

Algoritmos e Estruturas de Dados II

relatório do Trabalho Prático I

Artur Mendes | 14870 | LESI PL

# Contextualização

Este documento foi desenvolvido no âmbito da UC de Algoritmos e Estruturas de Dados II, com o intuito de aplicar os conhecimentos lecionados ao longo do semestre até à data.

Todo o projeto foi concretizado em C, através de máquina virtual Linux (Parrot OS).

Pretende-se implementar um sistema capaz de fornecer informações sobre conjuntos de peças **Lego**, que ajude as pessoas que possuem peças de Lego, a descobrir quais os conjutos que podem construir com essas peças.  
  
Os dados estão organizados em 3 ficheiros de texto, onde cada linha contém um conjunto de colunas separadas por um caracter de tabulação (\t).   
**NOTA:** A primeira linha dos ficheiros inclui a informação sobre cada coluna.  
  
**Ficheiros**

**- Sets.tsv**

**(set\_num, year, theme)**Contém os conjuntos oficiais da Lego, indicando o seu código, o nome de comércio, o ano de lançamento e o tema em que se insere *(duplo, technic, etc).*

**- Parts.tsv**

**(part\_num, name, class, stock)**Contém os dados das diversas peças existentes, indicando a sua identificação (num), nome, classe (tipo de peça), e a quantidade em stock.

**- Parts\_sets.tsv**

**(set\_num, quantity, part\_num)**  
Contém os dados necessários para a definição da constituição de cada conjunto oficial da LEGO, detalhando o conjunto, a quantidade de cada peça que é utilizada, e a identificação da peça.

# Objetivos

Com os conjuntos de dados acima mencionados, pretende-se que o sistema seja capaz de calcular:

1 – Quais os conjuntos de determinado tema (ordenados pelo ano);

2 – As peças de **determinado tipo** em determinado **conjunto**;

3 – Quais as peças necessárias para construir um dado conjunto, indicando os dados de cada peça e respetiva quantidade

4 – O **total** de peças em Stock;

5 – O **total** de peças num **determinado conjunto**;

6 – A peça usada em **mais conjuntos diferentes**, independentemente da quantidade em cada um deles;

7 – A lista dos conjuntos que se conseguem construir com o **stock existente**.  
  
Para além dos dados lidos a partir dos ficheiros fornecidos, pretende-se que haja também opções para:  
  
- **Alterar** o nº de peças em **Stock**;

- A **adição de stock** com base no identificador de um conjunto (adicionar as peças que esse conjunto tem ao stock total de peças)

- **Remover todas as peças** de determinada classe (tipo de peça);

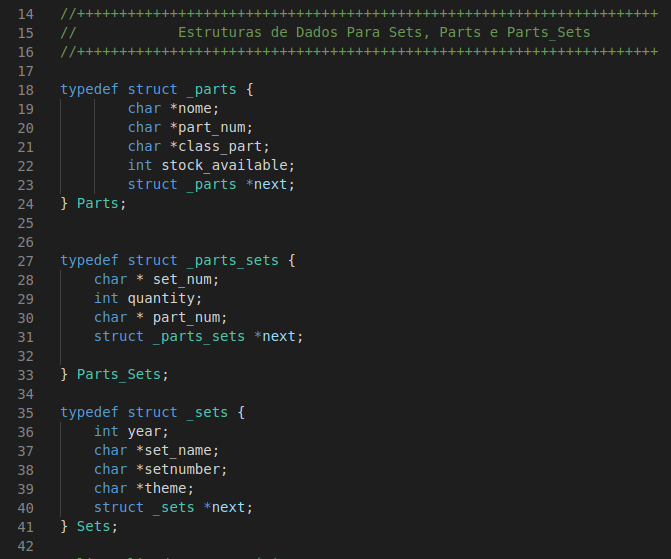
- **Remover todos os sets** de determinado tema;

# Desenvolvimento

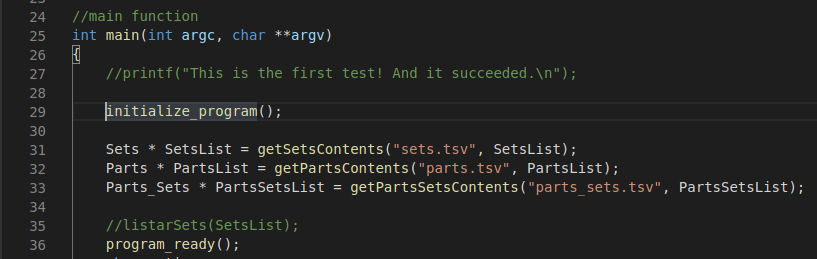
O processo de desenvolvimento foi baseado na “tentativa erro”, com desenvolvimento, testes e ajustes conforme o que achava que seria o mais conveniente para a quantidade de dados que cada ficheiro contém.

Para a leitura dos ficheiros Tsv para o sistema foram implementadas 3 listas simplesmente Ligadas, onde cada uma contém a informação vinda de cada ficheiro processado individualmente.

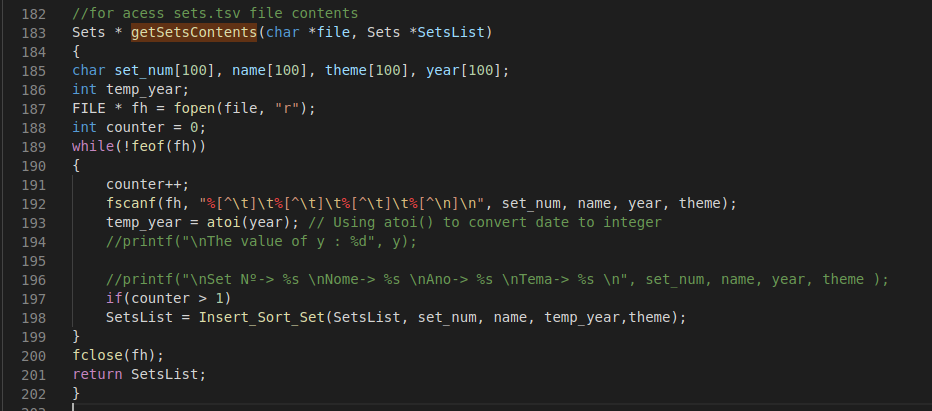
O projeto está dividido em 3 ficheiros, main.c, functions.c e functions.h



Img. 1 - Definição de Estruturas de dados das Listas Simplesmente Ligadas



Img. 2 - Definição das Listas no Main.c



Img. 3 - Função getSetsContents

getSetsContents – Acede ao conteúdo do ficheiro Sets.tsv, separa os dados que estão divididos através do “\t” e insere na lista ligada SetsList (de forma ordenada, na função Insert\_Sort\_Set), para garantir que a lista de Sets está ordenada por ano no ex.1.