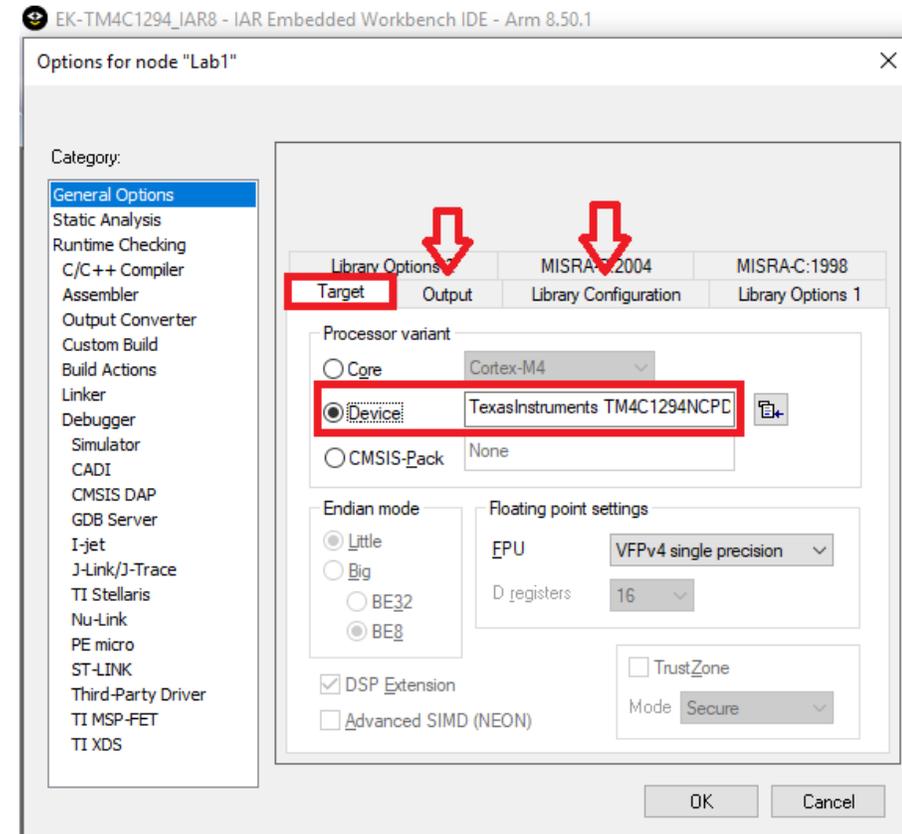
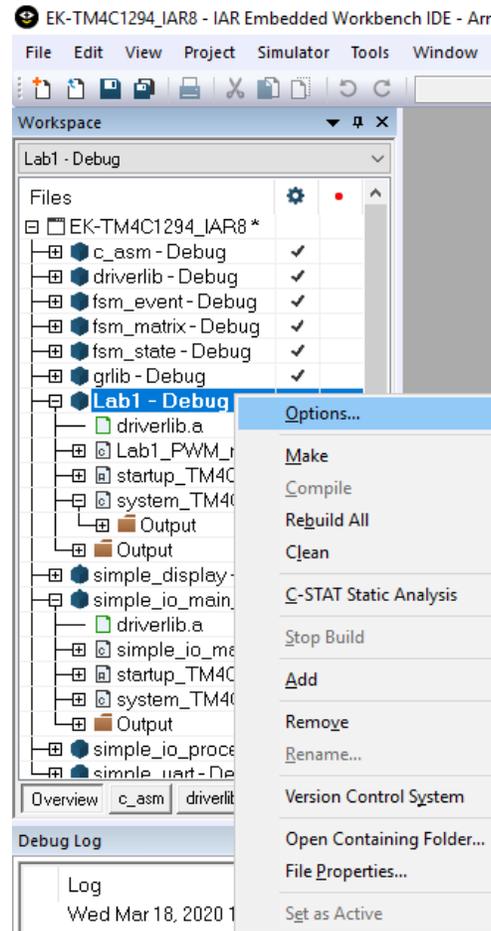
Arquivos do projeto

- Se a biblioteca “**driverlib**” for utilizada no projeto, adicionar o seu código-objeto : – **driverlib.a**
 - Adiciona o arquivo igual foi feito no passo anterior
 - Lembre-se de colocar “**all files**” para aparecer o arquivo
- Observação: a localização do código-objeto da biblioteca pode ser encontrada no projeto “**simple_io_main_sp**”.
 - Quando for abrir basta copiar o caminho onde encontra-se o arquivo e colar na barra quando for adiciona-lo



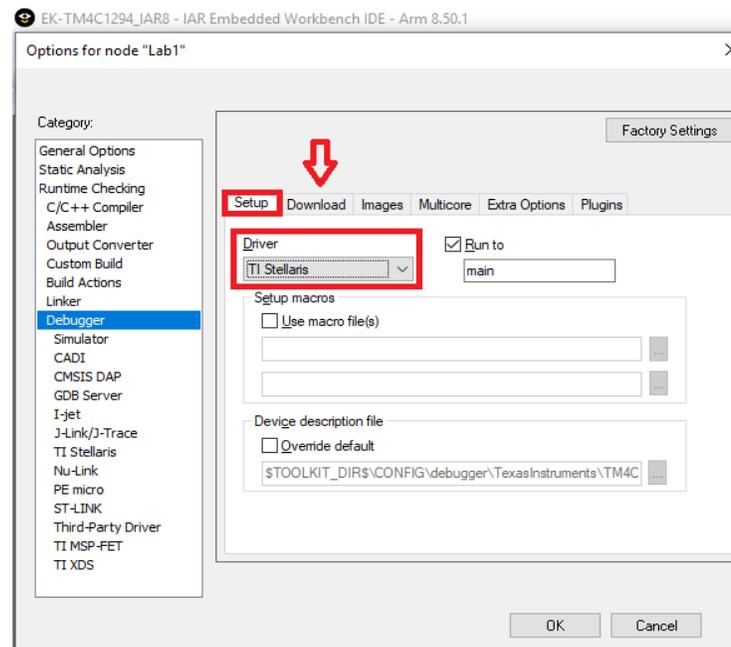
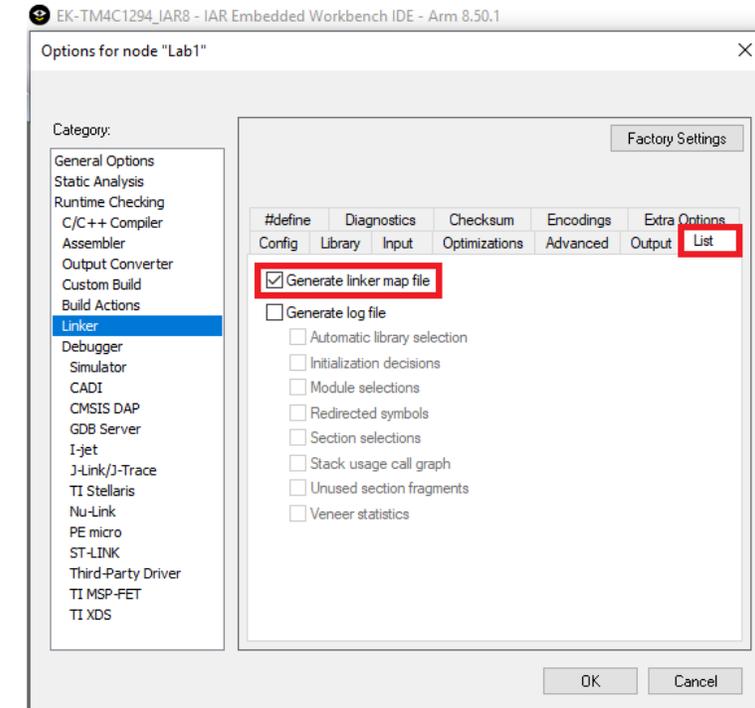
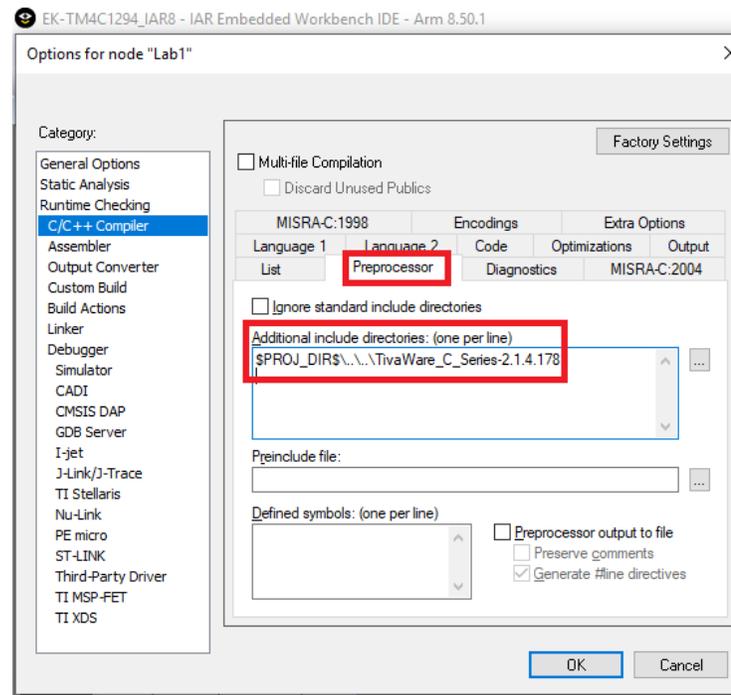
Opções do projeto

- Clicar com o botão direito do mouse sobre o projeto recém criado e selecionar **Options...**
- Menu **General Options**
 - **Target** → **Device:** Texas Instruments TM4C1294NCPDT
 - **Output** → **Executable**
 - **Library Configuration** → **Library:** Normal
 - **Library Configuration** → **CMSIS:** Use CMSIS



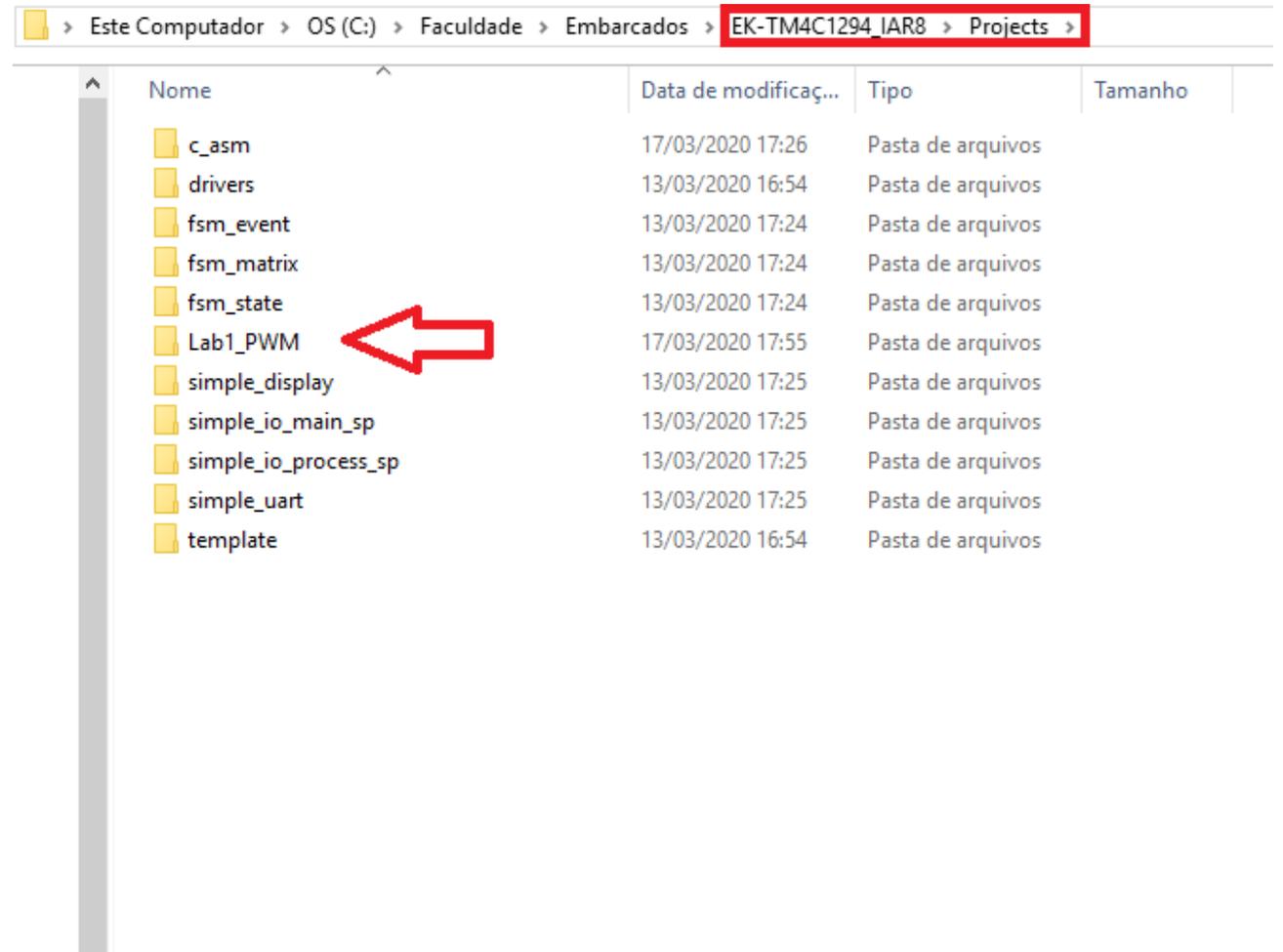
Opções do projeto

- Menu **C/C++ Compiler**
 - – Preprocessor → Additional include directories:
\$PROJ_DIR\$\\.\.\TivaWare_C_Series-2.1.4.178
- Menu **Linker**
 - List: Generate linker map file
- Menu **Debugger**
 - Setup → Driver: TI Stellaris
 - Setup → Download: Use flash loader(s)



Compilando o Projeto

- Depois de tudo isso basta compilar no IAR
 - Clicar com o botão direito em cima do projeto e apertar em “**Rebuild All**”
- Se após compilar não aparecer nenhum erro, teu projeto base está pronto.
- Agora é só dar o *upload* na plataforma GitHub e enviar para o professor



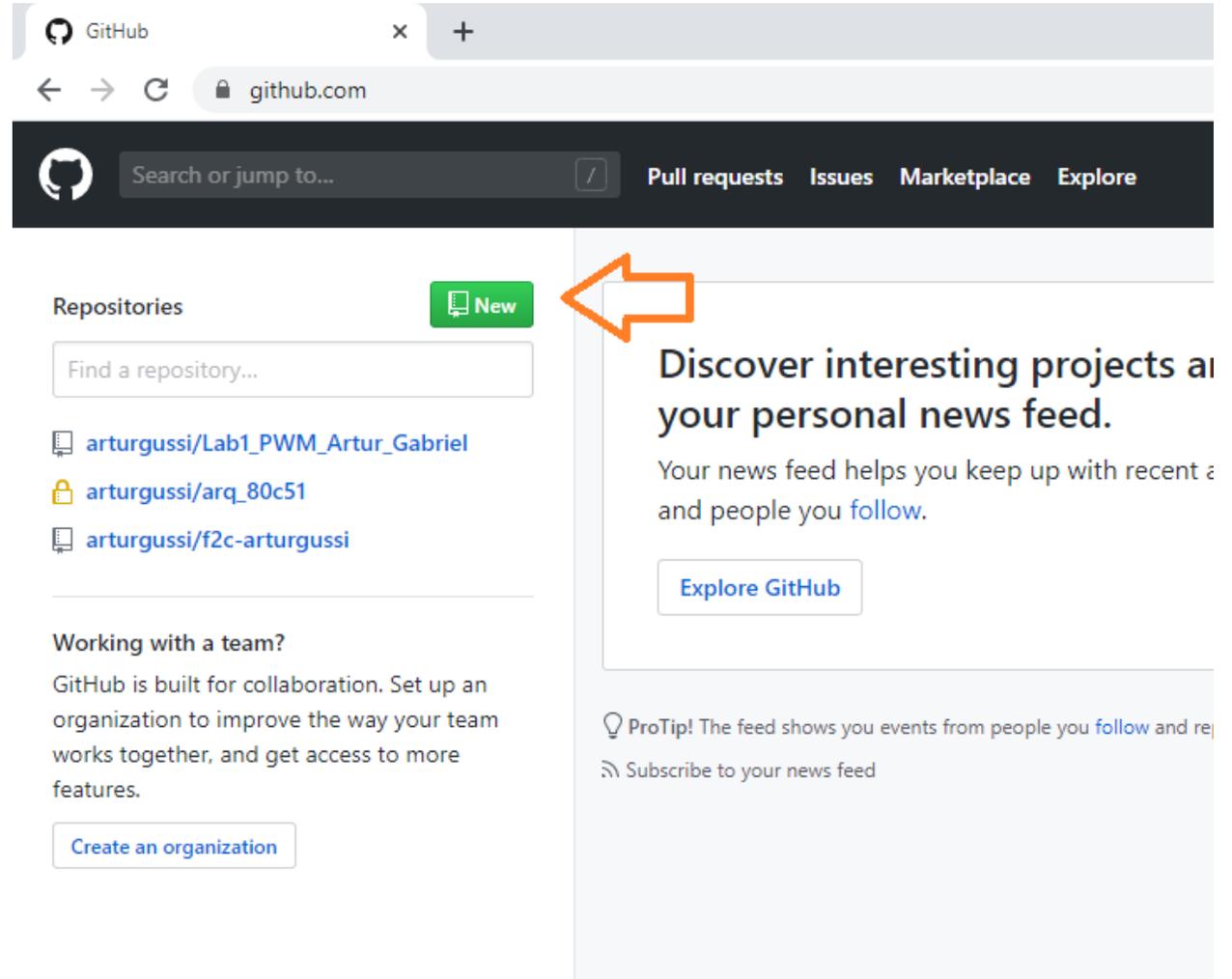
Criando repositório no GitHub



GitHub

Web

- Na página inicial do GitHub, clicar no botão “**New**” ao lado esquerdo da tela, como mostrado na imagem ao lado



Web

- No primeiro box, colocar o nome do repositório
 - Talvez colocar o nome dos participantes pode ajudar ao professor saber quais alunos estão mexendo no mesmo
- No segundo box coloque uma pequena descrição do que é aquele repositório. É opcional
- No terceiro box, deixe como público para o professor ter acesso aos arquivos
- Depois clicar em criar o repositório

Create a New Repository x +

github.com/new

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner: arturgussi

Repository name *
Lab1_PWM ✓

Great repository names are short and memorable. Need inspiration? How about [congenial-meme?](#)

Description (optional)
Primeiro laboratório desenvolvido na aula de Sistemas Embarcados

Public
Anyone can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

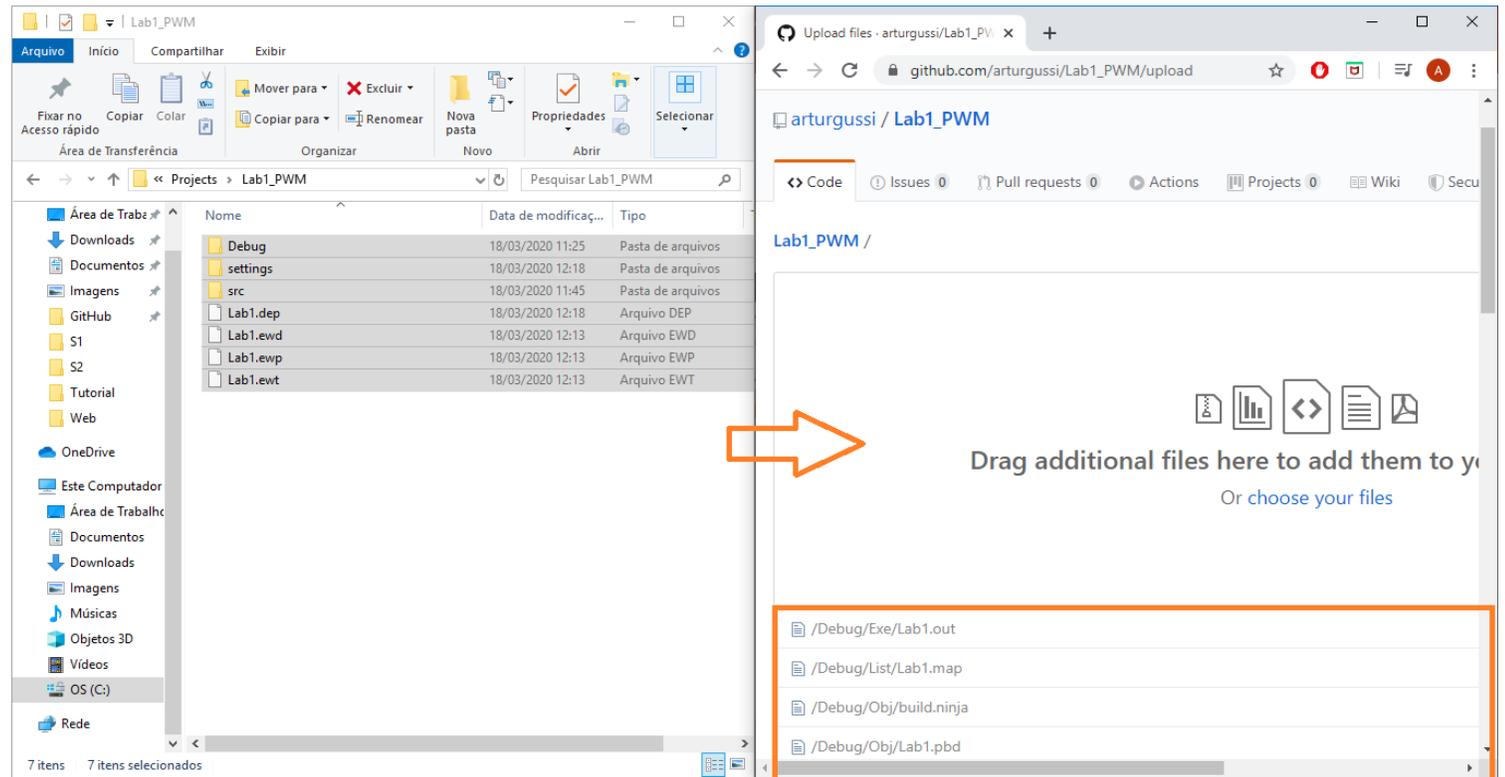
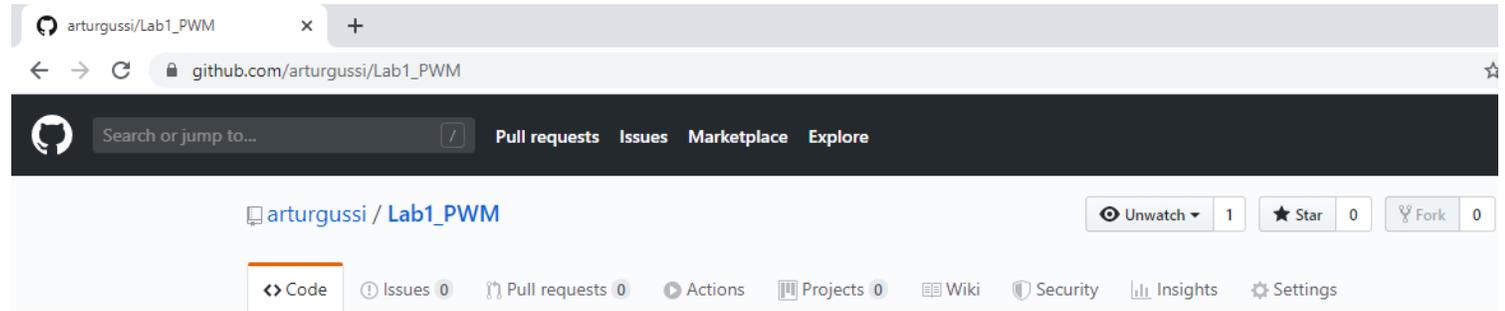
Initialize this repository with a README
This will let you immediately clone the repository to your computer.

Add .gitignore: None | Add a license: None ⓘ

Create repository ←

Web

- O link no box é o link para acessar o repositório, cujo deve ser enviado ao professor
- Clique em “**uploading na existing file**” para adicionar os primeiros arquivos ao repositório
- Agora basta adicionar os arquivos ao repositório, onde pode ser arrastando ou selecionar clicando em “**choose your files**” e espere carregar



Web

- Ao final da página terá essa imagem
- No primeiro box, coloque um comentário curto do que é esse commit que está realizando, neste caso é o nosso primeiro commit do projeto
- No segundo box pode por uma observação maior sobre mudanças no programa e informações importantes. No nosso caso é o arquivo base para começar um projeto
- Após isso é só clicar em **Commit changes** e seus arquivos já estarão no repositório para acesso de outros devs



Commit changes

Commit Inicial

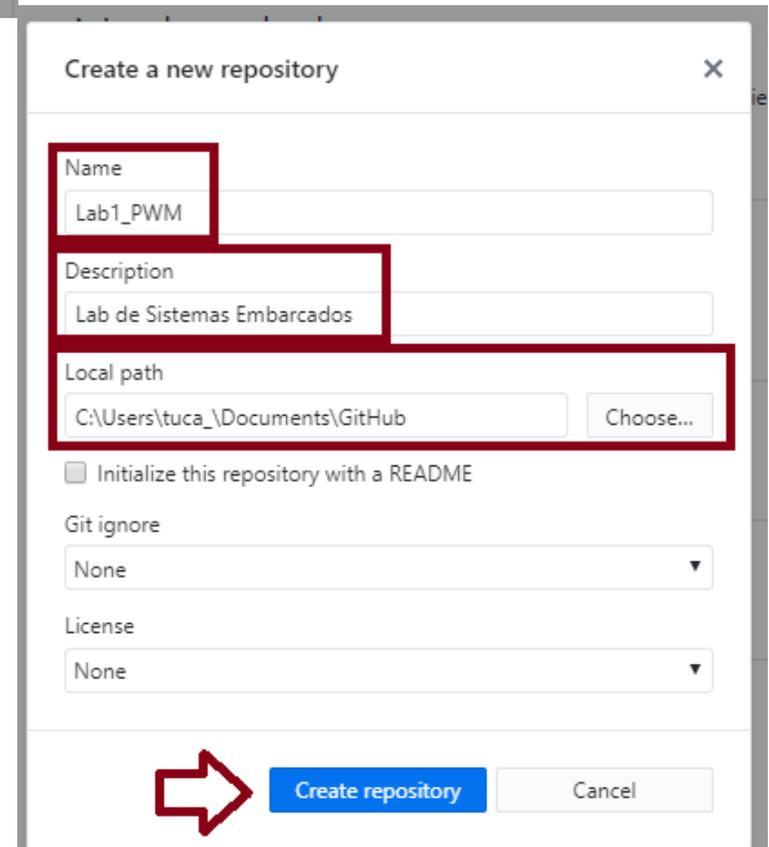
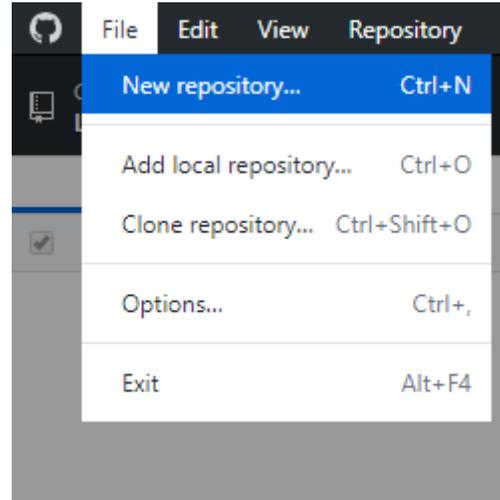
Arquivos base do projeto

Commit changes

Cancel

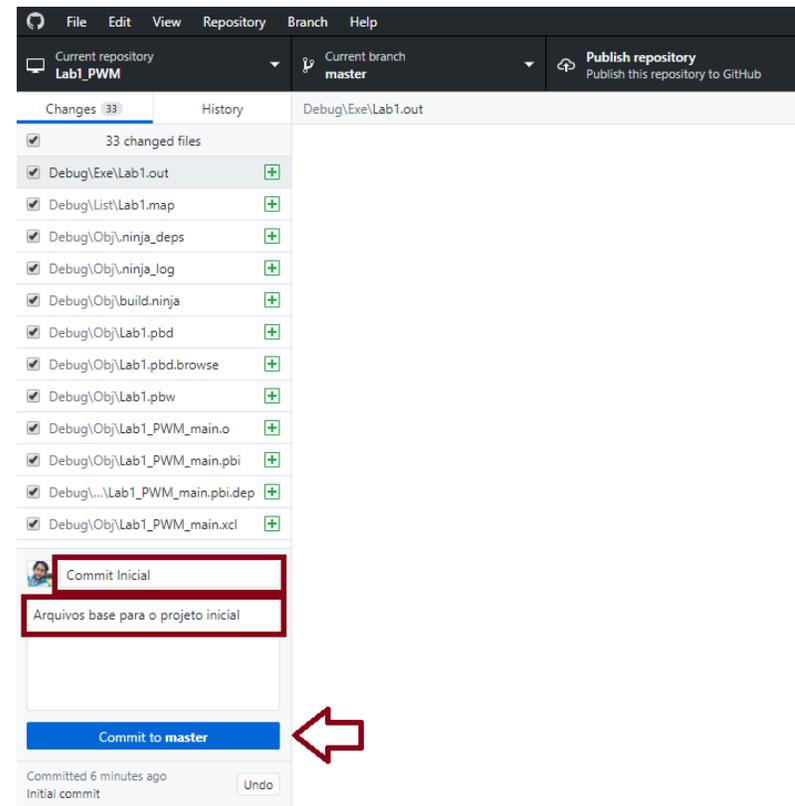
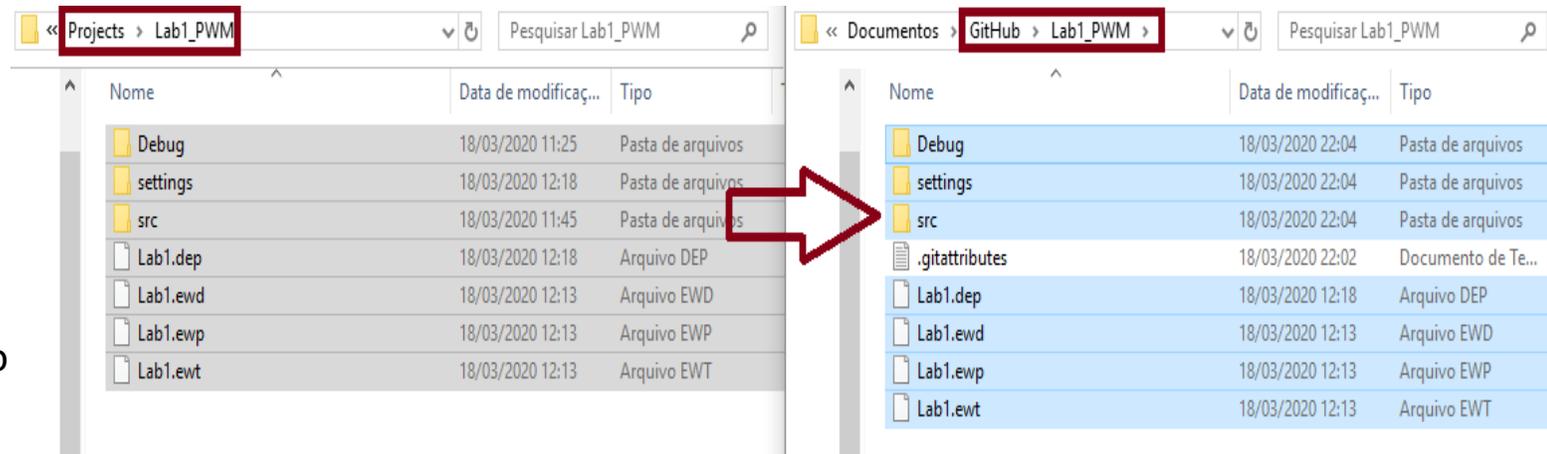
GitHub Desktop

- Na tela inicial do GitHub, clicar no botão:
 - “File” -> “New Repository”
- No box que abrirá, preencher com:
 - Nome do repositório
 - Descrição do repositório
 - Qual pasta ficará o repositório
 - Utilizo a do próprio Git para ficar só os projetos que quero no Git juntos
- Depois só clicar em “Create repository”
 - Ele criará um repositório local na sua máquina



GitHub Desktop

- No caminho em que você indicou o *path* estará uma pasta física do teu repositório
- Agora colocaremos os arquivos que queremos nesta pasta
- Feito isto, o próprio GitHub Desktop perceberá que foram alterados arquivos na pasta
- No primeiro box, coloque algo curto do commit realizado
 - Nosso exemplo é o commit inicial
- No segundo box, acrescente observações sobre o commit, as alterações que estão sendo feitas para lembrar depois
- Após clique em “**Commit to master**” e os arquivos serão commitados no branch master em sua máquina



GitHub Desktop

- Agora para publicar no servidor do GitHub (mandar para o site) basta clicar em um dos “**Publish Repository**” e está pronto
- Na janela que aparecerá, lembre-se de desmarcar a caixa “**Keep this code private**” para o professor ter acesso

