

Plano de uso do controle de versões e controle de modificações no projeto	Date: <07 /12/2018>

Alocação de Espaços

Plano de uso do controle de versões e controle de modificações no projeto.

1. Versionamento

Para o versionamento foi adotado o uso do sistema de controle de versões distribuído, Git, utilizando os repositórios da plataforma GitHub.

O uso de um controle de versão distribuído se deu por conta da própria estratégia de desenvolvimento do projeto (Consultar Plano de Projeto), dessa forma, a equipe pode trabalhar de forma repartida, sem a necessidade (e perigo) de um único repositório se comunicando com as várias Áreas de Trabalho.

A cada alteração realizada por um membro da equipe, será feito um *commit* dessa modificação, gerando uma nova revisão no repositório. A identificação dessas revisões (para não haver conflito de numeração por um ou mais membros da equipe) será feita utilizando uma hash SHA-1, gerando um identificador único e específico para cada revisão. Cada revisão deve ser verificada e validada por um membro da equipe, diferente do que desenvolveu essa revisão.

Revisões concorrentes (quando membros diferentes estiverem trabalhando no mesmo ponto) criaram *merges*, que também devem ser verificados e validados por outro membro da equipe, diferente dos responsáveis por essa revisão.

Revisões que não foram validadas com o estado de “Aceito”, devem conter uma sinalização explicitando o por que não foi aceita, e ser excluída da linha principal de versionamento, sendo direcionada para o desenvolvedor, a fim de corrigir ou realmente excluir essa revisão.

Versões serão no formato “RESERVA_X.Y”, onde X corresponde a versão principal e Y a versão secundária.

Versão principal é aquela a partir do primeiro release do projeto, e as próximas serão incrementadas sequencialmente, exemplo V1.1 → V2.0.

Uma nova versão principal é criada a cada grande alteração e atualização de funcionalidades consideradas com nível de prioridade acima de Alto (vide formulário “Avaliação de mudança” → <https://goo.gl/forms/fvk4nkEHlrXvY3Ur1>), após a alteração/atualização ser implementada (vide formulário “Implementação de mudança” → <https://goo.gl/forms/i2aKuSaeBNYJ2Jey1>) vai surgir um novo release, e consequentemente, o número da versão principal será incrementado.

Uma versão secundária é criada a cada alteração/atualização de funcionalidades consideradas com o nível de prioridade de “Médio” para baixo (OBS.: Alterações de nível médio, dependendo da avaliação e impacto também podem gerar uma versão principal). após a alteração/atualização ser implementada e avaliada, o número de versão secundária será incrementado.

Podemos ter versões específicas para clientes específicos, seja por mais diversos motivos, manter a compatibilidade com Hardware, fluidez do sistema, ou até mesmo a remoção/alteração de uma funcionalidade essencial para certos clientes.

Plano de uso do controle de versões e controle de modificações no projeto	Date: <07 /12/2018>

1.1 Estratégia

CodeLines serão versionadas por tags no commit e serão separadas por repositórios. Representarão cada componente do sistema sendo a priori: Um para servidor, uma aplicação web e uma para documentos. Sendo assim, para criar uma BaseLine será criada uma nova Ramificação que representará uma versão do projeto com todas as CodeLines na versão necessária para aquela função. Um release ocorrerá toda vez que uma BaseLine for suficiente para entregar ao Cliente.

1.1.1 CodeLines

Code lines serão versionadas por tags nos seus commits e serão separadas por diretórios. Tags terão o seguinte padrão NOMEREPOSITORIO-X-Y (Exemplo de tag: SERVER-1-1).

1.1.2 BaseLines

BaseLines se darão por Branches e seu versionamento se dará pelo nome da Branch. (Exemplo: V-1.0). O surgimento de baselines deve ser discutido com o cliente, para, a partir das necessidades iniciais, estabelecermos a base de implementação que será o guia inicial do desenvolvimento. Novas baselines podem surgir no meio de uma implementação seja uma entrega de funcionalidades específicas para um cliente, ou até mesmo a alteração parcial/completa de uma funcionalidade.

1.1.3 Release

Uma Release se dará toda vez que ocorrer uma entrega programada com cliente, ou a partir de alterações/atualizações de funcionalidades. Será versionado como um Branch com o nome Release-X.Y . (Exemplo: Release-1.1).

1.1.4 Comentários em commits

Cada comentário em um commit será usado para descrever as mudanças feitas naquele commit. E outras informações relevantes para o mesmo.

2. Mudanças

Para a realização de mudanças no projeto deve-se verificar e preencher o seguinte formulário:LINK para assim a gerência do projeto tomar decisão de aplicação ou não da mudança.

Para a mudança deve ser estudado caso a caso, o que foi preenchido no formulário. toda mudança solicitada no formulário deverá constar no git em documentos.