

## GUIÃO 09 – O TAD PERSON-SET – 2ª VERSÃO

Pretende-se desenvolver o tipo abstrato de dados **PERSON-SET**, que permitirá representar um conjunto de instâncias do tipo **PERSON** desenvolvido anteriormente.

Mas, neste guião, esse conjunto será representado usando uma lista ordenada de instâncias do tipo **PERSON**, sem elementos duplicados.

### TAREFAS

- Comece por analisar o conteúdo dos ficheiros disponibilizados e as funcionalidades definidas para cada tipo de dados.
- **SORTED-LIST** – Analisar, completar e testar o tipo de dados.
- **PERSON-SET** – Analisar, completar e testar esta **nova versão** do tipo de dados.
- Teste de modo incremental todo o código desenvolvido.
- Pode executar o **valgrind** para verificar se tem “*memory leaks*” ou outros problemas relacionados com a alocação dinâmica de memória.

### SORTED-LIST

- Analise as **funcionalidades** disponibilizadas e a **estrutura de dados** que permite representar **uma lista ordenada**.
- Complete as **funções incompletas**, assinaladas no ficheiro **SortedList.c**.
- Analise o ficheiro **SortedListTest1.c** e teste as funções que completou. Que **instâncias** são armazenadas na lista ordenada? Como são feitas as **comparações** entre essas instâncias?
- Analise o ficheiro **SortedListTest2.c** e teste as funções que completou. Que **instâncias** são armazenadas na lista ordenada? Como são feitas as **comparações** entre essas instâncias?

### PERSON-SET

- Analise as **funcionalidades** disponibilizadas – que são as mesmas do guião anterior – e o modo **como é representado um conjunto de instâncias do tipo PERSON**.
- Termine o desenvolvimento do tipo de dados: as **funções incompletas** estão assinaladas no ficheiro **PersonSet.c**.
- Analise o ficheiro **Tests.c** e teste as funções que completou.