



## LABORATÓRIO DE PROGRAMAÇÃO CC2013

---

# Projeto 1 - Quadro de Kanban

---

*Realizado por:*

Rui Pacheco (up201906505)

Pedro Santos (up201907254)

## Índice

1	Esquema com as estruturas de dados utilizadas	2
2	Estrutura Geral do Programa	2
3	Estrutura dos ficheiros	5
4	Como executar o programa	5

## 1 Esquema com as estruturas de dados utilizadas

Com o intuito de auxiliar o desenvolvimento do projeto foi criada uma estrutura “Cartao” dando origem a um novo tipo de variáveis, do tipo “card”, esta armazena todos os dados necessários de uma tarefa ao longo de qualquer fase da pipeline. O id, a prioridade e a data de criação de um cartão são guardados logo quando a tarefa é inserida na lista “To Do” nas variáveis “nrid” e “prio” do tipo “int”, respetivamente e, por sua vez, a data de criação é originada automaticamente tendo em conta a data atual do momento em que a tarefa foi inserida pela primeira vez, na variável “date\_criac” do tipo “char[]”. Quando estas tarefas são movidas para a lista “Doing” é-lhes associado um prazo e uma pessoa sendo estas armazenadas nas variáveis “prazo” e “pessoa” ambas do tipo “char[]” uma vez que tanto o prazo como o nome da pessoa são passados como um conjuntos de caracteres. Por fim, quando uma tarefa é passada para a lista “Done” a mesma origina uma data de conclusão, gerada automaticamente segundo a data atual, que é armazenada na variável “date\_conc” do tipo “char[]”. Uma outra estrutura criada com o objetivo de facilitar a implementação das diferentes fases da pipeline foi a “lnode”, esta estrutura contém 2 parâmetros distintos, a “info” onde vai ser guardada toda a informação de uma tarefa pois esta é do tipo “card”, estrutura anteriormente referida, e um apontador “next”, do tipo “List”, onde vai ser guardado o endereço de memória referente ao próximo “card” da lista, caso este seja “NULL” então significa que esta seria a última tarefa da lista atual. Por fim, foi criada uma estrutura “List” que contém um apontador para a primeira tarefa da lista em questão.

## 2 Estrutura Geral do Programa

A implementação deste algoritmo encontra-se dividida em 3 partes. Primeiramente foi criada a biblioteca “Lists.h” onde são declaradas as estruturas, os seus parâmetros e todas as funções criadas com o intuito de as manipular. De seguida foi elaborado o “lists.c” onde estão as implementadas todas as funções declaradas no ficheiro “Lists.h”. Por fim é criado o ficheiro “main.c” onde foi implementado todo o código propriamente dito, código este que utiliza funções e estruturas declaradas e implementadas nos ficheiros “Lists.h” e “lists.c”, anteriormente referidos, bem como estruturas e funções predefinidas da linguagem “c” utilizada para a construção deste algoritmo.

Ao longo de todo o algoritmo existem diversas formas de controlo sobre as escolhas do utilizador, por exemplo, na grande maioria das vezes as opções que o utilizador deseja usar são controladas por “flags” e “switch cases” que indicam ao algoritmo qual a parte do código que deve correr, bem como em praticamente todos os “inputs”,

existem ciclos cuja função se baseia em verificar os “inputs” passados pelo utilizador e comparar com os possíveis “inputs” esperados pois, desta forma, previne o programa de originar “bugs”, “crashes”, ou até mesmo “loops” infinitos.

Ao iniciar o programa o utilizador depara-se imediatamente com a possibilidade de carregar dados que já estivessem armazenados tendo ainda a opção de ignorar os mesmos e, assim, começar do zero. Após isto é impresso ao utilizador o menu que contém todas as opções necessárias para a gestão das suas tabelas.

### 1. Inserir uma nova tarefa

Caso o utilizador opte por esta opção será então criada uma nova tarefa, com o próximo “id”, “id” este que é incrementado automaticamente, a prioridade passada pelo próprio utilizador, uma data de criação gerada automaticamente tendo em conta a data e hora atuais e por fim uma descrição da tarefa que será também inserida pelo utilizador e que pode contar com no máximo 17 caracteres.

### 2. Mover cartões entre as listas “To Do” e “Doing”

Esta opção possui ainda 2 opções distintas:

#### (a) Mover cartões de “To Do” para “Doing”

Com esta opção o utilizador tem a possibilidade de mover uma tarefa da lista “To Do” para a lista “Doing”, ou seja, tem a possibilidade de começar uma tarefa, tarefa esta que para ser identificada o utilizador deve passar o “id” da respetiva tarefa, como esta tarefa foi iniciada tem de possuir uma pessoa associada que será responsável pela mesma e um prazo máximo em que esta tarefa tem de estar pronta, tanto a pessoa responsável como o prazo máximo têm de ser passados pelo utilizador.

#### (b) Mover cartões de “Doing” para “To Do”

Contrariamente com esta opção o utilizador tem a possibilidade de mover uma tarefa da lista “Doing” para a lista “To Do”, ou seja, tem a possibilidade de parar uma tarefa que esteja em andamento, tarefa esta que para ser identificada o utilizador deve passar o “id” da respetiva tarefa e onde a prazo e o nome anteriormente passados pelo utilizador vão agora perder o seu efeito e, por isso, estas variáveis vão ambas tomar o valor “Não está atribuído”.

### 3. Alterar a pessoa responsável por um cartão em “Doing”

Neste caso o utilizador tem a possibilidade de alterar a pessoa responsável por uma tarefa que já esteja em andamento, ou seja, uma tarefa pertencente à lista “Doing”, deste modo a variável “pessoa” pertencente a uma tarefa, que

será identificada pelo “id” passado pelo utilizador, irá ser sobreposta pelo novo “input” passado.

4. **Fechar tarefa**

Caso o utilizador opte por esta opção é possível terminar uma tarefa, ou seja, mover uma tarefa da lista “Doing” para a lista “Done”, inserindo o “id” da tarefa desejada, neste caso será apenas gerado uma data de conclusão que vai tomar o valor da data e horas atuais.

5. **Reabrir tarefa**

Esta opção permite ao utilizador reabrir uma tarefa, isto é, mover uma tarefa da lista “Done” para a lista “To Do”, assim e tal como na maioria das opções a tarefa será identificada pelo “id” inserido pelo utilizador, também aqui tal como quando é inserida uma nova tarefa vai ser necessário que o utilizador passe uma prioridade para esta que acabou de ser inserida.

6. **Visualizar o quadro**

Esta opção permite a visualização do quadro em todas as diferentes fases de “pipeline” permitindo assim ao utilizador ver as diferentes tarefas e em que estado elas se encontram e, desta forma, permite uma melhor perceção do que terá de ser feito a seguir. Reiterar ainda que diferentes listas estão ordenadas de diferentes formas, a lista “To Do” está ordenada por prioridade e em caso de possuírem prioridades iguais ordena por data de criação sendo que a primeira é a mais antiga, a lista “Doing” aparece ordenada pelo nome das pessoas e a lista “Done” pela data de conclusão.

7. **Visualizar todas as tarefas de uma pessoa**

Esta opção permite ao utilizador visualizar todas as tarefas de uma determinada “pessoa” passado pelo próprio como “input” separadas pelas diferentes fases de pipeline, ter em conta que só são pesquisadas tarefas com esta “pessoa” associada nas listas “Doing” e “Done”, pois na lista “To Do” as tarefas ainda não possuem uma “pessoa” associada.

8. **Visualizar todas as tarefas ordenadas por data de criação**

Caso o utilizador opte por esta opção todas as tarefas são ordenadas e impressas conforme a data de criação, sendo que são ainda separadas consoante as diferentes fases de “pipeline”.

9. **Eliminar uma tarefa**

Esta opção foi adicionada embora não seja expressamente pedida, deste modo é possível eliminar uma tarefa sendo que o utilizador apenas passa o “id” que vai servir para a identificar e posteriormente para a eliminar.

10. **Visualizar o quadro com todas as informações**

Esta opção, analogamente à anterior, embora não seja expressamente pedida foi ainda adicionada com o intuito de imprimir todas os parâmetros que cada tarefa tem associada nas diferentes fases de “pipeline”.

#### 11. **Guardar alterações**

Esta opção tal como o próprio nome indica serve apenas para guardar as alterações mas continuar a correr o programa.

#### 12. **Sair**

Esta opção tem o intuito de terminar o programa podendo ainda, caso o utilizador assim o pretenda, gravar as alterações.

Tal como é possível verificar foram adicionadas algumas opções que embora não sejam necessárias completam e tornam mais versátil o programa como, por exemplo, as opções 9 e 10.

Reiterar ainda que ao longo do programa, entre muitas outras opções, o utilizador tem a possibilidade de cancelar que lhe permite abortar uma opção que esteja em funcionamento pois caso contrário poderiam surgir determinados casos que iriam obrigar o utilizador a terminar o programa abruptamente o que poderia resultar na perda de informação.

### 3 Estrutura dos ficheiros

No algoritmo estão presentes 2 funções “load\_content” e “save\_changes” que se relacionam diretamente com os ficheiros com o intuito de carregar e gravar dados respetivamente, deste modo, e caso o utilizador assim o pretenda pode carregar informações já existentes partindo do ponto onde ficou. Cada atributo da estrutura “Card” que foi utilizada para armazenar os parâmetros de cada tarefa, irá ser guardado em linhas diferentes permitindo, deste modo, carregar os diferentes parâmetros de uma forma mais fácil e precisa pois o conteúdo de cada um estará separado por um “\n”.

### 4 Como executar o programa

Como já foi referido anteriormente o algoritmo encontra-se dividido em 3 partes e, tendo em conta que “Lists.h” é uma nova biblioteca a forma de a incluir nos ficheiros será diferente e, por isso, para executar o programa antes de correr o comum “gcc” para compilar, esta nova biblioteca deverá ser gerada, para isso serão utilizados os

seguintes comandos:

- `gcc -Wall -c lists.c`
- `ar -rc liblists.a lists.o`
- `ar -t liblists.a`
- `gcc -Wall -o main main.c -L. -llists`

Após estes comandos a nova biblioteca já foi gerada e está pronta a ser utilizada bem como os ficheiros “lists.c” e “main.c” que já foram compilados e estão prontos a executar, para isto vamos utilizar o comando “./main” que irá iniciar a execução do algoritmo. Ressaltar ainda que os ficheiros de teste não podem ser utilizados através do comando “./main <NomedoFicheiro.txt” sob risco de este não funcionar corretamente, deste modo o input dever ser copiado e colado diretamente na ”shell” ou então através da interação direta com o programa.