

# Quadro de KanBan



[This Photo](#) by Unknown Author is licensed under [CC BY-SA-NC](#)

## Laboratório de Programação

Grupo:

Murilo Marotta Reis Rosa (201900689)

Márcia Silva Dias (201704466)

Tiago Jorge Moutinho Gonçalves (201905179)

## Índice

|                                                |   |
|------------------------------------------------|---|
| O que é o quadro de KanBan?                    | 3 |
| Estruturas de dados utilizadas                 | 3 |
| Estrutura geral do programa                    | 4 |
| Estrutura dos ficheiros                        | 4 |
| Breve explicação de como o programa se executa | 4 |

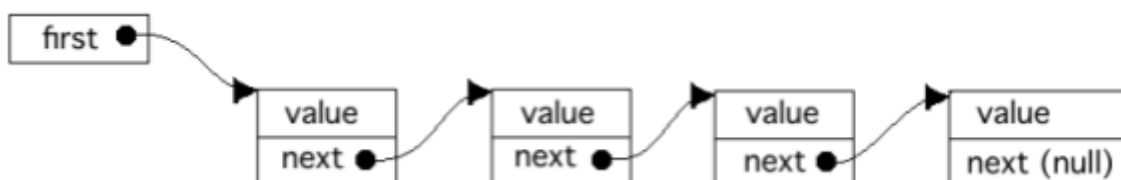
## O que é o quadro de KanBan?

O quadro de KanBan é uma ferramenta de ajuda para organização de trabalho. Desenhado para ajudar a visualizar, limitar e maximizar a eficiência no trabalho, o quadro de KanBan permite uma visão geral sobre a carga de trabalho através de cartões que descrevem as tarefas, repartindo-se por colunas, (respectivamente: "Fazer", "Em execução", "Feito"), que mostram o estado atual da tarefa e determina tempos limite para tarefas em execução.

## Estruturas de dados utilizadas

Neste trabalho o nosso grupo decidiu utilizar apenas listas ligadas. Estas estruturas contêm células que estão interligadas através de nós, ou seja, cada célula tem um nó que aponta para o endereço de memória da próxima célula. Deste modo, as células da estrutura não precisam estar em posições contíguas da memória. Isto faz com que esta estrutura seja dinâmica, pois a qualquer momento podemos colocar novas células na lista. Usamos listas ligadas porque trazem vantagens para este trabalho, podemos remover e inserir elementos sem ter que alterar as posições dos outros e não temos que definir, ao criar a lista, o número máximo de elementos que esta poderá ter, ou seja, é possível alocar a memória dinamicamente.

Sendo  $n$ , o número de elementos de uma lista ligada, no pior caso, para inserir e remover a complexidade é  $O(n)$ .



[This Photo](#) by Unknown Author is licensed under [CC BY-SA](#)

## Estrutura geral do programa

Divimos o trabalho em vários ficheiros, temos o main.c, main.h, card.c, card.h, list.c, list.h.

No ficheiro main.h temos todas as funções declaradas e no main.c temos a criação de ficheiros, temos o menu e temos as funções essenciais para interação entre as listas, os cartões e a interface com usuário.

No ficheiro card.h temos a estrutura de cada cartão e a estrutura de cada lista de cartões. No card.c temos a implementação das funções para manipulação dos cartões.

No ficheiro list.h temos apenas a declaração das funções que estão no ficheiro list.c. Aqui temos funções de ordenação de listas, remoção e inserção de elementos, criação e obtenção de campos.

## Estrutura dos ficheiros

O método de armazenamento de dados foi desenvolvido pela utilização de 3 ficheiros, o "cards.bin", o "authors.txt" e a "text.txt". Cada ficheiro guarda as informações sobre, os cartões, os autores e a descrição da tarefa respetivamente. O ficheiro "cards.bin" corresponde a um ficheiro binário onde guardamos os cartões. É através de um "id" que vamos buscar a posição inicial do cartão neste ficheiro. Temos um campo "texto" na estrutura do cartão que é onde guardamos a posição inicial da descrição no ficheiro "text.txt". O mesmo acontece com o autor, temos um campo "autor" onde guardamos a posição do autor no ficheiro "author.txt".

## Breve explicação de como o programa se executa

Inicialmente temos um menu, onde o utilizador pode escolher o que deseja fazer.

1. Pode inserir uma tarefa em "Fazer". Aqui temos um função onde obtemos a referência do cartão e a grava em memória e no ficheiro binário "cards.bin".
2. Pode mover o cartão de "Fazer" para "Em execução". Aqui definimos um autor para o cartão e uma data de vencimento.
3. Pode alterar o autor. Primeiro verificamos se já existe o autor no ficheiro "author.txt" e, se existe, apenas escrevemos a posição desse autor no campo respetivo do cartão. Se o autor não existir, escrevemos o novo autor no final do ficheiro e colocámos a posição dele no cartão do campo "autor".
4. Pode fechar a tarefa. Isto implica passar a tarefa de "Em execução" para "Feito". Alteramos o campo "column" para "Feito" e removemos na lista ligada da coluna "Em Execução" e inserimos na lista ligada da coluna "Feito".
5. Pode reabrir tarefa. Aqui passamos a tarefa de "Feito" para "Fazer". Atualizamos as listas ligadas e obtemos e redefinimos a prioridade da tarefa.
6. Pode visualizar o quadro. Nessa visualização, cada coluna já organizada da maneira que foi pedida. Isto é, na coluna "Fazer" temos organizado primeiro pela prioridade e depois pela data de criação. Na coluna "Em execução" estão organizadas pelo nome da pessoa. E a coluna "Feito" está organizada pela data de conclusão.
7. Pode ver as tarefas de uma pessoa. Temos também aqui já uma lista ligada em que os cartões estão agregados por autor.
8. Pode ver tarefas por ordem de criação. Temos também uma lista ligada, ordenada pela data de criação a qual iteramos sobre e printamos.

