

HERRAMIENTA DE DEMOSTRACIÓN Y AUTOAPRENDIZAJE PARA ESTRUCTURAS DE DATOS

Carlos Herrero

Departamento de Sistemas Informáticos y Computación

Universidad Politécnica de Valencia

Resumen

Se presenta una herramienta de autoaprendizaje para la asignatura “Estructuras de datos y algoritmos” de la Escuela Superior de Informática Aplicada en la Universidad Politécnica de Valencia. En esta aplicación el alumno puede ver representadas de manera gráfica y pormenorizada las estructuras que a lo largo del curso se describen e implementan en la asignatura. Además, puede interactuar con las instrucciones habituales de inserción y borrado o, en el caso de la estructura Grafo, efectuar los recorridos más habituales y comprobar su resultado.

Palabras Clave: *autoaprendizaje, demostración, programación, interactivo.*

1. Introducción

La asignatura de Estructuras de Datos y Algoritmos es una asignatura anual y obligatoria dentro del segundo curso de las dos titulaciones de Ingeniería Técnica en Informática de Gestión e Ingeniería Técnica en Informática de Sistemas de la Escuela Superior de Informática Aplicada. Consta de 12 créditos (6 Teoría, 3 problemas y 3 prácticas)

Pese a que no existen prerequisites para acceder a su matriculación, sí que se recomienda haber cursado y superado al menos las asignaturas de Programación, Análisis Matemático y Matemática Discreta y Álgebra de primer curso. Los objetivos de la asignatura consisten entre otros en mostrar los fundamentos de las estructuras de datos más habituales y su implementación en Java [1][2][3][4][5] y [6]

En este centro se lleva a cabo un proyecto de innovación docente destinado a fomentar la participación y evitar el absentismo que una asignatura de carácter anual y con la envergadura de esta tiende a presentar. Para ello se establece el denominado “Contrato de Aprendizaje” que obliga a los alumnos a mantener la asistencia a las clases tanto de teoría como de prácticas incidiendo en la exigencia de su aprovechamiento con la obtención de una nota mínima en cada parte para el mantenimiento de la vigencia de dicho contrato en los términos en que fue inscrito. Así pues, se realiza una evaluación continua representada por tres parciales a lo largo del curso evaluando la teoría y dos pruebas de aptitud para evaluar la parte práctica. Esta evaluación por contrato de aprendizaje se desarrolla de forma paralela a la evaluación habitual con un examen final en junio de teoría y una prueba de prácticas, que sólo deben realizar los alumnos que no suscriban el contrato de aprendizaje o que lo hayan abandonado a lo largo del curso de manera voluntaria o por incumplimiento de sus normas.

Además, la asignatura EDA es también pionera en el uso de la plataforma Poliformat en la presentación de contenidos para el autoaprendizaje y como medio de comunicación entre los alumnos y profesores, con el uso del chat y de los foros de discusión.

Dentro de todas estas actividades de innovación, el autor desarrolló como experiencia piloto en la propia aula una aplicación que permitía a los alumnos comprobar el funcionamiento de sus estructuras de datos así como ver que las clases java por ellos diseñadas se comportaban tal y como se esperaba de las mismas. En particular, se pretendía que los detalles de la implementación que quedan ocultos a nivel de modelos de estructuras de datos, quedasen patentes en una representación gráfica de las estructuras internas que implementan dichos modelos, para que en un momento dado, el alumno pudiese comprobar que la representación real concidía con lo que teóricamente debiera ocurrir. Dicho de otro modo, que en un ejercicio sobre

papel de una traza de ejecución, tuviesen una herramienta que les permitiese no sólo aprender el funcionamiento real de su programa, sino también comprobar la corrección de sus resultados.

2. La Herramienta Mostrador.

2.1 Lenguaje y Formatos de la herramienta.

Esta primera versión, que podría denominarse la versión Beta de la aplicación, no busca la representación estética de las diferentes estructuras de datos representadas, sino que su objetivo primordial es dar a conocer de manera sencilla su “aspecto” real en cuanto a datos y referencias.

Por esta razón, no se ha escogido un lenguaje más adecuado para la representación como una presentación flash o un video interactivo, sino una aplicación Java que utiliza las propias clases que definen los alumnos y que usan en su aprendizaje y prácticas. Además, la intención es que los alumnos puedan utilizar la herramienta de manera independiente de la plataforma o sistema operativo que utilicen normalmente y por tanto se optó por ejecutar un archivo “.jar” desde la utilidad Java Web Start, [5] que permite a los usuarios trabajar de manera local con tan sólo el intérprete java instalado (cosa que por otra parte necesitan para poder realizar las prácticas y los ejercicios). Por añadidura, esta utilidad tiene una interesante característica y es que permite actualizar el archivo de manera transparente al usuario, así, si se añaden nuevas características a la aplicación, el alumno al abrirla verá cómo el programa se ha actualizado automáticamente. Además, para una de las opciones de la aplicación es precisa la utilización de un fichero de datos que realizará el alumno. Para ello es necesario que se acepte un certificado emitido por el autor y que autoriza a abrir un fichero local en la ubicación que el usuario solicite.

2.2 La Herramienta.

La herramienta tiene un aspecto simple y carente de artificios . En principio se presenta como una ventana interactiva, con una barra de menú y un mensaje de bienvenida indicando al usuario que escoja la opción para empezar a trabajar. con el siguiente menú de opciones

El menú principal de la herramienta (Fig. 1) permite acceder a nueve opciones diferentes, que coinciden con nueve representaciones de diversos modelos de Estructuras de datos. A saber:

- Pila, (con un array de datos)
- Cola, (con array de datos en versión de array circular)
- Lista con Punto de interés, (con lista enlazada y tres referencias)
- Cola de Prioridad, (con un Árbol Binario de Búsqueda con elementos repetidos)
- Diccionario, (con un Árbol Binario de Búsqueda sin elementos repetidos)
- Cola de Prioridad,(con un Montículo Binario o Heap en un array).
- Diccionario, (con una tabla hash con exploración lineal)
- Diccionario, (con una tabla hash con exploración cuadrática)
- Grafo, con su representación por lista de adyacencia y por matriz de adyacencia.

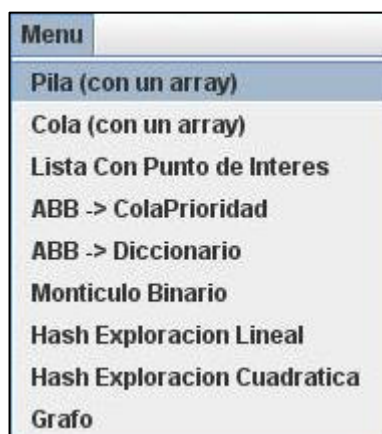


Fig. 1: Menú de opciones.

2.3 La Opción Pila(con un array)

Una pila con un array se representa con los datos situados en posiciones crecientes del mismo. Por simplicidad, los datos son siempre objetos de la clase Integer, y para facilitar su introducción, en la parte inferior de la pantalla se facilitan una serie de botones con los enteros del 0 al 149. En todo momento se muestra el array (con sus índices) los valores enteros que contiene, y la posición del tope de la pila, así como el resultado de las operaciones consultoras “esVacia()” y “tope()”. También se proporciona un botón para desapilar.

En (Fig 2) puede verse el resultado de haber introducido los valores 3, 31, 57, 33, 58 y 60 en una pila, con el consultor “esVacia()” a falso y el consultor “tope()” con un valor 60

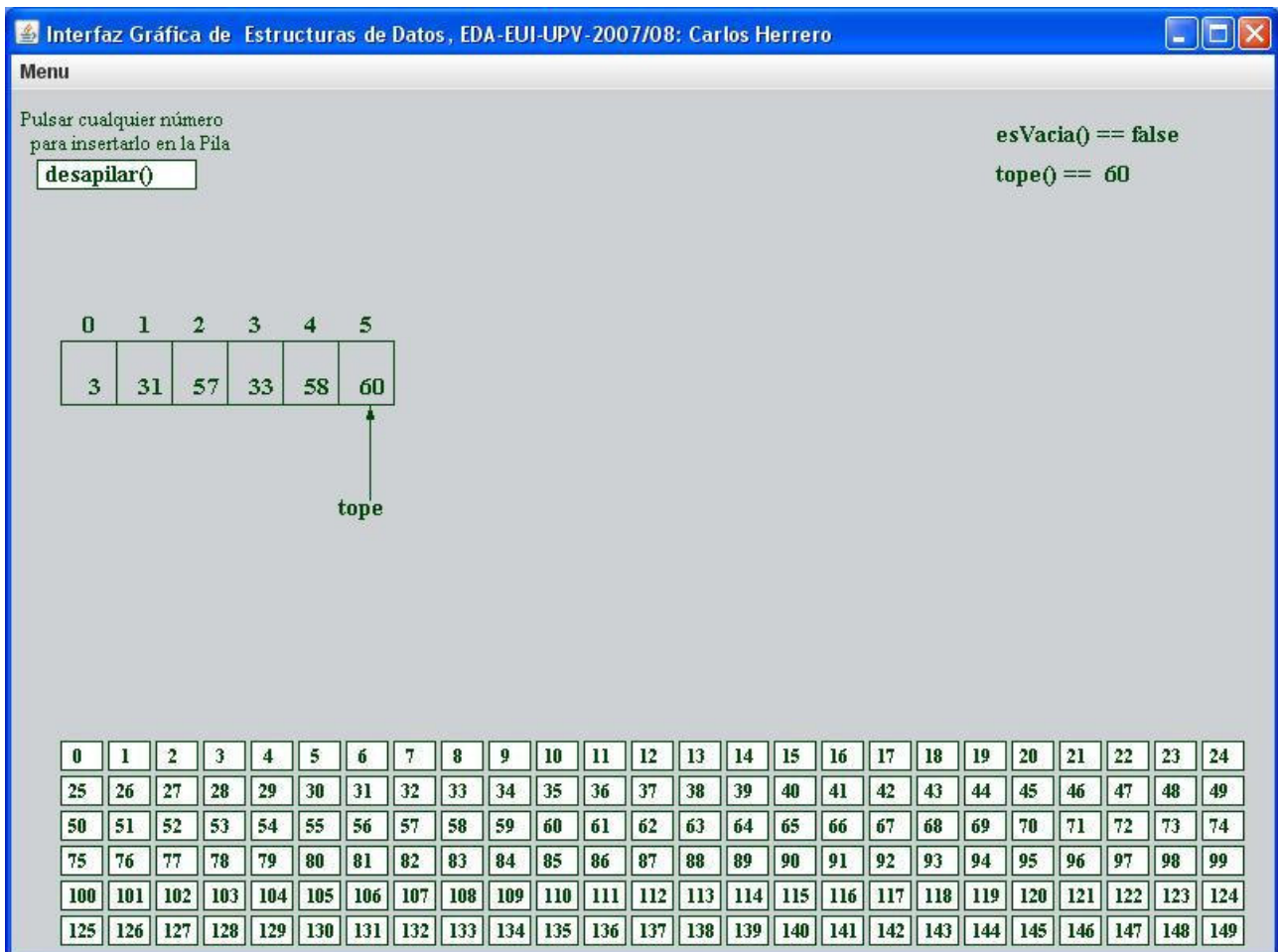


Fig. 2: Pila con un Array.

En esta pantalla, el alumno tiene completa autonomía para interactuar con la pila, incluso puede provocar posibles errores en la aplicación, como tratar de desapilar elementos cuando la pila esté vacía. En este caso, la aplicación muestra una ventana emergente con el mensaje de error correspondiente (ver Fig. 3):

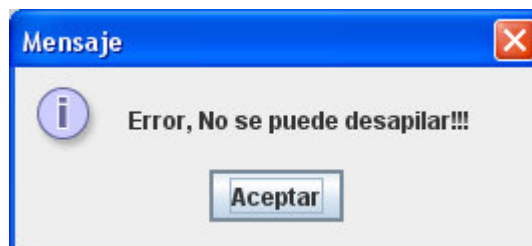


Fig. 3: Error al tratar de desapilar una pila vacía.

2.4 La Opción Cola(con un array)

Una cola con un array se representa simulando un array circular para evitar desplazamientos en el caso de que se eliminen elementos. Para ello, se considera que el siguiente a la última posición es la posición 0 del array. Representado este hecho, se dibuja el array con sus índices de manera circular. Cuando se añaden elementos a la cola, se incrementa el índice "ult", y cuando se eliminan elementos de la cola se incrementa el índice "pri". Por simplicidad, los datos son siempre objetos de la clase Integer, y para facilitar su introducción, se presentan los mismos botones que en la opción previa. En todo momento se muestra el array (con sus índices) los valores enteros que contiene, y las posiciones de "pri" y "ult" así como el resultado de las operaciones consultoras "esVacia()" y "tope()". También se proporciona un botón para "desencolar".

En (Fig 4) puede verse el resultado de haber introducido en una cola los valores 10,7,62,61,62,35,58 y habiendo "desencolado" dos elementos con el consultor "esVacia()" a falso y el consultor "primero()" con un valor 62

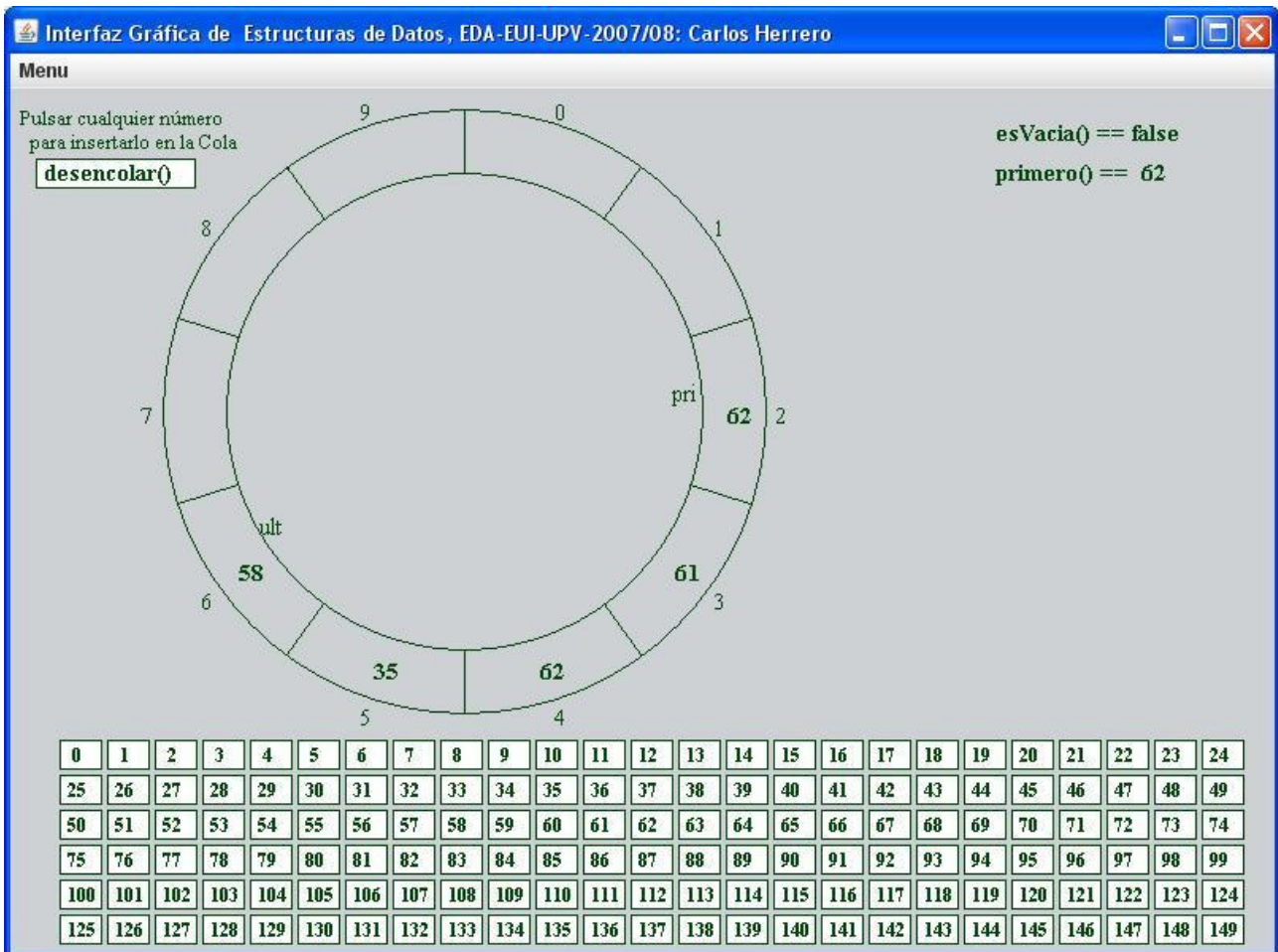


Fig. 4: Cola con un array.

Igual que antes si el alumno trata de eliminar un elemento de una cola vacía aparece el mensaje de error (Fig. 5)

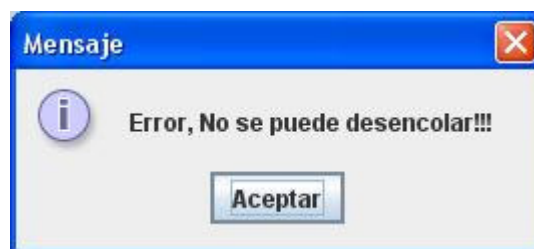


Fig. 5: Error al tratar de "desencolar" una cola vacía.

2.5 La Opción Lista con Punto de Interés

Una cola con un array se representa con una representación enlazada con un nodo inicial vacío (ficticio) y tres referencias, el primero que apunta al nodo ficticio, el segundo una referencia al nodo anterior al del punto de interés (para facilitar inserciones y borrados) y por último otra referencia al último nodo de la estructura. Al igual que en la cola y la pila los datos son Integer y se facilitan botones para introducirlos. Además, se presentan más botones en la línea de menú con el resto de operaciones de la estructura de datos.

En (Fig 6) puede verse el resultado de haber introducido en una lista con punto de interés secuencialmente los valores 63,37, 60,61,111, habiendo pulsado el botón inicio, y seguidamente introduciendo los secuencialmente los valores 30,56,32,39. y habiendo pulsado "siguiente" dos veces más. Se presenta el estado inicial de la lista, los botones "borrar()", "inicio()", "fin()", "siguiente()", y las consultoras "esfin()", "esVacia()" y "recuperar()"

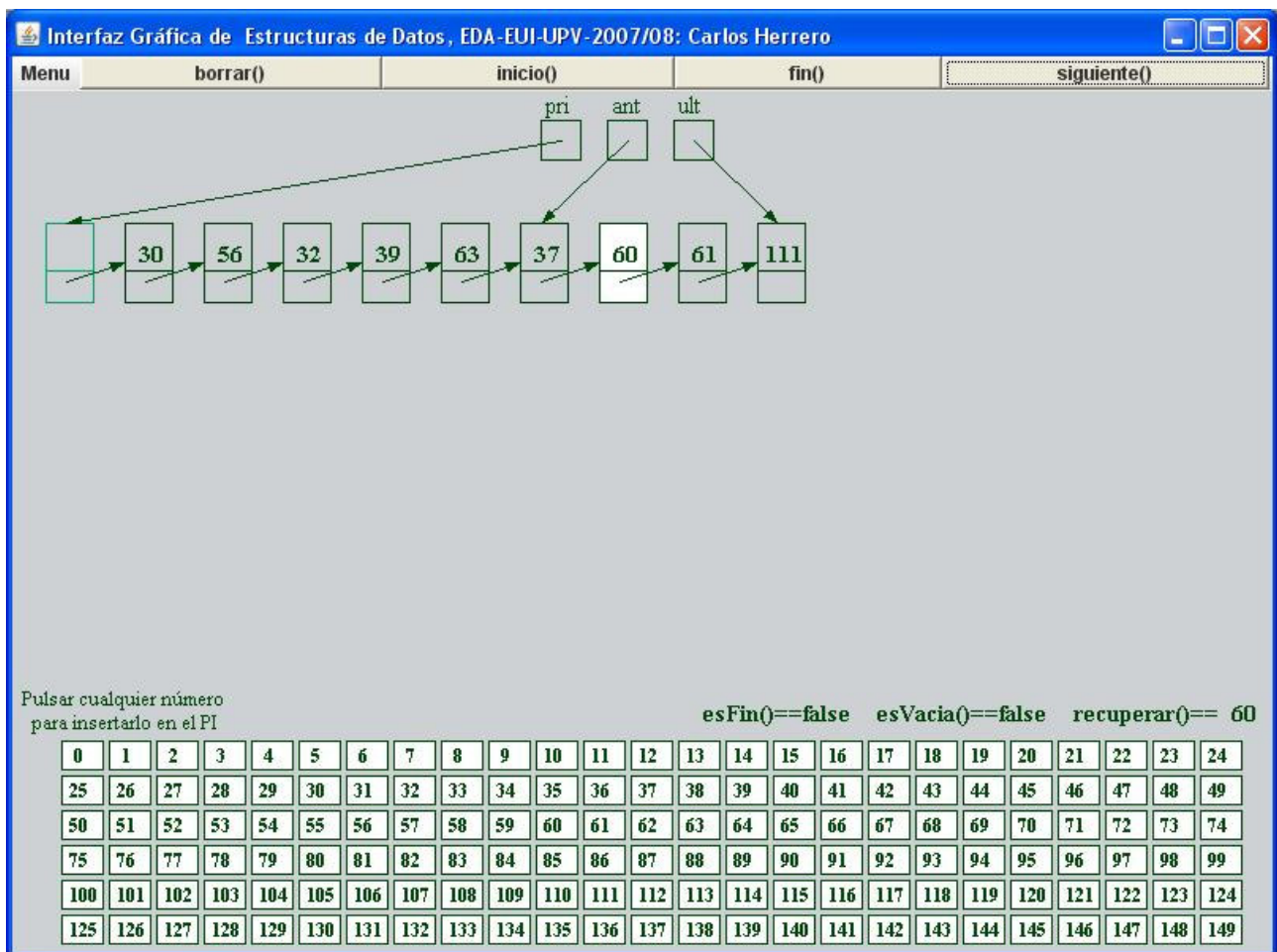


Fig. 6: Lista con Punto de Interés.

Si el alumno tratase de eliminar en una posición incorrecta o avanzar más allá de la última posición, la aplicación abriría ventanas emergentes con sendos mensajes de error indicándolo lo que pueden verse en (Fig.7)

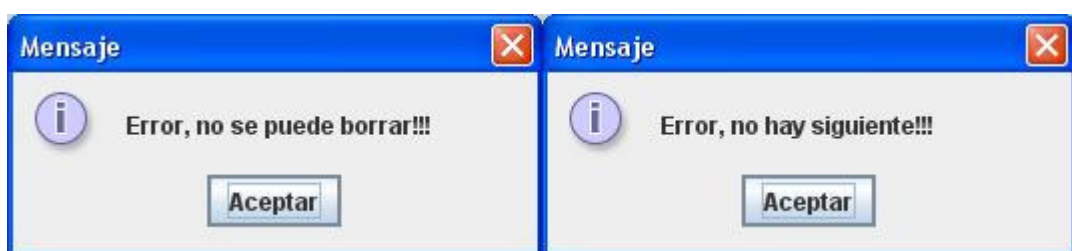


Fig. 7: Errores al tratar de eliminar o avanzar el punto de interés

2.6 La Opción ABB (Cola de Prioridad)

Una cola de prioridad puede representarse como un Árbol Binario de Búsqueda (ABB) con elementos repetidos y las operaciones de insertar() y eliminarMin(). Al igual que en las EDAs anteriores los datos son objetos de la clase Integer y se facilitan botones para introducirlos. También se facilita a un alumno un botón para eliminar el mínimo y consultores para “esVacia()” y para “buscarMin()”

En (Fig 8) puede verse el resultado de haber introducido en una Cola de Prioridad con un ABB secuencialmente los valores 11, 6, 35, 2, 10, 18, 39, 0, 3, 18, 29, 45 y 18. Los consultores “esVacia()” Y “buscarMin()” muestran los valores falso y 0 respectivamente.

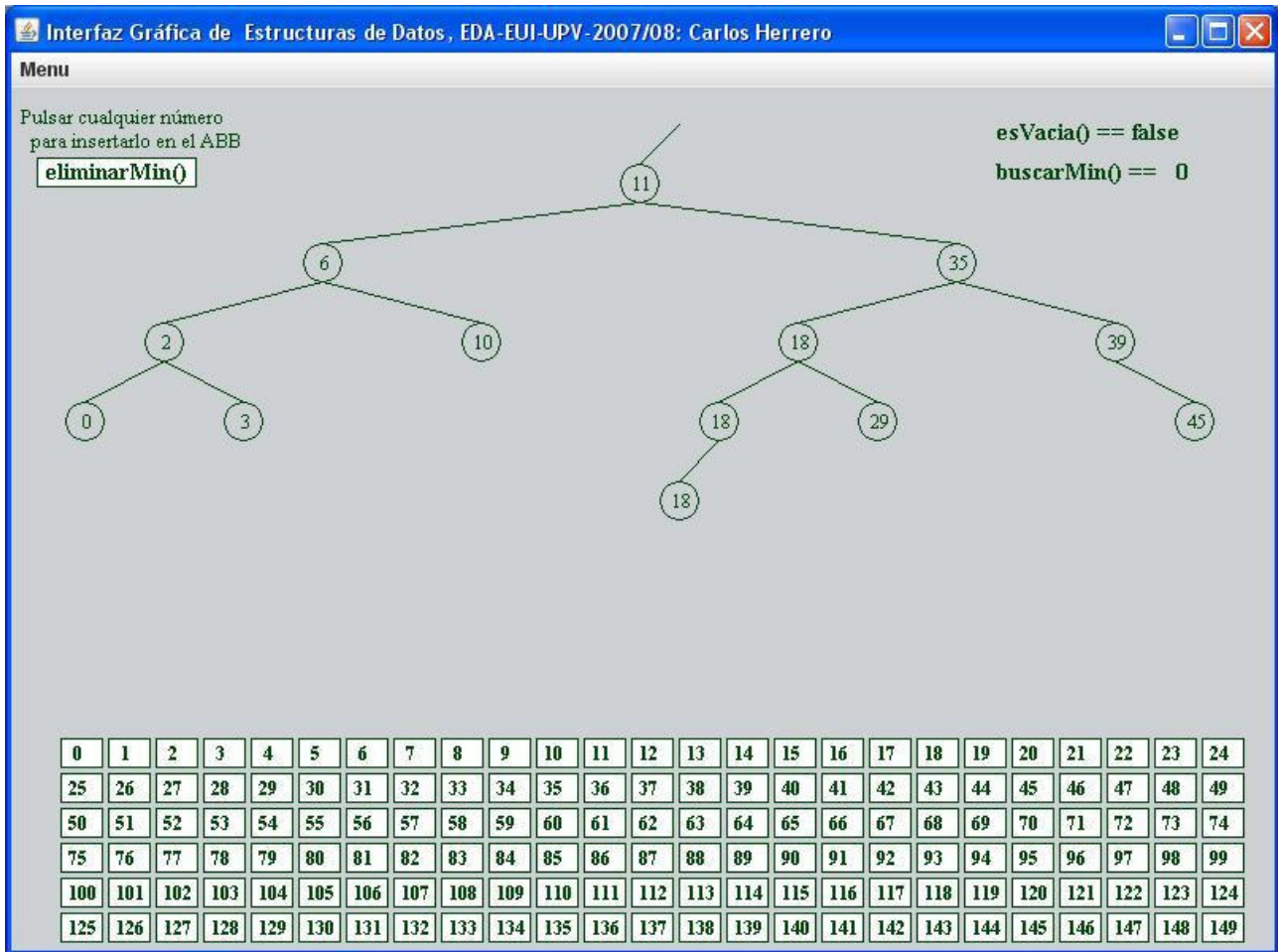


Fig. 8: ABB (Cola de Prioridad)

Si el alumno tratase de eliminar el mínimo en una Cola de Prioridad Vacía, la herramienta abriría una ventana emergente indicando el error como puede verse en (Fig. 9).

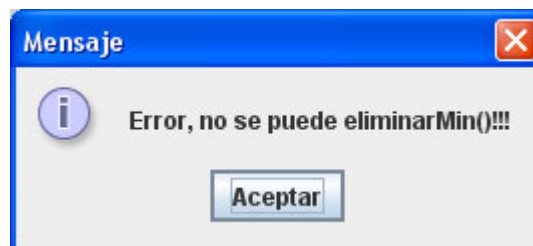


Fig. 9: Error al tratar de “eliminarMin()” en un ABBColaPrioridad vacío.

2.7 La Opción ABB (Diccionario)

Una Diccionario puede representarse como un Árbol Binario de Búsqueda (ABB) sin elementos repetidos y las operaciones de insertar y eliminar. Al igual que en las EDAs anteriores los datos son objetos de la clase Integer y se facilitan botones para introducirlos . También se facilita a un alumno un botón para conmutar de modo insertar/eliminar y consultores para “esVacio()”

En (Fig 10) puede verse el resultado de haber introducido en un Diccionario con un ABB secuencialmente los valores 9, 2, 36, 0, 4, 59, 1, 3, 7, 37 y 88. El consultor “esVacio()” muestran el valor falso

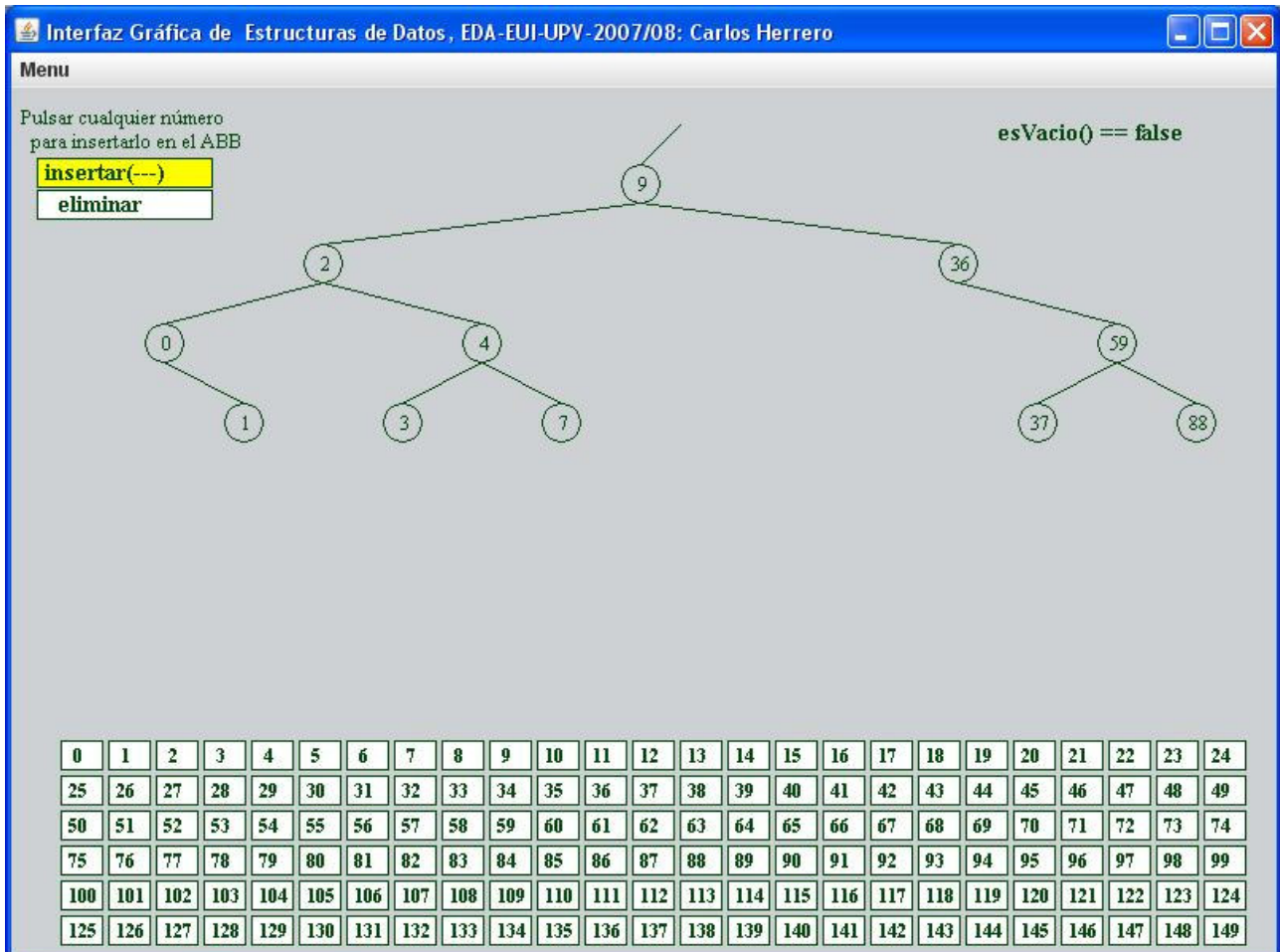


Fig. 10: ABB (Diccionario)

Si el alumno tratase de eliminar un valor inexistente (por ejemplo el 10) o añadir un elemento repetido (por ejemplo el 36) , la herramienta abriría sendas ventanas emergentes con los correspondientes mensajes:de error como puede verse en (Fig. 11).

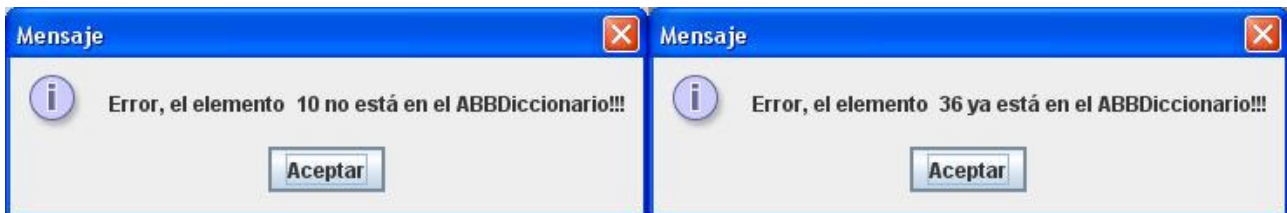


Fig. 11: Errores al tratar de eliminar o insertar en un ABB Diccionario.

2.8 La Opción Montículo Binario

Una cola de prioridad puede representarse más eficientemente como un Montículo Binario, que puede verse como un AB completo en su representación en un array. La herramienta presenta ambas opciones. Al igual que en las EDAs anteriores los datos son objetos de la clase Integer y se facilitan botones para introducirlos. También se facilita a un alumno un botón para eliminar el mínimo y consultores para “esVacía()” y para “buscarMin()”

En (Fig 12) puede verse el resultado de haber introducido en una Cola de Prioridad con un Montículo Binario secuencialmente los valores 11, 6, 35, 2, 10, 18, 39, 0, 3, 18, 29, 45 y 18. Los consultores “esVacía()” Y “buscarMin()” muestran los valores falso y 0 respectivamente.

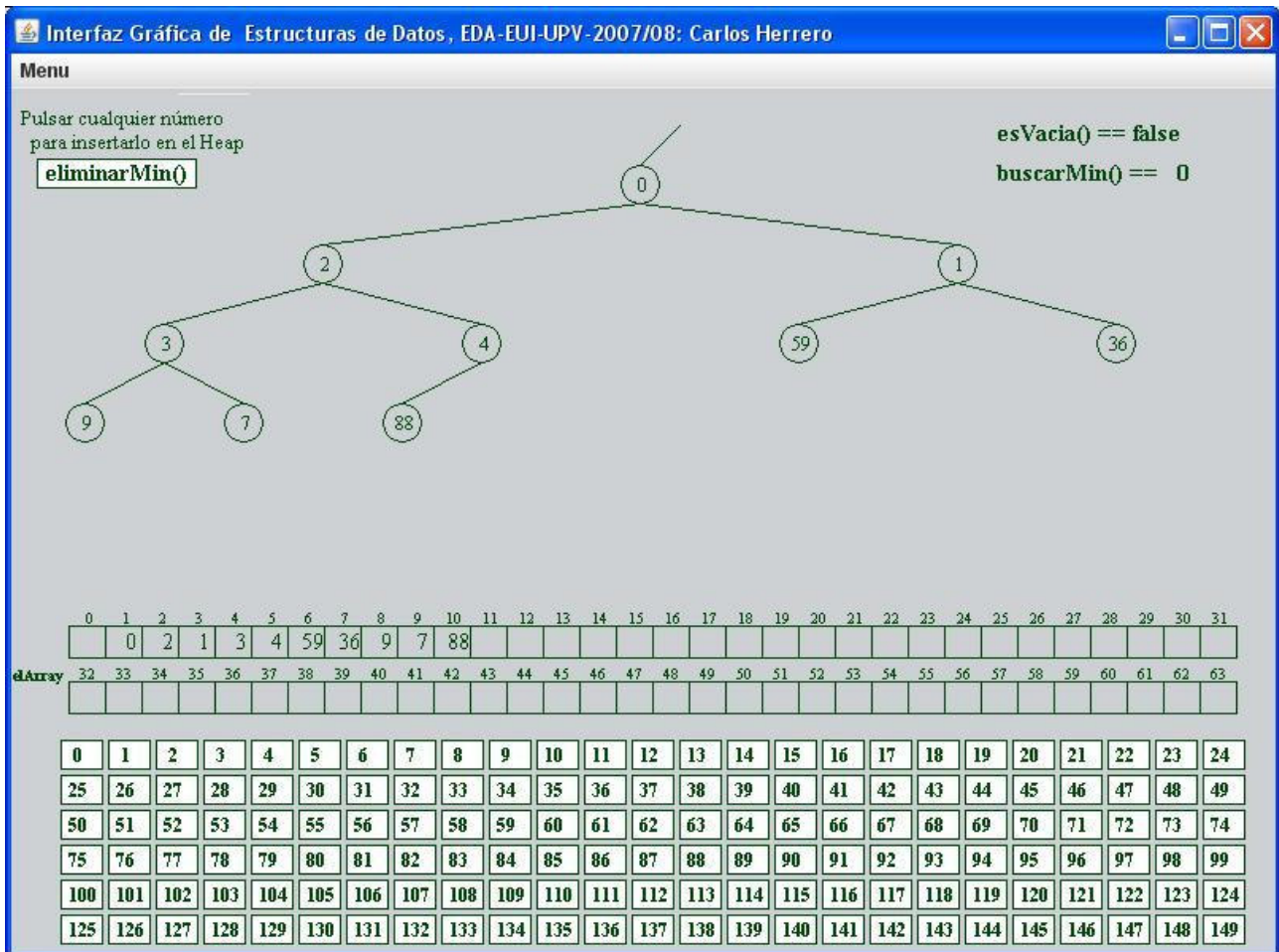


Fig. 12: Montículo Binario

Si el alumno tratase de eliminar el mínimo en una Cola de Prioridad Vacía, la herramienta abriría una ventana emergente indicando el error como puede verse en (Fig. 9). vista anteriormente.

Cabe destacar, que el alumno aprenderá que por criterios de eficiencia es preferible el uso de esta aproximación para definir una cola de prioridad antes que usar la estructura vista en con el ABB. La aplicación, es este caso, resulta muy útil al comprobar que los pasos necesarios para mantener la estructura eficiente en ambas estructuras son muy diferentes y se decantan claramente por el montículo binario.

Además, este último al presentar una estructura de AB completo, no tiene los problemas que la inserción ordenada puede provocar en un ABB como son la degeneración del mismo y que deben ser resueltos adicionalmente con diversas técnicas como los AVL o los árboles rojos y negros.

2.9 La Opción Tabla Hash Exploración Lineal

Una Diccionario puede representarse como una tabla Hash con exploración lineal, que provoca encadenamiento primario. Al igual que en las EDAs anteriores los datos son objetos de la clase Integer y se facilitan botones para introducirlos . También se facilita a un alumno un botón para conmutar de modo insertar/eliminar y consultores para “esVacio()”

En (Fig 13) puede verse el resultado de haber introducido en Diccionario representado con una tabla Hash con exploración lineal secuencialmente los valores 27, 4, 54, 73, 96 y 119. El consultor “esVacio()” muestran el valor falso. Además se muestran valores de la lógica interna de la tabla hash, como la longitud de la tabla tras su duplicación, el número de activas (la eliminación es perezosa) y el número de ocupadas.

Menu

Pulsar cualquier número para insertarlo en la Tabla

insertar(---)

eliminar

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	
				4	27	50	73	96	119														

esVacio() = false longitud de la tabla=23 Activas =6 Ocupadas =6

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74
75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124
125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149

Fig. 13: Tabla Hash con exploración Lineal

Si el alumno tratase de eliminar un valor inexistente o añadir un elemento repetido , la herramienta abriría sendas ventanas emergentes con los correspondientes mensajes:de error como ya pudo verse en (Fig. 11).

Nótese que el encadenamiento primario implica que las colisiones tienden a agruparse en zonas mucho más probables para presentar conflictos. Esto se soluciona parcialmente con las tablas hash con exploración cuadrática y no tienen incidencia en las tablas hash enlazadas.

La misma secuencia presenta un comportamiento distinto en la siguiente estructura.

2.10 La Opción Tabla Hash Exploración Lineal

Una Diccionario puede representarse como una tabla Hash con exploración lineal, que provoca encadenamiento secundario. Al igual que en las EDAs anteriores los datos son objetos de la clase Integer y se facilitan botones para introducirlos . También se facilita a un alumno un botón para conmutar de modo insertar/eliminar y consultores para “esVacio()”

En (Fig 14) puede verse el resultado de haber introducido en Diccionario representado con una tabla Hash con exploración cuadrática secuencialmente los valores 27, 4, 54, 73, 96 y 119. El consultor “esVacio()” muestran el valor falso. Además se muestran valores de la lógica interna de la tabla hash, como la longitud de la tabla tras su duplicación, el número de activas (la eliminación es perezosa) y el número de ocupadas.

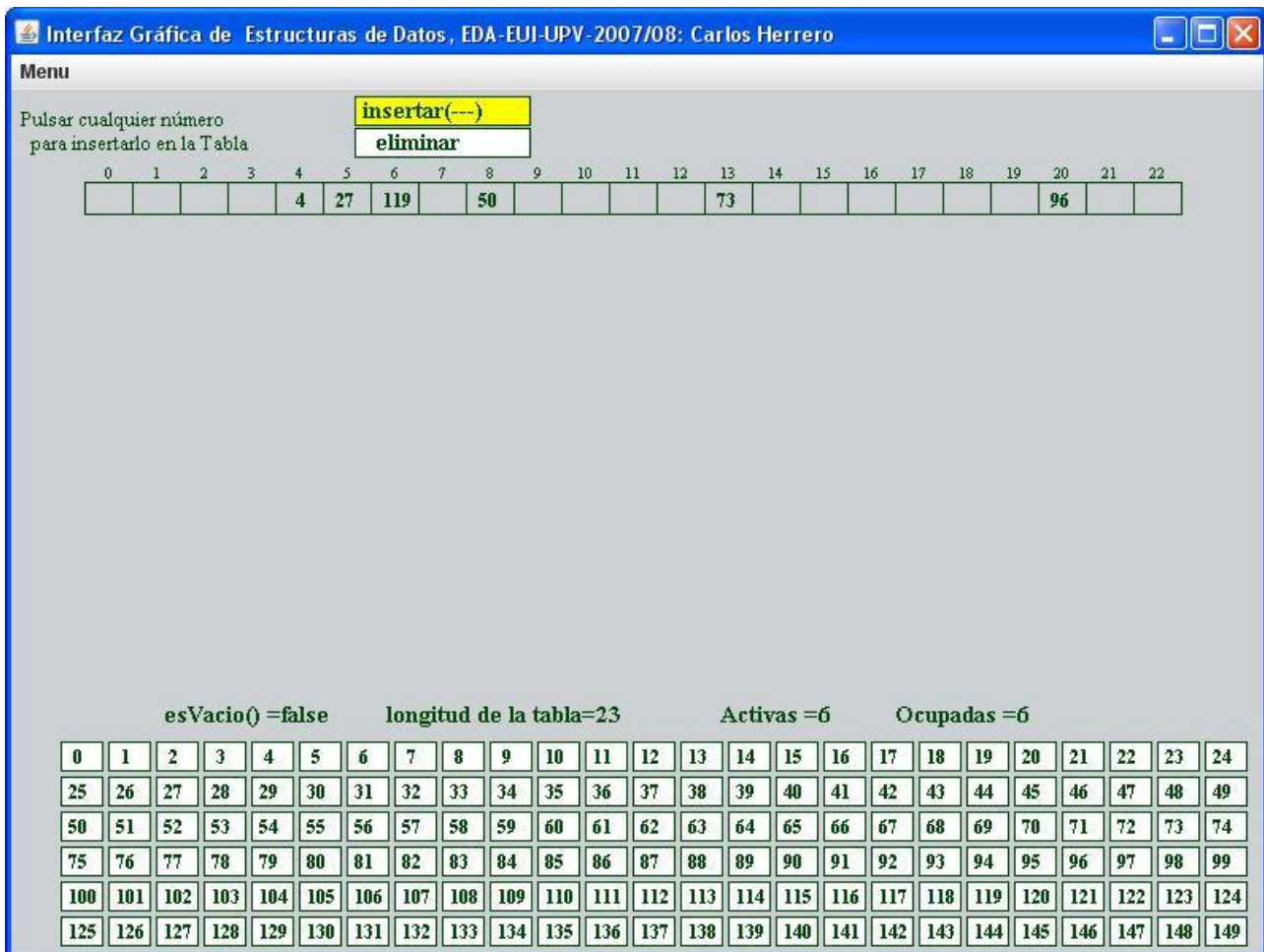


Fig. 14: Tabla Hash con exploración cuadrática

Si el alumno tratase de eliminar un valor inexistente o añadir un elemento repetido , la herramienta abriría sendas ventanas emergentes con los correspondientes mensajes:de error como ya pudo verse en (Fig. 11).

Nótese que el encadenamiento secundario provoca problemas mucho más leves que el encadenamiento primario, y es que las colisiones tienden a agruparse en zonas mucho más probables para presentar conflictos, pero estas zonas están más dispersas.

2.11 La Opción Grafo

Una grafo, puede representarse con una lista de adyacencia o una matriz de adyacencia. Adicionalmente se utiliza un Diccionario para almacenar los nombres de los vértices que son codificados con un entero. En la herramienta, dado que hay mucha información que introducir para

cada grafo, se ha optado por leer los datos directamente de un fichero En (Fig.15) puede verse el fichero "c:/e0.txt".que contiene los datos de un grafo.

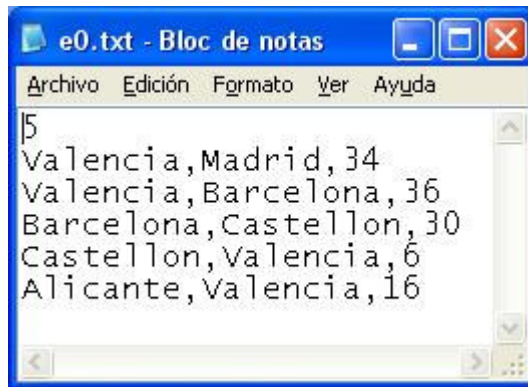


Fig. 15: fichero "e0.txt" que contiene un grafo

El fichero contiene. El número de vértices del grafo y las aristas del grafo con el nombre de vértice origen, nombre de vértice destino y coste.

En (Fig. 16) puede verse el resultado de leer ese fichero desde la aplicaciónl grafo representado que muestra la representación gráfica, la representación por lista de adyacencia, la representación por Matriz de adyacencia así como diversos caminos mínimos sin y con pesos y el orden topológico del grafo.

Menu c:/e0.txt Cargar Fichero

Graph representation (Nodes: 0, 1, 2, 3, 4):

- 0 (Vale) connects to 1 (Madr) with weight 34, 2 (Barc) with weight 36, and 3 (Cast) with weight 6.
- 1 (Madr) connects to 0 (Vale) with weight 34.
- 2 (Barc) connects to 0 (Vale) with weight 36 and 3 (Cast) with weight 30.
- 3 (Cast) connects to 0 (Vale) with weight 6 and 4 (Alic) with weight 20.
- 4 (Alic) connects to 0 (Vale) with weight 16.

Adjacency List:

0 Vale	0	→ 2 36	→ 1 34
1 Madr	1	→ Vacia	
2 Barc	2	→ 3 30	
3 Cast	3	→ 0 6	
4 Alic	4	→ 0 16	

Adjacency Matrix (M):

M	0	1	2	3	4
0		34	36		
1					
2				30	
3	6				
4	16				

Search Results:

- C.M.(sp) desde 0: <0, 1> c:1; <0, 2> c:1; <0, 2, 3> c:2;
- C.M.(cp) desde 0: <0, 1> c:34; <0, 2> c:36; <0, 2, 3> c:66;
- O.T.: Alicante Valencia Madrid Barcelona Castellon

Fig. 16: Tabla Hash con exploración cuadrática

2.12 Conclusiones y Trabajo Futuro.

Se ha presentado una herramienta destinada a la demostración de las Estructuras de Datos básicas y al autoaprendizaje del alumno vía la prueba y la visión gráfica de las mismas.

La herramienta está desarrollada íntegramente en Java, aprovechando las clases que los alumnos desarrollan y utilizan en el aula y laboratorios, de manera que hay una correspondencia directa entre las implementaciones que los alumnos aprenden y la herramienta que muestra gráficamente dichas implementaciones.

Los alumnos acceden a la herramienta a través de un archivo jnlp, que gracias a la aplicación Java Web Start permite la actualización automática de la herramienta y, por tanto, ofrecer a los alumnos de manera incremental las distintas “demos” de las que consta, con arreglo al desarrollo del temario en el aula.

En la actualidad, dos alumnos están desarrollando un proyecto final de carrera basado en la herramienta presentada, con la intención de mejorar su aspecto gráfico, incluir animaciones en las trazas, y añadir nuevas estructuras de datos como las Tablas Hash enlazadas.

Referencias

- [1] AHO, A.V., HOPCROFT, J.E., ULLMAN, J.E.. “Data structures and algorithms” *Addison-Wesley*, 1983
- [2] BRASSARD, G., BRATLEY, T. ”Fundamentos de algoritmia *Prentice Hall*, 1997
- [3] COHOON, J. P., DAVIDSON, J.W. “Programación en Java 5.0” *McGraw-Hill*, 2006
- [4] CORMEN, T. H. “Introduction to algorithms 2nd. ed”.*MIT Press : McGraw-Hill*, 2001
- [5] <http://java.sun.com>
- [6] WEISS, MARK ALLEN “Estructuras de datos en JAVA : compatible con JAVA 2” *Addison-Wesley*, 2006