

Projeto de Arquitetura e Tecnologia de Computadores I

Gestor de Ginásio – “*Get Up*”



Universidade do Minho

João Pedro Martins Abreu A80560

João Pedro Vieira Leitão A88270

Miguel Nunes Reina A90065

Rui Miguel Lobo Resende Bessa Fernandes A89381

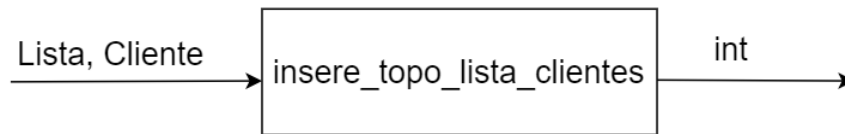
1. Introdução

Este relatório tem como objetivo explicar de forma simplificada em que consiste o projeto “Get Up”. Foi um trabalho que nos foi proposto pelo docente da UC de ATC1, onde a escolha do tema daria aos alunos alguma liberdade. O projeto foi composto por duas etapas: inicialmente foi utilizada memória estática (a qual já tínhamos conhecimento) e na segunda e última fase foi utilizada memória dinâmica. A nossa escolha recaiu sobre um gestor/aplicação de ginásio.

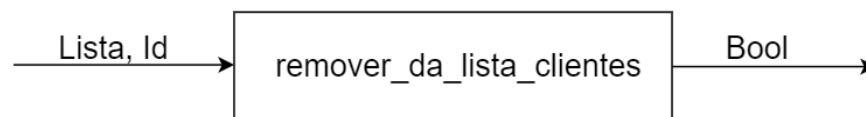
2. Problema

O nosso problema consistiu em simular a atividade/gestão de um ginásio, ao qual foi designado, ficticiamente, por “Get Up”. Por sua vez, para tal acontecer, decidimos implementar um gestor que contém dois tipos de interação: Admin/Cliente/Staff. Estes dois modos contêm um conjunto de funcionalidades de modo a permitir uma aproximação da realidade de tal estabelecimento de trabalho.

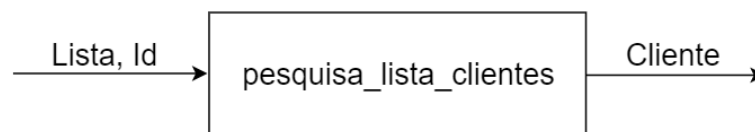
3. Análise e Design (funções da lista de staff são iguais, apenas alteram no seu nome)



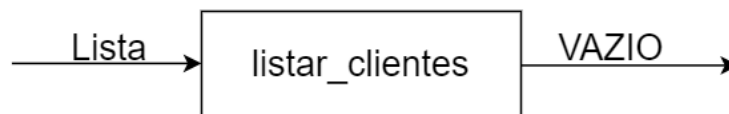
(1)Descrição: Dada a lista de clientes e o cliente (struct), esta função cria um novo nó do cliente e insere-o no topo da lista de clientes. Retorna 1 se conseguir inserir, senão retorna 0.



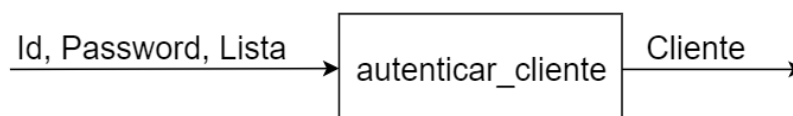
(2)Descrição: Dada a lista de clientes e o id do cliente que se quer remover, esta função apaga um nó se o id do nó do cliente que se está a verificar for igual ao id recebido como parâmetro. Retorna true se remover.



(3)Descrição: Dada a lista de clientes e o id do cliente que se quer pesquisar na lista, esta função retorna um cliente se o id do nó do cliente que se está a verificar for igual ao id recebido como parâmetro.



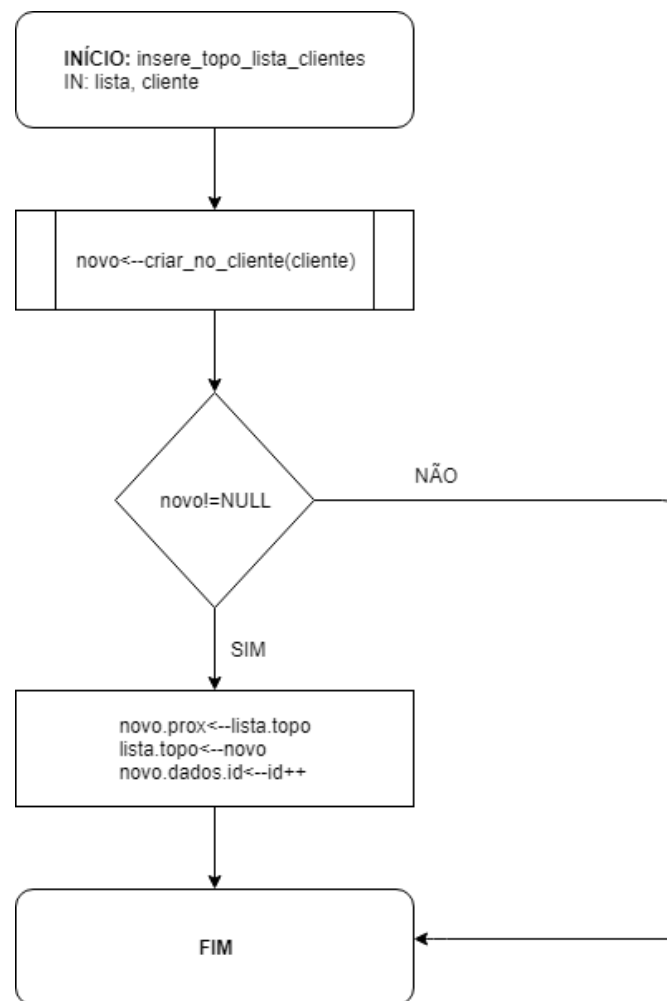
(4)Descrição: Dada a lista de clientes, esta função percorre a lista (nó a nó) imprimindo a informação do cliente se esta existir.

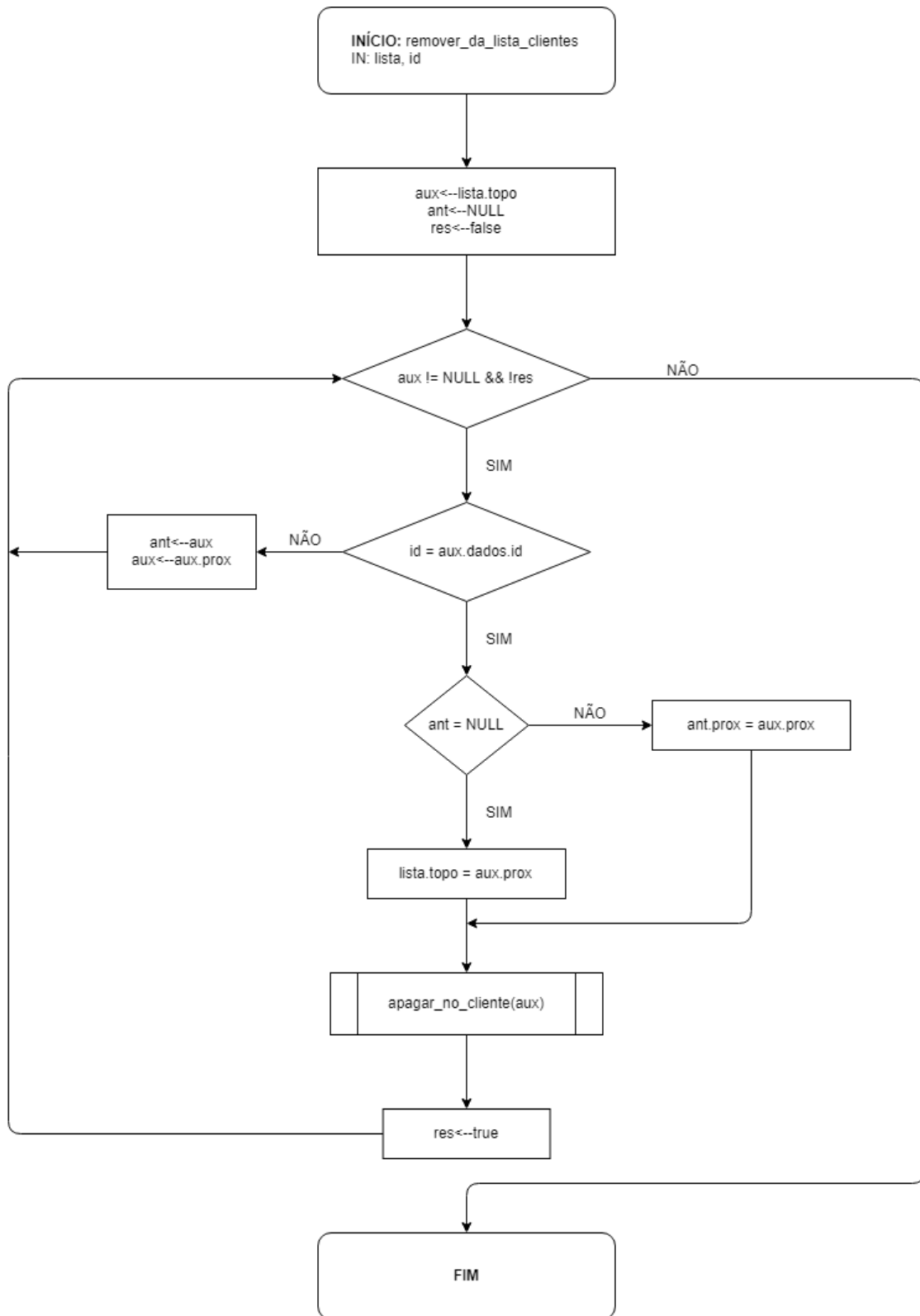


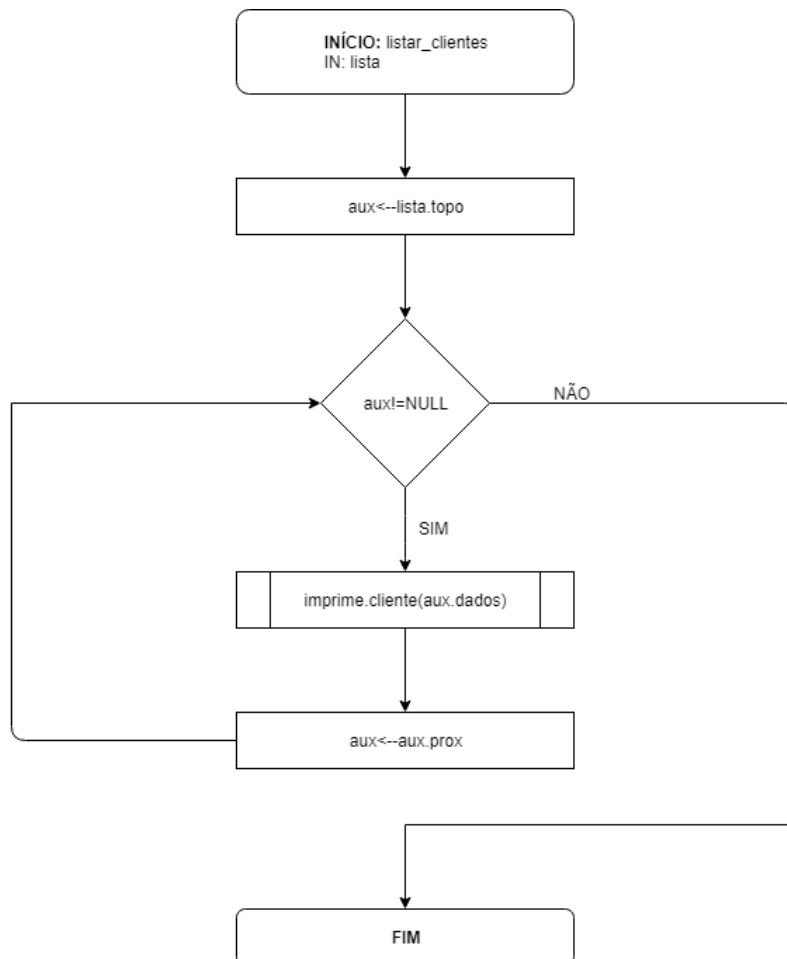
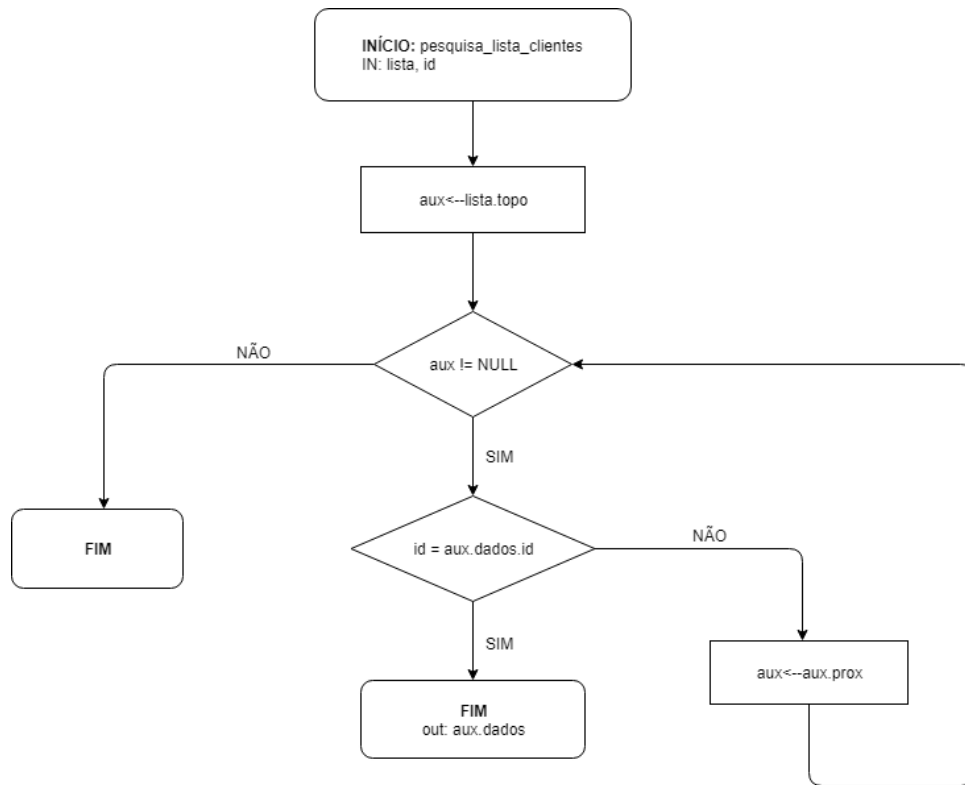
(5)Descrição: Dada a lista de clientes, o id do cliente e a password do cliente, esta função declara um cliente (chamando a função `pesquisa_lista_clientes`) e retorna o cliente se este não estiver vazio e se a password recebida como parâmetro (password de login) coincidir com a password de registo.

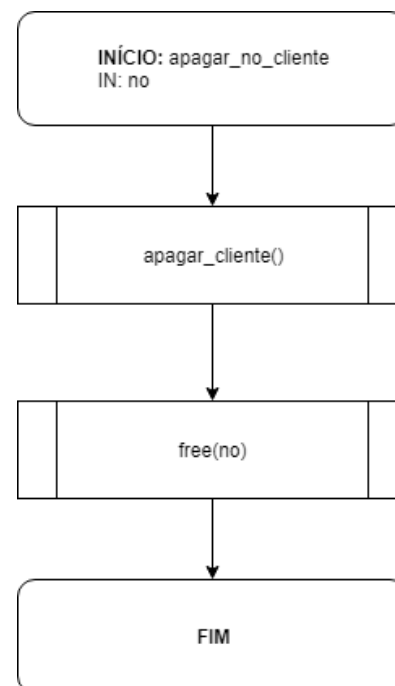
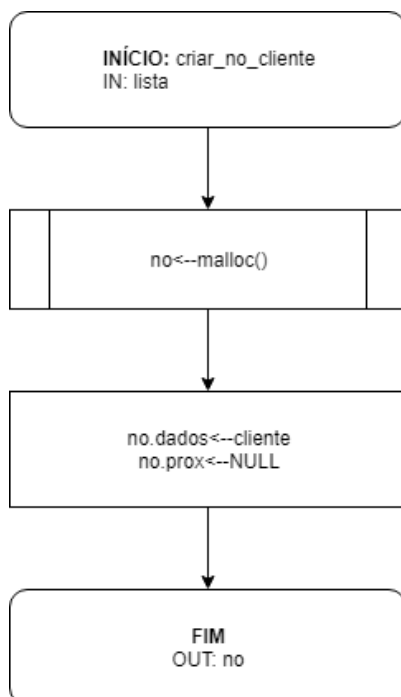
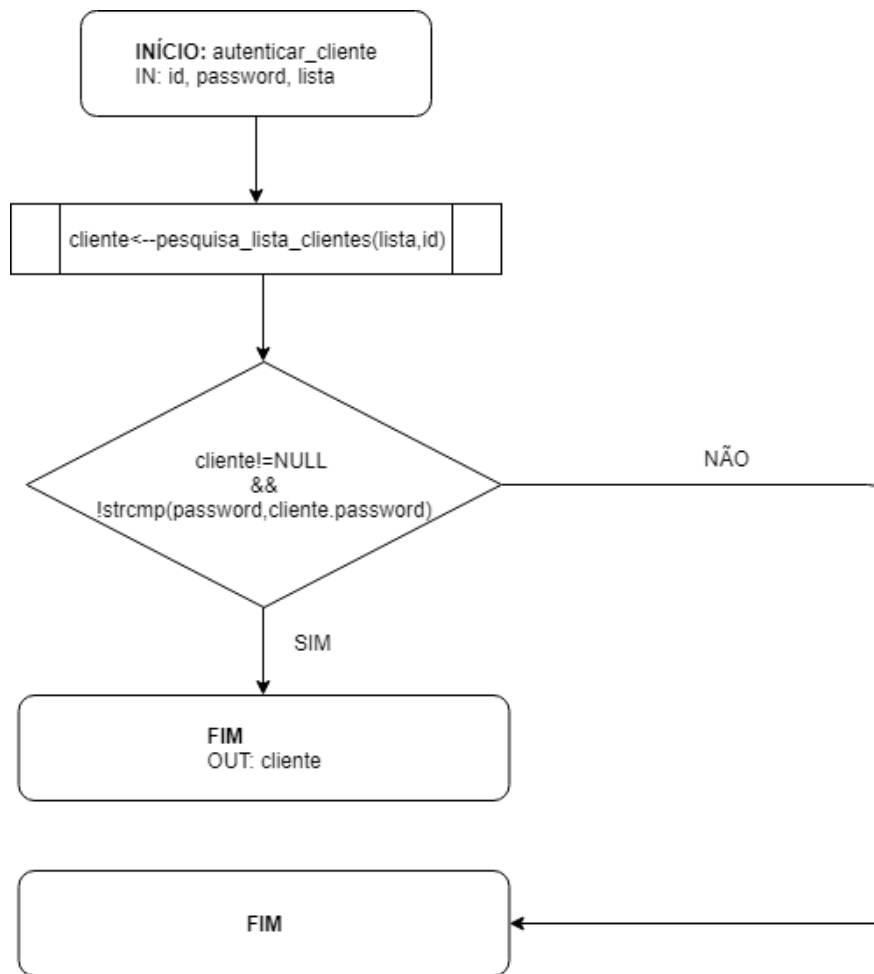
4. Fluxogramas (funções da lista de staff são iguais, apenas alteram no seu nome)

Apenas se colocaram fluxogramas de funções relativas às listas pois consideramos serem as mais importantes, principalmente as funções de inserir e remover, onde estão implementados os respectivos algoritmos. Estas funções serão “chamadas” nas nossas funções de interface desenvolvidas.









5. Interface

5.1 Admin

Menu inicial: É possível iniciar sessão como admin, cliente ou staff.

```
||||| GET UP |||||
1-Login Admin
2-Login Cliente
3-Login Staff
0-Sair
```

Se pretendermos iniciar sessão como admin, a password é “1”:

```
Introduza a palavra pass de admin: 1
```

Depois de iniciarmos sessão como admin, é possível realizar as tarefas descritas na imagem:

```
1-Registar cliente
2-Editar cliente
3-Listar clientes
4-Remover cliente

5-Registar funcionario
6-Editar funcionario
7-Listar funcionarios
8-Remover funcionario

0-Voltar ao menu anterior
```

Os clientes e os funcionários criados, são guardados em ficheiros binários, “salvando” assim o estado do programa.

Se pretendermos registar um cliente ou editar, é pedida a seguinte informação (exemplo):

```
Introduza o seu nome: Manuel Carvalho
Introduza a sua altura (m): 1.9
Introduza o seu peso (kg): 88
Introduza o seu numero de telefone: 930928719
Introduza a sua massa gorda (%): 20
Introduza a sua massa_magra (%): 80
Introduza a frequencia semanal (dias): 7
Introduza o seu objetivo:
1-Perder peso
2-Hipertrofia muscular
3-Manutencao
2
Introduza a sua password: gym
Repita a sua password: gym

Id: 1
```

Se pretendermos listar os clientes, aparece a seguinte informação (exemplo):

```
INFORMACAO DO CLIENTE COM O ID 2
Nome: Manuel Carvalho
Altura: 1.90
Peso: 88.00
Telefone: 930928719
Magorda(%): 20.00
Mmagra(%): 80.00
Password: gym
Frequencia Semanal: 7
Objetivo: Hipertrofia muscular

INFORMACAO DO CLIENTE COM O ID 1
Nome: Maria Miranda
Altura: 1.60
Peso: 55.00
Telefone: 967829182
Magorda(%): 40.00
Mmagra(%): 60.00
Password: gym
Frequencia Semanal: 4
Objetivo: Manuetencao
```

Ainda dentro do login do admin, é possível fazer exatamente as mesmas tarefas para o staff, porém este apenas possui nome, número de telefone, salário e password.

5.2 Cliente

Se pretendemos iniciar sessão como cliente, o id e a password serão aqueles que foram atribuídos no registo:

```
Introduza o id de cliente: 1  
Introduza a sua palavra pass: gym
```

Depois de iniciarmos sessão como cliente, dispomos do seguinte menu:

```
1-Editar os meus dados  
2-Ver planos de treino  
3-Ver planos alimentares  
4-Ver os meus dados  
  
0- Terminar sessao
```

De referir que os planos de treino são atribuídos com base na frequência semanal (dias) e os planos alimentares são atribuídos com base no objetivo do cliente.

5.3 Staff

Se pretendemos iniciar sessão como staff, o id e a password serão aqueles que foram atribuídos no registo:

```
Introduza o id de staff: 1  
Introduza a sua palavra pass: gym
```

Depois de iniciarmos sessão como staff, dispomos do seguinte menu:

```
1-Ver planos de treino do cliente  
2-Ver planos alimentares do cliente  
3-Editar os dados do cliente  
0-Terminar sessao
```

De referir que o staff apenas consegue editar os dados físicos do cliente.

6.Conclusão

Neste projeto conseguimos aprender de uma maneira mais aprofundada como funciona a memória dinâmica em C. Aprendemos a lidar com listas ligadas e o seu conceito ficou mais solidificado.