

Práctica 3

PacVac: Gestión de Pacientes y Vacunación

Algoritmos y Estructuras de Datos - Grado en Ingeniería Informática (USC)

25 de septiembre de 2022

Objetivos

En esta práctica trataremos de:

- Reutilizar los TADs conocidos (árbol binario de búsqueda, lista, cola).
- Valorar qué TAD es el más adecuado para el diseño de las diferentes tareas de un programa.

1. Versión 1 de la práctica

En esta primera versión de la práctica utilizaremos el TAD árbol binario de búsqueda para guardar los datos de un conjunto de pacientes. Los datos que tendremos para cada paciente son: Apellidos_Nombre y lista de vacunas que tienen pendiente de administrar.

El programa se iniciará creando la base de datos de pacientes vacía. Para la base de datos se utilizará un TAD árbol binario de búsqueda que pueda almacenar tantos pacientes como sea necesario. A continuación se ofrecerá el menú de selección de opciones que se repetirá continuamente hasta que el usuario elija la opción de salir. Por último, se destruirá la base de datos. En el menú se mostrarán las siguientes opciones:

- **A. Añadir paciente:** esta opción permitirá añadir un nuevo paciente a la base de datos. Al añadir un nuevo paciente se preguntará su nombre (con formato Apellidos, Nombre) y las vacunas que va a recibir, que se almacenarán en un TAD lista. Para cada vacuna se preguntará su nombre y el número de dosis totales de la pauta de vacunación (el número de dosis administradas se inicializa a 0 por defecto). El proceso de pregunta sobre las vacunas finaliza cuando el nombre de la vacuna sea "fin".
- **L. Listado alfabético de pacientes:** Se imprimirán los datos de los pacientes de la base de datos en orden alfabético. También se imprimirá de forma detallada la información de **todas** sus vacunas. Para cada vacuna se imprimirá el nombre de la misma, junto con el máximo de dosis de la pauta de vacunación y el número de dosis administradas.
- **E. Eliminar paciente:** esta opción permitirá eliminar un paciente de la base de datos. Se preguntará su nombre y, si está en la base de datos, se informará de su baja. Si no está en la base de datos, también se informará por pantalla.
- **S. Salir:** esta opción sale del menú, finalizando el programa.

Temporización recomendada

A continuación se presenta una temporización recomendada para realizar cada una de las fases del proyecto. Tened en cuenta que esto es sólo una estrategia de implementación: tenéis completa libertad para implementar las funcionalidades que se piden en el enunciado, siempre y cuando utilicéis el TAD `abb` y el TAD `lista`.

Fase 1: Preparación del proyecto.

En primer lugar, debéis crear un proyecto en Netbeans (o en otro IDE) con el nombre `Vacunodromo_1` y decirle que sí queréis crear `main` (fijaos que a la derecha esté seleccionado C, no C++). Netbeans crea su propio `makefile`, os recomendamos que trabajéis con él para poder utilizar las herramientas de depuración. Para poder trabajar con el terminal, tened en cuenta que Netbeans creará el ejecutable en la carpeta `./dist/Debug/Cygwin-Windows`, por lo que podéis abrir el terminal en Netbeans y cambiarnos a esa carpeta para probar las entradas por línea de comandos que se proponen en la versión 2.

Todas las funciones que se llaman desde el menú principal deben estar implementadas en un archivo `vacunodromo.c` con su correspondiente interfaz `vacunodromo.h`, que iremos ampliando en las distintas versiones del programa.

Fase 2: Preparación de los TAD

El programa deberá implementar obligatoriamente el árbol y la lista indicadas con los TAD correspondientes, cuya descripción y funcionalidades se han explicado en clase de teoría. Antes de iniciar el desarrollo de la práctica debéis repasar detalladamente las diapositivas y notas de teoría de los temas correspondientes y, de forma especial, los ejemplos, para recordar el funcionamiento de los TAD.

NO PODRÁ REALIZARSE NINGUNA MODIFICACIÓN SOBRE LOS FICHEROS DE LOS TAD PROPORCIONADOS excepto las indicadas a continuación, además de las que necesitéis para añadir los `#include` necesarios para modificar los tipos de elemento de los TAD, así como definir constantes para las longitudes de las cadenas de caracteres.

Preparación del TAD `abb` (de pacientes)

Para la base de datos de pacientes se utilizará el TAD `abb` implementado en los ficheros adjuntos (`abb.c` y `abb.h`) en el que `TIPOELEMENTOABB` debe ser una estructura con la información del paciente, campos `ApellidosNombre` (podéis usar un array estático definiendo su longitud como una constante) y una `TLISTA` de vacunas. También debéis modificar `TIPOCLAVE` para indicar cuál va a ser vuestro campo de ordenación en el `abb`, si ordenamos por orden alfabético será un `char *`.

Además, debéis modificar las funciones `_clave_elem()`, `_comparar_claves()` y `_destruir_elem()` en `abb.c` para adaptarlas a vuestro `TIPOELEMENTOABB`. Recordad que para usar `strcmp()` debéis incluir la librería `string.h`.

Preparación del TAD `lista` (de vacunas)

Para la lista de vacunas se utilizará el TAD `Lista` implementado en los ficheros adjuntos (`lista.c` y `lista.h`) en el que el `TIPOELEMENTOLISTA` debe definirse como una estructura con los campos `nombreVacuna` (podéis usar un array estático, pero entonces debéis definir la longitud como una constante), y dos enteros que almacenen el número máximo de dosis y las dosis administradas.

Fase 3: Implementación del menú

Por último, deberéis implementar el menú del programa. Las opciones del menú que debería haber implementadas son las siguientes:

- **Añadir paciente:** tendréis que llamar a la función `añadirPaciente()` que realiza la toma de datos por teclado de la información del paciente a añadir a la base de datos. Tenéis que tener en cuenta que debéis implementar un bucle para poder tomar información sobre las vacunas mientras el nombre de la misma sea distinta de “fin” (puede resultar útil aquí la función `strcmp()` incluida en la librería `string.h`).
- **Listado alfabético de pacientes:** Tenéis que tener en cuenta que las funciones para imprimir podéis implementarlas en cadena, es decir, `listadoPacientes()` llamaría a `_imprimirPaciente()` e `_imprimirPaciente()` llamaría a `_imprimirVacunas()`.
- **Eliminar paciente:** tendréis que llamar a la función `eliminarPaciente()` que solicitará el nombre del paciente a eliminar y, si está en la base de datos, procederá a eliminarlo, informando por pantalla del resultado de la acción.
- **Salir:** tendrá como efecto salir del bucle del menú.

Una vez salimos del menú, habrá que proceder con la destrucción de la base de datos de memoria.

2. Versión 2: Lectura/Grabación de la base de datos de pacientes

En esta versión leeremos los datos de los pacientes de un archivo que se introducirá por línea de comandos y también actualizaremos la base de datos guardando los datos actualizados en el mismo archivo. La lectura del archivo se realizará antes del menú y la grabación se realizará al finalizar el menú, antes de destruir el árbol.

Lectura de datos desde un fichero

Se deberá permitir la lectura de un fichero de pacientes y sus vacunas y la inicialización de la base de datos a partir de este fichero. El fichero deberá tener el siguiente formato: *PACIENTE/VACUNA_1;DOSIS_VACUNA_1;MAXDOSIS_VACUNA_1;VACUNA_2;DOSIS_VACUNA_2;MAXDOSIS_VACUNA_2;*. Un ejemplo de fichero se muestra en el siguiente fragmento de fichero:

```
1 Vazquez Perez , Manuel|Comirnaty;1;2;Chiroflu;2;3;Pneumovax;0;1;
2 Diaz Couselo , Maria|Comirnaty;0;1;
3 Cea Bermudez , Juan|Chiroflu;0;2;Pneumovax;0;1;
```

Se asume que el fichero es siempre correcto, para no sobrecargar vuestro proyecto con comprobaciones defensivas. El programa deberá llamarse desde línea de comandos con el siguiente formato: `./ejecutable -f NOMBRE_DEL_FICHERO`.

Para poder procesar este fichero debéis tener en cuenta las siguientes consideraciones:

- En primer lugar, debéis procesar los argumentos por línea de comandos.
- En segundo lugar, debéis leer el fichero línea a línea. Para ello podéis utilizar la función `fgets()`, que tiene el problema de que añade el retorno de carro a la cadena. Para eliminarlo, podéis utilizar la siguiente función (necesitaréis la librería `string.h`).

```

1 void _strip_line(char * linea){
2     linea[strcspn(linea, "\r\n")] = 0;
3 }

```

- En tercer lugar, debéis procesar los campos de la línea leída. Para eso, necesitaréis dos tipos de llamadas a la función `strtok()`. En el primer tipo de llamada, partiréis la línea a partir del carácter “|”. Para eso, es suficiente con hacer dos llamadas a `strtok()` tal y como sigue:

```

1 char * nombre_paciente = strtok(linea, "|");
2 char * lista_vacunas = strtok(NULL, "|");

```

Después, tendréis que llamar a `strtok()` sobre el fragmento que contiene las vacunas (`listaVacunas`) usando el carácter “;” como delimitador. Debéis almacenar una vacuna nueva por cada par de palabras procesadas por la función. También debéis convertir la cadena de caracteres correspondiente al número de dosis administradas y al número máximo de dosis a un entero (utilizad la función `atoi()`). A continuación tenéis un ejemplo de uso de `strtok()`:

```

1 char *cadena="Hola,mundo,soy,una,cadena,separada,por,comas";
2 char *token=strtok(cadena, ",");
3 while(token!=NULL){
4     printf("Token: %s\n", token);
5     token=strtok(NULL, ",");
6 }

```

Grabación de datos en un fichero

El fichero debe actualizarse con los últimos datos disponibles por lo que, si tenéis un fichero precargado que habéis introducido por línea de comandos, debéis actualizarlo abriéndolo para escritura y escribiendo los datos del árbol de pacientes. Tened cuidado con el recorrido que hacéis para escribir, pues si elegís un recorrido inorden los datos se guardarán en el archivo por orden alfabético y por tanto la siguiente vez que los leamos el `abb` estará totalmente desequilibrado. En caso de que no hayáis precargado un archivo por línea de comandos, debéis preguntar al usuario el nombre del fichero de salida con el que quiere guardar la base de datos, y así servirá para siguientes ejecuciones.

3. Versión 3: Generación automática de la cola de pacientes del vacunódromo

Esta versión la subiré más adelante, pues será una nueva entrega. Generaremos el conjunto de pacientes que se vacunarán cada día y los insertaremos en una cola. Una vez que se vacunen, deberemos actualizar sus datos en la base de datos de pacientes.

Entregables y calificaciones

Deberá realizarse la entrega por el Campus Virtual, donde están especificadas las fechas de entrega particulares para cada grupo y los criterios de evaluación. Las instrucciones para generar el fichero para su entrega son las siguientes:

- Debéis subir un único fichero comprimido con el nombre `Apellido1Apellido2Nombre_3.zip`.

- Se incluirán todos los ficheros .c y .h necesarios para compilar el proyecto junto con los archivos de los TADs utilizados. Se asumirá que todos los archivos necesarios para compilar están en la misma carpeta.

CUALQUIER EJERCICIO QUE NO COMPILE SERÁ EVALUADO CON LA CALIFICACIÓN DE 0. ESTE CRITERIO SE MANTENDRÁ EN EL RESTO DE PRÁCTICAS DE LA ASIGNATURA.

Ejemplo de ejecución

A continuación se muestran ejemplos de ejecución del programa. Tened en cuenta que el formato de vuestro menú puede diferir de lo aquí expuesto.

Versión 1

```
1 > ./vacunodromo_1
2
3 Bienvenid@ al programa de gestion PacVac
4 A. Anhadir paciente
5 L. Listar pacientes
6 E. Eliminar paciente
7 S. Salir
8 Seleccione una opcion: A
9
10 Nombre del paciente: Martinez Couselo, Maria
11     Nombre vacuna (fin para finalizar): Comirnaty
12     Numero de dosis maximas: 2
13     Nombre vacuna (fin para finalizar): Chiroflu
14     Numero de dosis maximas: 3
15     Nombre vacuna (fin para finalizar): Pneumovax
16     Numero de dosis maximas: 1
17     Nombre vacuna (fin para finalizar): fin
18
19 Bienvenid@ al programa de gestion PacVac
20 A. Anhadir paciente
21 L. Listar pacientes
22 E. Eliminar paciente
23 S. Salir
24 Seleccione una opcion: a
25
26 Nombre del paciente: Paz Marin, Pedro
27     Nombre vacuna (fin para finalizar): Comirnaty
28     Numero de dosis maximas: 1
29     Nombre vacuna (fin para finalizar): fin
30
31 Bienvenid@ al programa de gestion PacVac
32 A. Anhadir paciente
33 L. Listar pacientes
34 E. Eliminar paciente
35 S. Salir
36 Seleccione una opcion: A
37
38 Nombre del paciente: Alvarez Castro, Noa
39     Nombre vacuna (fin para finalizar): Chiroflu
40     Numero de dosis maximas: 2
41     Nombre vacuna (fin para finalizar): Pneumovax
42     Numero de dosis maximas: 1
43     Nombre vacuna (fin para finalizar): fin
44
```

```

45 Bienvenid@ al programa de gestion PacVac
46 A. Anhadir paciente
47 L. Listar pacientes
48 E. Eliminar paciente
49 S. Salir
50 Seleccione una opcion: L
51
52 Alvarez Castro, Noa
53     Vacuna:   Chiroflu      Dosis 0 de 2
54     Vacuna:   Pneumovax     Dosis 0 de 1
55
56 Martinez Couselo, Maria
57     Vacuna:   Comirnaty     Dosis 0 de 2
58     Vacuna:   Chiroflu      Dosis 0 de 3
59     Vacuna:   Pneumovax     Dosis 0 de 1
60
61 Paz Marin, Pedro
62     Vacuna:   Comirnaty     Dosis 0 de 1
63
64 Bienvenid@ al programa de gestion PacVac
65 A. Anhadir paciente
66 L. Listar pacientes
67 E. Eliminar paciente
68 S. Salir
69 Seleccione una opcion: E
70
71 Nombre del paciente a eliminar: Paz Marin, Pedro
72 El paciente ha sido eliminado de la base de datos
73
74 Bienvenid@ al programa de gestion PacVac
75 A. Anhadir paciente
76 L. Listar pacientes
77 E. Eliminar paciente
78 S. Salir
79 Seleccione una opcion: E
80
81 Nombre del paciente a eliminar: Perez Vazquez, Manuel
82 Este paciente no áest registrado en la base de datos
83
84 Bienvenid@ al programa de gestion PacVac
85 A. Anhadir paciente
86 L. Listar pacientes
87 E. Eliminar paciente
88 S. Salir
89 Seleccione una opcion: l
90
91 Alvarez Castro, Noa
92     Vacuna:   Chiroflu      Dosis 0 de 2
93     Vacuna:   Pneumovax     Dosis 0 de 1
94
95 Martinez Couselo, Maria
96     Vacuna:   Comirnaty     Dosis 0 de 2
97     Vacuna:   Chiroflu      Dosis 0 de 3
98     Vacuna:   Pneumovax     Dosis 0 de 1
99
100 Bienvenid@ al programa de gestion PacVac
101 A. Anhadir paciente
102 L. Listar pacientes
103 E. Eliminar paciente
104 S. Salir
105 Seleccione una opcion: s
106
107 Gracias por usar nuestro sistema PacVac

```

108 |
109 >

Versión 2 (lectura/escritura de archivos)

```
1 > cat archivo_pacientes.txt
2 Martinez Couselo, Maria|Comirnaty;1;2;Chiroflu;2;3;Pneumovax;0;1;
3 Paz Marin, Pedro|Comirnaty;0;1;
4 Alvarez Castro, Noa|Chiroflu;0;2;Pneumovax;0;1;
5
6 > ./vacunodromo_2 -f archivo_pacientes.txt
7
8 Bienvenid@ al programa de gestion PacVac
9 A. Anhadir paciente
10 L. Listar pacientes
11 E. Eliminar paciente
12 S. Salir
13 Seleccione una opcion: L
14
15 Alvarez Castro, Noa
16     Vacuna:   Chiroflu           Dosis 0 de 2
17     Vacuna:   Pneumovax         Dosis 0 de 1
18
19 Martinez Couselo, Maria
20     Vacuna:   Comirnaty          Dosis 1 de 2
21     Vacuna:   Chiroflu           Dosis 2 de 3
22     Vacuna:   Pneumovax         Dosis 0 de 1
23
24 Paz Marin, Pedro
25     Vacuna:   Comirnaty          Dosis 0 de 1
26
27 Bienvenid@ al programa de gestion PacVac
28 A. Anhadir paciente
29 L. Listar pacientes
30 E. Eliminar paciente
31 S. Salir
32 Seleccione una opcion: A
33
34 Nombre del paciente: Bouzas Perez, Pablo
35     Nombre vacuna (fin para finalizar): Comirnaty
36     Numero de dosis maximas: 2
37     Nombre vacuna (fin para finalizar): fin
38
39 Bienvenid@ al programa de gestion PacVac
40 A. Anhadir paciente
41 L. Listar pacientes
42 E. Eliminar paciente
43 S. Salir
44 Seleccione una opcion: S
45
46 Gracias por usar nuestro sistema PacVac
47
48 >
49 > cat archivo_pacientes.txt
50 Martinez Couselo, Maria|Comirnaty;1;2;Chiroflu;2;3;Pneumovax;0;1;
51 Alvarez Castro, Noa|Chiroflu;0;2;Pneumovax;0;1;
52 Bouzas Perez, Pablo|Comirnaty;0;2;
53 Paz Marin, Pedro|Comirnaty;0;1;
54
55 >
```