

## Orientações de Uso do GitHub

- 1) **Criar três milestones equivalentes as três releases:**
  - a. **As releases ficaram agendadas com as datas:**  
Primeira Release - 18/02 a 22/02  
Segunda Release - 08/04 a 12/04  
Terceira Release - 06/05 a 10/05
- 2) **Criar label para as sprints: Uma sprint equivale a uma semana de desenvolvimento. O padrão de nomenclatura deverá se mes\_diaInicial\_diaFinal:**  
Ex: fev\_02\_08 (perceba que ela sempre começa no sábado e encerra na sexta)
- 3) **Criar label para as disciplinas.**
  - a. BDII
  - b. ESW
  - c. PI
  - d. LPII
  - e. WEB
- 4) **Utilizar o label de tarefas já disponibilizado pelo GITHUB.**
  - a. bug quando for a correção de uma falha
  - b. enhancement quando for uma tarefa nova
  - c. wont fix quando não for feita a correção
  - d. duplicate quando for uma tarefa repetida que não será feita
  - e. question quando for uma issue de discussão do projeto
  - f. invalid quando ela não for ser relacionada ao projeto)
- 5) **Para criar uma Issue**
  - a. O nome da tarefa deve seguir o padrão: UCXXX – Nome da Tarefa. XXX deve ser trocado pelo numero do caso de uso. UC é abreviação para USE CASE (caso de Uso).
  - b. Lembre-se de dividir as tarefas em tarefas pequenas para que possam ser estimadas. Ex de tarefas:
    - i. UC001 – Criar DAO insert , UC001 – Criar GUI, UC001 – Criar Validação de Tela, UC001 – Criar BD do Cliente, UC001 – Criar Classe Controle do Cliente , UC001 – Criar teste do Cliente, entre outros.
  - c. Todo o comentário deve ser feito na ISSUE
  - d. **Vocês podem comentar inclusive uma parte específica do código. Em caso de dúvida utilizem este recurso para discutir soluções de implementação e documentar soluções de códigos utilizadas. EVITE COMENTAR O CÓDIGO**
  - e. Uma issue PODE ter mais de um commit, e pode commitar mais de uma classe, sem problema.
  - f. **NÃO ESQUEÇA DE ASSOCIAR A ISSUE A UM DESENVOLVEDOR**

## **6) PULL REQUEST**

- a. Quando ocorrer um BUG (falha), e for necessário realizar uma correção, utilize o recurso de enviar o pull request, discutir a solução para a falha, e integrar a uma issue.
  - i. NÃO SE ESQUEÇA DE RELACIONAR O PULL REQUEST COM A ISSUE – ISTO É POSSÍVEL E DEVE SER FEITO

## **7) WIKI – Documentação das atividades DE CADA DESENVOLVEDOR**

- a. Cada desenvolvedor deve criar sua WIKI no projeto
- b. Na sua WIKI, durante as férias descreva:
  - i. O que você estudou/fez, o que você aprendeu, quais são suas dificuldades
- c. Após o retorno das aulas vamos simular o Daily Meeting da seguinte forma: Todas os Sábados, antes de começar a nova SPRINT, descreva na sua WIKI registrando a Data atual, e respondendo as seguintes questões:
  - i. O que eu fiz esta semana? O que eu vou fazer semana que vem? O que eu aprendi esta semana? Quais foram minhas dificuldades? Como resolvi minhas dificuldades

## **8) WIKI – Documentação das atividades DO PROJETO**

- a. Existe uma página HOME do projeto. Nesta página vocês devem:
  - i. Documentar os Gráficos do SCRUM que discutimos na sala de aula
  - ii. Documentar as conversas de video e/ou chat feitas externamente
  - iii. Qualquer informação relevante do projeto
  - iv. Documentação do projeto
  - v. Decisões de projeto
  - vi. Documentar as métricas ágeis / e quadros discutidos e vistos na sala de aula (exemplo: Comparação de linhas de código / testes)

### **Alguns lembretes.**

- Uma *Issue* é igual a uma Tarefa;
- Cada commit - deve conter breve descrição, tag da release, e associar a issue (ver com o prof. Reginaldo em projeto Integrador como será o padrão da descrição, e como associar a uma issue – bem como fechá-la automaticamente)
- Para mencionar um desenvolvedor do projeto em uma mensagem postada em uma issue, basta utilizar @LogindoDesenvolvedor.
- Para mencionar uma issue basta colocar #numeroDaIssue
- Toda comunicação e discussão sobre as tarefas DEVEM SER FEITAS no GITHUB. Qualquer comunicação EXTRA (ex uma chamada de video para discutir o projeto remotamente, deve ser gravada e postada na pagina HOME do projeto que pode ser editada no WIKI)

**Adicione os professores no projeto do GitHub:**

igorwiese – Prof. Igor Wiese – Engenharia de Software

campiolo – Prof. Rodrigo Campiolo – Web

Igvalent – Prof. Lucio Valente – BDII e LPII

reginaldo-re - Prof. Reginaldo Ré – Projeto Integrador I

juanfms – Aluno Juan Fonseca - que vai auxiliar durante a disciplina discutindo testes, métricas.

**Para a estimativa das tarefas:**

Durante as férias vocês vão começar a estimar as tarefas. Lembrando que ficou combinado uma carga de trabalho de 10 horas de trabalho por semana + as horas da disciplina de projeto integrador, totalizando 15 horas de dedicação semanal POR INTEGRANTE DO PROJETO. Portanto vocês terão . Também foi combinado que vocês não poderão exceder 330 horas de ESTIMATIVA do projeto, que é o valor obtido somando as 30 horas da equipe por todas as semanas do semestre.

Vocês devem construir uma tabela como a abaixo e postar na WIKI HOME do projeto logo após concluírem o documento de casos de uso na disciplina de Projeto Integrador.

Tarefa	Estimativa #Dev1	Estimativa #Dev2	Média da Estimativa
<b>UC001 – Cliente</b>			
UC001 – Criar Dao Insert do Cliente			
UC001 – Criar BD do Cliente			
UC001 – Classe Controle Cliente			
UC001 – Criar Validação GUI			
UC001 – Criar teste Cliente			
<b>UC002 – XXXXX</b>			
UCXXX – ZZZZZ			
UCXXX –YYYYYYY			
UCXXX – HHHHHH			
UCXXX– ZZZZZZ			
UCXXX– IIIIIIIIIII			

Lembre-se de fazer a estimativa individual, utilizando a escala de 30 minutos, 1 hora, 2 horas e 4 horas. Em caso de divergências na estimativa, discutam para chegar em um consenso.

Qualquer dúvida no processo de estimativa, entrem em contato.