



Taller 1

Trabajo de Laboratorio

Curso nocturno 2017/18

Licenciatura en Informática

Anexo - Punto 3

“Pseudocódigo detallado para los tres comandos”

Tutor: Gómez, Federico.

Integrantes:	Pías, Richard	- 1.924.591-2
	Segovia, Joaquín	- 4.739.544-4
	Torres, Mathias	- 4.223.291-4

```
1  ///UDE, Lic. En Informática, Curso 2017/18 - Nocturno - Taller 1 - Tutor: Gómez, Federico
2  ///Integrantes Grupo FG2: Pías, Richard - 1.924.591-2 ; Segovia, Joaquín - 4.739.544-4 ; Torres, Mathias - 4.223.291-4
3
4
5  Levantar la línea escrita en un String
6
7  Partir la línea en subString.
8
9  Si (el primer subString contiene MAYÚSCULAS)
10
11      Mensaje de error ("El comando ingresado solo puede contener letras minúsculas");
12      volver al Prompt del programa;
13
14  Sino Si(el primer subString contiene NÚMEROS)
15
16      Mensaje de error ("El comando ingresado no puede contener números");
17      volver al Prompt del programa;
18
19  Sino Si(la cantidad de subString = 1)
20
21      Si (el primer substring NO es "salir")
22
23          Mensaje de error ("El comando ingresado no es valido, el único comando con un solo argumento debe ser 'salir'");
24          volver al Prompt del programa;
25
26      Sino
27
28          Salgo del Programa
29
30      FinSi
31
32  Sino Si(la cantidad de subString = 2)
33
```

```
34     Si (el primer subString contiene MAYÚSCULAS)
35
36         Mensaje de error ("El comando ingresado solo puede contener letras minúsculas");
37         volver al Prompt del programa;
38
39     Sino Si(el primer subString contiene NÚMEROS)
40
41         Mensaje de error ("El comando ingresado no puede contener números);
42         volver al Prompt del programa;
43
44     Sino
45
46         Si (el primer substring es "compilar")
47
48             Si (el segundo substring contiene MAYÚSCULAS)
49
50                 Mensaje de error ("El argumento (nombre de programa) a compilar ingresado solo puede contener letras
51                 minúsculas");
52                 volver al Prompt del programa;
53
54             Sino Si(el segundo substring contiene NÚMEROS)
55
56                 Mensaje de error ("El argumento (nombre de programa) a compilar ingresado no puede contener números);
57                 volver al Prompt del programa;
58
59             Sino Si(no existe el archivo)
60
61                 Mensaje de error ("No existe el archivo de código fuente con el nombre de programa "nombreArchivo" a
62                 compilar, por favor asegúrese de crearlo previamente");
63                 volver al Prompt del programa;
64
65     Sino
```

```
65  /* CÓDIGO CORRESPONDIENTE A compilar */-----
66  -----
67      creo el ABB en memoria
68
69      creo la lista en memoria
70
71      Guardo en una variable varArchivo el nombre para comparar posteriormente el substring luego de 'PROGRAMA'
72
73      Abro el archivo.csim
74
75      Levantar la línea escrita en un String
76
77      Partir la línea en subString.
78
79      Si (el primer subString contiene minusculas)
80
81          Mensaje de error ("Error de compilacion: La instruccion de la primera seccion del codigo fuente solo
82          puede contener letras mayusculas");
83          volver al Prompt del programa;
84
85      Sino Si(el primer subString contiene NÚMEROS)
86
87          Mensaje de error ("Error de compilacion: La instruccion de la primera seccion del codigo fuente no puede
88          contener números);
89          volver al Prompt del programa;
90
91      Sino Si(el primer subString no es "PROGRAMA")
92
93          Mensaje de error ("Error de compilacion: La instruccion de la primera seccion del codigo fuente debe ser
94          PROGRAMA");
95          volver al Prompt del programa;
96
97      Sino
```

```
95
96         Si (la cantidad de subString != 2)
97
98             Mensaje de error ("Error de compilacion: La instruccion de la primera seccion del codigo fuente no es
99             valido, se permite solo 1 argumento");
100             volver al Prompt del programa;
101
102         Sino Si (el segundo subString contiene MAYÚSCULAS)
103
104             Mensaje de error ("Error de compilacion: El argumento de la instruccion de la primera seccion del
105             codigo fuente solo puede contener letras minúsculas");
106             volver al Prompt del programa;
107
108         Sino Si (el segundo subString contiene NÚMEROS)
109
110             Mensaje de error ("Error de compilacion: El argumento de la instruccion de la primera seccion del
111             codigo fuente no puede contener números");
112             volver al Prompt del programa;
113
114         Sino Si (el segundo subString (nombre de programa) no coincide con el contenido de la varaible varArchivo)
115
116             Mensaje de error ("Error de compilacion: El argumento de la instruccion de la primera seccion del
117             codigo fuente tiene que coincidir con el nombre del archivo.csim a compilar");
118             volver al Prompt del programa;
119
120         Sino
121
122             Levantar la siguiente línea escrita en un String
123
124             Partir la línea en subString.
```

```
125         Si (el primer subString contiene minusculas)
```

```
124
125         Mensaje de error ("Error de compilacion: La instruccion de la segunda seccion del codigo fuente
126         solo puede contener letras mayusculas");
127         volver al Prompt del programa;
128     Sino Si(el primer subString contiene NÚMEROS)
129
130         Mensaje de error ("Error de compilacion: La instruccion de la segunda seccion del codigo fuente
131         no puede contener números);
132         volver al Prompt del programa;
133     Sino Si(el primer subString no es "VARIABLES")
134
135         Mensaje de error ("Error de compilacion: La instruccion de la segunda seccion del codigo fuente
136         debe ser VARIABLES");
137         volver al Prompt del programa;
138     Sino
139
140         Mientras (existan SubStrings en la linea levantada (en la lista))
141
142             Si (el subString contiene MAYÚSCULAS)
143
144                 Mensaje de error ("Error de compilacion: Las variables del codigo fuente solo puede
145                 contener letras minúsculas");
146                 volver al Prompt del programa;
147             Sino Si(el subString contiene NÚMEROS)
148
149                 Mensaje de error ("Error de compilacion: Las variables del codigo fuente no puede
150                 contener números");
151                 volver al Prompt del programa;
```

```
152         Sino
153
154             Cargo el SubString como una hoja del ABB (o sea nombre de variable y valor 0)
155
156         FinSi
157
158     Fin Mientras
159
160     Levantar la siguiente línea escrita en un String
161
162     Partir la línea en subString.
163
164
165
166     Si (la cantidad de subString = 1)
167
168         Si (el primer substring no es "INSTRUCCIONES")
169
170             Mensaje de error ("Error de compilacion: La instruccion de la tercer seccion del codigo
171             fuente debe ser 'INSTRUCCIONES'");
172             volver al Prompt del programa;
173
174         Sino
175
176             Mientras (existan lineas en el codigo fuente (no sea el final del archivo))
177
178                 Levantar la línea escrita en un String
179
180                 Partir la línea en subString.
181
182                 Si (la cantidad de subString = 1)
183
```

```
184      Mensaje de error ("Error de compilacion: Todas las instrucciones dentro de la
185      seccion INSTRUCCIONES del codigo fuente requieren al menos un argumento");
186      volver al Prompt del programa;
187      Sino
188
189      Si (el primer substring no es "LEER" o "MOSTRAR" o no es una variable de la
seccion VARIABLES(si no esta en el arbol))
190
191          Mensaje de error ("Error de compilacion: Las intrucciones posibles son LEER,
MOSTRAR, o asignacion a una variable");
192
193          volver al Prompt del programa;
194
195      Sino
196
197      Si (el primer substring es LEER) //CASO 1
198
199          Si (el segundo substring no es una variable de la seccion VARIABLES(si no
esta en el arbol))
200
201              Mensaje de error ("Error de compilacion: para la instruccion LEER
debe utilizar una variable declarada en la seccion de VARIABLES");
202
203              volver al Prompt del programa;
204
205          Sino
206
207              Guardo las intruccion en la lista de instrucciones
208
209          FinSi
210
211      Sino Si (el primer substring es MOSTRAR) //CASO 2
```



```
212
213 Si (el segundo substring no es una variable de la seccion VARIABLES(si no
214 esta en el arbol))
215 Mensaje de error ("Error de compilacion: para la instruccion MOSTRAR
216 debe utilizar una variable declarada en la seccion de VARIABLES");
217 volver al Prompt del programa;
218
219 Sino
220 Guardo las intruccion en la lista de instrucciones
221
222 FinSi
223
224 Sino //se que es una variable valida //CASO 3-8
225 Si (el segundo substring no el signo de =)
226 Mensaje de error ("Error de compilacion: para la asignacion a una
227 variable se debe utilizar el signo de =");
228 volver al Prompt del programa;
229
230 Sino
231 Si (el tecer substring no es entero o una var del arbol o si no es
232 una FUNC) //FUNC es SUM o RES o DIV o MUL
233 Mensaje de error ("Error de compilacion: luego del signo de =
234 solo puede haber un entero, una variable declarada o una de las
235 FUNC");
236
237
238
```

```
239         volver al Prompt del programa;
240
241     Sino
242
243         Si (el tecer substring NO es una FUNC) //caso 3 o 4
244
245             Guardo las intruccion en la lista de instrucciones
246
247         Sino
248
249             Si( la cantidad de substring es mayor a 5)
250
251
252                 Mensaje de error ("Error de compilacion: la asignacion
253                     con FUNC solo admite como maximo 2 argumentos");
254
255                 volver al Prompt del programa;
256
257         Sino
258
259             Guardo las intruccion en la lista de instrucciones, le
260             paso la lista de substring y se debera validar
261             correctamente todas las combinaciones, tanto LEER,
262             MOSTRAR y asiganciones, dadas en el diseño.
263             Utilizando la estructura y haciendo énfasis en la union
264             para las combinaciones en la asignacion.
265
266         FinSi
267     FinSi
268 FinSi
```

```
267
268             FinSi
269
270             FinSi
271
272             FinSi
273
274             FinSi
275
276
277             Fin Mientras
278
279             Cierro el archivo.csim
280
281             Bajo el arbol al archivo .vars
282
283             Bajo la lista al archivo .inst
284
285             libero memoria del ABB
286
287             libero memoria de la lista
288
289             FinSi
290
291             FinSi
292
293
294             FinSi
295
296             FinSi
297
298             FinSi
299
```

```
300         volver al Prompt del programa;
301
302     FinSi
303
304     Sino Si (el primer substring es "ejecutar")
305
306         Si (el segundo substring contiene MAYÚSCULAS)
307
308             Mensaje de error ("El argumento (nombre de programa) a ejecutar ingresado solo puede contener letras
309             minúsculas");
310             volver al Prompt del programa;
311
312         Sino Si(el segundo substring contiene NÚMEROS)
313
314             Mensaje de error ("El argumento (nombre de programa) a ejecutar ingresado no puede contener números);
315             volver al Prompt del programa;
316
317         Sino Si(si no existen los dos archivos)
318
319             Mensaje de error ("No existe el argumento (nombre de programa) a ejecutar, con el nombre de programa
320             "nombreArchivo", por favor asegúrese de haberlo compilado previamente");
321             volver al Prompt del programa;
322
323     Sino
324
325     /* CÓDIGO CORRESPONDIENTE A ejecutar */-----
326     -----
327     creo el ABB en memoria
328     creo la lista en memoria
329     Abro el archivo.vars
330
```

```
331
332         levanto y armo el arbol en memoria
333
334         Cierro el archivo.vars
335
336         Abro el archivo.inst
337
338         levanto y armo la lista en memoria
339
340         Cierro el archivo.inst
341
342         Mientras (haya intrucciones en la lista)
343
344             Segun (el tipo de instruccion) sea
345
346                 0) LEER:
347
348                     Pedir valor entero al usuario por teclado para la variable
349
350                     Paso el string leído a una funcion que recorra todo ese string, caracter por caracter y que
351                     devuelva true si tiene algun caracter que no este entre el codigos ascii 0-9
352
353                     Si (si es true, o sea que el valor no esta entre los codigos ascii 0-9)
354
355                         Mensaje de error ("Error de ejecucion: No se puede leer y guardar letras o caracteres raros,
356                         debe ingresar solo valores enteros");
357                         volver al Prompt del programa;
358
359                     Sino
360
361                         Paso el string a una funcion que me lo parsee y me devuelva un entero
362                         Buscar variable en el arbol
```

```
362             Asignar el valor entero a la variable
363
364         Fin Si
365
366     1) MOSTRAR:
367
368         Buscar variable en el arbol
369         Imprimir nombre y valor de variable por pantalla
370
371     2) AS1:
372
373         Buscar variable en el arbol
374         Asignar el valor entero a la variable
375
376     3) AS2:
377
378         Buscar variableA en el arbol
379         Asignar el valor de la variableB a la variableA
380
381     4) AS3:
382
383         Si (FUNC = DIV y el segundo entero es = 0)
384
385             Mensaje de error ("Error de ejecucion: No se puede realizar la division entre 0");
386             volver al Prompt del programa;
387
388         Sino
389
390             Segun (tipo de operador aritmetico) sea
391
392                 0) SUM:
393
394                     resultado = primer entero + segundo entero
```

```
395
396             1) RES:
397
398                 resultado = primer entero - segundo entero
399
400             2) MUL:
401
402                 resultado = primer entero * segundo entero
403
404             3) DIV:
405
406                 resultado = primer entero / segundo entero
407
408             FinSegun
409
410             Buscar variableA en el arbol
411             Asignar el valor resultante de la FUNC correspondiente a la variableA
412
413             FinSi
414
415     5) AS4:
416
417         Si (FUNC = DIV y la segunda variable (variableB) es = 0)
418
419             Mensaje de error ("Error de ejecucion: No se puede realizar la division entre 0");
420             volver al Prompt del programa;
421
422         Sino
423
424             Segun (tipo de operador aritmetico) sea
425
426                 0) SUM:
427
```

```
428             resultado = entero + segunda variableB
429
430         1) RES:
431
432             resultado = entero - segunda variableB
433
434         2) MUL:
435
436             resultado = entero * segunda variableB
437
438         3) DIV:
439
440             resultado = entero / segunda variableB
441
442         FinSegun
443
444         Buscar variableA en el arbol
445         Asignar el valor resultante de la FUNC correspondiente a la variableA
446
447     FinSi
448
449 6) AS5:
450
451     Si (FUNC = DIV y el segundo entero es = 0)
452
453         Mensaje de error ("Error de ejecucion: No se puede realizar la division entre 0");
454         volver al Prompt del programa;
455
456     Sino
457
458         Segun (tipo de operador aritmetico) sea
459
460             0) SUM:
```



```
461
462             resultado = segunda variableB + entero
463
464         1) RES:
465
466             resultado = segunda variableB - entero
467
468         2) MUL:
469
470             resultado = psegunda variableB * entero
471
472         3) DIV:
473
474             resultado = segunda variableB / entero
475
476         FinSegun
477
478         Buscar variableA en el arbol
479         Asignar el valor resultante de la FUNC correspondiente a la variableA
480
481     FinSi
482
483 7) AS6:
484
485     Si (FUNC = DIV y la segunda variable (variableB) es = 0)
486
487         Mensaje de error ("Error de ejecucion: No se puede realizar la division entre 0");
488         volver al Prompt del programa;
489
490     Sino
491
492         Segun (tipo de operador aritmetico) sea
493
```

```
494          0) SUM:
495
496              resultado = segunda variableB + tercer variableC
497
498          1) RES:
499
500              resultado = segunda variableB - tercer variableC
501
502          2) MUL:
503
504              resultado = psegunda variableB * tercer variableC
505
506          3) DIV:
507
508              resultado = segunda variableB / tercer variableC
509
510          FinSegun
511
512          Buscar variableA en el arbol
513          Asignar el valor resultante de la FUNC correspondiente a la variableA
514
515          FinSi
516
517          FinSegun
518
519      Fin Mientras
520
521      libero memoria del ABB
522
523      libero memoria de la lista
524
525      volver al Prompt del programa;
526
```

```
527         FinSi
528
529     Sino
530
531         Mensaje de error ("El comando ingresado no es válido, solo existen los comandos 'compilar' o 'ejecutar' para un
532         argumento (nombre de programa)");
533         volver al Prompt del programa;
534     Fin Si
535
536 Sino
537
538     Mensaje de error ("El comando ingresado no es valido, no existen comandos con un argumento (nombre de programa) compuesto
539     por mas de una palabra");
540     volver al Prompt del programa;
541 FinSi
542
543
544 ////UDE, Lic. En Informática, Curso 2017/18 - Nocturno - Taller 1 - Tutor: Gómez, Federico
545 ////Integrantes Grupo FG2: Pías, Richard - 1.924.591-2 ; Segovia, Joaquín - 4.739.544-4 ; Torres, Mathias - 4.223.291-4
```

