

Curso: Git e GitHub

1º Dia - Caracterizando o problema

Ricardo da Fonseca e Michel Farah

UNESP

23 de novembro de 2011

Conteúdo

1 Introdução

- Escrevendo um trabalho
- Formas de controlar um trabalho

2 Controlando as versões

- Para que serve o controle de versão?
- Como funciona?

3 Objetivo do Curso

4 Cronograma

Problemas

- Alguém já sobrescreveu um trabalho por acidente e acabou perdendo as alterações?
- Tem dificuldade em saber quais as alterações efetuadas e quando foram feitas?
- Tem dificuldade em recuperar uma versão anterior a qual está trabalhando?
- Tem problemas em manter várias versões do trabalho?

Problemas

- Alguém já sobrescreveu um trabalho por acidente e acabou perdendo as alterações?
- Tem dificuldade em saber quais as alterações efetuadas e quando foram feitas?
- Tem dificuldade em recuperar uma versão anterior a qual está trabalhando?
- Tem problemas em manter várias versões do trabalho?

Problemas

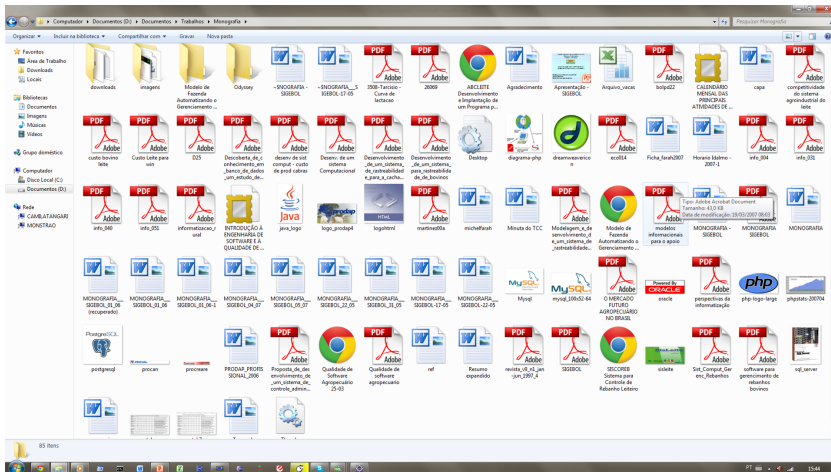
- Alguém já sobrescreveu um trabalho por acidente e acabou perdendo as alterações?
- Tem dificuldade em saber quais as alterações efetuadas e quando foram feitas?
- Tem dificuldade em recuperar uma versão anterior a qual está trabalhando?
- Tem problemas em manter várias versões do trabalho?

Problemas

- Alguém já sobrescreveu um trabalho por acidente e acabou perdendo as alterações?
- Tem dificuldade em saber quais as alterações efetuadas e quando foram feitas?
- Tem dificuldade em recuperar uma versão anterior a qual está trabalhando?
- Tem problemas em manter várias versões do trabalho?

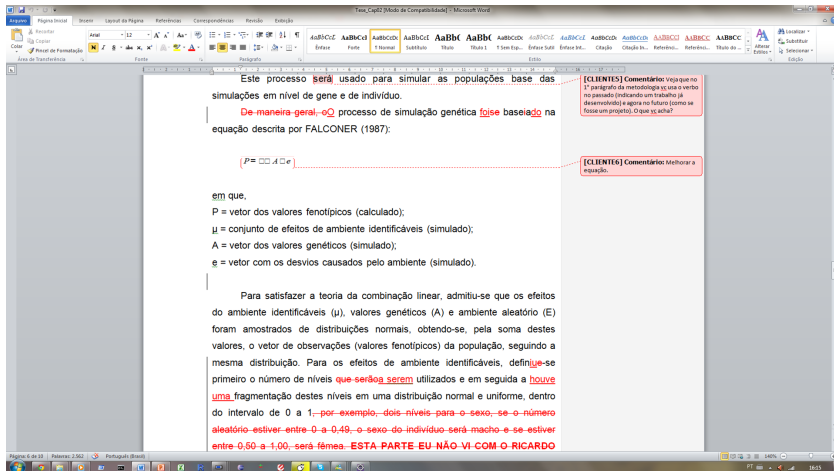
Formas de escrever um Trabalho:

- Como organizamos nossas Dissertações ou Teses?



Formas de escrever um Trabalho:

- Como corrigimos nossos trabalhos?



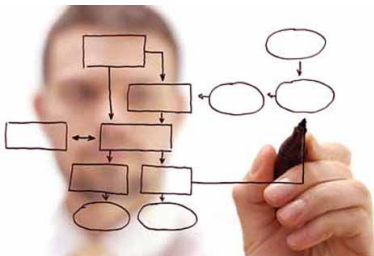
Trabalho em grupo:

- Como organizamos um trabalho em grupo?



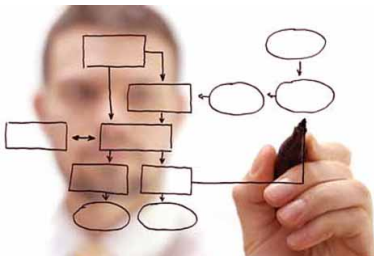
Exemplos:

- Desenvolvimento de Programas;
- Desenvolvimento de Algoritmos complexos;
- Escrever trabalhos;
- Entre outros.



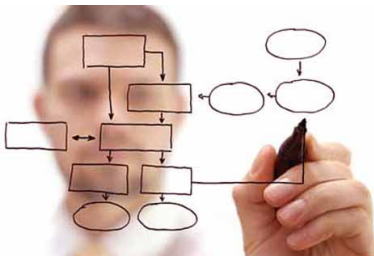
Exemplos:

- Desenvolvimento de Programas;
- Desenvolvimento de Algoritmos complexos;
- Escrever trabalhos;
- Entre outros.



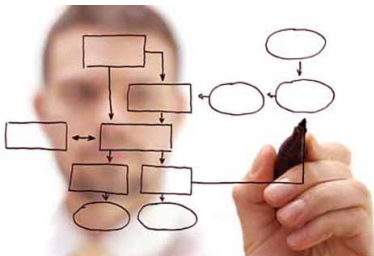
Exemplos:

- Desenvolvimento de Programas;
- Desenvolvimento de Algoritmos complexos;
- Escrever trabalhos;
- Entre outros.



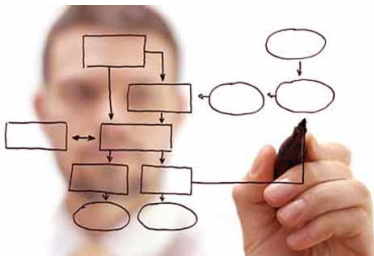
Exemplos:

- Desenvolvimento de Programas;
- Desenvolvimento de Algoritmos complexos;
- Escrever trabalhos;
- Entre outros.



Exemplos:

- Desenvolvimento de Programas;
- Desenvolvimento de Algoritmos complexos;
- Escrever trabalhos;
- Entre outros.



O que é um controlador de versão?

- É um software com finalidade de gerenciar diferentes versões no desenvolvimento de um documento qualquer.
- Muito utilizado por empresas de desenvolvimento de softwares e é útil em diversos aspectos, tanto para projetos pequenos e pessoais como também para grandes projetos comerciais.
- Os mais comuns são:
 - Git;
 - SVN;
 - SourceSafe (da Microsoft);
 - ClearCase (da IBM);
 - Entre muitos outros.

O que é um controlador de versão?

- É um software com finalidade de gerenciar diferentes versões no desenvolvimento de um documento qualquer.
- Muito utilizado por empresas de desenvolvimento de softwares e é útil em diversos aspectos, tanto para projetos pequenos e pessoais como também para grandes projetos comerciais.
- Os mais comuns são:
 - Git;
 - SVN;
 - SourceSafe (da Microsoft);
 - ClearCase (da IBM);
 - Entre muitos outros.

O que é um controlador de versão?

- É um software com finalidade de gerenciar diferentes versões no desenvolvimento de um documento qualquer.
- Muito utilizado por empresas de desenvolvimento de softwares e é útil em diversos aspectos, tanto para projetos pequenos e pessoais como também para grandes projetos comerciais.
- Os mais comuns são:
 - Git;
 - SVN;
 - SourceSafe (da Microsoft);
 - ClearCase (da IBM);
 - Entre muitos outros.

O que é um controlador de versão?

- É um software com finalidade de gerenciar diferentes versões no desenvolvimento de um documento qualquer.
- Muito utilizado por empresas de desenvolvimento de softwares e é útil em diversos aspectos, tanto para projetos pequenos e pessoais como também para grandes projetos comerciais.
- Os mais comuns são:
 - Git;
 - SVN;
 - SourceSafe (da Microsoft);
 - ClearCase (da IBM);
 - Entre muitos outros.

O que é um controlador de versão?

- É um software com finalidade de gerenciar diferentes versões no desenvolvimento de um documento qualquer.
- Muito utilizado por empresas de desenvolvimento de softwares e é útil em diversos aspectos, tanto para projetos pequenos e pessoais como também para grandes projetos comerciais.
- Os mais comuns são:
 - Git;
 - SVN;
 - SourceSafe (da Microsoft);
 - ClearCase (da IBM);
 - Entre muitos outros.

O que é um controlador de versão?

- É um software com finalidade de gerenciar diferentes versões no desenvolvimento de um documento qualquer.
- Muito utilizado por empresas de desenvolvimento de softwares e é útil em diversos aspectos, tanto para projetos pequenos e pessoais como também para grandes projetos comerciais.
- Os mais comuns são:
 - Git;
 - SVN;
 - SourceSafe (da Microsoft);
 - ClearCase (da IBM);
 - Entre muitos outros.

Para que serve?

- **Histórico** - Registra toda a evolução do projeto, cada alteração em cada arquivo. Ajuda a responder algumas questões básicas:

- 1 “quem?”
- 2 “o quê?”
- 3 “onde?”
- 4 “quando?”
- 5 “por que?”
- 6 “como?”
- 7 “histórico”



Para que serve?

- **Histórico** - Registra toda a evolução do projeto, cada alteração em cada arquivo. Ajuda a responder algumas questões básicas:

- 1 “quem?”
- 2 “o quê?”
- 3 “onde?”
- 4 “quando?”
- 5 “por que?”
- 6 “como?”
- 7 “histórico”



Para que serve?

- **Histórico** - Registra toda a evolução do projeto, cada alteração em cada arquivo. Ajuda a responder algumas questões básicas:

- 1 “quem?”
- 2 “o quê?”
- 3 “onde?”
- 4 “quando?”
- 5 “por que?”
- 6 “como?”
- 7 “histórico”



Para que serve?

- **Histórico** - Registra toda a evolução do projeto, cada alteração em cada arquivo. Ajuda a responder algumas questões básicas:

- 1 “quem?”
- 2 “o quê?”
- 3 “onde?”
- 4 “quando?”
- 5 “por que?”
- 6 “como?”
- 7 “histórico”



Para que serve?

- **Histórico** - Registra toda a evolução do projeto, cada alteração em cada arquivo. Ajuda a responder algumas questões básicas:

- 1 “quem?”
- 2 “o quê?”
- 3 “onde?”
- 4 “quando?”
- 5 “por que?”
- 6 “como?”
- 7 “histórico”



Para que serve?

- **Histórico** - Registra toda a evolução do projeto, cada alteração em cada arquivo. Ajuda a responder algumas questões básicas:

- 1 “quem?”
- 2 “o quê?”
- 3 “onde?”
- 4 “quando?”
- 5 “por que?”
- 6 “como?”
- 7 “histórico”



Para que serve?

- **Histórico** - Registra toda a evolução do projeto, cada alteração em cada arquivo. Ajuda a responder algumas questões básicas:

- 1 “quem?”
- 2 “o quê?”
- 3 “onde?”
- 4 “quando?”
- 5 “por que?”
- 6 “como?”
- 7 “histórico”



Para que serve?

- **Histórico** - Registra toda a evolução do projeto, cada alteração em cada arquivo. Ajuda a responder algumas questões básicas:

- 1 “quem?”
- 2 “o quê?”
- 3 “onde?”
- 4 “quando?”
- 5 “por que?”
- 6 “como?”
- 7 “histórico”



Para que serve?

- **Colaboração** - Possibilita que várias pessoas trabalhem em paralelo sobre os mesmos arquivos sem que um sobrescreva o trabalho do outro.



Para que serve?

- **Variações no Projeto** - Mantém linhas diferentes de evolução do mesmo projeto



Como funciona?

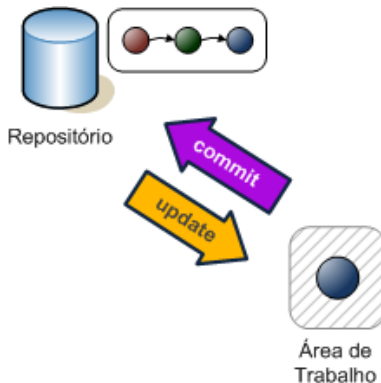


Figura: Estrutura de um controle de versão é composta por repositório e área de trabalho. A comunicação entre elas é através de operações como *commit* e *update*

Como funciona?

- Cada *commit* gera uma **revisão** no repositório, contendo as modificações feitas, data e autor.
- Cada **revisão** gera uma “foto” dos arquivos e diretórios daquele *commit*, gerando o histórico do projeto. Estas “fotos” podem ser recuperadas e analisadas sempre que desejado.



Objetivos:

1 O que é o Git?

- ▶ Instalação;
- ▶ Funções básicas.

2 O que é GitHub?

- ▶ Funções básicas.



Objetivos:

1 O que é o Git?

- ▶ Instalação;
- ▶ Funções básicas.

2 O que é GitHub?

- ▶ Funções básicas.



Objetivos:

1 O que é o Git?

- ▶ Instalação;
- ▶ Funções básicas.

2 O que é GitHub?

- ▶ Funções básicas.



Objetivos:

1 O que é o Git?

- ▶ Instalação;
- ▶ Funções básicas.

2 O que é GitHub?

- ▶ Funções básicas.



Cronograma:

Período	Segunda	Terça	Quarta	Quinta
8:00-10:00	Apresentação	Git	Git	GitHub
10:00-10:30	Café	Café	Café	Café
10:30-12:00	Exercícios	Exercícios	Exercícios	Exercícios