RELATÓRIO
Trabalho Prático- Parte 1

Docente: Arsénio Monteiro Reis
Unidade Curricular: Programação funcional

Elaborado por: Filipe Afonso Nunes (Al79057)
Tiago Isidro Sousa da Silva (Al78417)

Introdução:

No âmbito da Unidade Curricular de programação funcional foi-nos proposto a realização de um programa, que permita a leitura dos três ficheiros de texto disponibilizados pelo professor e que fosse capaz de:

- Apresentar todos os alunos que estão inscritos em cada unidade curricular;
- Apresentar todas as unidades curriculares às quais cada aluno está inscrito;
- Apresentar as listagens indicadas no tópico 1, filtrando por apenas uma unidade curricular;
- Apresentar as listagens indicadas no tópico 2, filtrando por apenas um aluno;
- Apresentar um menu que apresente as possibilidades de listagem indicadas nos pontos anteriores, e permita ao utilizador introduzir as opções que pretende.

Explicação da solução proposta:

Para a realização destes tópicos foi necessário criar uma função capaz de ler os ficheiros de texto fornecidos e dividi-los em linhas e letras:

```
fileSplit :: FilePath -> IO [[String]]
fileSplit filePath = do
  fileContents <- readFile filePath
  let linesList = lines fileContents
  let wordsList = map words linesList
  return wordsList
```

Funções acrescentadas ao programa:

```
getUcs :: [[String]] -> [[String]] -> [[String]] -> String -> [String]
getUcs inscr ucs listaAl option =
 case option of
  "pf" -> [head ucs !! 2 ++ " " ++ head ucs !! 3]
  "c" -> [ucs !! 1 !! 2]
  "t" -> [ucs !! 2 !! 2]
  "f" -> [ucs !! 3 !! 2]

getAl:: [[String]] -> [[String]] -> [[String]] -> String -> [String]
getAl inscr ucs listaAl option =
 case option of
  "al001" -> [head listaAl !! 2 ++ " " ++ head listaAl !! 3]
  "al002" -> [listaAl !! 1 !! 2 ++ " " ++ listaAl !! 1 !! 3]
  "al003" -> [listaAl !! 2 !! 2 ++ " " ++ listaAl !! 2 !! 3]
  "al004" -> [listaAl !! 3 !! 2 ++ " " ++ listaAl !! 3 !! 3]
```

Estas duas funções foram adicionadas com o intuito de buscar o conteúdo dos ficheiros de texto e criar uma lista com um nome só, por exemplo ["programacao funcional"] ou ["Rui Silva"], desta maneira obtemos uma demonstração no terminal mais simplificada e fácil de ler

Menu:

```
main :: IO ()
main = menu

menu :: IO()
menu = do
 putStrLn "////// MENU / / / / / /"
 putStrLn "1. Alunos inscritos em cada UC"
 putStrLn "2. Inscriçoes de cada aluno"
 putStrLn "3. Alunos inscrito numa UC em especifico"
 putStrLn "4. Inscriçoes de um aluno específico"
 putStrLn "Selecione uma opçao: "
 input <- getLine
 case input of
   "1" -> tar1
   "2" -> tar2
   "3" -> tar3
   "4" -> tar4
   "5" -> exitSuccess
```

Terminal

```
/ / / / / / MENU / / / / / /
1. Alunos inscritos em cada UC
2. Inscrições de cada aluno
3. Alunos inscritos numa UC em específico
4. Inscriçoes de um aluno específico
5. Encerrar o programa
Selecione uma opçao:
|
```

TÓPICO 1:

```
tar1 :: IO()
tar1 = do
 system "cls"
 ucs <- fileSplir "ucs.txt"
 listaAl <- fileSplit "listaalunos.txt"
 inscr <- fileSplit "inscricoes.txt"
 let progFunc = getUcs inscr ucs  listaAl "pf" ++ getAl inscr ucs listaAl "al001"
 (continuação) ++ getAl inscr ucs listaAl "al002"
 print progFunc
 let comp = getUcs inscr ucs listaAl "c" ++ getAl inscr ucs listaAl "al001" ++
 (continuação) getAl inscr ucs listaAL "al002" ++ getAl incsr ucs listaAl "al003"
 print comp
 let topicos = getUcs  inscr ucs listaAl "t" ++ getAl inscr ucs listaAl "al004"
 print topicos
 let fisica = getUcs  inscr ucs listaAl "f"
 print fisica
 system "pause"
 main
```

Terminal

```
["programacao funcional", "Rui Silva", "Maria Silva"]
["compiladores", "Rui Silva", "Maria Silva", "Tiago Silva"]
["topicos", "Sofia Silva"]
["fisica"]
Press any key to continue . . .
```

TÓPICO 2:

```
tar2 :: IO()
tar2 = do
 system "cls"
 ucs <- fileSplit "ucs.txt"
 listaAl <- fileSplit "listaalunos.txt"
 inscr <- fileSplit "incricoes.txt"
 let al001 = getAl inscr ucs listaAl "al001" ++ getUcs inscr ucs listaAl "pf" ++
 (continuação) getUcs inscr listaAl "c"
 print al001
 let al002 = getAl inscr ucs listaAl "al002" ++ getUcs inscr ucs listaAl "pf" ++
 (continuação) getUcs inscr listaAl "c"
 print al002
 let al003 = getAl inscr ucs listaAl "al003" ++ getUcs inscr ucs listaAl "c"
 print al003
 let al004 = getAl inscr ucs listaAl "al004" ++ getUcs inscr ucs listaAl "t"
 print al004
 system "pause"
 main
```

Terminal

```
["Rui Silva", "programacao funcional", "compiladores"]
["Maria Silva", "programacao funcional", "compiladores"]
["Tiago Silva", "compiladores"]
["Sofia Silva", "topicos"]
Press any key to continue . . .
```

TÓPICO 3:

```
tar3 :: IO()
tar3 = do
 system "cls"
 ucs <- fileSplit "ucs.txt"
 listaAl <- fileSplit "listaalunos.txt"
 inscr <- fileSplit "incricoes.txt"
 putStrLn "1. Programação Funcional"
 putStrLn "2. Compiladores"
 putStrLn "3. Tópicos"
 putStrLn "4. Fisica"
 putStrLn "Escolha a cadeira que pretende: "
 option <- getLine
 case option of
    "1" -> let progFunc = getUcs inscr ucs listaAl "pf" ++ getAl inscr ucs listaAl
        (continuação) "al001" ++ getAl inscr ucs listaAl "al002"
        in print progFunc
    "2" -> let comp = getUcs inscr ucs listaAl "c" ++ getAl inscr ucs listaAl
        (continuação) "al001" getAl inscr ucs listaAl "al002" ++ getAl inscr ucs
        (continuação) listaAl "al003"|
        in print comp
    "3" -> let topicos = getUcs inscr ucs listaAl "t" ++ getAl inscr ucs listaAl
        (continuação) "al004"
        in print topicos
    "4" -> let fisica = getUcs inscr ucs listaal "f"
        in print fisica
 system "pause"
 main
```

```
1. Programaçao Funcional
2. Compiladores
3. Tópicos
4. Física
Escolha a cadeira que pretende:
|
```

Terminal

```
1. Programaçao Funcional
2. Compiladores
3. Tópicos
4. Física
Escolha a cadeira que pretende:
3
["Sofia Silva"]
```

TÓPICO 4:

```
tar4 :: IO()
tar4 = do
 system "cls"
 ucs <- fileSplit "ucs.txt"
 listaAl <- fileSplit "listaalunos.txt"
 inscr <- fileSplit "incricoes.txt"
 putStrLn "1. al001 - Rui Silva"
 putStrLn "2. al002 - Maria Silva"
 putStrLn "3. al003 - Tiago Silva"
 putStrLn "4. al004 - Sofia Silva"
 putStrln "Escolha o aluno que pretende: "
 option <- getLine
 case option of
    "1" -> let al001 = getAl inscr ucs listaAl "al001" ++ getUcs inscr ucs listaAl
        (continuação) "pf" ++ getUcs inscr ucs listaAl "c"
        in print al001
    "2" -> let al002 = getAl inscr ucs listaAl "al002" ++ getUcs inscr ucs listaAl
        (continuação) "pf" ++ getUcs inscr ucs listaAl "c"
        in print al002
    "3" -> let al003 = getAl inscr ucs listaAl "al003" ++ getUcs inscr ucs listaal
        (continuação) "c"
        in print al003
    "4" -> let al004 = getAl inscr ucs listaal "al004" ++ getUcs inscr ucs listaal
        (continuação) "t"
        in print al004
 system "pause"
 main
```

Terminal

```
1. al001 - Rui Silva
2. al002 - Maria Silva
3. al003 - Tiago Silva
4. al004 - Sofia Silva
Escolha o aluno que pretende:
|
```

```
1. al001 - Rui Silva
2. al002 - Maria Silva
3. al003 - Tiago Silva
4. al004 - Sofia Silva
Escolha o aluno que pretende:
2
["programacao funcional", "compiladores"]
```