

UNIVERSIDADE ESTADUAL DO NORTE FLUMINENSE  
LABORATÓRIO DE ENGENHARIA E EXPLORAÇÃO DE PETRÓLEO

PROJETO ENGENHARIA  
DESENVOLVIMENTO DO SOFTWARE  
MODELOS DE DESLOCAMENTO IMISCÍVEL PARA RECUPERAÇÃO  
SECUNDÁRIA DE PETRÓLEO  
TRABALHO DA DISCIPLINA PROGRAMAÇÃO PRÁTICA

DAVID HENRIQUE LIMA DIAS  
JULIA RANGEL RIBEIRO  
MARCOS VINÍCIUS DE PAULA CHAIBEN  
- Versão 1:  
Prof. André Duarte Bueno

MACAÉ - RJ  
DEZEMBRO - 2021

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Escopo do problema . . . . .	1
1.2	Objetivos . . . . .	1
<b>2</b>	<b>Implementação</b>	<b>3</b>
2.1	Código fonte . . . . .	3
<b>3</b>	<b>Teste</b>	<b>7</b>
3.1	Teste 1: Descrição . . . . .	7
3.2	Teste 2: Descrição . . . . .	8
<b>4</b>	<b>Documentação</b>	<b>10</b>
4.1	Documentação do usuário . . . . .	10
4.1.1	Como instalar o software . . . . .	10
4.1.2	Como rodar o software . . . . .	10
4.2	Documentação para desenvolvedor . . . . .	10
4.2.1	Dependências . . . . .	11
4.2.2	Como gerar a documentação usando doxygen . . . . .	11
<b>5</b>	<b>Título do Apêndice</b>	<b>15</b>
5.1	Sub-Título do Apêndice . . . . .	15
<b>6</b>	<b>Título do Apêndice.. Usando Citações</b>	<b>16</b>
6.1	Roteiro Para Uso do Sistema de Citações Com Banco de Dados .bib . . . .	17
6.1.1	Citações no meio do texto . . . . .	18
6.1.2	Incluir nas referências bibliográficas (fim do documento), mas não citar . . . . .	18
6.2	Informações adicionais . . . . .	18
<b>7</b>	<b>Como criar seu projeto no github a partir do modelo do professor</b>	<b>19</b>
7.1	Roteiro para criar uma conta no github . . . . .	19
7.2	Roteiro para criar um repositório novo (vazio) no github . . . . .	19
7.3	Roteiro para clonar o repositório do projeto modelo em seu computador . .	21

7.4	Roteiro para clonar o repositório do projeto modelo diretamente (+simples)	22
-----	--	----

# Capítulo 1

## Introdução

Impulsionado pela importância que o petróleo tem sobre toda a humanidade, sendo ainda hoje uma das maiores fonte de energia em uso pelo ser humano, o presente trabalho de engenharia, desenvolve-se um projeto computacional em linguagem orientada a objeto C++ que tem como principal objetivo o gerenciamento de informações e realização de cálculos para estudo da recuperação de óleo resultante do deslocamento por um fluido imiscível.

### 1.1 Escopo do problema

No início de sua descoberta, os reservatórios de óleo e gás possuem uma certa quantidade de energia denominada energia primária. Com o avanço da vida produtiva, ocorre uma dissipação dessa energia primária resultando em um esgotamento da energia natural e uma queda no diferencial de pressão entre os limites do reservatório e os poços produtores. Com isso, o reservatório estaria destinado a uma baixa taxa de produção (ROSA ET AL.,2006).

Para contornar tal problema são usadas operações de manutenção de pressão, como a recuperação secundária. Este método consiste na recuperação por injeção de fluidos, como água e/ou gás, principalmente para fins de manutenção de pressão e eficiência de varredura volumétrica (SHENG, 2011). A eficiência deste método pode ser superior a 60%, embora o valor mais frequente seja de 30 a 50%, para os métodos convencionais (ROSA ET AL.,2006).

### 1.2 Objetivos

Os objetivos deste projeto de engenharia são:

- Objetivo geral:

- Desenvolver um software na área da engenharia de petróleo, mais especificamente, engenharia de reservatório;
  - Propor a solução para aplicação do método de recuperação secundária, a partir de água como fluido injetado;
  - Realizar análise para previsão de escoamento bifásico imiscível num reservatório;
- Objetivos específicos:
    - Solucionar o problema de permeabilidade relativa a partir do Modelo de Corey-Brooks;
    - Desenvolver a curva de fluxo fracionário para um reservatório bifásico;
    - Calcular a área invadida pela injeção;
    - Calcular e analisar o comportamento das pressões;
    - Aplicar o Modelo de Dykstra-Parsons (1950) e Stiles (1949) para:
      - \* Calcular da frente de avanço da camada em cada breakthrough (BT);
      - \* Calcular da eficiência vertical em cada BT;
      - \* Calcular do volume de óleo recuperado total;
      - \* Calcular do tempo necessário do BT na última camada (todo óleo recuperável possível por esse método de injeção);

# Capítulo 2

## Implementação

Neste capítulo do projeto de engenharia apresentamos os códigos fonte que foram desenvolvidos.

### 2.1 Código fonte

Apresenta-se a seguir um conjunto de classes (arquivos .h e .cpp) além do programa main dos softwares.

Apresenta-se na listagem ?? o arquivo com código da classe CFormaReservatorio.

Listing 2.1: Arquivo de cabeçalho da classe CFormaReservatorio.

---

```
1 #ifndef CCOREY_H_
2 #define CCOREY_H_
3
4 #include <vector>
5
6 class CCorey {
7
8     protected:
9
10         std::vector<double> sw, kro, krw;
11         double KOrw, KOrw, No, Nw;
12
13     public:
14
15         CCorey(){};
16
17         void calcSwn(double _swi, double _sor);
18         void calcKro(std::vector<double> _sw);
19         void calcKrw(std::vector<double> _sw);
```

---

```

20         void setK0rw(double _K0rw);
21         void setK0ro(double _K0ro);
22         void setNo(double _No);
23         void setNw(double _Nw);
24         std::vector<double> getSw();
25         std::vector<double> getKro();
26         std::vector<double> getKrw();
27
28         ~CCorey(){};
29
30 };
31
32 #endif

```

---

Apresenta-se na listagem ?? o arquivo de implementação da classe CFormaReservatorio.

Listing 2.2: Arquivo de implementação da classe CFormaReservatorio.

---

```

1 #include "CCorey.h"
2 #include <cmath>
3 #include <iostream>
4
5 void CCorey::calcSwn(double _swi, double _sor){
6     double a;
7     for (double i = _swi; i <=(1.0 - _sor)+.01; i+= .01){
8         a = (i - _swi)/(1.0 - _swi - _sor);
9         sw.push_back(a);
10    }
11
12 }
13
14 void CCorey::calcKro(std::vector<double> _sw){
15
16     for(double sw:_sw)
17         kro.push_back(K0ro*pow(1 - sw, No));
18
19 }
20
21 void CCorey::calcKrw(std::vector<double> _sw){
22
23     for(double sw:_sw)
24         krw.push_back(K0rw*pow(sw, Nw));
25

```

```
26 }
27
28 void CCorey::setK0rw(double _K0rw){
29
30     K0rw = _K0rw;
31
32 }
33
34 void CCorey::setK0ro(double _K0ro){
35
36     K0ro = _K0ro;
37
38 }
39
40 void CCorey::setNo(double _No){
41
42     No = _No;
43
44 }
45
46 void CCorey::setNw(double _Nw){
47
48     Nw = _Nw;
49
50 }
51
52 std::vector<double> CCorey::getSw(){
53
54     return sw;
55
56 }
57
58 std::vector<double> CCorey::getKro(){
59
60     return kro;
61
62 }
63
64 std::vector<double> CCorey::getKrw(){
65
66     return krw;
67
```



68 }

---

# Capítulo 3

## Teste

Todo projeto de engenharia passa por uma etapa de testes. Neste capítulo apresentamos alguns testes do software desenvolvido. Estes testes devem dar resposta aos diagramas de caso de uso inicialmente apresentados (diagramas de caso de uso geral e específicos).

### 3.1 Teste 1: Descrição

No início apresente texto explicativo do teste:

- O que esta sendo testado?
- Como o teste vai ser realizado?
- Como o programa será validado?

A seguir apresente texto explicando a sequência do teste e imagens do programa (captura de tela).

coloque aqui texto falando do diagrama de pacotes, referencie a figura. Veja Figura 3.1.

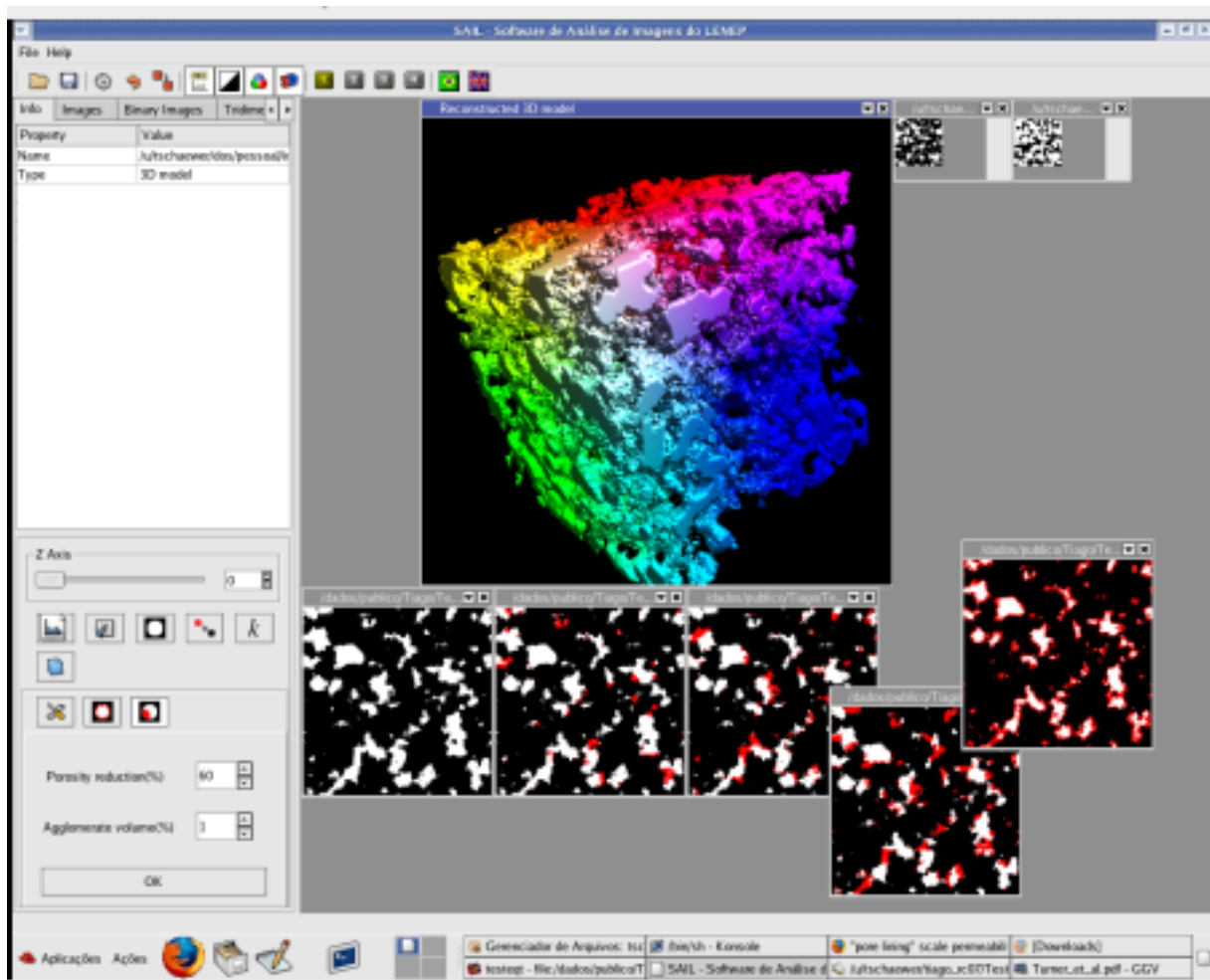


Figura 3.1: Tela do programa mostrando xxx

## 3.2 Teste 2: Descrição

No início apresente texto explicativo do teste:

- O que está sendo testado?
- Como o teste vai ser realizado?
- Como o programa será validado?

A seguir apresente texto explicando a sequência do teste e imagens do programa (captura de tela).

Coloque aqui texto falando do diagrama de pacotes, referencie a figura. Veja Figura 3.2.

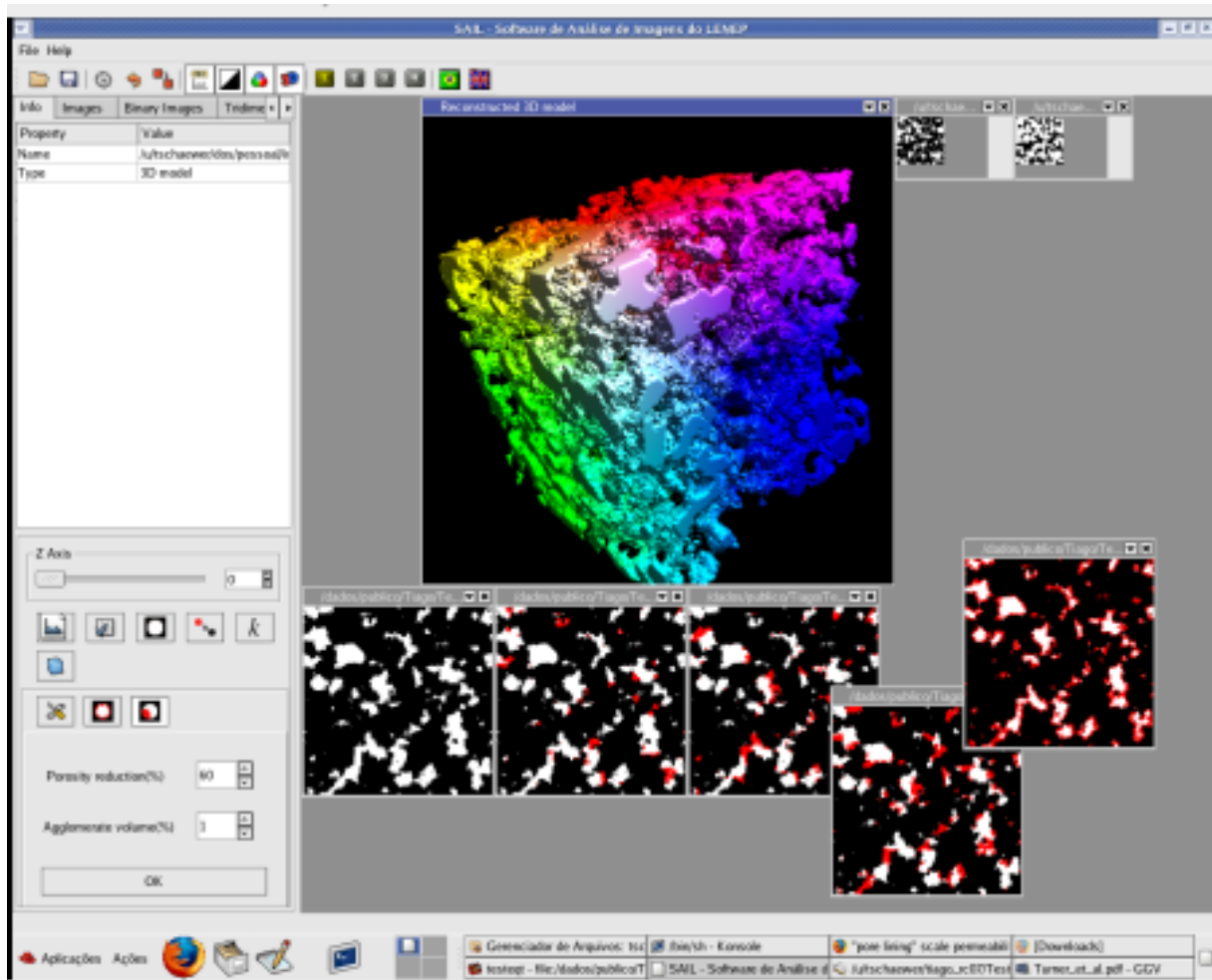


Figura 3.2: Tela do programa mostrando xxx

### Nota:

Não perca de vista a visão do todo; do projeto de engenharia como um todo. Cada capítulo, cada seção, cada parágrafo deve se encaixar. Este é um diferencial fundamental do engenheiro em relação ao técnico, a capacidade de desenvolver projetos, de ver o todo e suas diferentes partes, de modelar processos/sistemas/produtos de engenharia.

# Capítulo 4

## Documentação

Todo projeto de engenharia precisa ser bem documentado. Neste sentido, apresenta-se neste capítulo a documentação de uso do "software XXXX". Esta documentação tem o formato de uma apostila que explica passo a passo como usar o software.

### 4.1 Documentação do usuário

Descreve-se aqui o manual do usuário, um guia que explica, passo a passo a forma de instalação e uso do software desenvolvido.

#### 4.1.1 Como instalar o software

Para instalar o software execute o seguinte passo a passo:

- blablabla
- ..
- .

#### 4.1.2 Como rodar o software

Para rodar o software ....blablabla

Veja no Capítulo 3 - Teste, exemplos de uso do software.

### 4.2 Documentação para desenvolvedor

Apresenta-se nesta seção a documentação para o desenvolvedor, isto é, informações para usuários que queiram modificar, aperfeiçoar ou ampliar este software.

### 4.2.1 Dependências

Para compilar o software é necessário atender as seguintes dependências:

- Instalar o compilador `g++` da GNU disponível em <http://gcc.gnu.org>. Para instalar no GNU/Linux use o comando `yum install gcc`.
- Biblioteca `CGnuplot`; os arquivos para acesso a biblioteca `CGnuplot` devem estar no diretório com os códigos do software;
- O software `gnuplot`, disponível no endereço <http://www.gnuplot.info/>, deve estar instalado. É possível que haja necessidade de setar o caminho para execução do `gnuplot`.
- .
- .

### 4.2.2 Como gerar a documentação usando doxygen

A documentação do código do software deve ser feita usando o padrão JAVADOC, conforme apresentada no Capítulo - Documentação, do livro texto da disciplina. Depois de documentar o código, use o software `doxygen` para gerar a documentação do desenvolvedor no formato html. O software `doxygen` lê os arquivos com os códigos (\*.h e \*.cpp) e gera uma documentação muito útil e de fácil navegação no formato html.

- Veja informações sobre uso do formato JAVADOC em:
  - <http://www.stack.nl/~dimitri/doxygen/manual/docblocks.html>
- Veja informações sobre o software `doxygen` em
  - <http://www.stack.nl/~dimitri/doxygen/>

Passos para gerar a documentação usando o `doxygen`.

- Documente o código usando o formato JAVADOC. Um bom exemplo de código documentado é apresentado nos arquivos da biblioteca `CGnuplot`, abra os arquivos `CGnuplot.h` e `CGnuplot.cpp` no editor de texto e veja como o código foi documentado.
- Abra um terminal.
- Vá para o diretório onde esta o código.

```
cd /caminho/para/seu/codigo
```

- Peça para o **doxygen** gerar o arquivo de definições (arquivo que diz para o doxygen como deve ser a documentação).

```
doxygen -g
```

- Peça para o **doxygen** gerar a documentação.

```
doxygen
```

- Verifique a documentação gerada abrindo o arquivo `html/index.html`.

```
firefox html/index.html
```

ou

```
chrome html/index.html
```

Apresenta-se a seguir algumas imagens com as telas das saídas geradas pelo software **doxygen**.

**Nota:**

Não perca de vista a visão do todo; do projeto de engenharia como um todo. Cada capítulo, cada seção, cada parágrafo deve se encaixar. Este é um diferencial fundamental do engenheiro em relação ao técnico, a capacidade de desenvolver projetos, de ver o todo e suas diferentes partes, de modelar processos/sistemas/produtos de engenharia.





# Referências Bibliográficas

- [e Patrick W. Daly, 1995] e Patrick W. Daly, H. K. (1995). *A Guide to Latex 2e*. Addison-Wesley, New York, 2 edition. 17
- [Grossens et al., 1993] Grossens, M., Mittelbach, F., and Samarin, A. (1993). *Latex Companion*. Addison-Wesley, New York. 17
- [Karger, 2004] Karger, A. (2004). *O Tutorial de Lyx*. LyX Team - <http://www.lyx.org>. 17
- [Knuth, 1986] Knuth, D. E. (1986). *The TeXbook*. Addison-Wesley. 17
- [Lamport, 1985] Lamport, L. (1985). *Latex - A Document Preparation System*. Addison-Wesley. 17
- [LyX-Team, 2004a] LyX-Team, editor (2004a). *Extended LyX Features*. LyX Team - <http://www.lyx.org>. 17
- [LyX-Team, 2004b] LyX-Team, editor (2004b). *The LyX User's Guide*. LyX Team - <http://www.lyx.org>. 17
- [Steding-Jessen, 2000] Steding-Jessen, K. (2000). *Latex demo: Exemplo com Latex 2e*. 17

# Capítulo 5

## Título do Apêndice

Descreve-se neste apêndice ...

- Os anexos ou apêndices contém material auxiliar. Por exemplo, tabelas, gráficos, resultados de experimentos, algoritmos, códigos e simulações.
- Um apêndice pode incluir assuntos mais gerais (geral demais para estar no núcleo do trabalho) ou mais específicos (detalhado demais para estar no núcleo do trabalho).
- Pode conter um artigo de auxílio fundamental ao trabalho.
- Pode conter artigos publicados.
- [tudo aquilo que for importante para a tese mas não essencial, deve ser colocado em apêndices]
- [como exemplo, revisão de metodologias, técnicas, modelos matemáticos, itens desenvolvidos por terceiros]
- [algoritmos e programas devem ser colocados no apêndice]
- [imagens detalhadas de programas desenvolvidos devem ser colocados no apêndice]

### 5.1 Sub-Título do Apêndice

.....conteúdo..

## Capítulo 6

### Título do Apêndice.. Usando Citações

Descreve-se neste apêndice ...

[tudo aquilo que for importante para a tese mas não essencial, deve ser colocado em apêndices]

[como exemplo, revisão de metodologias, técnicas, modelos matemáticos, itens desenvolvidos por terceiros]

[algoritmos e programas devem ser colocados no apêndice]

[imagens de programas desenvolvidos/utilizados devem ser colocados no apêndice]

## 6.1 Roteiro Para Uso do Sistema de Citações Com Banco de Dados .bib

O sistema de referências usando bibtex é extremamente simples e muito prático. O mesmo é composto de uma base de dados (um arquivo .bib que contém a lista de referências a ser utilizada). Por exemplo, o arquivo andre.bib, inclui referências bibliográficas no formato bib (de uma olhada agora no arquivo andre.bib usando um editor de texto como o emacs). A seguir, você deve incluir no arquivo do lyx, o nome de sua base de dados. Finalmente, você precisa incluir as referências cruzadas.

Veja a seguir um roteiro:

1. Você deve fazer uma cópia do arquivo andre.bib com seu nome, e a seguir usar um editor qualquer (mas preferencialmente o emacs) para incluir suas referências bibliográficas. Ou seja, inclua no arquivo seuNome.bib todas as citações e referências bibliográficas a serem incluídas em sua tese (tudo que você leu, e que pode ser incluído na citação da tese e de outros artigos. É sua base de dados de citações).
  - (a) Você pode incluir itens no arquivo .bib que não irão fazer parte da tese, mas poderão ser citadas em artigos futuros.
2. Para fazer uma citação é necessário incluir no arquivo do lyx um "Insert-> Lists & Toc->Bibtex reference". Vai aparecer um diálogo pedindo para você incluir o nome do arquivo com a base de dados de citações (digite seuNome.bib).
3. Finalmente, faça referências cruzadas usando o item de menu "Insert Cross-Reference".
4. Aqui um exemplo, vou citar material sobre LyX e Latex. Veja maiores informações sobre latex em [Grossens et al., 1993, Knuth, 1986, Steding-Jessen, 2000, e Patrick W. Daly, 1995, Lamport, 1985, LyX-Team, 2004a, Karger, 2004, LyX-Team, 2004b].

### 6.1.1 Citações no meio do texto

### 6.1.2 Incluir nas referências bibliográficas (fim do documento), mas não citar

asldkjasldkajsdlkajsdlaksjd  
asldkjasldkajsdlkajsdlaksjd

## 6.2 Informações adicionais

- Manuais do LyX (precisa ler!)
  - <http://chem-e.org/comando-cite-e-citeonline-no-abntex/>
  - <http://win.ua.ac.be/~nschloe/content/bibtex-how-cite-website>.
  - <http://chem-e.org/comando-apud-e-apudonline-no-abntex/>.
  - [http://en.wikibooks.org/wiki/LaTeX/Bibliography\\_Management](http://en.wikibooks.org/wiki/LaTeX/Bibliography_Management)
- ...

## Capítulo 7

# Como criar seu projeto no github a partir do modelo do professor

### 7.1 Roteiro para criar uma conta no github

Objetivo: Criar uma conta no site github.

1. Entrar no site <http://github.com>.
2. Clicar em criar conta.
3. Entre com login.
4. Entre com senha.

### 7.2 Roteiro para criar um repositório novo (vazio) no github

Objetivo: criar um repositório novo, vazio, no site do github. Depois este repositório será clonado em seu computador, você vai adicionar os arquivos (`git add`), commitar (`git commit -m"mensagem"`) e então enviar para o servidor github (`git push`).

1. O primeiro passo é criar sua conta no github (se ainda não tem), veja 7.1. Entre no site <https://github.com> e crie sua conta, login e senha.
2. Após fazer o login no site <https://github.com> você vai encontrar, lá em cima, a direita, um ícone +; pressione ele e a seguir selecione "new repository". De o nome "ProjetoEngenharia-Ano-SeuNome". Não adicione nenhum arquivo no projeto, ou seja, não adicione Readme, nem nenhum outro arquivo. Click em "create repository". O github vai criar o repositório.

3. Agora vamos copiar/clonar o seu novo repositório do site do github para sua máquina.

Selecione o botão "copy" e então copie o endereço do projeto. Outra opção seria copiar o endereço do projeto na barra de comandos de seu navegador de internet.

Abra um terminal e digite a sequência:

```
cd
cd workdir
git clone https://github.com/seuLogin/ProjetoEngenharia-Ano-SeuNome
```

O programa `git` vai copiar os arquivos do servidor github para sua máquina local.

4. Agora temos de obter uma copia dos arquivos do modelo do professor. Veja na seção 7.3 como clonar o modelo de projeto.

Depois de clonar o modelo de projeto, dentro do diretório `workdir`, teremos dois subdiretórios, um com o projeto modelo do professor e outro com seu projeto

```
workdir/ModeloDocumento-ProjetoEngenharia-ProgramacaoPratica
workdir/ProjetoEngenharia-Ano-SeuNome
```

5. Bem, até aqui o seu projeto ainda esta vazio. Temos de copiar os arquivos do modelo para dentro do seu projeto.

Usando o navegador de arquivos você deve copiar os diretórios "imagens", "listagens", "lyx" e o arquivo `leiametext` para dentro do diretório "`workdir/ModeloDocumento-ProjetoEngenharia-ProgramacaoPratica`". Também pode fazer isso usando o terminal, veja comandos de terminal abaixo.

```
cd
cd workdir/ModeloDocumento-ProjetoEngenharia-ProgramacaoPratica
cp -R imagens listagens lyx leiametext ../ProjetoEngenharia-Ano-SeuNome
```

Entre no diretório `workdir/ProjetoEngenharia-Ano-SeuNome` e verifique se os arquivos foram copiados.

Neste momento o diretório `workdir/ProjetoEngenharia-Ano-SeuNome` já tem diversos arquivos novos, mas estes arquivos ainda não fazem parte do repositório.

6. Agora você precisa pedir ao `git` para adicionar estes arquivos novos no seu repositório local. Possível sequência de comandos:

```
cd ~/workdir/ProjetoEngenharia-Ano-SeuNome
git status
git add *
```

7. Precisa comitar os arquivos, ou seja, enviar para o repositório local:

```
git commit -m "Adicionados arquivos do modelo no ProjetoEngenharia-Titulo-
```

8. Finalmente precisa enviar os arquivos do repositório local para o repositório no site github. Comandos:

```
git push
```

é possível que peça seu login e senha de acesso ao site github.

9. Neste momento você pode usar seu navegador e verificar se os arquivos foram carregados no site do seu repositório

Veja: <https://github.com/seuLogin/ProjetoEngenharia-Ano-SeuNome>

Se os arquivos não foram carregados é possível que você tenha de configurar seu ssh e copiar os dados de acesso no site do github. Tem vídeos na internet que mostram como fazer. Na videoaula que disponibilizamos mostramos como fazer. Tem instruções no site do github, veja <https://docs.github.com/en/github/authenticating-to-github/connecting-to-github-with-ssh>.

10. Estando tudo correto, você poderá modificar os arquivos, adicionar novos arquivos, enviar os arquivos para o site do seu repositório no site do github. Ou seja, pode usar comandos como:

```
git status
git add Class1.h Class1.cpp
git commit -m "mensagem"
git push
```

## 7.3 Roteiro para clonar o repositório do projeto modelo em seu computador

Objetivo: neste caso queremos ter, em nosso computador, uma cópia do modelo do projeto da disciplina de projeto de engenharia (programação prática).

1. Copiar (clonar) o modelo de projeto da disciplina em seu computador

Abra um terminal e faça o download do modelo de projeto, abaixo os comandos:



```
cd
mkdir workdir
cd workdir
git clone https://github.com/ldsc/ModeloDocumento-ProjetoEngenharia-ProgramacaoPratica
```

Após alguns segundos/minutos o projeto estará em seu computador na pasta  
~/workdir/ModeloDocumento-ProjetoEngenharia-ProgramacaoPratica.

Note que você clonou um projeto pré-existente, criado pelo professor. Você não pode usar ele como o seu projeto, pois o site github não vai aceitar o envio de modificações feitas por você. Ou seja, você clonou o projeto do professor e tem acesso apenas leitura. Mas você pode copiar estes arquivos para dentro do repositório workdir/ProjetoEngenharia-Ano-SeuNome e manipular como sendo seus.

## 7.4 Roteiro para clonar o repositório do projeto modelo diretamente (+simples)

Atualmente o github permite que um repositório público possa ser clonado diretamente. Veja as etapas:

1. Entrar no site <http://github.com>; Entre com seu login e senha.
2. Clicar no botão + e selecionar "import repository".
3. Abaixo de "Your old repository's clone URL" você deve colocar o endereço do repositório que será clonado. Cole ali o seguinte endereço:  
"https://github.com/ldsc/ModeloDocumento-ProjetoEngenharia-ProgramacaoPratica".
4. Marcar como publico.
5. Clicar em "Begin import". Com esta sequência você estará criando uma cópia do modelo do professor em sua conta pessoal, e poderá modificar a mesma.
6. Lembre que esta cópia está lá no servidor do github, e que você terá de copiar/clonar a mesma em seu computador local. Terá de executar uma sequência de comandos como esta abaixo:

```
cd
cd workdir
git clone https://github.com/seuLogin/ProjetoEngenharia-Ano-SeuNome
cd ProjetoEngenharia-Ano-SeuNome
```

# Índice Remissivo

## C

Citações, 17

Citações no meio do texto, 18

## I

Implementação, 3