

E-Fólio A

21077 - Linguagens de Programação 2018-19

Pedro Miguel Madaleno Brandão
Aluno nº 1100055

Introdução

Neste primeiro e-fólio foi-nos solicitado que desenvolvêssemos um programa na linguagem de programação OCaml. As funcionalidades a implementar incluíam a passagem dos ficheiros de *input* por parâmetro, leitura do conteúdo dos ficheiros, *parsing* de acordo com um formato pré-definido (tendo em conta casos de exceção como valores não divulgados), e o cálculo de diversos indicadores climáticos.

Este relatório descreve o processo de desenvolvimento adotado.

Ambiente de trabalho

Na altura da instalação do ambiente de programação no início do semestre, cheguei à conclusão que o suporte para OCaml é maior em Sistemas Operativos Unix/Linux, comparando com Windows. Mesmo assim, depois de experimentar várias alternativas cheguei à combinação Visual Studio Code + OCPWin, com a execução de programas a ser feita através da linha de comandos do Windows. Por exemplo :

```
>ocaml.exe ex1.ml
```

Solução mais do que suficiente para praticar os exercícios disponibilizados e completar o eFólio.

Dificuldades encontradas

Durante a minha primeira passagem pelo ensino superior, tive contato com várias linguagens de alto nível e declarativas, tais como Lisp, Scheme (dialecto do Lisp) e Prolog.

Talvez devido ao fato de a minha carreira na indústria profissional incidir há 10 anos apenas na área das linguagens imperativas e programação orientada a objetos, e possivelmente ter alguns “vícios”, acabei por ter desta vez mais dificuldade em me adaptar ao paradigma funcional - apesar de o OCaml disponibilizar tanto o paradigma funcional como o imperativo/OOs.

Este fator foi ainda agravado, pois não adquiri a bibliografia opcional, e a documentação e exemplos online não são dos mais extensos ou claros.

Penso que por vezes o compilador de OCaml poderia ser mais explícito. No caso de certos erros de sintaxe, apenas nos é indicada a linha onde o mesmo ocorre e uma mensagem genérica, sem nenhuma informação adicional. Estes erros até podem estar a ser causados por problemas noutras linhas do programa, e a resolução dos mesmos pode levar bastante tempo no caso de programas ou funções de grande complexidade ou tamanho. Utilizadores mais experientes na linguagem certamente conseguirão verificar mais rapidamente qual o problema. Também contribui para isto o fato de existirem diversas maneiras de terminar uma instrução, tais como o ponto e vírgula, duplo ponto e vírgula, a *keyword* `in` no caso dos *lets*, ou mesmo sem nenhum carácter de término de instrução.

Solução

A solução para este exercício consiste num programa modularizado, que é composto por uma função principal com o ciclo de análise e output final, e duas funções auxiliares, uma para a leitura dos ficheiros (**read_lines**) e outra para dividir uma string em partes de acordo com um carácter separador (**split_on_char**).

A função principal executa os seguintes passos :

- Começa com um ciclo que percorre todas as linhas dos ficheiros de temperatura e precipitação, de acordo com o número de linhas dos mesmos, e termina quando processa a última linha;
- Para cada linha processada, declara variáveis auxiliares com o conteúdo dos ficheiros, e outras utilizadas no processamento do texto e dos diversos indicadores;
- Divide cada linha de acordo com o separador ‘;’ , produzindo uma lista com as correspondentes sub-strings;
- Inicia novo ciclo, com número de *loops* igual ao número de *tokens* (sub *strings*) da lista mencionada anteriormente;
- Se o *token* for o primeiro, imprime o nome da cidade;
- Se o *token* for o segundo, imprime o ano;
- Se o *token* for o terceiro, o mesmo é ignorado, pois é apenas uma indicação do tipo de indicador (Temp/Prec);
- Se for o *token* número 3 em diante, efetua diversos cálculos, incluindo uma nova chamada à função de divisão de *strings* de acordo com o separador ‘:’ , ignora valores ‘n/d’, e atualiza valores médios e totais de vários indicadores.
- As ultimas instruções, já fora do ciclo secundário, consistem em calcular os valores finais dos 4 indicadores (disponíveis em variáveis no *scope* do ciclo principal), e imprimir os mesmos.

Testes

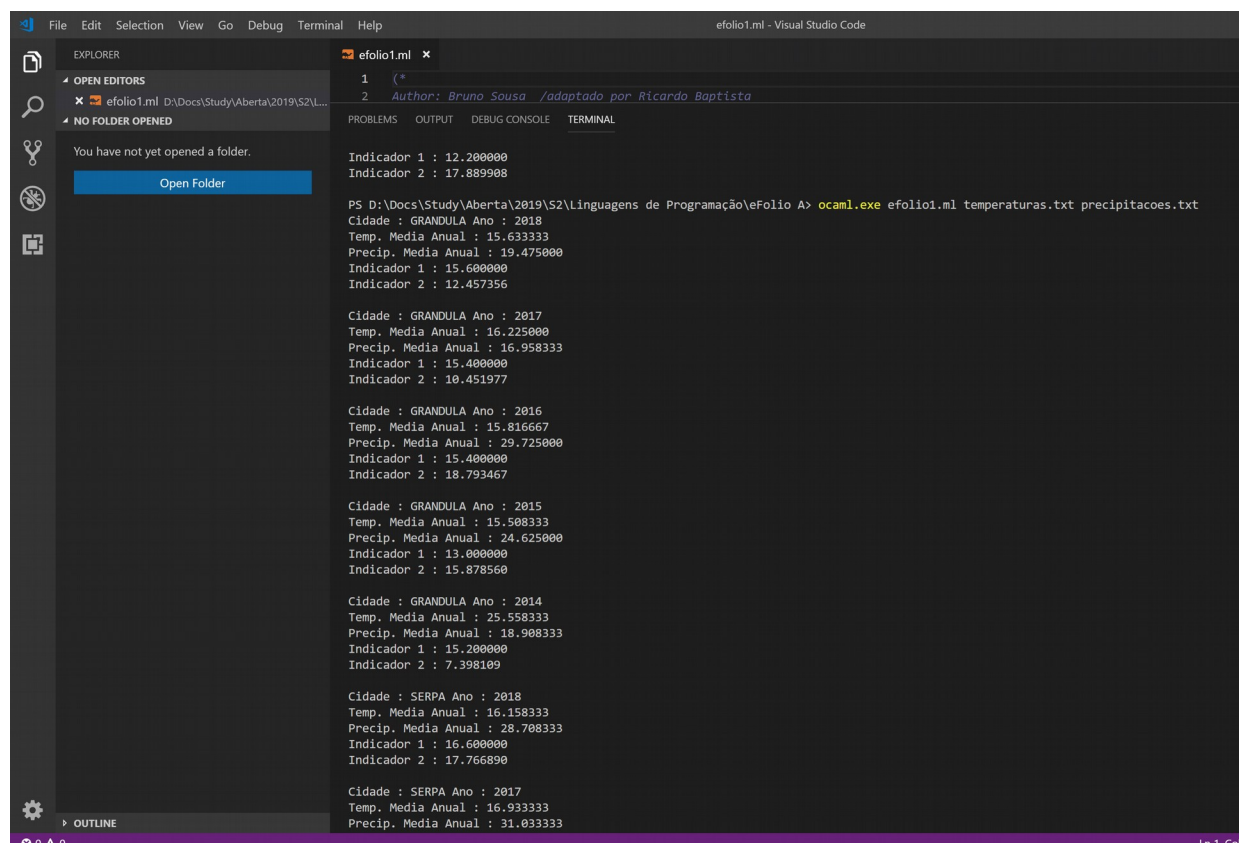
À medida que fui completando diferentes partes da solução, e corrigindo erros de execução e bugs no código, fui também testando a mesma.

Inicialmente apenas imprimindo diferentes strings, para ter a certeza que estava a consumir corretamente os dados dos dois ficheiros de texto. E de seguida calculando valores, formatando os resultados, e imprimindo os diversos dados correspondentes à informação pedida : cidade, ano, nome do indicador, valor.

Nos testes finais, verifiquei que os dados batiam certo com tudo o que era pedido, e comparei os valores obtidos pelo programa com os valores obtidos através de cálculos manuais feitos por mim.

O resultado final é uma solução completa e, penso eu, correta, conforme se pode verificar no screenshot seguinte :

```
> ocaml.exe base-efolio1.ml temperaturas.txt precipitacoes.txt
```



```
efolio1.ml x
1 (*)
2 Author: Bruno Sousa /adaptado por Ricardo Baptista

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Indicador 1 : 12.200000
Indicador 2 : 17.889908

PS D:\Docs\Study\Aberta\2019\S2\Linguagens de Programação\efolio A> ocaml.exe efolio1.ml temperaturas.txt precipitacoes.txt
Cidade : GRANDULA Ano : 2018
Temp. Media Anual : 15.633333
Precip. Media Anual : 19.475000
Indicador 1 : 15.600000
Indicador 2 : 12.457356

Cidade : GRANDULA Ano : 2017
Temp. Media Anual : 16.225000
Precip. Media Anual : 16.958333
Indicador 1 : 15.400000
Indicador 2 : 10.451977

Cidade : GRANDULA Ano : 2016
Temp. Media Anual : 15.816667
Precip. Media Anual : 29.725000
Indicador 1 : 15.400000
Indicador 2 : 18.793467

Cidade : GRANDULA Ano : 2015
Temp. Media Anual : 15.508333
Precip. Media Anual : 24.625000
Indicador 1 : 13.000000
Indicador 2 : 15.878560

Cidade : GRANDULA Ano : 2014
Temp. Media Anual : 25.558333
Precip. Media Anual : 18.908333
Indicador 1 : 15.200000
Indicador 2 : 7.398109

Cidade : SERPA Ano : 2018
Temp. Media Anual : 16.158333
Precip. Media Anual : 28.708333
Indicador 1 : 16.600000
Indicador 2 : 17.766890

Cidade : SERPA Ano : 2017
Temp. Media Anual : 16.933333
Precip. Media Anual : 31.033333
```