

Atividade avaliativa: Plano de gerenciamento de configuração.

Deverá ser configurado o Redmine para controle de atividades relacionadas ao projeto, deverá ser configurado um projeto principal com dois subprojetos, um subprojeto para gerenciamento de tarefas relacionadas às implementações de funcionalidades , desenvolvimentos em geral e correção de bugs, outro subprojeto deverá ser configurado para gerenciamento de releases. No subprojeto dedicado ao desenvolvimentos serão configurados os usuários que fazem parte da equipe de desenvolvimentos, stakeholders e testers, para os tipos de tarefa selecionáveis ao criar uma nova tarefa, serão adicionados os seguintes itens: *Nova funcionalidade*, que será dedicada a criação de uma nova tela do software por exemplo. *Implementação*, quando uma nova função ou pequena adição de funcionalidade for adicionada a uma tela ou módulo já existente, como um campo de formulário por exemplo. *Correção de bug*, quando uma tarefa for aberta para corrigir um bug. Os status para controle de tarefas deverão ser mudados a cada interação, os tipos de status serão os seguintes: *Nova*, quando for uma tarefa recém criada, *Em desenvolvimento* , quando um desenvolvedor atribuir seu nome a tarefa e começar a desenvolvê-la, *Em teste*, quando o usuário terminar de desenvolvê-la deverá mudar seu status para ser testada pelo usuário tester. *Devolvida*, quando uma tarefa reprovar no processo de teste e validação, deverá ser devolvida ao desenvolvedor. *Concluída*, após ser validada e testada deverá ser mudada para este status. Além dos campos de tipo e status de tarefa deverá ser configurado um campo definindo a quantidade de pontos da tarefa, para uso de métricas, os seguintes itens devem ser configurados para este campo, *1 Ponto*: tarefas que não ultrapasse duas horas de desenvolvimento, *2 pontos*: tarefas entre duas e 4 horas, *3 Pontos*: tarefas entre 4 e 6 horas, e *4 Pontos*: para tarefas que levem em média 8 horas. Ainda se tratando de tarefas, uma configuração de transição de status de tarefa deverá ser adicionada, tarefas novas poderão mudar apenas para o status de desenvolvimento, tarefas em desenvolvimento poderão ser mudadas apenas para teste, e tarefas com status teste poderão ser mudadas apenas para os status de devolvida e concluída. Outro item importante a ser configurado no Redmine é a integração com o repositório, que deverá possibilitar a integração entre os commits e as tarefas.

Para controle de versão deverá ser usado o Github, as seguintes branches iniciais deverão ser criadas: *Produção*, *Homologação*, *Release*, *Desenvolvimento*, *Hotfix*, no início de cada tarefa uma nova branch deve ser gerada a partir da branch *Desenvolvimento* onde está contido os códigos do próximo deploy, cada feature desenvolvida a partir da branch *Desenvolvimento*, deverá retornar a ela. Na branch de produção ficará as implementações testadas e aprovadas, onde apenas um usuário com acesso específico terá controle sobre ela. A branch de *Homologação* ficará atualizada com as tarefas onde os testers terão acesso para realizarem os testes. A branch de *release* ficarão as últimas releases já testadas e aprovadas, na branch *Hotfix*, serão comitadas as tarefas de correção de bugs, esta branch é puxada a partir da *Produção* e mescladas com a *Desenvolvimento* e a *Produção*. Deverá ser instruído aos desenvolvedores para que nas suas mensagens de commit sejam incluso os códigos das tarefas relacionadas , exemplo #0201, para serem rastreadas e acompanhadas a partir do Redmine.

Para controle de documentação e artefatos deverá ser usado o Google docs, deverão ser criados arquivos para requisitos do projeto, documento descritivo para cada nova release gerada contendo informações das novas funcionalidades e descrições, os manuais e instruções do produto final, além de outros artefatos relacionados a mockups, diagramas e instruções.

O endereço dos documentos devem ficar na descrição do projeto que foi criado no redmine, para fácil localização.