Qual é o repositório do seu grupo?

Nosso repositório está localizado no Github de forma privada.

Como está estruturado o seu repositório?

1. **Quais artefatos do projeto (código-fonte, documentação, scripts de build, etc.) serão os itens de configuração?.**

   Nossa política de gerência de configuração preza pela adição de grande parte de nossa documentação (Formulários, diagramas, tabela de dependências e durações, script SQL, projeto java spring, etc.). Porém, não é feita a inclusão de nossos Mockups, que devem permanecer alocados apenas no Figma, e o quadro Kanban que foi produzido no Trello.

2.

   2.1 Quando abrir uma Branch (por tarefa ou por desenvolvedor, ou outro critério)?

   Por tarefa é o nosso critério principal.

   2.2 A nomenclatura dos branches?

   Elas seguem nomes relacionados às tarefas, por exemplo a branch ‘coding’ é destinada ao desenvolvimento do código.

   2.3 Todos os branches devem ser baseados na (principal)?

   Sim, adiciona-se que nossa branch principal é chamada ‘main’.

3.

   3.1 Quando realiza, e quem realiza (quantos pessoas realizam?

   O merge é realizado assim que o trabalho em uma branch é finalizado e testado. Os realizadores do merge são os próprios membros responsáveis pela tarefa executada nesta outra branch, caso necessário, outra pessoa pode também ser chamada.

   3.2 Como tratam os Conflitos de merge?

   Os autores das alterações conflitantes se reúnem para discutir o que ocasionou o conflito, quais correções são necessárias para corrigir o conflito e se a versão final após tais atualizações realmente é adequada e funciona.

4. **Inserem alguma mensagem no commit para descrever a modificação e referenciando a tarefa??? qual é o padrão**

   Sim, as mensagens seguem um formato padrão básico, sendo “Marcador: corpo descritivo”. Nesse sentido, o marcador se refere ao que de fato consiste aquele commit sobre os itens de configuração, podendo ser uma Adição, Exclusão, Correção, Organização e Teste.

Assim, um exemplo de mensagem de commit seria “Organização: reorganizando os arquivos em pastas de acordo com a etapa a que se referem”. Além disso, outro exemplo seria “Teste: esse commit constitui-se em um teste de primeiro uso da ferramenta git”.

5.

5.1 As versões do software seguirão o padrão de versionamento semântico?

Sim, elas seguirão a ideia do versionamento semântico, especificando modificações Major (para incompatibilidade com versões anteriores), Minor (para o lançamento de novas funcionalidades) e Patch (para correção de bugs e erros).

5.2 Quando é lançada uma release? quem aprova a release?

Uma release só será lançada quando todas as funcionalidades propostas até então forem implementadas, toda a documentação necessária esteja pronta e testes tenham sido executados, a fim de detectar bugs críticos. A release será aprovada por algum membro, após o sim de todos do grupo.

5.3 Como você identifica os elementos release?

O versionamento semântico será implementado por meio de tags, além disso cada release contará com uma descrição própria que explica o motivo de ter sido lançada.

6.

- O repositório será organizado por (mainline), e .
- : Contém o código de desenvolvimento ativo.
- : Criados para grandes evoluções ou correções específicas.
- : Usados para identificar releases específicas.

7.

- Alterações nos itens de configuração devem ser iniciadas por uma aprovada pelo Comitê de Controle de Configuração.
- Cada mudança passará por uma , que inclui a análise dos custos, cronograma e esforço.
- O plano de mudanças será documentado e revisado antes da implementação.