

Segona pràctica d' Estructures de dades

Matrícula en el GEI

Integrants del Grup:

Bernat Bosca Candel
Albert Cañellas Solé

Grau:

Doble grau Biotecnologia e Enginyeria Informàtica

Professor de Pràctiques:

Maria Ferré Bergadà

Índex

Disseny de la solució del problema proposat	3
Especificació TADLlistaGenerica pre/post	5
Anàlisi de cost de les operacions	6
Anàlisi del temps d'execució de l'estructura de dades	7
Anàlisi del temps d'execució de les diferents operacions	9
Joc de proves	15

Disseny de la solució del problema proposat

Per realitzar la pràctica es va fer un estudi previ, per tal de tenir clar el que ens demanava l'enunciat i poder fer totes les funcions que ens demanaven de la multillista.

La solució plantejada consta d'un programa que deixa triar a l'usuari quina consulta es vol fer de quatre possibilitats:

- A partir del codi d'alumne, mostrar per pantalla les dades de les assignatures que ha matriculat i els crèdits totals.
 - Solució: crear un mètode que retorni una llista de les assignatures cursades per un alumne.
- A partir d'un codi d'assignatura, mostrar per pantalla quins alumnes l'han matriculat i mostrar el numero total d'alumnes matriculats.
 - Solució: crear un mètode que retorni una llista dels alumnes matriculats en aquella assignatura.
- Mostrar per pantalla les dades dels alumnes que han matriculat X crèdits o menys.
 - Solució: amb el mètode anterior que ens retorna una llista de assignatures per cada alumne sumar els crèdits de totes les assignatures cursades i si la suma es igual o menor, escriure la informació d'aquell estudiant per pantalla.
- Mostrar per pantalla les dades de les assignatures que tenen un mínim de Y estudiants matriculats.
 - Solució: amb el mètode anterior que ens retorna una llista d'alumnes per cada assignatura mirem la dimensió de la llista retornada, si la dimensió es major, mostrar la informació del alumne per la pantalla.

En cada consulta es mostrarà el temps que ha tardat en executar-se igual que a l'hora de carregar les dades.

El programa dissenyat consta de 4 paquets:

- Aplicació: conte el main i les funcions principals.
 - Aplicacio: classe que implementa el menú, els mètodes per carregar les dades i els mètodes per la consulta.
- Dades.base: conte les implementacions de les classe base.
 - Alumne: classe que conte la informació a guardar de l'alumne e implementa els mètodes necessaris per gestionar-la.
 - Assignatura: classe que conte la informació a guardar de l'assignatura e implementa els mètodes necessaris per gestionar-la.
 - Matricula: classe que conte la informació que relaciona l'assignatura i l'alumne.
- Dades.Llistes: conte les implementacions de les llistes.
 - LlistaDinamica: classe que implementa tots els mètodes necessaris per la llista dinàmica.
 - Node: nodes necessaris per la implementació de la llista dinàmica.
 - LlistaEstatica: classe que implementa tots els mètodes necessaris per la llista estàtica.
 - LlistaJava: classe opcional de java.util que implementa la crida a tots els mètodes necessaris de la llista.
 - TADLlistaGenerica: interfície dels mètodes generals de les llistes.
- Exceptions: Les excepcions del programa.
 - LlistaPlena: excepció quan la llista esta plena.
 - LlistaBuida: excepció quan la llista esta buida.

Especificació TADLlistaGenerica pre/post

TADLlistaGenerica

```
* Crea una llista buida del tamany indicat (el constructor
* de la classe)
* @pre cert
* @post La llista construïda correspon a la llista buida
Crear (int dimensió);

* Mètode que afegeix un element
* @pre cert
* @post la llista final es la llista inicial afegint elem
* al final
* @error si la llista esta plena no es pot afegir un
* element
void afegirElement (T elem) throws LlistaPlena;

* Mètode que consulta l'element de la posició
* @pre cert
* @post retorna l'element en la posició i
* @error si la llista esta buida no pot retornar un element
T consultarPosicio (int i) throws LlistaBuida;

* Mètode que retorna el numero d'elements
* @pre cert
* @post retorna el numero d'elements
int numElems();

* Mètode toString
* @pre cert
* @post retorna un string que mostra totes les variables de
* la llista
String toString();

* Mètode iterador
* @pre cert
* @post retorna el iterator corresponent
Iterator<T> iterator();
```

Anàlisi de cost de les operacions

Implementació Dinàmica:

- Crear: $\theta(1)$
- AfegirElement: $\theta(2n)$
- ConsultarPosicio: $\theta(2n)$
- NumElems: $\theta(1)$
- toString: $\theta(1)$
- Iterator: $\theta(n)$

Implementació Estàtica:

- Crear: $\theta(1)$
- AfegirElement: $\theta(n)$
- ConsultarPosicio: $\theta(1)$
- NumElems: $\theta(1)$
- toString: $\theta(1)$
- Iterator: $\theta(n)$

Implementació Java:

- Crear: $\theta(1)$
- AfegirElement: $\theta(n)$
- ConsultarPosicio: $\theta(1)$
- NumElems: $\theta(1)$
- toString: $\theta(1)$
- Iterator: $\theta(n)$

Anàlisi del temps d'execució de l'estructura de dades

S'han analitzat els diferents temps d'execució en els diferents tipus d'estructures de dades per a implementar les llistes (Estàtica, Dinàmica i implementada per Java), i amb aquestes implementar la multillista.

S'ha realitzat la mitja de 3 temps de construcció de les estructures de dades, és a dir, el que es tarda en carregar les llistes i la multillista. Hem realitzat la mitja de 3 temps a cada tipus de implementació, per obtenir un valor temporal més real.

I com dades per construir l'estructura de dades hem emprat el document:

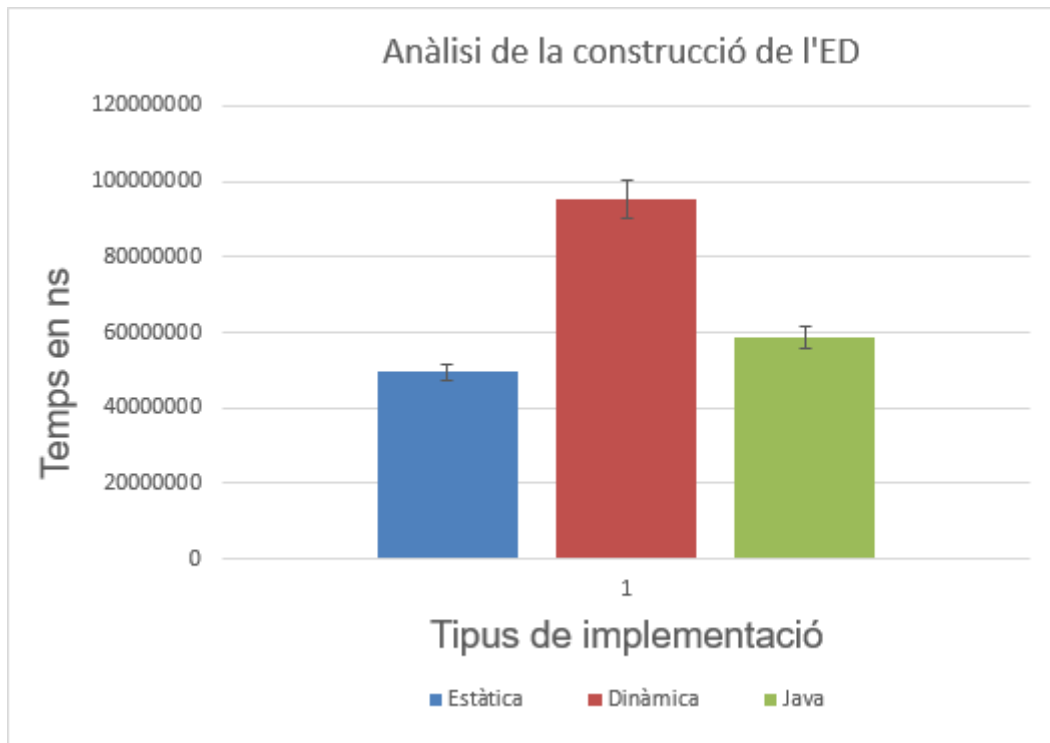
DadesMatricula.csv

NOTA: El document 'text.txt' sols l'hem creat per realitzar les proves necessàries per al correcte desenvolupament de les operacions demanades. No per al anàlisi dels costos.

Hem obtingut les següents dades:

nº intent	Estàtica	Dinàmica	Java
1	47265297	96659323	58970742
2	49946077	99273874	55676382
3	51306549	89452324	61390023
Mitja	49505974	95128507	58679049
Desviació	2056258	5086576	2867967

I hem realitzat el següent gràfic comparatiu amb aquestes:



Com a conclusions d'aquesta comparació podem dir que:

- El cost per a construir les estructures de dades mínim és amb la implementació estàtica.
- La implementació Java la podem considerar gairebé com l'estàtica, ja que la implementació escollida del `java.collection` és una implementació estàtica, i la diferencia entre aquesta implementació i la estàtica nostra pot deures les operacions que hem de fer per a que quadre la interfície amb l'estructura del `java.collection`.
- El tipus de llista que major cost té al contruir l'estructura de dades és la dinàmica ja que ha de crear i anar modificant per a que queden ordenades gran quantitat de referències.
- El cost promig de la llista dinàmica es quasi el doble que el cost promig de la llista estàtica. Això es deu al tenir que desplaçar el punter de la classe (`elem`) fins l'última posició sempre, i en el pijor dels casos podem tenir un cost n per a desplaçar-lo des de la primera posició fins a l'última i després en buscar la posició del nou element tornar a recórrer tota la llista ja que li corresponia la primera posició obtenint un cost final de $2n$, el doble que en l'estàtica.

Anàlisi del temps d'execució de les diferents operacions

S'han analitzat les diferents operacions del menú amb 3 intents per fer la mitja del temps cadascuna, en cada tipus de implementació.

I com dades per construir l'estructura de dades hem emprat el document:
DadesMatricula.csv

En la primera opció del menú: hem utilitzat sempre el mateix codi d'Alumne, el següent: URV_0006. I obtingut les següents dades:

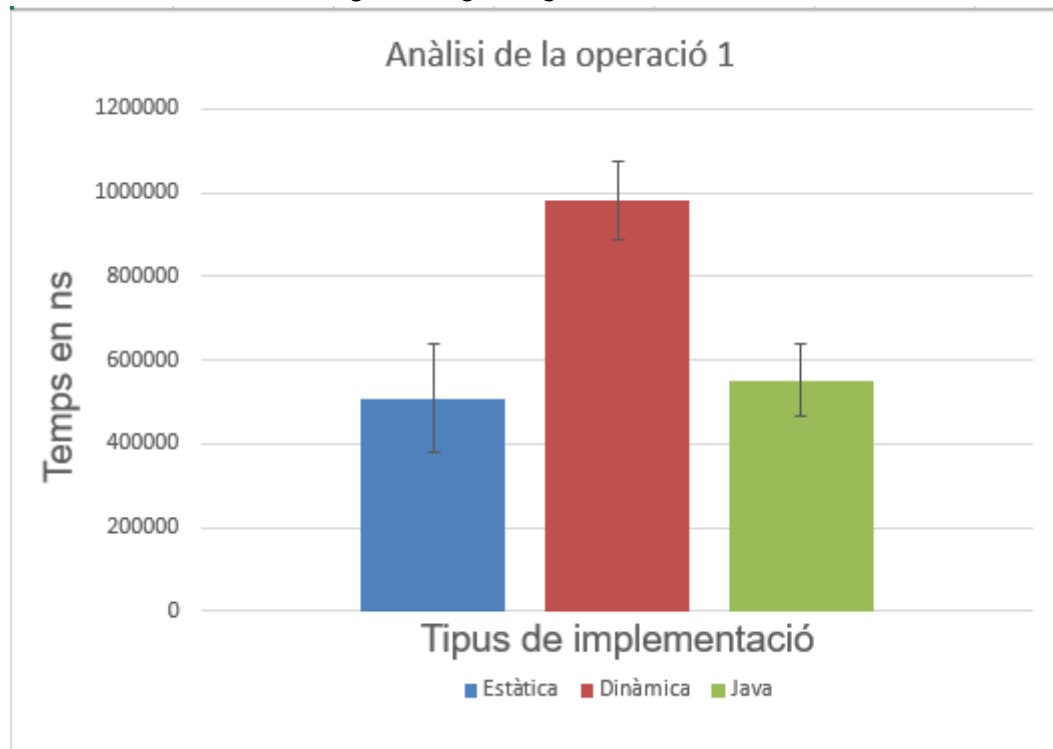
nº intent	Estàtica	Dinàmica	Java		Operació 1
1	637081	1089146	624262		URV_0006
2	508895	929343	572988		
3	379428	927206	455912		
Mitja	508468	981898	551054		
Desviació	128827	92885	86292		

I per consola el resultat independentment del tipus d'implementació és:

1. Assignatura: ARQUITECTURA DE COMPUTADORS. Amb codi: 17234109, credits: 6, curs: 3, quadrimestres: 1
2. Assignatura: ESTRUCTURA DE SISTEMES OPERATIUS. Amb codi: 17234113, credits: 6, curs: 3, quadrimestres: 1
3. Assignatura: INTERACCIO PERSONA-ORDINADOR. Amb codi: 17234104, credits: 6, curs: 3, quadrimestres: 1
4. Assignatura: TECNIQUES AVANÇADES DE PROGRAMACIO. Amb codi: 17234117, credits: 6, curs: 3, quadrimestres: 1
5. Assignatura: LLENGUATGES FORMALS. Amb codi: 17234110, credits: 6, curs: 3, quadrimestres: 2
6. Assignatura: SEGURETAT EN XARXES. Amb codi: 17234120, credits: 6, curs: 3, quadrimestres: 2
7. Assignatura: SISTEMES DISTRIBUITS. Amb codi: 17234106, credits: 6, curs: 3, quadrimestres: 2

Número de crèdits totals: 42

I com a resultat hem obtingut el següent gràfic:



En la segona opció del menú: hem utilitzat sempre el mateix codi d'Assignatura, el següent: 17234006. I obtingut les següents dades:

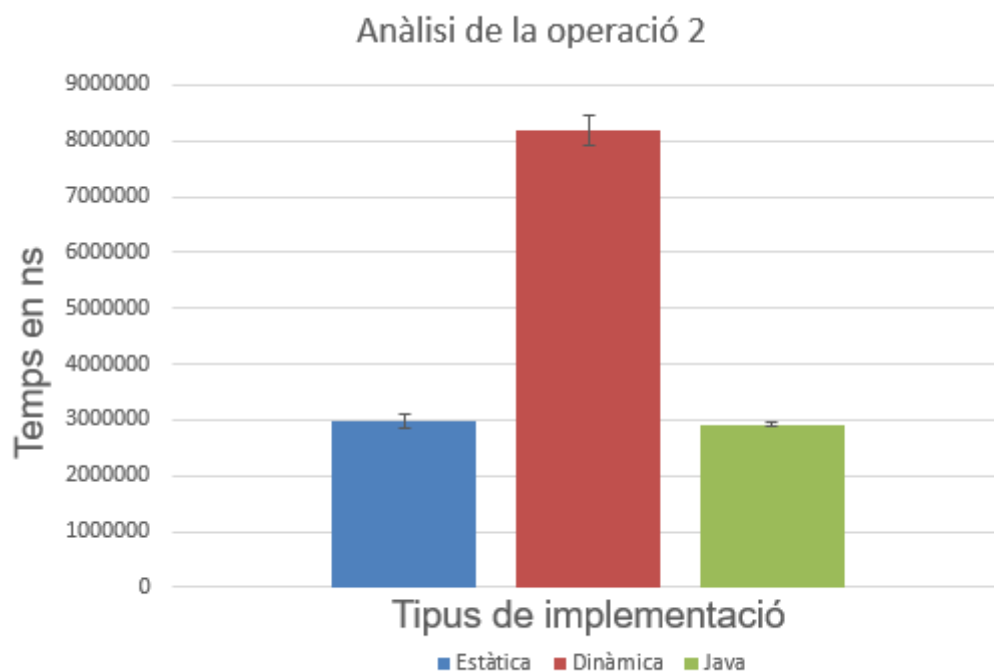
nº intent	Estàtica	Dinàmica	Java		Operació 2
1	3124728	8031380	2956378		17234006
2	2901258	8492850	2908522		
3	2902968	8039501	2900147		
Mitja	2976318	8187910	2921682		
Desviació	128530	264117	30338		

I per consola el resultat independentment del tipus d'implementació és:

```
55. Alumne: JOAN. Amb codi: URV_0175
56. Alumne: JOAN. Amb codi: URV_0190
57. Alumne: JOEL. Amb codi: URV_0054
58. Alumne: JORDI. Amb codi: URV_0021
59. Alumne: JORDI. Amb codi: URV_0187
60. Alumne: JOSEP. Amb codi: URV_0125
61. Alumne: JOSEP. Amb codi: URV_0192
62. Alumne: JULIA. Amb codi: URV_0072
63. Alumne: JULIAN. Amb codi: URV_0081
64. Alumne: LIDIA. Amb codi: URV_0170
65. Alumne: MANEL. Amb codi: URV_0141
66. Alumne: MIQUEL. Amb codi: URV_0109
67. Alumne: OCTAVI. Amb codi: URV_0095
68. Alumne: ORIOL. Amb codi: URV_0039
69. Alumne: ORIOL. Amb codi: URV_0064
70. Alumne: ORIOL. Amb codi: URV_0143
71. Alumne: PABLO. Amb codi: URV_0176
72. Alumne: RAUL. Amb codi: URV_0166
73. Alumne: SERGI. Amb codi: URV_0063
74. Alumne: SERGI. Amb codi: URV_0074
75. Alumne: SERGI. Amb codi: URV_0133
76. Alumne: SERGI. Amb codi: URV_0154
77. Alumne: SERGI. Amb codi: URV_0174
78. Alumne: SERGIO. Amb codi: URV_0003
79. Alumne: SERGIO. Amb codi: URV_0017
80. Alumne: VICENTE. Amb codi: URV_0094
81. Alumne: VICTOR. Amb codi: URV_0075
82. Alumne: VICTOR. Amb codi: URV_0196
83. Alumne: XAVIER. Amb codi: URV_0119
84. Alumne: XAVIER. Amb codi: URV_0188
```

Total d'alumnes: 84

I com a resultat hem obtingut el següent gràfic:



En la tercera opció del menú: hem utilitzat sempre el mateix número de crèdits 20 o menys dels que pot estar matriculat un alumne. I obtingut les següents dades:

nº intent	Estàtica	Dinàmica	Java		Operació 3
1	2520548	8336310	2729824		20
2	2887585	8781431	2033063		
3	2088137	7979528	2286016		
Mitja	2498757	8365756	2349634		
Desviació	400169	401762	352710		

I per consola el resultat independentment del tipus d'implementació és:

1. Alumne: CRISTIAN. Amb codi: URV_0155 i crèdits totals: 18
2. Alumne: JUAN JAVIER. Amb codi: URV_0079 i crèdits totals: 12
3. Alumne: MANUEL ALEJANDRO. Amb codi: URV_0128 i crèdits totals: 12
4. Alumne: PAU. Amb codi: URV_0059 i crèdits totals: 18

I com a resultat hem obtingut el següent gràfic:



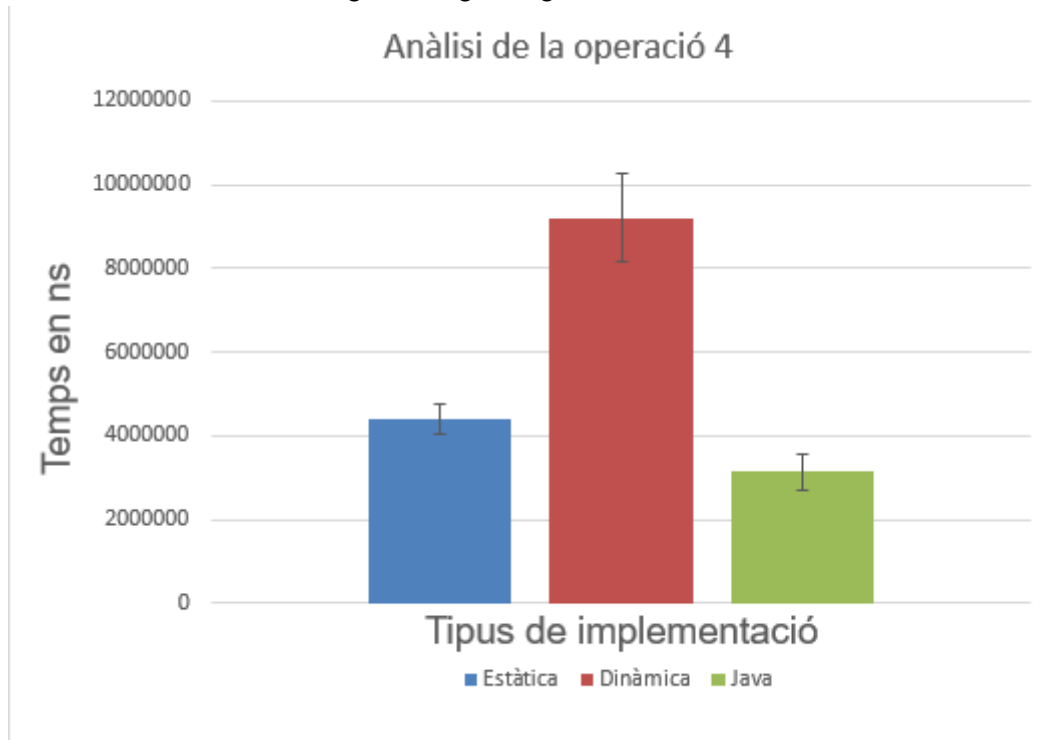
En la quarta opció del menú: hem utilitzat sempre el mateix número d'Alumnes 15 mínim que ha de tenir una assignatura. I obtingut les següents dades:

nº intent	Estàtica	Dinàmica	Java		Operació 4
1	4688587	9138750	3184547		15
2	4527928	8198725	3539193		
3	3975450	10300962	2705135		
Mitja	4397322	9212812	3142958		
Desviació	374078	1053074	418581		

I per consola el resultat independentment del tipus d'implementació és:

6. Assignatura: FONAMENTS DE COMPUTADORS. Amb codi: 17234002, credits: 6, curs: 1, quadrimestres: 1 i estudiants totals: 76
7. Assignatura: FONAMENTS DE PROGRAMACIO. Amb codi: 17234001, credits: 6, curs: 1, quadrimestres: 1 i estudiants totals: 68
8. Assignatura: ANALISI MATEMATICA II. Amb codi: 17234006, credits: 6, curs: 1, quadrimestres: 2 i estudiants totals: 84
9. Assignatura: ANGLES TECNIC. Amb codi: 17234102, credits: 6, curs: 1, quadrimestres: 2 i estudiants totals: 60
10. Assignatura: FISICA II. Amb codi: 17234004, credits: 6, curs: 1, quadrimestres: 2 i estudiants totals: 66
11. Assignatura: FONAMENTS DE COMPUTADORS. Amb codi: 17234002, credits: 6, curs: 1, quadrimestres: 2 i estudiants totals: 76
12. Assignatura: COMPUTADORS. Amb codi: 17234107, credits: 6, curs: 2, quadrimestres: 1 i estudiants totals: 57
13. Assignatura: FONAMENTS DE SISTEMES OPERATIUS. Amb codi: 17234112, credits: 6, curs: 2, quadrimestres: 1 i estudiants totals: 48
14. Assignatura: MATEMATICA DISCRETA I. Amb codi: 17234009, credits: 6, curs: 2, quadrimestres: 1 i estudiants totals: 41
15. Assignatura: METODOLOGIES DE LA PROGRAMACIO. Amb codi: 17234116, credits: 6, curs: 2, quadrimestres: 1 i estudiants totals: 38
16. Assignatura: PROGRAMACIO. Amb codi: 17234114, credits: 6, curs: 2, quadrimestres: 1 i estudiants totals: 40
17. Assignatura: BASES DE DADES. Amb codi: 17234103, credits: 6, curs: 2, quadrimestres: 2 i estudiants totals: 58
18. Assignatura: ESTADISTICA. Amb codi: 17234011, credits: 6, curs: 2, quadrimestres: 2 i estudiants totals: 49
19. Assignatura: ESTRUCTURA DE COMPUTADORS. Amb codi: 17234108, credits: 6, curs: 2, quadrimestres: 2 i estudiants totals: 51
20. Assignatura: ESTRUCTURES DE DADES. Amb codi: 17234115, credits: 6, curs: 2, quadrimestres: 2 i estudiants totals: 65
21. Assignatura: MATEMATICA DISCRETA II. Amb codi: 17234010, credits: 6, curs: 2, quadrimestres: 2 i estudiants totals: 39
22. Assignatura: ARQUITECTURA DE COMPUTADORS. Amb codi: 17234109, credits: 6, curs: 3, quadrimestres: 1 i estudiants totals: 36
23. Assignatura: ESTRUCTURA DE SISTEMES OPERATIUS. Amb codi: 17234113, credits: 6, curs: 3, quadrimestres: 1 i estudiants totals: 28
24. Assignatura: INTERACCIO PERSONA-ORDINADOR. Amb codi: 17234104, credits: 6, curs: 3, quadrimestres: 1 i estudiants totals: 34
25. Assignatura: TECNiques AVANCADES DE PROGRAMACIO. Amb codi: 17234117, credits: 6, curs: 3, quadrimestres: 1 i estudiants totals: 23
26. Assignatura: XARXES DE DADES. Amb codi: 17234118, credits: 6, curs: 3, quadrimestres: 1 i estudiants totals: 31
27. Assignatura: ANALISI I DISSENY D'APLICACIONS. Amb codi: 17234105, credits: 6, curs: 3, quadrimestres: 2 i estudiants totals: 17
28. Assignatura: GESTIO DE SISTEMES I XARXES. Amb codi: 17234119, credits: 6, curs: 3, quadrimestres: 2 i estudiants totals: 27
29. Assignatura: LLENGUATGES FORMALS. Amb codi: 17234110, credits: 6, curs: 3, quadrimestres: 2 i estudiants totals: 27
30. Assignatura: SEGURETAT EN XARXES. Amb codi: 17234120, credits: 6, curs: 3, quadrimestres: 2 i estudiants totals: 28
31. Assignatura: MODELITZACIO I VISUALITZACIO. Amb codi: 17234213, credits: 6, curs: 4, quadrimestres: 1 i estudiants totals: 21
32. Assignatura: PROJECTES DE SISTEMES INFORMATICS. Amb codi: 17234111, credits: 6, curs: 4, quadrimestres: 1 i estudiants totals: 16
33. Assignatura: SISTEMES DE TEMPS REAL. Amb codi: 17234124, credits: 6, curs: 4, quadrimestres: 1 i estudiants totals: 20
34. Assignatura: COMPUTACIO PARAL·LELA I MASSIVA. Amb codi: 17234129, credits: 6, curs: 4, quadrimestres: 2 i estudiants totals: 22
35. Assignatura: SISTEMES D'INFORMACIO EN LES ORGANITZACIONS. Amb codi: 17234122, credits: 6, curs: 4, quadrimestres: 2 i estudiants totals: 20
36. Assignatura: SISTEMES DE COMERÇ ELECTRONIC. Amb codi: 17234211, credits: 6, curs: 4, quadrimestres: 2 i estudiants totals: 21

I com a resultat hem obtingut el següent gràfic:



Com a conclusions d'aquestes comparacions entre els diferents tipus d'implementacions a les 4 possibles consultes, podem dir que:

- Cada operació té un cost diferent de les altres operacions, però totes coincideixen en que el tipus d'estructura de dades més ineficient és la llista dinàmica.
- El fet de que la llista dinàmica sigui la més ineficient és deu a que per consultar un element de la llista a les altres dos accedim directament a la posició de memòria directament d'on es troba l'element, però en canvi a la llista dinàmica ens hem de recórrer tots els elements(en el pitjor dels casos) de la llista a través de les referències fins aplegar al element que volem.
- No hi ha gairebé diferència entre la implementació de la llista estàtica implementada per nosaltres amb la que ja té implementada Java.
- Sempre i quan no ens trobem limitats per la memòria i necessitem una aplicació d'aquest estil serà millor utilitzar llistes estàtiques que no pas, llistes dinàmiques.

Joc de proves

Hem realitzat les següents proves:

Prova	Resultat
Si el fitxer no existeix?	Demana el nom de un fitxer que existeixi.
Si el valor del tipus de llista es incorrecte?	Demana un nou valor, fins que es correspongui amb alguna implementació.
Opció incorrecta en el menú	Demana una nova opció.
Menú correcte	Realitza les opcions pertinents fins que es selecciona la opció 5, sortir del menú.
Codi del alumne correcte?	Comprova que el codi del alumne comença per URV_0 i contingui altres 3 valors numèrics (sols 3 valors, més és considera incorrecte també).
I si aquest 3 dígit no es corresponen en cap alumne?	Dirà que el alumne no es troba matriculat en cap assignatura de la URV.
Codi assignatura correcte?	És considera que el codi de una assignatura és correcte sempre que comença per 17234 i contingui altres 3 valors numèrics (sols 3 valors, més és considera incorrecte també).
I si aquest 3 dígit no es corresponen en cap assignatura?	Dirà que aquesta assignatura no existeix o no té matriculat cap alumne.
Al demanar els crèdits màxims que pot tenir un alumne?	Comprova que el valor introduït sigui numèric, sinó ho és demana un nou valor.
Si el valor és correcte?	Mostra les dades dels alumnes que tenen aquest valor de crèdits o menys. I si no hi ha cap, mostra un missatge dient que no hi ha cap alumne amb X crèdits o menys.
Al demanar els alumnes mínims que pot tenir una assignatura?	Comprova que el valor introduït sigui numèric, si no ho és demana un nou valor.
Si el valor és correcte?	Mostra les dades de les assignatures que tenen aquest valor d'alumnes com a mínim. I si no hi ha cap, mostra un missatge dient que no hi ha cap assignatura amb un mínim de X alumnes.
Es repeteix el menú?	Sempre que no s'introdueixi un valor correcte. I després de realitzar una opció que no sigui la de sortir(5).

Warnings al JavaDoc	Tots solucionats
Opera correctament amb LlistaEstàtica?	Si
Opera correctament amb LlistaDinàmica?	Si
Opera correctament amb LlistaJava?	Si
Si consultem el nombre d'alumnes que es troben matriculats d'un màxim de 800 crèdits?	Els mostra tots, per tant, és correcte.
Si consultem el nombre d'assignatures que tenen un mínim de 0 alumnes matriculats?	Les mostra totes, per tant, és correcte.