

Pràctiques - Programació declarativa. Aplicacions

Curs 2022/23

GEINF

Pràctica ViquipediaSimil

Angel Molero 41512839R – u1967593@campus.udg.edu

Gabriel Lopez 41511387K – u1968394@campus.udg.edu

Professor: Mateu Villaret Auselle

Índex

A	Introducció	2
B	Descripció de les classes	2
C	Extractes comentats del codi	2
C.1	Primera Part	2
C.2	Segona Part	3
C.2.1	Modificació MapReduce	4
D	Càlculs realitzats	4
E	Jocs de proves	7
E.1	Resultats Execucions Primera Part	7
E.1.1	Freqüència de paraules	7
E.1.2	Sense stop-words	7
E.1.3	Distribució de paraules	8
E.1.4	Ngrames	8
E.1.5	Vector space model	8
E.2	Resultats Execucions Segona Part	9
F	Taula de rendiment	10
F.1	Especificacions	11
F.2	Taula dels actors	11
G	Alltres consideracions	11

A Introducció

L'objectiu de la pràctica consisteix en analitzar el contingut de diversos fitxers i comprovar la seva similitud. S'ha realitzat de dues formes diferents, la primera de forma iterativa i la segona mitjançant mappers i reducers. Encara que l'objectiu sigui el mateix, el resultat final serà diferent, ja que, a la primera s'utilitzen un número reduït de documents a analitzar, mentre que a la segona s'utilitzen 5000.

Dins el directori **primeraPart** es pot veure el codi realitzar per la primera part i un document del mateix.

B Descripció de les classes

- **Main.scala**: Dins l'objecte **Main** trobem els mètodes principals dels diferents mappers i reducers, i on s'executa el codi inicial.
- **InfoFicheros.scala**: Dins l'objecte **InfoFicheros** guardem aspectes dels diferents fitxers analitzats i alguns mètodes auxiliars. Així calculem alguns aspectes un únic cop i millorem el rendiment.
- **MapReduceActors.scala**: En aquest fitxer trobem les classes **Mapper**, **Reducer** i **MapReduce** on s'han modificat per complir amb els objectius de la pràctica.

C Extractes comentats del codi

C.1 Primera Part

S'han agafat les funcions d'ordre superior més rellevants.

- **toFloat**: Mètode utilitzat per convertir el valor a float. Un exemple d'ús:

```
private def calculFreq(numParaules: Int, i: Int): BigDecimal = {  
    BigDecimal((i.toFloat / numParaules) * 100)  
}
```

- **foldLeft**: Mètode que agafa una funció d'ordre superior i l'utilitzarà per col·lapsar la col·lecció. Un exemple d'ús, per calcular la freqüència total de les paraules. Ens serveix per recorre i guardar el valor anterior:

```
private def raizSumatorio(freq1: List[(String, Double)]): Double = {  
    Math.sqrt(freq1.foldLeft(0.0)((x, y) => x + y._2 * y._2).doubleValue)  
}
```

- **map**: Mètode que aplica una funció a cada element de la col·lecció. L'hem utilitzat per calcular la freqüència normal. Exemple d'ús:

```
private def frecuenciaNormal(l1: List[(String, Int)], freqNormMax1: Double, sumaTotal: Double): List[(String, Double)] = {  
    l1.map(x => (x._1, redondear(BigDecimal((x._2.toFloat / sumaTotal) * 100), ndigits = 2) / freqNormMax1))  
}
```

- **setScale**: Mètode utilitzat per redondejar. Exemple d'ús:

```
def redondear(num: BigDecimal, ndigits: Int): Double = {
  num.setScale(ndigits, BigDecimal.RoundingMode.HALF_UP).doubleValue
}
```

- **compareTo**: Mètode comparar dues cadenes. L'utilitzem per saber quina de les dues cadenes hem d'avançar. Exemple d'ús:

```
def vectorAlignment(v1: List[(String, Double)], v2: List[(String, Double)], result: Double): Double = {
  if (v1.isEmpty || v2.isEmpty) result
  else {
    val comp = v1.head._1.compareTo(v2.head._1)
    if (comp == 0) vectorAlignment(v1.tail, v2.tail, result + v1.head._2 * v2.head._2)
    else if (comp < 0) vectorAlignment(v1.tail, v2, result)
    else vectorAlignment(v1, v2.tail, result)
  }
}
```

- **sliding**: Utilitzada per dividir la llista en n grames. Exemple d'ús:

```
def ngrames(n: Int, texto: String): List[(String, Int)] = {
  val modtexto = texto.toLowerCase().replaceAll( regex = "[^a-z]", replacement = " ").split( regex = " ").filter(_ != "")
  val grames = modtexto.sliding(n).toList
  val nrep = grames.map(x => (x.mkString(" "), 0))
  nrep.groupBy(_._1).map(x => (x._1, x._2.size)).toList.sortWith(_._2 > _._2)
}
```

- **groupBy**: Utilitza el predicat rebut per agrupar i formar un Map. L'utilitzem agafant els nGrames (1 2) i formats com una cadena i els agrupem per key i així agrupem per nGrama. Exemple d'ús:

```
def ngrames(n: Int, texto: String): List[(String, Int)] = {
  val modtexto = texto.toLowerCase().replaceAll( regex = "[^a-z]", replacement = " ").split( regex = " ").filter(_ != "")
  val grames = modtexto.sliding(n).toList
  val nrep = grames.map(x => (x.mkString(" "), 0))
  nrep.groupBy(_._1).map(x => (x._1, x._2.size)).toList.sortWith(_._2 > _._2)
}
```

- **distinct**: Aquest mètode l'utilitzem per guardar només les paraules úniques, i així no repetir. Exemple d'ús:

```
def nonstopfreq(texto: String): List[(String, Int)] = {
  val stopwords = scala.io.Source.fromFile("src/main/scala/english-stop.txt").mkString
  val stop = stopwords.split( regex = "\n").toList
  val modtexto = texto.toLowerCase().replaceAll( regex = "[^a-z]", replacement = " ").split( regex = " ").filter(_ != "")
  val f = modtexto.filter(!stop.contains(_))
  f.distinct.map(p => (p, f.count(x => x == p))).sortWith(_._2 > _._2).toList
}
```

C.2 Segona Part

A la segona part no cal destacar cap funció d'ordre superior, totes les utilitzades són molt comunes.

C.2.1 Modificació MapReduce

- S'han afegit dos paràmetres nous **numappers** i **numreducers**.
- A la línia 73 s'assigna numappers a nmappers.
- A la línia 108 s'assigna la longitud de l'entrada a mappersPendants.
- A la línia 125 s'assigna numreducers a nreducers.

D Càlculs realitzats

- **input:** List[(K1,List[V1])]
- **mapping:** (K1,List[V1]) \Rightarrow List[(K2,V2)]
- **reducing:** (K2,List[V2]) \Rightarrow (K2,V3)
 - **mappingLlegir i reducingLlegir:** Aquestes dos funcions ens permeten obtenir a partir del nom del fitxer que li passem com a V1, per obtenir la seva informació i així guardar-la en el objecte que s'ha creat com a diccionari InfoFicheros, més endavant. Això ens permet no tornar a cridar la funció parseViquipediaFile cada vegada que necessitem alguna informació d'algun fitxer. Per tant, en el mapping retornarem com a key el títol, y el valor una tupla de llistes de la informació que té aquell fitxer.
 - * **K1:** Int, id que no necessitem.
 - * **V1:** String, del nom del fitxer.
 - * **K2:** String, nom del fitxer
 - * **V2:** Tupla, de llista de Strings, amb cada una de la informació del fitxer.
 - * **V3:** Llista de tuples de llista de Strings. Retornem el mateix que ha rebut, però la segona posició de la tupla, serà una List, ja que després ens serà més fàcil de treballar, en cas que li tinguem que enviar al MapReduce.
 - **mappingRef i reducingRef:** Aquestes funcions ens permetran calcular el nombre de vegades que surt una referència entre tots els documents. Per això, retornem com a key 2, el nom de referència i un comptador per després en el reducing sumar-lo.
 - * **K1:** String, títol del document. No el farem servir ja que ens interessa només les referències que té.
 - * **V1:** String, nom de la referència, que utilitzarem com a K2.
 - * **K2:** String, es el nom de la referència, ja que ens interessa agrupar per nom de la referència.
 - * **V2:** Int, serà un 1, perquè es com un contador que després sumarem tots en el reducing i així saber quantes vegades surt.
 - * **V3:** Int, la suma de quantes vegades ha sortit la referència K2.
 - **mappingCombNoRef i reducingCombNoRef:** Aquestes funcions son importants, perquè ens permeten combinar els documents sense repetir-se entre ells i sense referenciar-se mútuament. Per fer el mapping, li passem el títol amb el seu contingut. Però el que ens interessa es el títol, perquè a partir del diccionari, concretament la variable InfoFicheros.fitxTratar, que tenim ordenat prèviament, podem saber en quina posició està. Volem saber la posició que està ja que, ens permet agafar del InfoFicheros.fitxTratar, a partir d'aquella posició i ens permet no repetir amb els anteriors de la llista. Això ens permet tindre millor rendiment i eficiència. Dins d'aquest mètode recorrem la llista sobrant de InfoFicheros.fitxTratar, i comprovem que no sigui el mateix nom, ja que no ens interessa i que no tingui una referència d'aquell fitxer, ja que així no ens interessaria guardar-lo.

- * **K1**: String, títol amb el que combinarem.
 - * **V1**: String, paraules del que no es rellevant.
 - * **K2**: String, títol serà el mateix que el K1, en cas de que es pugui combinar sinó serà un espai en blanc. El espai en blanc el borraré quan s'hagi acabat tot el MapReduce.
 - * **V2**: String, títol en cas de que es pugui combinar, sinó estarà buit.
 - * **V3**: List[String], que conté totes les combinacions possibles amb K2.
- **mappingWC i reduccingWC**: Aquestes funcions ens permetran contar les ocurrences de cada paraula en cada fitxer. És important que guardem com a K2, una tupla de títol i paraula, per agrupar-les. El value serà un comptador. Tindrà la mateixa idea que nombre de referències.
- * **K1**: String, títol del document.
 - * **V1**: String, paraula que conté el document.
 - * **K2**: Tupla, de títol i paraula.
 - * **V2**: Int, comptador que utilitzarem per sumar.
 - * **V3**: Int, suma de valors que apareix aquella paraula en el document.
- **mappingTF i reduccingTF**: Aquesta funció és va pensar per reestructurar les ocurrences de les paraules, és a dir, ara tenim un Map amb una tupla com a Key i la ocurrencia com a valor. Aleshores, aquestes dos funcions ens permetran tindre com a key, el títol del document, i com a value una llista de tuples de paraula i ocurrencia que té aquest document.
- * **K1**: Tupla, títol i paraula.
 - * **V1**: Int, Ocurrencia que surt aquesta paraula en aquest títol, K1.
 - * **K2**: String, títol del document, per així poder agrupar en paraules i ocurrences.
 - * **V2**: Tupla, amb la paraula i la ocurrencia.
 - * **V3**: Tupla, amb la paraula i la ocurrencia.
- **mappingTfidf i reduccingTfidf**: Aquestes dos funcions ens permetrà calcular el tfidf. Per això ens interessa agrupar per paraules, i que el reduccing rebi la paraula i una llista dels documents a on surt amb la seva ocurrencia. Així, ja sabrem a quants documents surt aquella paraula.
- * **K1**: String, títol del document.
 - * **V1**: Tupla, paraula i ocurrencia
 - * **K2**: String, paraula
 - * **V2**: Tupla, títol i ocurrencia
 - * **V3**: Tupla, títol i valor tfidf.
- **mappingGirar i reduccingGirar**: Aquestes dos funcions s'utilitzen per reestructurar la informació que ja tenim dels tfidf. Per tant convertirem de paraula (key), i tupla títol, tfidf a títol (key), paraula i tfidf. Això ens permet tindre una millor estructura per treballar més tard.
- * **K1**: String, paraula dels diferents documents.
 - * **V1**: Tupla, títol i tfidf.
 - * **K2**: String, títol del document.
 - * **V2**: Tupla, paraula i tfidf.
 - * **V3**: Tupla, paraula i tfidf. És important que ordenem ja que quan calculem el cosinisme, els tenim que tindre els elements.

- **mappingRaizSumatorio i reduccingRaizSumatorio:** Aquestes dues funcions, és van crear per calcular la part del denominador del càlcul del cosinesim de similitud, ja que, es fer la arrel del sumatori del mateix document. Aquest resultat el guardem al diccionari, per utilitzar-lo en el cosinesim.
 - * **K1:** String, títol del document.
 - * **V1:** Tupla, paraula i tfidf.
 - * **K2:** String, títol del document
 - * **V2:** Double, valor del tfidf.
 - * **V3:** Double, resultat del càlcul del arrel del sumatori
- **mappingCosinoSimil i reduccingCosinoSimil:** En aquestes dues funcions els utilitzem per calcular el valor del cosinesim. El mapping rebrà el títol del document i la llista de documents amb el que pot combinar. Per tant, a partir d'aquí amb l'ajuda del diccionari agafarem la informació que necessitem per calcular-lo.
 - * **K1:** String, títol del document
 - * **V1:** String, títol del document.
 - * **K2:** String, títol (K1) del document.
 - * **V2:** Tupla, títol del document combinat i el valor del cosinesim.
 - * **V3:** List[(String,Double)], títol del document combinat i el valor del cosinesim.
- **mappingFotos i reduccingFotos:** Aquestes dues funcions ens serveixen per a partir de totes les imatges que tenen els diferents documents, calcular el nombre mitjà de fotos.
 - * **K1:** String, titol del document.
 - * **V1:** String, nom de la foto.
 - * **K2:** Int, és una Key per poder-los agrupar i fer la suma després.
 - * **V2:** Int, quantitat de fotos que hi ha en un document.
 - * **V3:** Tupla, K2 i nombre mitjà.
- **mappingNombrePromRef i reduccingNombrePromRef:** Aquestes dues funcions ens serveixen per a partir de totes les referències que tenen els diferents documents, calcular el nombre mitjà de referències.
 - * **K1:** String, titol del document.
 - * **V1:** String, referència.
 - * **K2:** Int, és una Key per poder-los agrupar i fer la suma després.
 - * **V2:** Int, quantitat de referències que hi ha en un document.
 - * **V3:** Tupla, K2 i nombre mitjà.

E Jocs de proves

E.1 Resultats Execucions Primera Part

E.1.1 Freqüència de paraules

```
Num de Paraules: 30419 Diferents: 3007
Paraules ocurrences frequencia
-----
the 1818 5.98
and 940 3.09
to 809 2.66
a 690 2.27
of 631 2.07
it 610 2.01
she 553 1.82
i 545 1.79
you 481 1.58
said 462 1.52
```

E.1.2 Sense stop-words

```
Num de Paraules: 10038 Diferents: 2623
Paraules ocurrences frequencia
-----
alice 403 4.01
gutenberg 93 0.93
project 87 0.87
queen 75 0.75
thought 74 0.74
time 71 0.71
king 63 0.63
turtle 59 0.59
began 58 0.58
tm 57 0.57
```


E.1.3 Distribució de paraules

```

Les 10 freqüències més freqüents:
1330 paraules apareixen 1 vegades
468 paraules apareixen 2 vegades
264 paraules apareixen 3 vegades
176 paraules apareixen 4 vegades
101 paraules apareixen 5 vegades
74 paraules apareixen 8 vegades
72 paraules apareixen 6 vegades
66 paraules apareixen 7 vegades
39 paraules apareixen 9 vegades
35 paraules apareixen 10 vegades

Les 5 freqüències menys freqüents:
1 paraules apareixen 68 vegades
1 paraules apareixen 178 vegades
1 paraules apareixen 227 vegades
1 paraules apareixen 940 vegades
1 paraules apareixen 100 vegades

```

E.1.4 Ngrams

```

(project gutenber tm,57)
(the mock turtle,53)
(i don t,31)
(the march hare,30)
(said the king,29)
(the project gutenber,29)
(said the hatter,21)
(the white rabbit,21)
(said the mock,19)
(said to herself,19)

```

E.1.5 Vector space model

- Resultat execució ($n = 0$):

	pg11-net.txt	pg11.txt	pg12-net.txt	pg12.txt	pg2500-net.txt	pg2500.txt	pg74-net.txt	pg74.txt
pg11-net.txt	1,000	0,951	0,864	0,831	0,213	0,210	0,219	0,217
pg11.txt	0,951	1,000	0,823	0,876	0,209	0,278	0,214	0,258
pg12-net.txt	0,864	0,823	1,000	0,962	0,202	0,198	0,208	0,207
pg12.txt	0,831	0,876	0,962	1,000	0,202	0,262	0,208	0,247
pg2500-net.txt	0,213	0,209	0,202	0,202	1,000	0,971	0,269	0,268
pg2500.txt	0,210	0,278	0,198	0,262	0,971	1,000	0,265	0,299
pg74-net.txt	0,219	0,214	0,208	0,208	0,269	0,265	1,000	0,988
pg74.txt	0,217	0,258	0,207	0,247	0,268	0,299	0,988	1,000

- Resultat execució ($n = 2$):

	pg11-net.txt	pg11.txt	pg12-net.txt	pg12.txt	pg2500-net.txt	pg2500.txt	pg74-net.txt	pg74.txt
pg11-net.txt	1,000	0,965	0,771	0,748	0,425	0,432	0,582	0,581
pg11.txt	0,965	1,000	0,752	0,792	0,439	0,493	0,585	0,613
pg12-net.txt	0,771	0,752	1,000	0,962	0,443	0,446	0,629	0,626
pg12.txt	0,748	0,792	0,962	1,000	0,453	0,504	0,626	0,651
pg2500-net.txt	0,425	0,439	0,443	0,453	1,000	0,933	0,633	0,636
pg2500.txt	0,432	0,493	0,446	0,504	0,933	1,000	0,644	0,671
pg74-net.txt	0,582	0,585	0,629	0,626	0,633	0,644	1,000	0,990
pg74.txt	0,581	0,613	0,626	0,651	0,636	0,671	0,990	1,000

- Resultat execució ($n = 3$):

	pg11-net.txt	pg11.txt	pg12-net.txt	pg12.txt	pg2500-net.txt	pg2500.txt	pg74-net.txt	pg74.txt
pg11-net.txt	1,000	0,894	0,320	0,284	0,061	0,057	0,209	0,198
pg11.txt	0,894	1,000	0,284	0,434	0,055	0,225	0,185	0,293
pg12-net.txt	0,320	0,284	1,000	0,993	0,071	0,065	0,243	0,230
pg12.txt	0,284	0,434	0,993	1,000	0,065	0,227	0,214	0,313
pg2500-net.txt	0,061	0,055	0,071	0,065	1,000	0,620	0,142	0,134
pg2500.txt	0,057	0,225	0,065	0,227	0,620	1,000	0,139	0,265
pg74-net.txt	0,209	0,185	0,243	0,214	0,142	0,139	1,000	0,943
pg74.txt	0,198	0,293	0,230	0,313	0,134	0,265	0,943	1,000

E.2 Resultats Execucions Segona Part

- numappers = 1, numreducers = 1 i 5000 fitxers:

```
----- MapReduce de Referencias -----
Awaiting result...
-----Resultado-----
Tiempo de ejecución: 1.191846601 segundos
Cantidad de ficheros devueltos: 3033
Titulo: Segunda Guerra Mundial -> 5089 referencias
Titulo: França -> 1907 referencias
Titulo: Alemania -> 1735 referencias
Titulo: 1945 -> 1519 referencias
Titulo: Primera Guerra Mundial -> 1363 referencias
Titulo: Regne Unit -> 1278 referencias
Titulo: 1944 -> 1237 referencias
Titulo: 1941 -> 1114 referencias
Titulo: París -> 1095 referencias
Titulo: Londres -> 998 referencias
```

```
-----Resultado-----
Tiempo de ejecución: 0.0216461 segundos
Numero promedio de referencias: 142.0
----- MapReduce de nombre promedio de todas las fotos -----
Awaiting result...
-----Resultado-----
Tiempo de ejecución: 0.0167531 segundos
Numero promedio de fotos: 6.0
```

```
-----Resultado-----
Numero de documentos: 4692
Tiempo de ejecución: 273.457643001 segundos
Campeones de dobles femenins del Torneig de Roland Garros - Campeones de dobles masculins del Torneig de Roland Garros -> 0.9971373979401987
Campeones de dobles masculins del Torneig de Roland Garros - Campeones de dobles mixts del Torneig de Roland Garros -> 0.9947961234145675
Campeones de dobles femenins del Torneig de Wimbledon - Campeones de dobles mixts del Torneig de Wimbledon -> 0.9938116456236963
Campeones d'individual femeni del Torneig de Wimbledon - Campeons d'individual masculí del Torneig de Wimbledon -> 0.9786429805112784
Viquipèdia:llista dels 1000 articles fonamentals - Viquipèdia:llista dels 1000 articles fonamentals/1-1-09 -> 0.8462952970781231
```

Tiempo total de toda la ejecución: 437.55533560099997 segundos

- numappers = 20, numreducers = 12 i 5000 fitxers:

```

----- MapReduce de Referencias -----
Awaiting result...
-----Resultado-----
Tiempo de ejecución: 1.1024984 segundos
Cantidad de ficheros devueltos: 3033
Titulo: Segona Guerra Mundial -> 5089 referencias
Titulo: França -> 1907 referencias
Titulo: Alemanya -> 1735 referencias
Titulo: 1945 -> 1519 referencias
Titulo: Primera Guerra Mundial -> 1363 referencias
Titulo: Regne Unit -> 1278 referencias
Titulo: 1944 -> 1237 referencias
Titulo: 1941 -> 1114 referencias
Titulo: Paris -> 1095 referencias
Titulo: Londres -> 998 referencias

```

```

-----Resultado-----
Tiempo de ejecución: 0.0226217 segundos
Numero promedio de referencias: 142.0
----- MapReduce de nombre promedio de todas las fotos -----
Awaiting result...
-----Resultado-----
Tiempo de ejecución: 0.0146227 segundos
Numero promedio de fotos: 6.0

```

```

----- MapReduce Similitud documentos -----
Awaiting result...
-----Resultado-----
Numero de documentos: 4493
Tiempo de ejecución: 72.0853306 segundos
Campeones de dobles femenins del Torneig de Roland Garros - Campeons de dobles masculins del Torneig de Roland Garros -> 0.9971385151319517
Campeones de dobles masculins del Torneig de Roland Garros - Campeons de dobles mixts del Torneig de Roland Garros -> 0.9947074244433085
Campeones de dobles femenins del Torneig de Wimbledon - Campeons de dobles mixts del Torneig de Wimbledon -> 0.9938116656236963
Campeones d'individual femení del Torneig de Wimbledon - Campeons d'individual masculí del Torneig de Wimbledon -> 0.8784469123774281
Viquipèdia:llista dels 1000 articles fonamentals - Viquipèdia:llista dels 1000 articles fonamentals/1-1-09 -> 0.844993611871774
----- MapReduce de nombre promedio de referencias todas las paginas -----

```

Tiempo total de toda la ejecución: 160.247642402 segundos

- numappers = 12, numreducers = 8 i 1000 fitxers:

```

----- MapReduce Similitud documentos -----
Awaiting result...
-----Resultado-----
Numero de documentos: 1000
Tiempo de ejecución: 2.525797401 segundos
Partit Democràtic Lliure - Unió Demòcrata Cristiana d'Alemanya -> 0.06979036570366583
Keith Park - Oceania -> 0.028326444778930555
Fred Zinnemann - Gregory Peck -> 0.011460729480043922
Ludwig Wittgenstein - The New Yorker -> 0.009051718059681052
Bundeswehr - Erich Raeder -> 0.008553595541528984
----- MapReduce de nombre promedio de referencias todas las paginas -----

```

Tiempo total de toda la ejecución: 31.156884402000003 segundos

F Taula de rendiment

Per la realització de les diferents proves, s'ha utilitzat l'equip amb les següents especificacions.

F.1 Especificacions

Nombre del dispositivo	DESKTOP-MHA7BS0
Procesador	Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz 2.59 GHz
RAM instalada	16,0 GB (15,8 GB usable)

60 segundos			
Uso	Velocidad	Velocidad de base:	2,59 GHz
9%	3,22 GHz	Sockets:	1
Procesos	Subprocesos	Núcleos:	6
275	3876	Procesadores lógicos:	12
Identificadores		Virtualización:	Habilitado
148879		Caché L1:	384 kB
Tiempo activo		Caché L2:	1,5 MB
24:09:39:23		Caché L3:	12,0 MB

[Monitor de recursos](#)

F.2 Taula dels actors

Les proves s'han realitzat amb tots els fitxers disponibles, i la taula representa per cada número d'actors, quants segons triga.

Número d'actors			
1	4	10	20
437.55	173.42	133.28	160.24

G Altres consideracions

Com s'indiquen a les proves anteriors, s'han realitzat amb tots els fitxers disponibles i s'ha aconseguit un molt bon temps. Per tant, molt contents amb el resultat final.