Laboratórios de Programação

Relatório - Quadro de kanban

FCUP 2020/2021

Filipe Colla David - up201810097

João Gabriel Ferreira Alves - up201810087

Índice

Introdução

Um quadro Kanban é uma ferramenta que pode ser usada para gerir qualquer tipo de projeto, seja este pessoal ou empresarial.
Um quadro Kanban representa graficamente o estado do projeto em diferentes etapas, usando cartões para representar cada tarefa. O quadro está dividido em 3 colunas (ToDo, Doing, Done), em que cada uma representa uma etapa. Os cartões são movidos da esquerda para a direita para mostrar o progresso e ajudar a coordenar a execução do projeto.

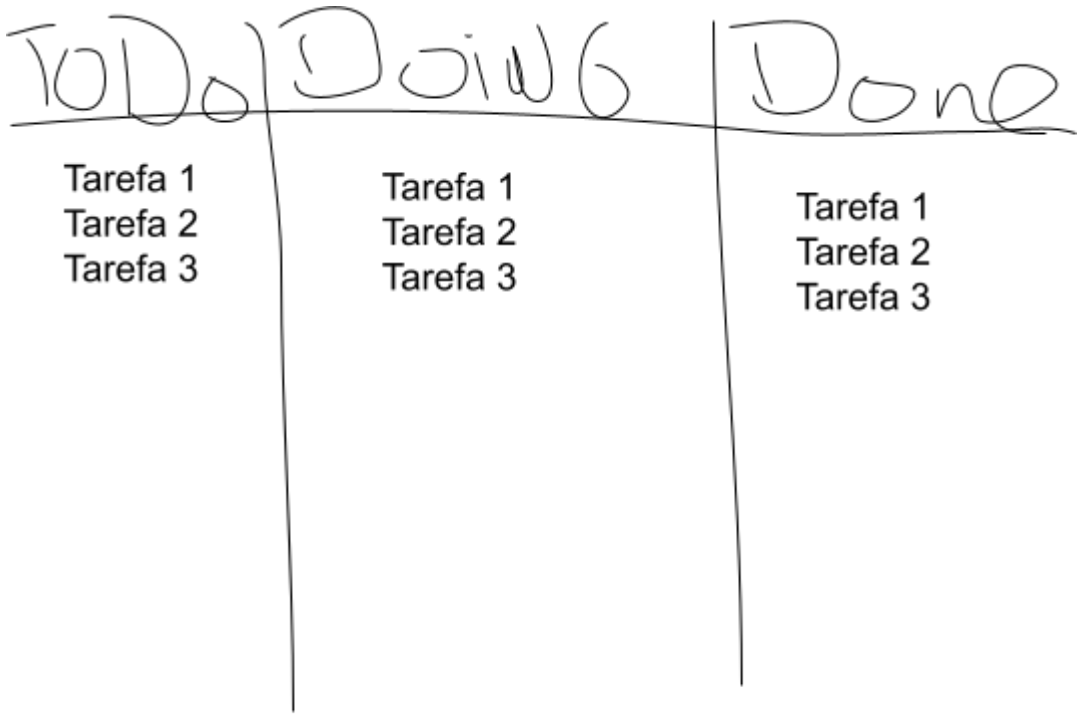

Figura 1. Sketch feito pelo grupo de um quadro de Kanban

Estruturas de dados utilizadas

Cada tarefa é representada por um struct *TASK* constituído por:

-2 *char[]* (*name,owner*), um para o nome da tarefa, outro para o dono

- 2 *int* (*id,priority*), um para o id, outro para a prioridade da tarefa

- 3 structs do tipo *DATA*

O struct *DATA* é constituído por 3 *int*, um para o dia, um para o mês e o último para o ano.

Por fim, temos um struct LIST, que representa uma lista ligada constituída por uma *TASK* e um nó *next* que aponta para o nó seguinte da lista.

No início do programa são inicializadas 3 listas, e cada lista é inicializada com uma tarefa chamada *t_head* que tem os elementos inicializados a valores inválidos (ie. name e owner = "\0", id = -1,priority = -1, e as datas inicializadas a (0,0,0)).

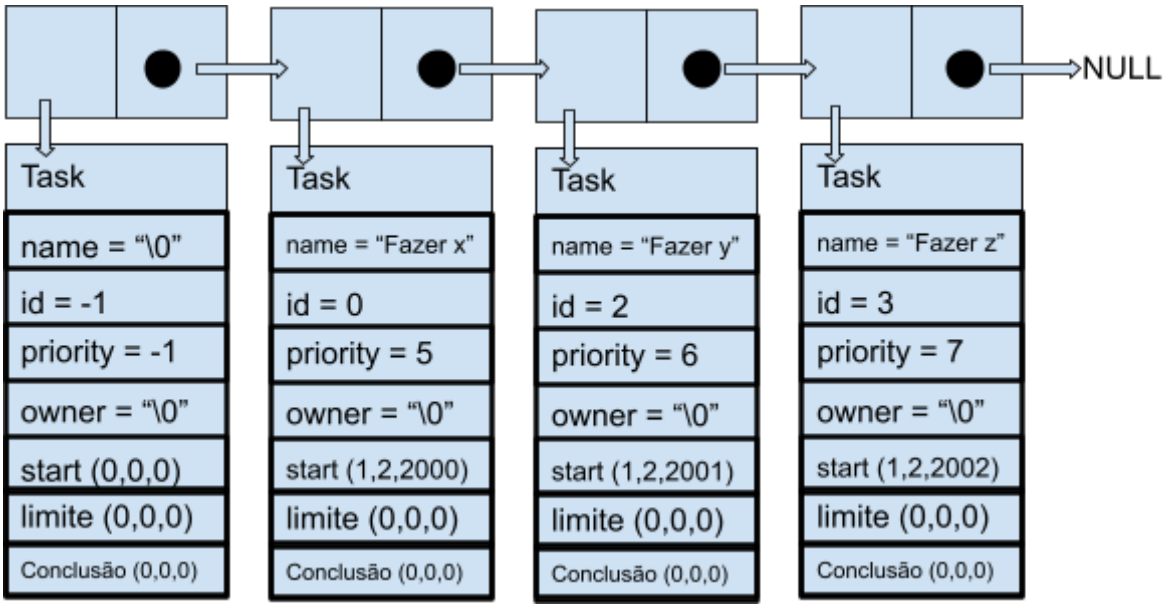

Figura 2. Representação gráfica de uma possível lista ToDo

Estrutura geral do programa e funcionamento

O programa começa por inicializar as 3 listas como referido anteriormente, e segue o seguinte diagrama:

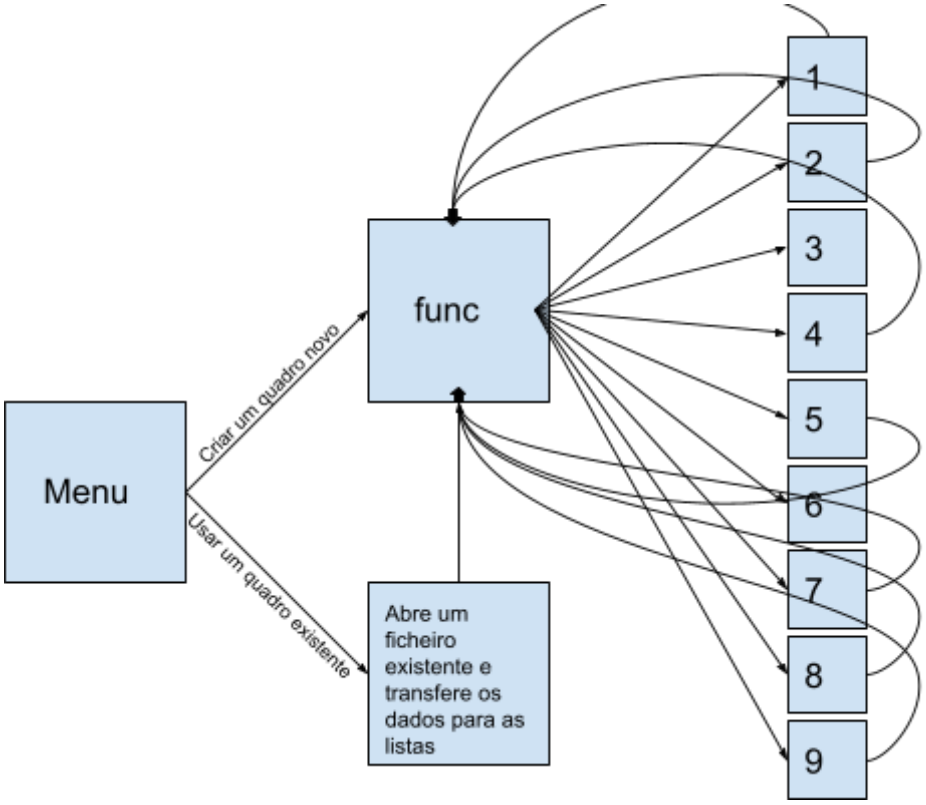

Figura 3. Diagrama do fluxo do programa

Após seguir para a função func, é pedido ao utilizador que introduza um carácter de 1 a 9, aceitando apenas essas opções, *prev* para voltar para o menu, *q* para sair e *h* para um texto de ajuda. Sempre que uma tarefa é inserida ou atualizada o programa grava num ficheiro de texto. Todas as funções que se seguem têm proteção de input, só aceitam input que seja válido para cada situação, *prev* para voltar para trás, *h* para um texto de ajuda e *q* para sair.

1. **addToDoInterface()**: É pedido ao utilizador que insira o nome de um tarefa que quer adicionar ao quadro para a coluna toDo, sendo perguntado a seguir ao utilizador para inserir a prioridade da tarefa, sendo adicionada uma tarefa à lista ToDo com essas características. Repete o processo até que o utilizador insira *prev*, neste caso o programa volta para func, ou *q* o que leva o programa a fechar.
2. **toDo2Doing()**: É imprimido na consola a lista toDo com um índice associado a cada tarefa e é pedido ao utilizador que escolha uma tarefa

da lista imprimida para adicionar a Doing. Caso seja introduzido um valor que não existe na lista, o programa pede para inserir de novo o valor, indicando que não existe o índice que pediu, caso contrário é pedido ao utilizador que introduza o dono da tarefa, seguido da data limite de conclusão. A data tem que ser futura ou a data actual, não há possibilidade de pedir tarefas para ontem ;)

3. **doing2ToDo()**: É imprimido na consola a lista Doing, e da mesma forma que a função 2 é pedido ao utilizador que escolha um índice, de modo a adicionar uma tarefa de Doing a toDo. Quando a tarefa é escolhida corretamente, esta é removida da lista Doing, o valor de owner e data limite de conclusão são revertidos aos valores default (owner = "\0" e data limite = (0,0,0) e é adicionada a toDo.
4. **doing2Done()**: É imprimido na consola a lista Doing, e é pedido ao utilizador para escolher uma tarefa que queira passar para a lista Done, isto é, que tarefa quer terminar. Esta é removida da lista Doing e é passada para a lista Done, atualizando a data de conclusão dEnd, com a data atual do sistema.
5. **done2Doing()**: É imprimido na consola a lista Done, e é pedido ao utilizador para escolher uma tarefa que queira passar da lista Done, para a lista Doing. Quando escolhida a tarefa, os valores de data de conclusão (*dEnd*) são revertidos para os valores default (0,0,0)
6. **change_owner()**: Esta função permite ao utilizador mudar o dono de uma tarefa que se encontre em Doing imprimindo a lista Doing, para que o utilizador consiga escolher a tarefa que quer mudar o nome, esta é removida da lista, é actualizada com o novo valor, sendo inserida de novo na mesma lista.
7. **PrintTable():** Esta função imprime as tarefas de cada uma das 3 tabelas (To Do,Doing,Done). Nota: Na tabela To Do só está ordenada por prioridade.
8. **pTasksOfOwner():** Esta função pede o nome de uma pessoa responsável por tarefas e imprime todas as tarefas em que está associado.
9. **printByDate()** Esta função imprime todas as tarefas ordenadas por ordem de criação.

Sempre que o utilizador escreve *q*, o programa vai para uma função *user_exit()*, que pergunta ao utilizador se quer guardar o ficheiro, caso queira, o programa

pede ao utilizador para inserir o nome do ficheiro, guarda todo o conteúdo das listas num ficheiro txt, e acaba a sua execução. Caso contrário apaga o ficheiro temporário que é criado e actualizado ao longo da execução do programa, e acaba a sua execução.

Estrutura dos ficheiros

Os ficheiros de gravação de dados estão divididos em 3 partes:

1ª - Um inteiro que indica o número de elementos da lista ToDo (nToDo), seguido de 2 linhas, na primeira linha está contido o nome da tarefa, o id e a prioridade separados por um separador especifico, neste caso, "*_*". Na segunda linha encontra-se a data de criação da tarefa separada por espaços. Isto acontece durante nToDo vezes, ou seja, enquanto houver elementos na lista ToDo.

2ª - Um inteiro que indica o número de elementos da lista Doing (nDoing), seguido de 3 linhas, na primeira linha está contido o nome da tarefa, o dono, o id e a prioridade separados pelo mesmo separadorde 1ª, seguido de duas datas, a data de criação da tarefa e data limite. Da mesma forma que a alínea anterior, isto acontece durante nDoing vezes, enquanto houver elementos na lista Doing.

3ª - Um inteiro que indica o número de elementos da lista Done (nDone), seguido de 4 linhas, na primeira lina está contido o nome da tarefa, o dono, o id e a prioridade, separados pelo menos separador de 1ª e 2ª, seguido de 3 datas, a data de criação a data limite e a data de conclusão.

Estes dados são armazenados no ficheiro após a chamada da função save() que tem como atributo o nome do ficheiro (temp.txt, por default), que imprime como referido acima o conteúdo de cada lista.

Exemplo:

```
2
Fazer k*_*2*_*2
9 4 2021
Fazer z*_*3*_*10
9 4 2021
1
Fazer y*_*Filipe*_*1*_*10
9 4 2021
```

```
12 12 2021
1
Fazer x*_*Joao*_*0*_*5
9 4 2021
31 12 2021
9 4 2021
```