Identificação de Itens de configuração

Padrao de nomeação de itens de configuração

**Documento de Requisitos**

**Documento de escopo**

**Proposta técnica**

**Plano geral de gestão de Configuração**

Definir um Padrão para nomeação dos itens de configuração

Definir uma pasta

Como vamos nomear o documento de requisitos e indicar o directorio

Padrão de nomenclatura

Situações comuns durante o desenvolvimento de software

1. Essa mudança não deu certo, vamos voltar com estava antes
2. O que mudou nessa rotina? Quando foi essa mudança? Quem é o responsável ?
3. Esse defeito já tinha sido corrigido, como reapareceu?
4. A nova função que implementei ontem era para estar aqui mas sumiu
5. Preciso daquela versão que está no cliente. Cade os arquivos?
6. Como fazer os testes sem parar o desenvolvimento?

O controle de versão precisa responder:

- o que mudou?

- Quem mudou?

- Quando mudou?

Mecanismos de controle de Concorrencia

- travamento;

-Mesclagem de arquivos;

- Ramos individuais.

Controle de versão é uma das ferramentas mais importantes do desenvolvimento de software

Serve para:

* Manter historico do projecto;
* Controlr a concorrencia de ediçao (Trabalho)
* Mantrer variações do projecto

As ferramentas mais interessantes actualmente de controle de versão sao Subversion, Git e Mercurial.

e você ainda não escolheu uma, ficará feliz em saber que existe sim uma solução simples e eficaz dependendo do perfil da equipe e das necessidades do projeto. A surpresa está em saber que a melhor opção certamente será o [Subversion](https://subversion.apache.org/) ou o [Mercurial](http://mercurial.selenic.com/), mas dificilmente o [Git](http://git-scm.com/). Vamos analisar os porquês neste artigo.

Critério de escolha de ferramenta de controle de versão

<https://blog.pronus.io/posts/qual-a-melhor-ferramenta-de-controle-de-versao-subversion-git-ou-mercurial/>

Link para a download do Mercuri

<https://tortoisehg.bitbucket.io/download/index.html>