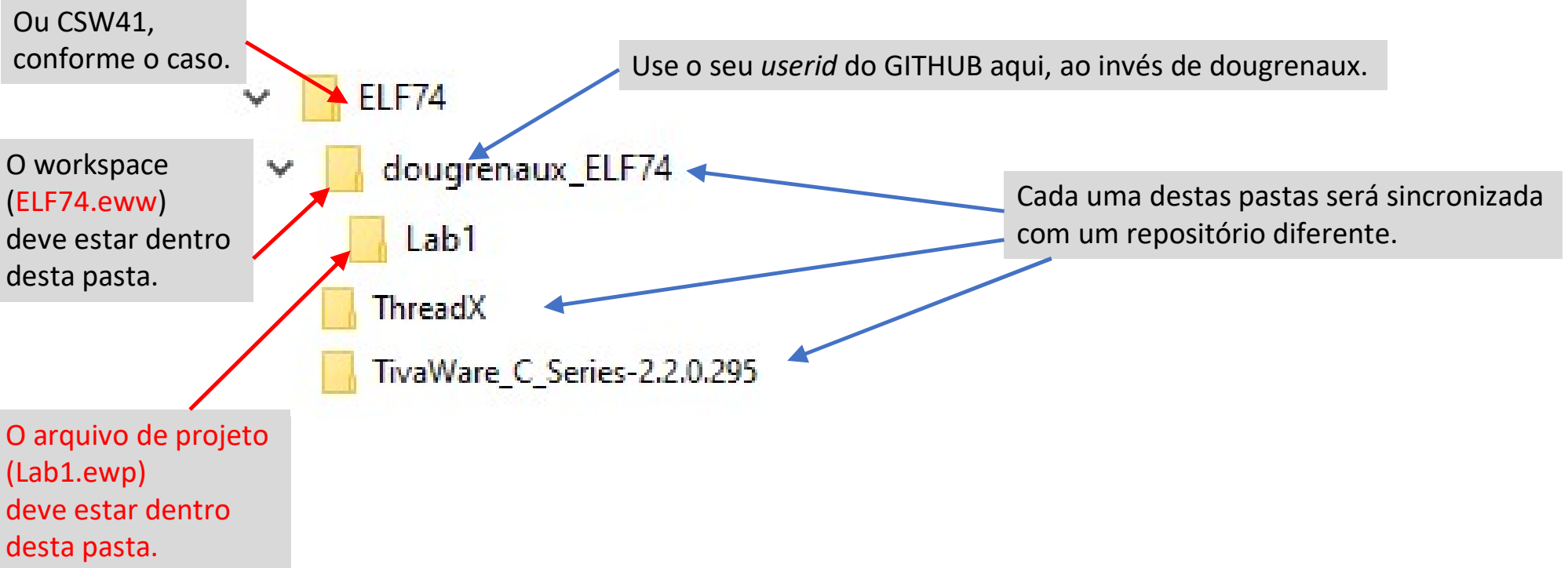


Lab 1 – passo a passo

- Objetivo:
validar a infraestrutura laboratorial necessária para os próximos experimentos: IAR e Kit da placa Tiva.
Opcionalmente: o Github e o Doxygen.
1. Incentivo vocês a usarem o Github para gerenciar versões do código e compartilharem com seu colega de equipe. Da mesma forma, a usar o Doxygen de forma a documentar o código e o resultado de cada lab no próprio código fonte. Contudo, o uso destas duas ferramentas é opcional. Caso deseje, use uma, outra ou ambas. Ver dicas de uso nos slides a seguir.
 2. Instalar o IDE da IAR: IAR Embedded Workbench for ARM versão 9.20. Há um passo-a-passo da instalação. Lembrar de ativar a licença do tipo “Code size limited”.

Lab 1 – passo a passo

3. Em uma pasta vazia no seu computador, planeje a estrutura futura de pastas para os Labs desta disciplina.



Lab 1 – passo a passo

4. Usando o IAR, criar um projeto novo na pasta Lab1. Se desejar, usar o template de projeto disponível no GitHub no usuário *dougrenaux*.
Usar a linguagem C++. Usar o `<iostream>` e não o `<stdio>`, portanto, usar *cout* e *cin*.

O propósito do código é de enviar e receber dados do terminal:

- Envie uma mensagem inicial.
- Apresente o valor de algumas símbolos pré-definidos (*predefined preprocessor symbols*), a exemplo de: `__cplusplus` `DATE` `TIME` `FILE` `LINE` `__STDC__`
`__STDC_VERSION__` `ARM_ARCH` `ARM_ARCH_ISA_THUMB`
`__ARM_SIZEOF_MINIMAL_ENUM` `__ARM_SIZEOF_WCHAR_T` `__ARMVFP__` `__CORE__`
- Ler (via *cin*) um valor em ponto flutuante (float)
- Somar a um outro valor em ponto flutuante (float)
- Apresentar o resultado da soma

Lab 1 – passo a passo

5. Nas opções de configuração de projeto:
Cortex-M4 com coprocessador numérico
Depurador usado: TI Stellaris.
O código deve ser executado na placa Tiva.
Obs: revise as opções de configuração do projeto verificando se estão adequadas.
6. Ao execute o programa,
 - onde aparecem as mensagem enviadas via cout ?
 - como entrar dados para leitura via cin ?
 - quais os valores dos símbolos pré-definidos que você escolheu ? o que significam ?
 - que tipo de instrução assembly foi usada para a soma dos dois valores float ?
qual configuração do projeto no IAR causou o uso desta instrução ?

Lab 1 – passo a passo

7. Entrega:
conforme prazo estabelecido no Classroom.
Entregar via Classroom; seja um link para seu repositório privado no github, seja um zip (limpar código objeto antes de zipar) da pasta seunome_ELF74. Repositórios privados devem autorizar o acesso pelo usuário dougrênaux para que eu possa acessar seus resultados.
Quando houver respostas textuais, a exemplo deste Lab, incluir a resposta no Readme.md, no github, na seção correspondente ao Lab1 ou na pasta Lab1 caso esteja enviando um zip.
Tanto para quem entrega via GitHub ou .zip, a documentação da entrega pode ser feita via Doxygen, numa pasta dentro da pasta Lab1. Favor informar este formato de entrega no Readme.md.

Obs: peço a gentileza de seguir cuidadosamente estes procedimentos; fica extremamente trabalhoso se eu tiver que formatar entregas feitas de forma distinta; então, terei que devolvê-las aos autores para correção.

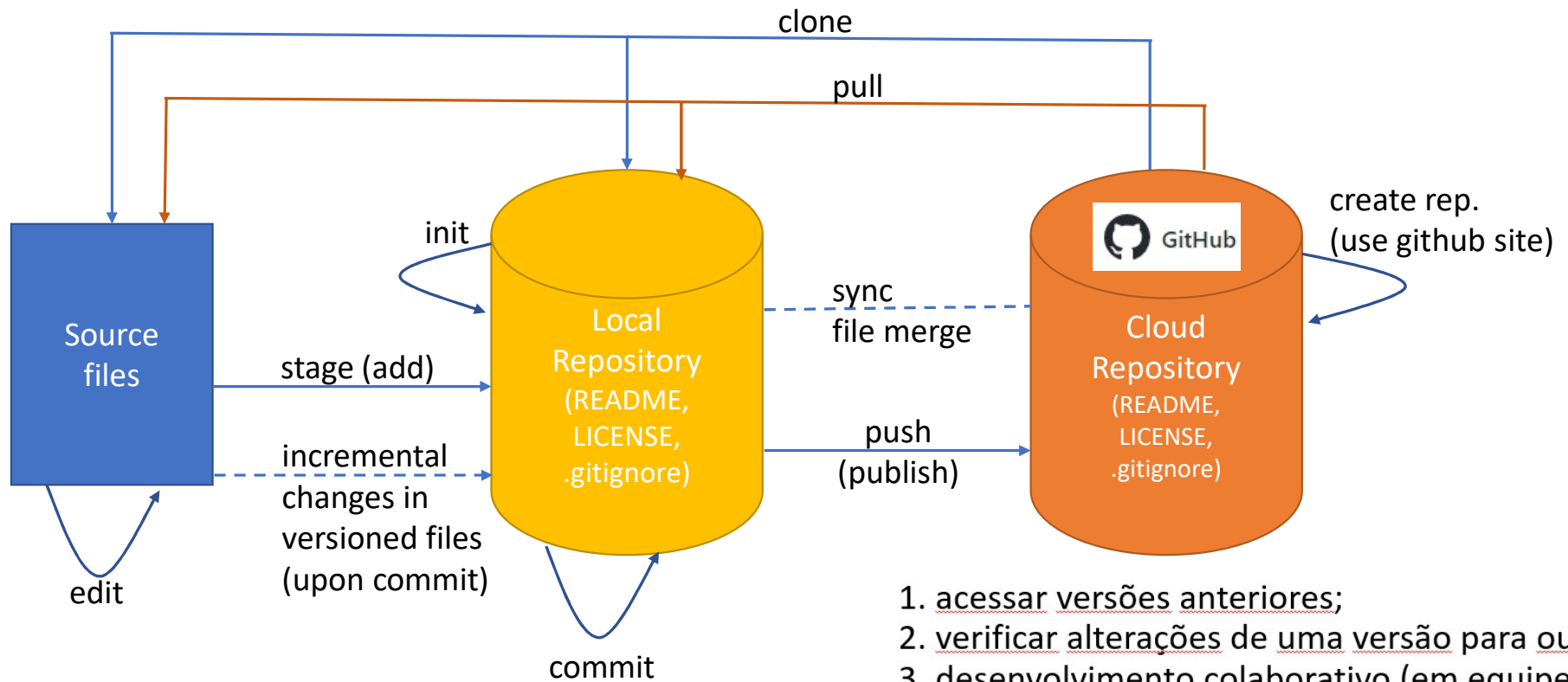
Git + Github

- O que é o Git ?
uma ferramenta para gerenciar versões de código permitindo:
 1. acessar versões anteriores;
 2. verificar alterações de uma versão para outra;
 3. desenvolvimento colaborativo (em equipe);
 4. compartilhamento de código.
- O que é o GitHub ?
um repositório de uso aberto. Acesso gratuito e/ou pago.

Instalação do Git

- Baixar a versão adequada a partir de:
<https://git-scm.com/downloads>
- Se desejar usar a interface gráfica de operação do Git/GitHub, baixar também o GitHubDesktop.
Obs: como eu preciso sincronizar com muitos repositórios de vocês estudantes, para mim, o GitHub agiliza o processo. Portanto, eu uso a interface gráfica e não a interface de linha de comando.
Escolham a que acharem mais adequada.
<https://desktop.github.com/>

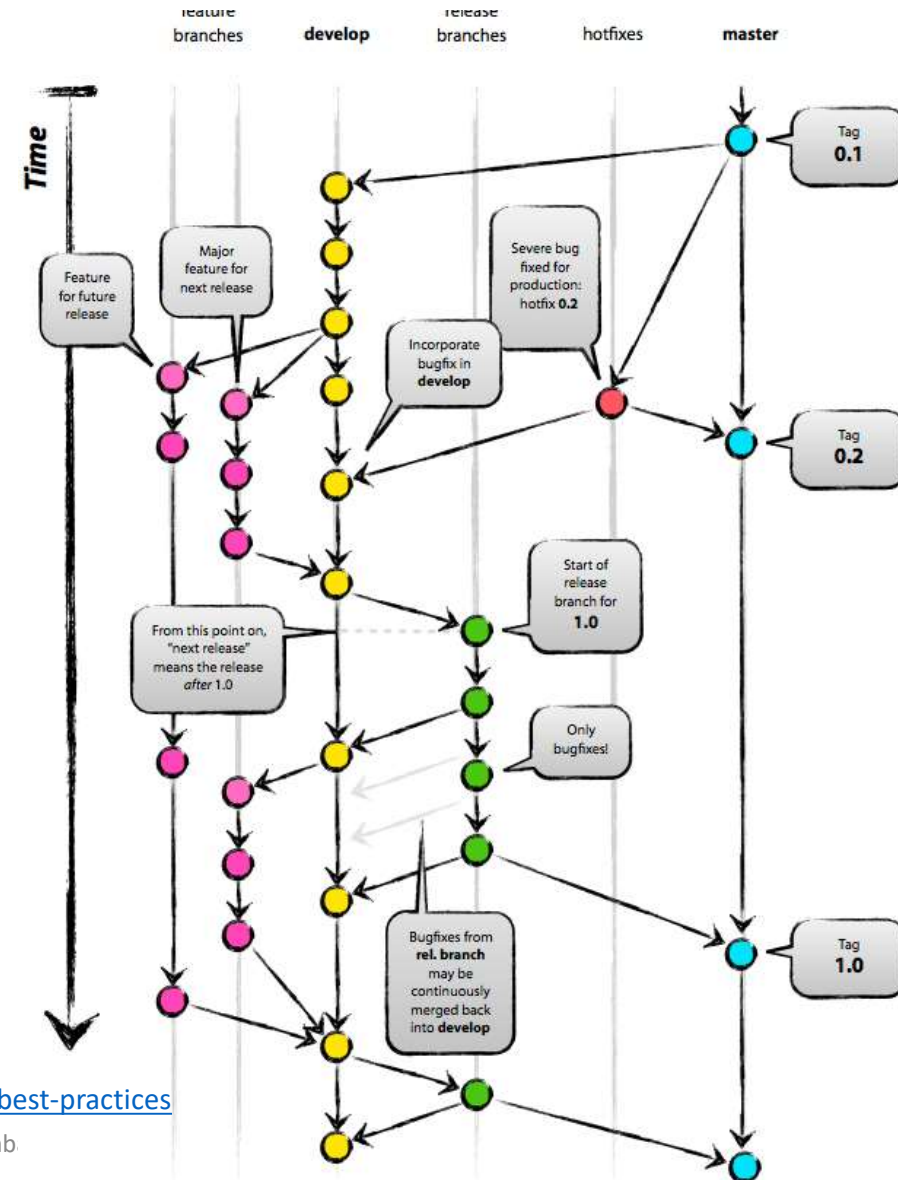
Controle de versão no Git + GitHub



1. acessar versões anteriores;
2. verificar alterações de uma versão para outra;
3. desenvolvimento colaborativo (em equipe);
4. compartilhamento de código.

Controle de versão no Git + Github

- Conceitos avançados
 - branch (e branch merge)
 - tags
 - ...



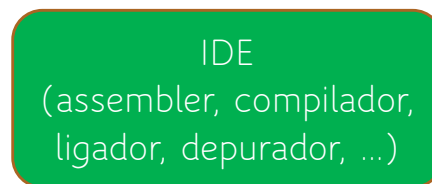
<https://softwareengineering.stackexchange.com/questions/165725/git-branching-and-tagging-best-practices>

Repositório - o que armazenar ?

Devem estar no repositório

.c
.h
.s
.txt
.eww
.ewp
.icf
Makefile
makedefs
...

arquivos de “entrada”
editados pelo programador
e/ou pela configuração da IDE



NÃO Devem estar no repositório
nem no .zip

folder settings
folder Debug
folder ewarm
.o
.a
.bin
.map
...

arquivos de “saída”
resultados de ações de ferramentas da IDE

Repositório Template

Independente de você optar por usar o Git+Github ou não, um possível ponto de partida para criar o projeto do Lab1 é usar o template que está no GitHub em:

https://github.com/dougrenaux/dr_Tiva_Template

Se você usa o Github, pode criar um repositório novo utilizando o repositório acima como template.

Se você não usa o Github, pode baixar este template como um arquivo .zip e descompactá-lo no seu computador.

Certifique-se que está acessando a versão mais recente do template.

Doxygen

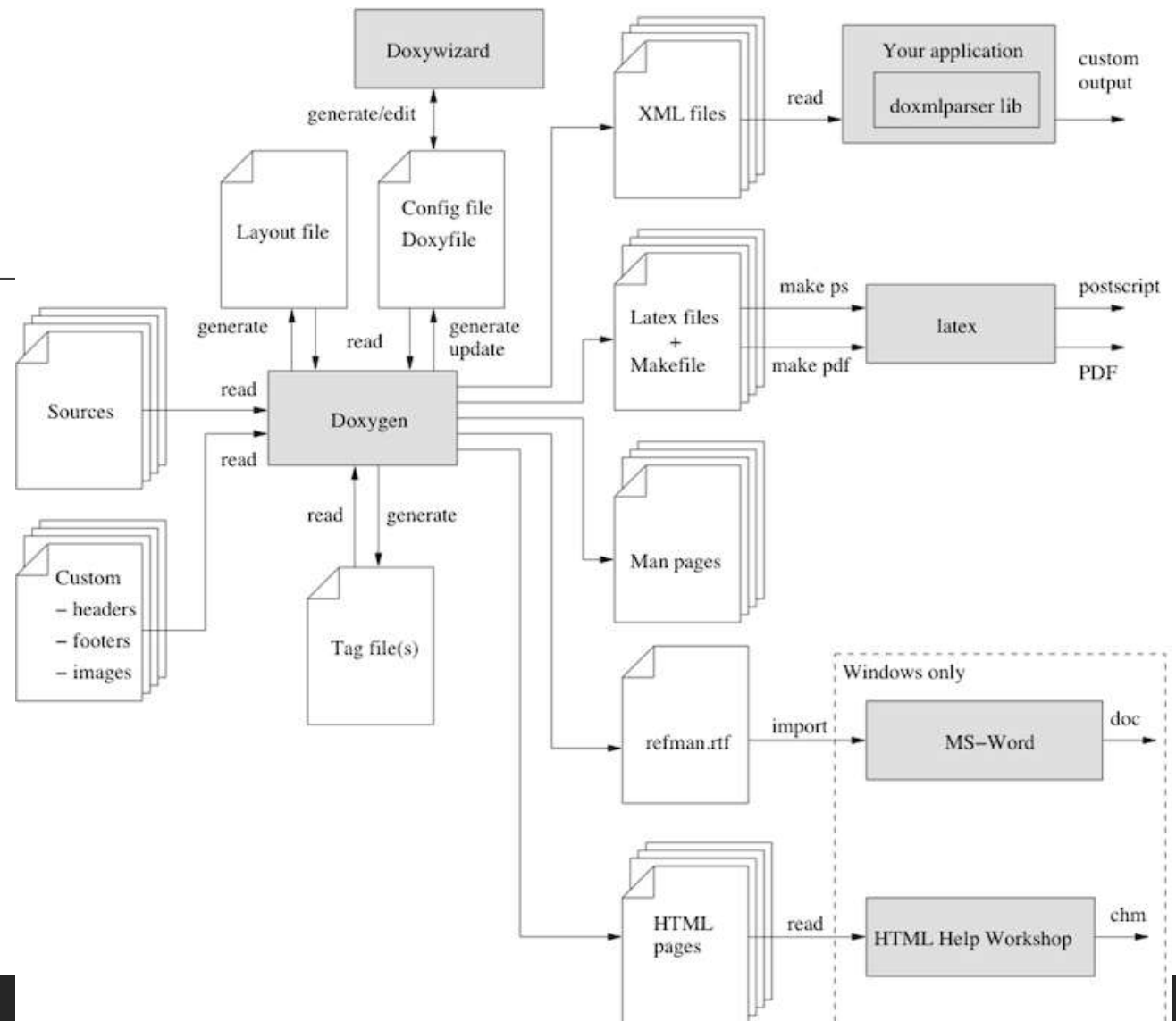
O que é o Doxygen ?

Uma ferramenta gratuita para documentação de código. Ela é suficientemente elaborada a ponto de permitir a geração automática de manuais de usuário. Por exemplo, o TivaWare, que será usado a partir do Lab2, tem seu manual gerado pelo Doxygen.

Instalação

Baixe o instalador a partir de doxygen.nl

Doxygen



Doxygen GUI frontend (C:/dr/_Embarcados_Tiva/ELF74/dr_Tiva_Template/Lab1/Doxyfile)

FileSettingsHelp

Specify the working directory from which doxygen will run

C:\dr_Embarcados_Tiva\ELF74\dr_Tiva_Template\Lab1

Configure doxygen using the Wizard and/or Expert tab, then switch to the Run tab to generate the documents

WizardExpertRun

Specify additional command line options for running doxygen

Run doxygenStatus: not running

Show HTML output

Output produced by doxygen

Generating directory doc
Generating index page...
Generating page index...
Generating module index...
Generating namespace ind
Generating namespace mem
Generating annotated com
Generating alphabetical
Generating hierarchical
Generating member index...
Generating file index...
Generating file member i
Generating example index
finalizing index lists...
writing tag file...
Running plantuml with JA
lookup cache used 2/6553
finished...

*** Doxygen has finished

Name

doxy
Doxyfile
Lab1.ewd
Lab1.ewp
Lab1.ewt
main.cpp
template.h

Lab1

Main PageFiles

Search

File List

Here is a list of all files with brief descriptions:

main.cpp

Solution to Lab1 of ELF74/CSW41 - UTFPR.
Tools instalation and validation procedure.
Show messages on terminal using std::cout.
Show current value of some predefined macros (preprocessor symbols).
Read float value from terminal using std::cin

template.h

This header file..

Generated by doxygen 1.9.1