

Aula Prática 1

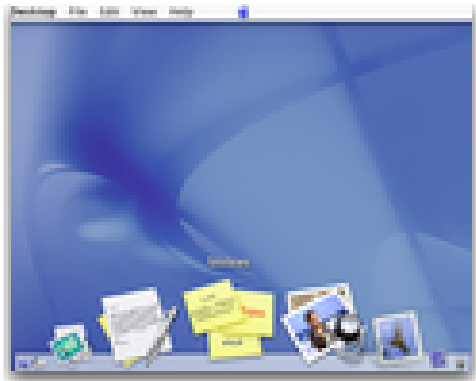
Paradigmas de Programação I / Programação Funcional

ESI/MCC 1º ano (2005/2006)

Nesta primeira aula pretende-se familiarizar os alunos com o ambiente de programação que será utilizado nestas disciplinas. Admite-se que os alunos já dispõem de algum tipo de contacto com a utilização de computadores.

1 Login e Apresentação do Sistema MacOS X

Após a introdução do *login* e da *password* (o docente informará qual é...) o computador exhibe um écran como o apresentado:



A barra no topo do écran diz respeito à aplicação activa e contém funcionalidades dessa aplicação (tais como abrir um novo ficheiro, gravar, etc).

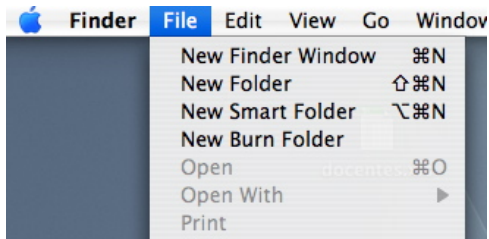
Na parte inferior do écran existe a **barra de ferramentas** que disponibiliza o acesso rápido a algumas aplicações (inicia aplicação com um *click*; arrastando um ficheiro sobre o icon: inicia a aplicação abrindo imediatamente o ficheiro em causa).

Na barra de ferramentas encontramos (entre outras) as aplicações:

- O **Finder** é a aplicação responsável por manter a sessão do utilizador. O seu aspecto mais visível é a interacção com o sistema de ficheiros. Por seu intermédio podemos aceder/manipular a informação contida no disco (ficheiros, directorias, etc.).
- O **Terminal** é a aplicação que permite interagir, em modo de comando, com o sistema operativo Mac OS X (o sistema Unix da Apple). A sua utilização requer portanto o conhecimento de alguns comandos desse sistema, tornando-se menos intuitiva do que a manipulação da interface gráfica do sistema. É no entanto muito mais flexível e permite-nos realizar operações não acessíveis pela interface gráfica.
- O **Safari** é um browser WWW. Permite consultar páginas Web (e.g. a página da disciplina).
- O **Trash** permite remover ficheiros/directorias, arrastando o que se pretende remover para cima do icon.

2 O Finder

Esta aplicação permite interagir de uma forma visual com o sistema de ficheiros. Algumas das operações básicas que se podem fazer são acessíveis na barra superior no item **File**.



Podemos abrir uma nova janela de visualização ou criar uma nova directoria.

Note-se que esta directoria será uma sub-directoria da que estamos neste momento a visualizar (se não estiver nenhuma janela do **Finder** aberta, será criada como uma sub-directoria da **Desktop**

Note a existência de uma directoria com icon  que se refere à sua directoria base (*Home*). A própria **Desktop** é uma sub-directoria dessa.

Tarefa 1 Crie na sua *Home* uma directoria chamada **Aula1**.

3 O Terminal

Através do **Terminal** também podemos interagir, de uma forma textual, com o sistema de ficheiros. Podemos por exemplo listar o conteúdo de uma directoria (**ls**), remover um ficheiro (**rm**), visualizar o conteúdo de um ficheiro (**cat**) ou mover/mudar o nome de um ficheiro (**mv**). Podemos além disso efectuar toda uma série de acções, incluindo o início de programas.

Tarefa 2 Inicie o Terminal e usando o comando **cd** (*change directory*) vá para a directoria **Aula1** criada atrás. Use ainda o comando **ls -la** para visualizar o conteúdo dessa directoria. Note que o comando **cd** quando usado sem argumentos o posiciona na sua *Home*.

Vamos agora usar um **editor de texto** para criar um ficheiro. Nesta primeira fase vamos usar o editor **vi**, comum a todos os sistemas Unix.

Quando estamos a usar o **vi** podemos estar em dois *modos de funcionamento*:

modo comando é o modo inicial e podemos invocar comandos: (**a**, **i**, e **o**) servem para passar a modo de inserção; **s** guarda as alterações efectuadas; (**:q**, **:x** e **ZZ**) terminam a execução do **vi**.

modo inserção onde podemos inserir o texto. Para voltar ao modo de comando devemos usar a tecla **Esc**.

4 Emacs

Um outro editor de texto muito utilizado em sistemas Unix é o **Emacs**. Tal como fizemos para o **vi**, vamos aqui descrever a utilização mais rudimentar deste editor de texto. Este programa pode ser encontrado na directoria **Applications** e tem como icon um gnu



Para abrir um determinado ficheiro **fich** usa-se o comando **CTRL-xCTRL-f**. Alternativamente, podemos usar o item **File** da barra superior do ecran. Aqui, uma das opções é abrir um ficheiro (*Open File*) ou uma nova janela (*New*). No final da edição devemos guardar o ficheiro editado, usando *Save* ou *Save As*, dependendo se o ficheiro já existia ou não.

5 Haskell

Tarefa 3 Usando um dos editores de texto acima referidos ou um qualquer outro do seu agrado, crie um ficheiro `Hello.hs` com o seguinte conteúdo:

```
module Hello where
main = putStrLn "Ola"
ultimonome n = head (reverse (words n))
```

Vamos agora começar a usar o interpretador de Haskell `ghci`. Para isso usa-se o comando `ghci` no Terminal.

Talvez o comando do `ghci` mais importante nesta altura é `:q` que permite terminar a sua execução. Além disso podem-se usar todas as funções pré-definidas em Haskell e que serão objecto de estudo ao longo desta disciplina.

Tarefa 4 Experimente as seguintes funções abaixo descritas. Tenha em atenção que para representar uma lista se delimitam os seus elementos por parêntesis rectos e se separam por vírgulas. Alternativamente, se se tratar de textos (listas de letras, pode-se fornecer este texto delimitado por aspas.

- `head`, calcula o primeiro elemento de uma lista
- `tail`, calcula a lista sem o primeiro elemento
- `reverse`, inverte uma lista
- `words`, dá como resultado a lista das palavras de um texto
- `unwords`, constrói um texto a partir de uma lista de palavras
- `take`, calcula os primeiros `n` elementos de uma lista
- `drop`, calcula a lista resultante de apagar os primeiros `n` elementos

Uma outra possibilidade consiste em *carregar* as definições constantes num ficheiro. Isso faz-se usando o comando `:l`.

Tarefa 5 Carregue as definições do ficheiro `Hello` e experimente-as. Nesse ficheiro foram definidas duas funções: `main` e `ultimonome`, esta última com um parâmetro que é um texto.

Uma forma mais expedita de usar o `ghci` consiste em usar duas janelas, uma para editar os ficheiros de definições e outra para executar o `ghci`. Como veremos mais à frente, existem editores de texto (e.g., o `emacs`) que permitem ter sub-janelas com a execução do `ghci`.