**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL**

ARTHUR KUNZLER (*arthurkunzler), EDUARDO BALLICO (EduardoBallico), GUILHERME DORNELLES (GuilhermeDornelles), GUILHERME* SANTOS (*guilhermxsantos*), ISMAEL FREITAS (*ismaelfvargas*), JOÃO VITTOR CAUMO (*joaovittorcaumo*), LEANDRO NÓBREGA (*29102001*), LUCAS ANDRADE (*AndradeHd*), MATHEUS STRECK (*MatheusStreck*) E VINICIUS BOFF (*VarBoff*)

**TRABALHO 1 - PRÁTICA COM GIT EM TIMES**

**PORTO ALEGRE**

**2021**

* Link para o Repositório

Segue abaixo o link para o repositório público no GitHub do grupo: <https://github.com/EduardoBallico/T1_GCS_AeS>

* Descrição do Fluxo de Trabalho

Após os ensinamentos da aula do dia 17/09/2021, sobre controle de versão, foi de decisão do grupo que o fluxo de trabalho escolhido seria a OneFlow, desenvolvida por Adam Ruka, cujo objetivo é simplificar as operações do dia a dia fazendo com que o projeto se torne mais limpo e de fácil leitura. Com esta ideia em mente adaptamos o fluxo desenvolvendo o programa inteiro na branch “develop” e após o término foi feito um merge para a branch “main”, com exceção de duas outras branches que por conta da dificuldade e erros que estávamos obtendo, foram criadas para evitarmos conflitos.

* Demonstração do Fluxo de Trabalho

Abaixo está uma imagem comprovando que o fluxo de OneFlow, escolhido pelo grupo, foi seguido com as adaptações propostas.

Uma imagem contendo objeto, relógio, bola, medidor

Descrição gerada automaticamente

Tela de celular

Descrição gerada automaticamente com confiança média

Uma imagem contendo Interface gráfica do usuário

Descrição gerada automaticamente

A primeira linha mostra a branch “main”, logo abaixo a reta azul equivale a branch “develop” na qual o trabalho inteiro foi desenvolvido e por último as branches de cores verde e roxo que foram utilizadas para testes e criações de features, visando minimizar o conflito e facilitar o trabalho do grupo.

Também foram utilizados comandos como git checkout, git push origin --delete remoteBranchName, git branch e git shortlog -s -n --all --no-merges para conferir o número de commits de cada integrante, como na imagem abaixo.

Texto

Descrição gerada automaticamente

* Tabela dos Requisitos

**R1** - Eu como gerente, gostaria que cada usuário tenha um identificador, um nome e seu tipo: funcionário ou administrador;

**R2** - Eu como usuário, gostaria que o sistema permita escolher qual o usuário atual a qualquer momento;

**R3** - Eu como gerente, quero que no sistema tenha diversos departamentos. 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);

**R4** - Eu como gerente, quero que cada funcionários esteja alocado em um destes departamentos. (o sistema deverá iniciar com dados já cadastrados de pelo menos 15 funcionários, associados aos seus departamentos.);

**R5** - Eu como gerente, quero que o sistema permita um funcionário ou administrador registrar um novo pedido de aquisição;

**R6** - Eu como usuário, quero que cada pedido de aquisição contenha:

**R6.1.** - O funcionário solicitante;

**R6.2.** - O departamento solicitante (deve ser o mesmo do funcionário no momento do cadastro);

**R6.3.** - A data do pedido;

**R6.4.** - A data de conclusão do pedido (preenchida quando os itens solicitados são entregues);

**R6.5.** - O status do pedido (aberto, aprovado, reprovado);

**R6.6.** - Lista de itens a serem adquiridos;

**R6.6.1** - Descrição do item, valor unitário, quantidade, total do item;

**R6.7.** - Valor total do pedido (deve ser igual ou inferior ao limite por pedido do respectivo departamento);

**R6.8.** - Somente um administrador pode avaliar um pedido que esteja aberto para aprová-lo ou rejeitá-lo. Permitir a um administrador;

**R6.8.1.** - Listar todos os pedidos entre duas datas;

**R6.8.2.** - Buscar pedidos por funcionário solicitante;

**R6.8.3.** - Buscar pedidos pela descrição de um item;

**R6.8.4.** - Visualizar os detalhes de um pedido para aprová-lo ou rejeitá-lo;

**R7** - Eu como gerente, quero que no sistema não seja possível reabrir um pedido depois de marcado como aprovado/reprovado/concluído;

**R8** - Eu como gerente, quero que um pedido ainda aberto poderá ser excluído, mas somente pelo funcionário que o criou;

**R9 -** Eu como gerente, gostaria que o sistema deverá permitir que um administrador tenha a opção de ver estatísticas gerais contendo;

**R9.1.** - Número de pedidos total, divididos entre aprovados e reprovados (com percentuais).

**R9.2.** - Número de pedidos nos últimos 30 dias e seu valor médio.

**R9.3.** - Valor total de cada categoria nos últimos 30 dias.

**R9.4.** - Detalhes do pedido de aquisição de maior valor ainda aberto.

* Tabela dos Componentes

* Conclusão

As principais dificuldades encontradas pelo grupo foram, o desenvolvimento do software junto com a organização e a separação de tarefas sempre tendo em mente o cuidado para que nenhum outro colega estivesse trabalhando ao mesmo tempo no programa por conta do fluxo de trabalho escolhido, que por ser em apenas uma branch não seria possível fazer “merge” antes de passar para a main. Com isso conseguimos aprimorar a nossa comunicação como uma equipe e a mantendo ativa para minimizar os erros e uma delimitação de responsabilidades para não sobrecarregar os membros da equipe.

É visto pelo grupo que o programa obedece aos requisitos solicitados, mas sem grandes avanços além disso, acreditando-se por parte do envolvidos uma nota 9,5 estaria de acordo. No entanto, é notável pelos integrantes que por parte de certos participantes desde o início do projeto não foi dado o máximo para a realização do trabalho, achando que pela parte de comunicação e envolvimento seria justa uma nota 7,0.