

Gerenciamento de Configuração de Software

Prof. Michael da Costa Móra, baseado em proposta elaborada pelo Prof. Daniel Callegari

Trabalho 1 – Prática com Git em Times

O trabalho consiste em observar os requisitos do trabalho, implementar uma solução seguindo os conceitos vistos em aula e produzir um relatório final.

O trabalho deverá ser realizado em times compostos por 4 à 6 membros, nem mais nem menos.

Atividades

Cada time deverá:

1. Cada grupo deve criar um único repositório no GitHub. O repositório deve ser público e seu endereço deve ser copiado para o relatório.
2. Definir o seu processo de *branching* (fluxo de trabalho), com base no Gitflow, estudado em aula.
3. Descrever as regras do fluxo de trabalho no relatório e segui-las ao longo da implementação.
 - a. O professor verificará a participação de cada membro do grupo através do log do git no repositório do time. Cada aluno deverá participar de pelo menos duas *features* ou correções de *bug*, com uma contribuição substancial.
 - b. As submissões ao repositório no GitHub deverão ocorrer com frequência, para que seja possível acompanhar o desenvolvimento do trabalho;
 - c. Os movimentos no repositório deverão acontecer até no máximo a data de entrega. Serão desconsiderados os trabalhos que tiverem movimentos depois da data de entrega.
4. Desenvolver o software a partir do repositório inicial criado de acordo com os requisitos e com o fluxo de trabalho definido pelo time.
5. Entregar um relatório via Moodle até a data limite. O relatório deverá conter:
 - a. Folha de rosto com nomes completos dos componentes do time e seus nomes de usuário no GitHub.
 - b. Link para o repositório público no GitHub.
 - c. Descrição clara e detalhada do fluxo de trabalho adotado.
 - d. Demonstração de que o time seguiu o fluxo definido. Deve conter capturas de tela e descrições que deixem isto claro. Sugere-se, por exemplo, capturar o Network graph do GitHub: Insights/Network.
 - e. Conclusão do grupo contendo reflexões sobre as dificuldades encontradas, como foram superadas e quais as lições aprendidas.

Critérios de Avaliação

Avaliação: 30% da nota é resultado do esforço individual e 70% da nota advém do resultado do grupo. Cada aluno deverá garantir que seu nome/usuário conste (em bom número de ocorrências) nos *commits* do histórico do *git*, confirmando a sua participação no processo de desenvolvimento, seguindo o fluxo de trabalho definido pelo grupo.

Dica: utilize o comando abaixo para uma métrica básica por usuário:

```
git shortlog -s -n --all --no-merges
```

Requisitos Gerais do Sistema a Desenvolver

1. Deseja-se desenvolver um **sistema de controle de aquisições** para uma empresa.
2. O sistema deverá ser implementado em **Java Console** /ou/ **Web Puro** (apenas HTML+CSS+Javascript). Não utilizar frameworks ou outras dependências. A ideia é ter a base de código o mais simples possível. Cuidar para que **todos** os membros do time conheçam a(s) linguagem(s) escolhida(s).
3. Não implementar um mecanismo de persistência de dados. O sistema deverá manter dados apenas em memória durante a execução.
4. Não implementar um mecanismo de login. No entanto, deverá ser possível identificar/alterar o usuário que está usando o sistema no momento. Para cada operador, deve-se saber o nome e as suas iniciais.
5. O sistema deverá iniciar com dados já preenchidos (em bom número e de boa qualidade), de forma a facilitar os testes.

Detalhamento

1. O sistema deverá permitir escolher o usuário atual a qualquer momento. Cada usuário tem um identificador, um nome e o seu tipo: funcionário ou administrador.
2. A empresa possui diversos departamentos (ex. Financeiro, RH, Engenharia, Manutenção, etc). Cada departamento tem um valor máximo permitido por pedido. [o sistema deverá iniciar com dados já cadastrados de pelo menos 5 departamentos; não é necessário implementar funções de inclusão nem exclusão de departamentos].
3. Cada funcionários está alocado em um destes departamentos. [o sistema deverá iniciar com dados já cadastrados de pelo menos 15 funcionários, associados aos seus departamentos.]
4. O sistema deverá permitir a um funcionário ou administrador registrar um novo pedido de aquisição.
5. Cada pedido de aquisição deverá conter:
 - 5.1. O funcionário solicitante
 - 5.2. O departamento solicitante (deve ser o mesmo do funcionário no momento do cadastro)
 - 5.3. A data do pedido
 - 5.4. A data de conclusão do pedido (preenchida quando os itens solicitados são entregues)
 - 5.5. O status do pedido (aberto, aprovado, reprovado)
 - 5.6. Lista de itens a serem adquiridos:
 - 5.6.1. Descrição do item, valor unitário, quantidade, total do item
 - 5.7. Valor total do pedido (deve ser igual ou inferior ao limite por pedido do respectivo departamento).
 - 5.8. Somente um administrador pode avaliar um pedido que esteja aberto para aprová-lo ou rejeitá-lo. Permitir a um administrador
 - 5.8.1. Listar todos os pedidos entre duas datas
 - 5.8.2. Buscar pedidos por funcionário solicitante
 - 5.8.3. Buscar pedidos pela descrição de um item
 - 5.8.4. Visualizar os detalhes de um pedido para aprová-lo ou rejeitá-lo.
6. Não deverá ser possível reabrir um pedido depois de marcado como aprovado/reprovado/concluído.
7. Um pedido ainda aberto poderá ser excluído, mas somente pelo funcionário que o criou.
8. O sistema deverá permitir a um administrador ver estatísticas gerais contendo:
 - 8.1. Número de pedidos total, divididos entre aprovados e reprovados (com percentuais).
 - 8.2. Número de pedidos nos últimos 30 dias e seu valor médio.
 - 8.3. Valor total de cada categoria nos últimos 30 dias.
 - 8.4. Detalhes do pedido de aquisição de maior valor ainda aberto.

Bom trabalho!