Projeto Prático – 2025/2026
Sistema de Ficheiros

Pretende-se um programa que faça a gestão de directorias e seus conteúdos (ficheiros e sub-directorias).

Nas funcionalidades que abaixo são pedidas, são apresentados os protótipos dos métodos a implementar, assumindo que tem de implementar as classes SistemaFicheiros; Directoria e Ficheiro (pode implementar outras que considere necessárias!!!)

Funcionalidades a implementar:

1. Correr o programa numa dada diretoria e esta (e seus conteúdos) serem carregados para memória (SistemaFicheiros); O método deve devolver true se a operação foi bem sucedida.
   bool SistemaFicheiros::Load(const string &path);
2. Contar o número de ficheiros que o *Sistema de Ficheiros* tem guardado em memória;
   int SistemaFicheiros::ContarFicheiros();
3. Contar o número de directorias que o *Sistema de Ficheiros* tem guardado em memória;
   int SistemaFicheiros::ContarDirectorias();
4. Determinar toda a memória ocupada;
   int SistemaFicheiros::Memoria();
5. Determinar qual a diretoria que tem mais informação (maior número de pastas e ficheiros); Se existir mais do que uma, deve devolver uma delas! (Não deve ser considerado o conteúdo das sub-directorias);
   string *SistemaFicheiros::DirectoriaMaisElementos ();
6. Determinar qual a diretoria que tem menos informação (menor número de pastas e ficheiros); Se existir mais do que uma, deve devolver uma delas!
   string *SistemaFicheiros::DirectoriaMenosElementos ();
7. Determinar o ficheiro que ocupa mais espaço, bem como o respetivo caminho;
   string *SistemaFicheiros::FicheiroMaior();
8. Determinar qual diretoria que está a ocupar mais espaço;
   string *SistemaFicheiros::DirectoriaMaisEspaco();
9. Pesquisar se existe um dado ficheiro ou diretoria, devolvendo o "caminho" completo até ele(a); Considere que se Tipo = 0 se pretende pesquisar um ficheiro e se Tipo = 1 se pretende uma directoria.
   string *SistemaFicheiros::Search(const string &s, int Tipo);
10. Remover todas as diretorias ou ficheiros; Se tipo = "DIR" remove directorias, senão remove ficheiros; Deve devolver true se a operação foi bem sucedida.
    bool SistemaFicheiros::RemoverAll(const string &s, const string &tipo);
11. Gravar para ficheiro em formato XML todo o *Sistema de Ficheiros*;
    void SistemaFicheiros::Escrever_XML(const string &s);
12. Ler de um ficheiro em formato XML todo o *Sistema de Ficheiros* (antes de ler deve ser apagado tudo o que estiver no *Sistema de Ficheiros*); Se leu corretamente devolve true, senão devolve false.
    bool SistemaFicheiros::Ler_XML(const string &s);

13. Mover um Ficheiro para outra diretoria; se existir mais do que um ficheiro, move o primeiro encontrado; Se o ficheiro já estiver em DirNova, não deve fazer nada. Se conseguir mover deve devolver **true** (**false** caso contrário);
    bool SistemaFicheiros::MoveFicheiro(const string &Fich, const string &DirNova);

14. Mover uma diretoria (e tudo o que lhe está associado!) para outra diretoria; se existir mais do que uma directoria, move a primeira encontrada. Assume-se que DirNew não é sub-directoria de DirOld. No caso de DirNew ser sub-directoria de DirOld, deve devolver false e não efetuar qualquer operação.
    bool SistemaFicheiros::MoverDirectoria(const string &DirOld, const string &DirNew);

15. Determinar a data de um dado ficheiro; se o ficheiro não existir deve devolver *NULL*; se existirem vários, deve devolver a data do primeiro ficheiro; (Exemplo de devolução de uma data "2017|10|5");
    string *SistemaFicheiros::DataFicheiro(const string &ficheiro);

16. À semelhança do comando do MS-DOS, implemente o método Tree; Deve considerar que a listagem deve sair na consola e num ficheiro. Se o método for chamado com um ficheiro o resultado deve ir para esse ficheiro. Se o método for chamado sem parâmetros, o resultado é apresentado na consola.
    void SistemaFicheiros::Tree(const string *fich = "tree.txt");

17. Pesquisa todas as directorias com nome <dir> e coloca-as em <lres>
    void SistemaFicheiros::PesquisarAllDirectorias(list<string> &lres, const string &dir);

18. Pesquisa todos os ficheiros com nome <file> e coloca em <lres>
    void SistemaFicheiros::PesquisarAllFicheiros(list<string> &lres, const string &file);

19. Renomear todos os ficheiros com um dado nome, para outro nome dado
    void SistemaFicheiros::RenomearFicheiros(const string &fich_old, const string &fich_new);

20. Verificar se Existem Ficheiros duplicados, mesmo nome;
    bool SistemaFicheiros::FicheiroDuplicados();

21. Copiar todos os ficheiros de uma directoria (e da respetiva estrutura de sub-directorias) cujo nome contenha uma determinada string para (a raiz de) outra diretoria; No caso de existirem ficheiros com o mesmo nome (obviamente em directorias diferentes) deve diferenciá-los acrescentando ao nome (antes da extensão) um número sequencial com 3 dígitos. Por exemplo: *xptopadraoABC**001**.ext*, *xptopadraoABC**002**.ext*.
    bool SistemaFicheiros::CopyBatch(const string &padrao, const string &DirOrigem, const string &DirDestino);

Observações:

- Os alunos não devem alterar os métodos apresentados acima!.
- Podem e devem criar novos métodos com os nomes que acharem por bem!
- Sempre na "filosofia" orientada a objetos!

Avaliação / Observações:

- Se a gestão da memória, não estiver correcta, haverá uma penalização de 5 valores;
- Se for detectado "copianço", trabalho anulado!
- Os grupos são constituídos no máximo por 3 alunos;
- Eventuais dúvidas serão esclarecidas pelos docentes da disciplina.

Entrega:

- 14/12/2025;
- 02/02/2026;