

# Apresentação do Curso de Laboratório de Gerência de Configuração

Leonardo Gresta Paulino Murta  
leomurta@ic.uff.br

# Apresentações

- Quem sou eu?
  - Leonardo Murta
  - <http://www.ic.uff.br/~leomurta>
- Quem são vocês?
  - Nome?
  - Emprego? Estágio? Projeto de Aplicação? Iniciação Científica?
  - O que acha de Engenharia de Software?
  - Expectativas para Laboratório de Gerência de Configuração?



# O que é Gerência de Configuração?

“GC é uma disciplina que visa **controlar a evolução** de sistemas de software”

Susan Dart (1991)

# GC na Engenharia de Software

Atividades Gerenciais

Planejamento de Projetos

Monitoração e Controle

Melhoria de Processos

Gerência de Riscos

Atividades de Desenvolvimento

Levantamento de Requisitos

Análise de Requisitos

Projeto

Codificação

Atividades de Apoio

Garantia da Qualidade

Medição e Análise

Verificação, Validação e Testes

Gerência de Configuração

Reutilização

# Objetivo do curso de Lab GC

- Foco em controle de versões
- Estudar sistemas de GC
  - SCCS, RCS e CVS (razões históricas)
  - Subversion
  - Mercurial
  - Git
- Fazer um sistema de controle de versões

# Conhecimento

Transferido para os  
alunos via aulas  
expositivas



Construído  
colaborativamente  
pelos alunos

ES 1

ES 2

GC

Lab GC

# Grupos

- As atividades são em grupo
  - Mas as notas são individuais!
- Deve ser o mesmo durante todo o curso
  - 2 participantes para graduação
  - Individual para mestrado e doutorado
- Definir na primeira semana de aula
  - Enviar por e-mail (assunto: Lab GC - Grupo) a matrícula e o nome completo dos participantes

# Dinâmica do curso

- Sextas de 9h às 11h
  - Apresentações gerais (professor)
  - Apresentações das leituras da semana (1 a 2 grupos por dia)
  - Discussão das leituras da semana (venham preparados!)
- Sextas de 11h às 13h
  - Seminários de andamento dos trabalhos (2 grupos por dia)
  - Discussões de design (todos devem colaborar!)
  - Apresentações voluntárias relacionadas a GC



# Avaliação

$$Média = \frac{(Apresentações de Leituras + Seminários + Trabalho)}{3}$$

- APROVADO

*Presença  $\geq$  75% E Média  $\geq$  6*

- VERIFICAÇÃO SUPLEMENTAR

*Aluno de Graduação E Presença  $\geq$  75% E  $4 \leq$  Média  $<$  6*

*Será aprovado na VS se tirar nota maior ou igual a 6*

- REPROVADO

*Caso contrário*

# Segundo o Regulamento dos Cursos de Graduação/Pós...

- Presença
  - 75% das aulas (Grad. Art. 80, §14 e Pós. Art. 20)
  - Nenhuma falta será abonada (Art. 80, §15)
- Segunda Chamada
  - Não será permitida a Avaliação de Aprendizagem em Caráter Excepcional (i.e., 2ª chamada)
  - Com exceção dos casos citados no Art. 87 (**congressos, competições ou serviço militar**), de acordo com os procedimentos do Art. 88 (**aviso na coordenação do curso com 30 dias de antecedência**).

# Leituras (+/- 35 pp. por leitura)

## 1ª leitura: sistemas históricos (SCCS, RCS e CVS)

- Marc J. Rochkind, 1975, “The Source Code Control System”, IEEE Transactions on Software Engineering (TSE), 1 (4), pp. 364-370
- Walter F. Tichy, 1984, “RCS - A System for Version Control”, Purdue University, TR 84-474
- Capítulo 1, “Overview”, de Cederqvist, 2005, “Version Management with CVS”

## 2ª leitura: Subversion (1 de 3)

- Prefácio, “What Is Subversion?”, de Ben Collins-Sussman, Brian W. Fitzpatrick, C. Michael Pilato, “Version Control with Subversion”
- Capítulo 1, “Fundamental Concepts”, de Ben Collins-Sussman, Brian W. Fitzpatrick, C. Michael Pilato, “Version Control with Subversion”
- Capítulo 2, “Basic Usage”, de Ben Collins-Sussman, Brian W. Fitzpatrick, C. Michael Pilato, “Version Control with Subversion”

## 3ª leitura: Subversion (2 de 3)

- Capítulo 4, “Branching and Merging”, de Ben Collins-Sussman, Brian W. Fitzpatrick, C. Michael Pilato, “Version Control with Subversion”

## 4ª leitura: Subversion (3 de 3)

- Capítulo 5, “Repository Administration”, de Ben Collins-Sussman, Brian W. Fitzpatrick, C. Michael Pilato, “Version Control with Subversion”

# Leituras (+/- 35 pp. por leitura)

## 5ª leitura: Mercurial (1 de 2)

- Capítulo 1, “How did we get here?”, de Bryan O’Sullivan, “Mercurial: The Definitive Guide”
- Capítulo 2, “A tour of Mercurial: the basics”, de Bryan O’Sullivan, “Mercurial: The Definitive Guide”
- Capítulo 3, “A tour of Mercurial: merging work”, de Bryan O’Sullivan, “Mercurial: The Definitive Guide”

## 6ª leitura: Mercurial (2 de 2)

- Capítulo 4, “Behind the scenes”, de Bryan O’Sullivan, “Mercurial: The Definitive Guide”
- Capítulo 6, “Collaborating with other people”, de Bryan O’Sullivan, “Mercurial: The Definitive Guide”
- Capítulo 8, “Managing releases and branchy development”, de Bryan O’Sullivan, “Mercurial: The Definitive Guide”

## 7ª leitura: Git (1 de 6)

- Capítulo 1, “Getting Started”, de Scott Chacon, 2010, “Pro Git”
- Capítulo 2, “Git Basics”, de Scott Chacon, 2010, “Pro Git”

## 8ª leitura: Git (2 de 6)

- Capítulo 3, “Git Branching”, de Scott Chacon, 2010, “Pro Git”

# Leituras (+/- 35 pp. por leitura)

## 9ª leitura: Git (3 de 6)

- Capítulo 4, “Git on the Server”, de Scott Chacon, 2010, “Pro Git”

## 10ª leitura: Git (4 de 6)

- Capítulo 5, “Distributed Git”, de Scott Chacon, 2010, “Pro Git”

## 11ª leitura: Git (5 de 6)

- Capítulo 6, “Git Tools”, de Scott Chacon, 2010, “Pro Git”

## 12ª leitura: Git (6 de 6)

- Capítulo 9, “Git Internals”, de Scott Chacon, 2010, “Pro Git”

# Apresentações de Leituras

- 1 a 2 grupos ficarão encarregado de apresentar as leituras da semana
  - Apresentação de 30 a 60 minutos
  - Uso de projetor e quadro branco
- Todos os grupos devem participar das discussões

# Trabalho

- Objetivo:
  - Construir um sistema de controle de versões
  - Cada grupo é responsável por módulos desse sistema
- Resultado do trabalho:
  - Seminários de andamento e final
  - Implementação
  - Artigo
- Atraso na entrega do trabalho terá 1 ponto de multa por dia

# Seminários

- Semanas iniciais
  - Contexto do trabalho
  - Detalhamento das interfaces
  - Metodologia a ser adotada na execução
- Semanas intermediárias
  - Andamento do trabalho
  - Resultado concreto, mesmo que ainda parcial
- Rodada final
  - Apresentação final do trabalho
  - Resultados obtidos
  - Relato de experiência



# Implementação

- Funcional
- Integrada com as demais **desde o início!**
- Linguagem: Java
- Repositório: GitHub

# Artigo

- Formato da SBC (de 8 a 10 páginas)
- Conteúdo
  - Introdução (motivação, objetivo e organização do trabalho)
  - Fundamentação teórica e trabalhos relacionados
  - Abordagem proposta
  - Aspectos da implementação
  - Avaliação experimental ou exemplo de utilização
  - Conclusão (contribuições, limitações e trabalhos futuros)

# Busquem por mais conhecimento...

- <http://scholar.google.com.br>
- <http://www.informatik.uni-trier.de/~ley/db>
- <http://ieeexplore.ieee.org>
- <http://portal.acm.org>
- <http://citeseer.ist.psu.edu>
  
- E usem uma ferramenta para controlar as suas referências: <http://www.zotero.org>

# Página do curso



**Leonardo Cresta Paulino Murta**  
 Assistant Professor (Professor Adjunto II), IC/UFF  
 D.Sc., COPPE/UFRJ, 2006  
 M.Sc., COPPE/UFRJ, 2002  
 B.Sc., IM/UFRJ, 1999

**Laboratório de Gerência de Configuração**

**Logística**

Disciplina: Laboratório de Gerência de Configuração  
 Data: quartas e sextas, de 9:00 às 11:00  
 Local: quartas na sala 331 do bloco D e sextas na sala 335 do bloco D  
 Lista: <http://groups.google.com/group/uff-labgc-2011-2> (importante: todos os alunos **devem** se cadastrar nessa lista)

**Avaliação**

Média = (Apresentação de Leituras + Resumos + Seminários + 2 x Trabalho) / 5

**APROVADO**  
 (Presença >= 75%) E (Média >= 6)

**VERIFICAÇÃO SUPLEMENTAR**  
 (Aluno de Graduação) E (Presença >= 75%) E (4 <= Média < 6)

Será aprovado na VS se tirar nota maior ou igual a 6.

**REPROVADO**  
 Caso contrário

**Grupos**

As atividades do curso serão feitas individualmente para alunos de doutorado e em dupla para alunos de mestrado e em trio para alunos de graduação, que devem ser constituídos na primeira semana de aula e ter a mesma formação até o final do curso. Apesar das atividades serem em grupo, a avaliação será individual. Entreguem por e-mail (assunto: Lab GC - Grupo) o número da matrícula e o nome completo de cada participante do grupo.

**Apresentação de Leituras**

Um cronograma de leituras de artigos e capítulos de livro é indicado para viabilizar a consolidação dos conhecimentos referentes à matéria.

Uma das aulas da semana (ver seção Cronograma) consiste na apresentação dos leituras e capítulos de livro lidos na semana, seguida de discussões sobre as leituras. Essa apresentação será feita por grupos previamente definidos (reserve o quanto antes as suas por e-mail). Os grupos responsáveis por apresentar a leitura da semana devem definir qual parte da leitura ficará a cargo de qual grupo. Além disso, os demais grupos ficarão encarregados de discutir aspectos da leitura. Cada apresentação deve ter duração de 20 a 30 minutos. A participação nessas atividades será considerada na composição da Média (item Apresentação de Leituras).

Os artigos e livros estão disponíveis em formato pdf no **Google Docs** para os alunos já cadastrados no **Google Groups**.

**1ª leitura: sistemas históricos (SCCS, RCS e CVS)**  
 Marc J. Rochkind, 1975, "The Source Code Control System", IEEE Transactions on Software Engineering (TSE), 1 (4), pp. 364-370

Leiam as **regras** do curso no site, anotem as **datas** e tragam as dúvidas na próxima aula!!!

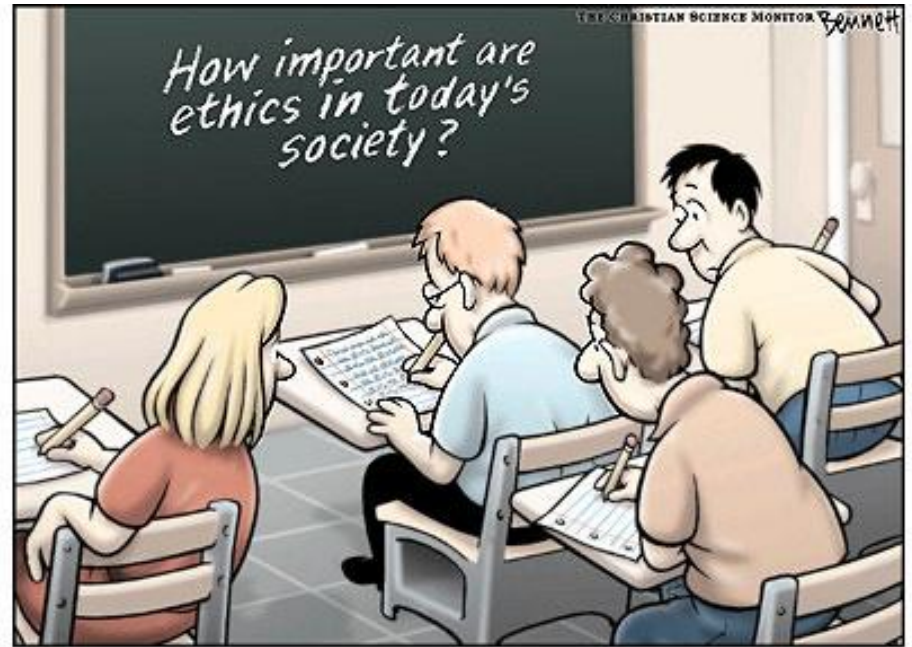
<http://www.ic.uff.br/~leomurta>

(dica: monitorem com <http://www.changedetection.com>)

**Importante:** cadastrem-se no Facebook informado no site!  
 (as leituras estão disponíveis lá)

# Fair Play!

- Não colar ou dar cola em provas
- Não plagiar o trabalho
- Não trapacear nas leituras e listas de exercício
- Não sobrecarregar os colegas do grupo
- Não assinar presença por colegas
- Dar crédito apropriado quando usar trabalhos de terceiros



<http://www.claybennett.com/pages/ethics.html>

# Apresentação do Curso de Laboratório de Gerência de Configuração

Leonardo Gresta Paulino Murta  
leomurta@ic.uff.br