**UNIVERSIDADE DO OESTE DE SANTA CATARINA**

**UNOESC – CAMPUS CHAPECÓ**

**FLUXOS DE TRABALHO GIT**

**Professor(a): Iskailer Inaian Rodrigues**

**Acadêmico(s): Julio Cesar Martins, Nome aluno 2, Nome aluno 3, Nome aluno 4**

**CHAPECÓ, SC**

**2017**

**LISTA DE ILUSTRAÇÕES**

Ilustração 1 - Xxxxxxxx xxxx xxxxxx xxxxxxxxxxx............................................................. 99

Ilustração 2 - Xxxxxxxxxx xxxx xxxxxx xxxxxxxxxxx......................................................... 99

Ilustração 3 - Xxxxxxxxxx xxxxxxxxx xxxxxxxxxxx xxxxxx xxxxxxxxxxxxx.................... 99

Ilustração 4 - Xxxxxx xxxxxxxxxxx....................................................................................... 99

Ilustração 5 - Xxxx xxxxxx xxxxxxxxxxx.............................................................................. 99

Espaço entre linhas 1,5

Espaçamento antes e depois de 0 pontos

Ilustrações são considerados:

· Desenhos

· Esquemas

· Fluxogramas

· Fotografias

· Gráficos

· Mapas

· Plantas

· Retratos

**LISTA DE QUADROS**

Quadro 1 - Xxxxxxxxxx xxxxxxxxx xxxxxxx xxxx xxxxxx xxxxxxxxxxx........................... 99

Quadro 2 - Xxxxxxxxxx xxxxxxxxx xxxxxx.......................................................................... 99

Quadro 3 - Xxxxxxxxxx xxxxxxxxx xxxxxxxxxxx xxxxxx xxxxxxxxxxxxx........................ 99

Quadro 4 - X xxxx xxxxxx xxxxxxxxxxx............................................................................... 99

Quadro 5 - Xxxxxxxxxx xxxxxxxxx xxxxx xxxx xxxxxx xxxxxxxxxxx............................... 99

Espaço entre linhas 1,5

Espaçamento antes e depois de 0 pontos

Quadros geralmente são utilizados para apresentar informações que não necessitam de nenhum auxílio matemático-estatístico. Geralmente em TI é utilizado para apresentar Código Fonte.

**LISTA DE TABELAS**

Tabela 1 - Xxxxxxxxxx xxxxxxxxx xxxxxxxxxxx xxxxxx xxxxxxxx xxxxxxxx x xx.................... 99

Tabela 2 - Xxxxx xxxx xxxxxx xxxxxxxxxxx.................................................................................. 99

Tabela 3 - Xxxxxxxxxx xxxxxxxxx xxxxxxxxxxx xxxxxx xxxxxxxxxxxxx................................... 99

Espaço entre linhas Simples 1,5

Espaçamento antes e depois de 0 pontos

Tabelas são compostas por células e apresentam fórmulas estatísticas e matemáticas para apresentar resultados como Médias, Somatórios, Medianas, etc.

**LISTA DE ABREVIAÇÕES E SIGLAS**

CD-ROM - Compact Disk Read Only Memory

DNS - Domain Name Server

DVD - Digital Video Disc

HD - Hard Disc

HDD - Hard Disc Drive

PIB - Produto Interno Bruto

RAM - Random Access Memory

ROM - Read Only Memory

Espaço entre linhas 1,5

Espaçamento antes e depois de 0 pontos

Classificar as Siglas/Abreviações em Ordem Alfabética de A → Z (crescente)

[1 INTRODUÇÃO 5](#_Toc496542168)

[2 DESENVOLVIMENTO 6](#_Toc496542169)

[2.1 Fluxo com Repositório central e somente um branch. 6](#_Toc496542170)

[2.2 Fluxo de Trabalho do Gerente de Integração. 7](#_Toc496542171)

[2.3 Fluxo de Ditador e Tenentes. 8](#_Toc496542172)

[3 CONCLUSÃO 9](#_Toc496542173)

[4 REFERÊNCIAS 10](#_Toc496542174)

# **1 INTRODUÇÃO**

Apresentar brevemente em um parágrafo de no mínimo 4 linhas e no máximo 10 uma contextualização sobre a sua pesquisa ou relatório. A contextualização é necessária para localizar o leitor do seu trabalho no assunto que irá abordar. Contextualizar tem origem no latim e significa "colocar alguém a par de algo" ou "situar um aspecto que está sendo relatado, no tempo e no espaço". Quando contextualizamos, queremos apresentar a conjuntura de uma situação, ou seja, as circunstâncias em torno de um acontecimento. Quando relatamos um acontecimento a uma pessoa, buscamos descrever o ambiente do fato, ou seja, capturamos o contexto. Já quando precisamos trazer o leitor para dentro do que vamos apresentar, dessa forma os argumentos tratados no trabalho ficam mais claros para o leitor.

Após realizada a contextualização do seu trabalho você deverá descrever a problematização dele. A problematização é construída em um parágrafo de no mínimo umas 6 linhas. O problema tem que ser definido de forma prática e clara. “Um problema implica uma ou mais dúvidas ou dificuldades em relação ao tema, que você se proporá a resolver. Formulá-lo, portanto, deve envolver perguntas, que o trabalho procurará responder” (MÁTTAR NETO, 2002, p. 143). A descrição do problema é o passo inicial para o desenvolvimento de qualquer trabalho. Se não tivermos um problema não temos um trabalho para desenvolver. O problema é a dúvida inicial que proporciona a busca da solução do mesmo. “A formulação do problema deve ser concisa e objetiva, a fim de permitir clareza” (STRIEDER, 2009, p. 37). O problema é sempre descrito na forma interrogativa, uma pergunta ou de um questionamento.

Depois de descrever a problematização você deverá descrever de forma breve, clara e concisa como resolveu o problema. Lembre-se de não apontar muitos detalhes aqui, pois os detalhes da solução do problema serão apresentados no próximo capítulo.

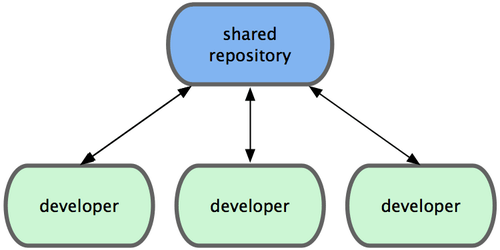
O último parágrafo da sua introdução deve ser um relato dos principais resultados que alcançou com o seu trabalho. Descreva aqui só o principal, os detalhes dos resultados estarão na conclusão.

# **2 DESENVOLVIMENTO**

## **2.1 Fluxo com Repositório central e somente um branch.**

Um dos fluxos mais simples, é utilizado apenas um repositório central com somente o branch master.

Neste modelo os desenvolvedores tem permissão para um push direto no repositório central, isso significa que os desenvolvedores fizerem um clone do repositório e o primeiro desenvolvedor fazer o push o segundo desenvolvedor terá que fazer um merge ou rebase do repositório para não sobrescrever o trabalho do primeiro desenvolvedor. Modelo normalmente utilizado para pequenas equipes



**Ilustração 1 – Repositório Centralizado**

Vantagens

* A simplicidade desse fluxo permite uma adoção mais tranquila para

quem está começando a utilizar.

* É mais fácil de adotar integração contínua, uma prática bastante comum

em projetos que usam metodologias ágeis.

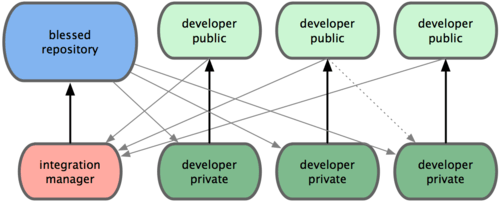
* É possível detectar conflitos a cada pull

Desvantagens

* Com esse fluxo, fica difícil separar o código da correção do defeito do código das novas funcionalidades, o que pode gerar mais defeitos e insatisfação nos clientes.
* Como tudo é comitado na branch master, as entregas são feitas com todo o código que está no repositório no momento da entrega. Não é possível entregar só parte das funcionalidades que foram compartilhadas com o repositório central.

## **2.2 Fluxo de Trabalho do Gerente de Integração**.

Esse fluxo funciona a partir de múltiplos repositórios, geralmente clonados do repositório “oficial”. Cada desenvolvedor cria seu próprio repositório, depois de “commitar” algumas modificações, pode ser feito um push para seu repositório. Quando o desenvolvedor estiver satisfeito com seu código, é possível enviar um pull request para o projeto oficial. O responsável pelo repositório oficial pode revisar o código desenvolvido e aplicar as melhorias ao repositório original. Modelo utilizado projetos open source de pequeno ou médio porte.



**Ilustração 2 – Gerente de Integralização.**

**Vantagens**

* Não é necessário dar permissões de push para todos os colaboradores do projeto.
* É um bom modelo para projetos open source de pequeno ou médio porte.

**Desvantagens**

* A integração das mudanças dos forks é feita de maneira bem tardia. Possíveis conflitos e/ou erros seriam descobertos apenas na hora de aplicarmos o pull request.
* Como só há um repositório original com, provavelmente apenas um mantenedor, o número de pull requests poderia ir acumulando. Para projetos open source muito grandes é necessária uma outra abordagem.

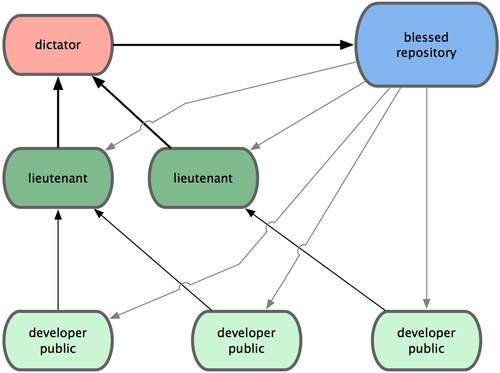
## **2.3 Fluxo de Ditador e Tenentes.**

Este fluxo funciona de forma parecida com o fluxo do gerente de integração, o mantenedor do projeto dica como um ditador, que tem a última palavra sobre o código do projeto, o ditador elege tenentes que mantem repositórios com partes do projeto, geralmente separado por módulos, esses eleitos servem como tenentes que recebem os pull request dos outros desenvolvedores e filtram. Um novo desenvolvedor teria de escolher um dos repositórios dos tenentes para fazer seu fork do projeto, provavelmente considerando o módulo em que quer colaborar. Depois de feitos seus commits, faria o push para seu repositório.

Então, poderia fazer um pull request para seu tenente, que faria uma revisão e daria um feedback.

Quando apropriado, o tenente faria pull requests para o repositório do ditador, sinalizando um pacote interessante de mudanças. O ditador fará das alterações para o repositório de referência, os tenentes possam fazer o rebase.

Modelo normalmente utilizado para projetos open source grandes, com milhares de colaboradores



**Ilustração 2 – Ditador e Tenentes.**

**Vantagens**

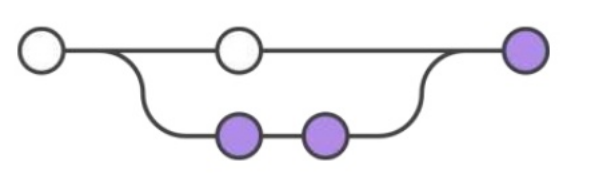
* Assim como no fluxo anterior, não são necessárias permissões de push para o repositório original, do ditador, nem dos tenentes.
* É um fluxo que funciona bem em projetos open source de grande porte.

**Desvantagens**

* É um fluxo de trabalho extremamente complicado, que requer muita familiaridade com o Git.
* A integração é feita de maneira tardia, só quando for aplicado o pull request (ou os patches recebidos por e-mail).

## **2.3 Fluxo de Feature Branch.**

Nesse modelo, podemos utilizar branches para isolar o código de novas funcionalidades ou alterações em funcionalidades existentes. A branch master fica estática, ficando estável durante todo o desenvolvimento do projeto. Só quando a funcionalidade estiver pronta, é feito um merge da branch da funcionalidade na master.

****

**Ilustração 2 – Feature Branch.**

# **3 CONCLUSÃO**

Procure fazer um [resumo](http://noticias.universia.com.br/destaque/noticia/2011/12/15/898676/como-twitter-pode-resumir-seu-curriculo.html) geral dentro da conclusão sobre o que foi abordado em todo seu trabalho. Faça isso de forma criativa e inovadora, para que não seja apenas um repetição do que já foi abordado.

Deve apresentar as respostas alcançadas durante a pesquisa. É na conclusão que o autor se mostra e expõe o seu ponto de vista sobre o tema estudado e sobre os resultados obtidos. Mostra a síntese de todo o conteúdo pesquisado, as informações relevantes, os apontamentos essenciais, as dificuldades encontradas

Deve apresentar uma redação direta, clara, relevante, empreendedora, clara e coesa, pois dever representar o fechamento do trabalho.

Não precisa ser extensa, basta que responda todas as questões expostas no objetivo. Ela pode ser feita em texto corrido ou através de tópicos. É importante não confundir conclusão com uma espécie de introdução ou resumo do trabalho, pois muita gente tem essa ideia e acaba errando em relação a conclusão. Lembre-se que tudo que precisava saber já está no restante do trabalho, não precisa ter as informações repetidas na conclusão.

Na tentativa de enfatizar a importância de sua [pesquisa](http://noticias.universia.com.br/destaque/noticia/2012/02/17/912323/como-fazer-uma-pesquisa-academica-em-7-passos.html), não seja sentimental, dramático, coitadinho ou sensacionalista.

# **4 REFERÊNCIAS**

São as informações que permitem a identificação de documentos utilizados, no todo ou em parte, pelo autor do trabalho. A correta e completa identificação de um texto facilitará o processo de sua localização e obtenção por um leitor interessado. É recomendável a consulta a NBR 6023 da ABNT, para a correta apresentação das referências bibliográficas.

ANDRADE, Maria Margarida de. **Introdução à metodologia do trabalho científico**: elaboração de trabalhos na graduação. 6.ed. São Paulo: Atlas, 2003

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS - ABNT. **Referências bibliográficas**: NBR 6023. Rio de Janeiro, ago. 2002.

\_\_\_\_\_\_. **Livros e folhetos**: NBR 6029. Rio de Janeiro, set. 2002.

\_\_\_\_\_\_. **Citações em documentos**: NBR 10520. Rio de Janeiro, ago. 2002.

\_\_\_\_\_\_. **Trabalhos acadêmicos**: NBR 14724. Rio de Janeiro, ago. 2002.

BEUREN, Ilse Maria (org.). **Como elaborar trabalho monográficos em contabilidade**: teoria e prática. São Paulo: Atlas, 2003.

CENTRO UNIVERSITÁRIO DE JARAGUÁ DO SUL – UNERJ. Orientações Metodológicas. Disponível em: <http://www.unerj.br/unerj/nap/orientacoes.doc>. Acesso em: 10 jun 2004.

LAKATOS, Eva Maria; MARCONI, Marina de Andrade. **Fundamentos de metodologia científica**. 6.ed. São Paulo: Atlas, 2007.

MARCONI, Marina de Andrade; LAKATOS, Eva Maria. **Fundamentos de metodologia**. 5.ed. São Paulo: Atlas, 2003.

MÁTTAR NETO, João Augusto. **Metodologia científica na era da informática**. São Paulo: Saraiva, 2002.

OLIVEIRA, Silvio Luiz de. **Tratado de metodologia científica**: projetos de pesquisas, TGI, TCC, monografias, dissertações e teses. São Paulo: Pioneira, 2000.

ROESCH, Sylvia Maria Azevedo. **Projetos de estágio e de pesquisa em administração**: guia para estágios, trabalhos de conclusão, dissertações e estudos de caso. 2.ed. São Paulo: Atlas, 1999.

STRIEDER, Roque. **Diretrizes para elaboração de projetos de pesquisa**. Joaçãba: Ed. Unoesc, 2009.