

Laboratorio 1 – Árboles de decisión

GUIÓN DEL LABORATORIO

1.- Objetivos del laboratorio

- Diseño de clases en C++
- Comprensión y uso del TAD Árbol

2.- Antes de asistir al laboratorio

Antes de asistir al laboratorio debes realizar las siguientes tareas:

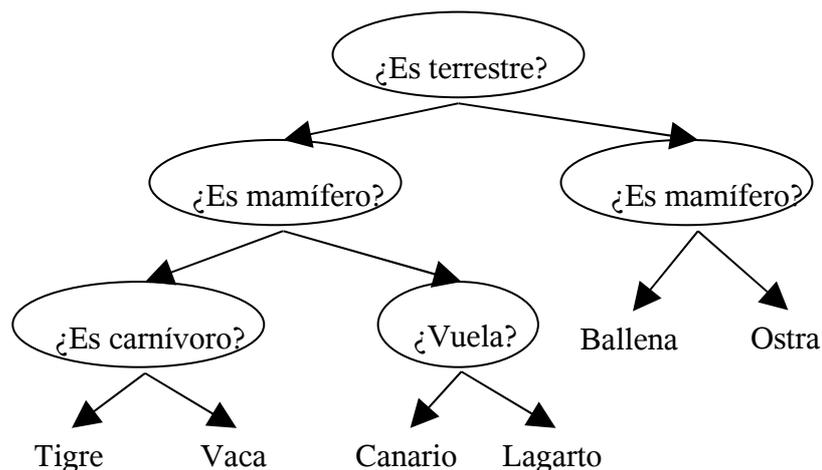
1. Leer los apuntes de la asignatura sobre el TAD Árbol.
2. Revisar el guión del laboratorio (este documento).
3. Realizar las Actividades indicadas en la hoja de Prerrequisitos. Recuerda que es preciso entregar esta actividad antes de la fecha prevista. En caso contrario, no podrás acceder al laboratorio ni entregar de ninguna manera la solución a esta práctica.

3.- Descripción del problema

Los árboles de decisión permiten representar el conjunto de todas las soluciones alternativas a un problema. De alguna manera, representan una clasificación de las soluciones en función de una serie de características.

Por ejemplo, si se plantea el problema de una clasificación animal, se podrían considerar características de los animales tales como: ¿Es terrestre?, ¿Es mamífero?, ¿Vuela?, ¿Es carnívoro?, etc. En ese caso, para realizar la clasificación, es posible utilizar un árbol binario de decisión que esté planteado de forma que se indique si el animal posee o no una determinada característica.

Ejemplo:



Como se puede ver en este ejemplo, los animales ocupan siempre los nodos terminales del árbol, mientras que las características (o preguntas) son nodos interiores del árbol.

4.- Actividades a realizar

En esta práctica se pide manipular un árbol de decisión que sigue el esquema del ejemplo anteriormente expuesto.

Actividad 1: Generar el árbol binario de decisión a partir del archivo “animales.arb”.

Se pide escribir un programa que incluya una función que genere el árbol binario de decisión cuya estructura está almacenada en el archivo “animales.arb”. Se trata de un archivo texto en el que el árbol se ha almacenado según el siguiente formato: cada línea representa la información de un nodo del árbol (la longitud máxima de una línea es 25 caracteres), si la información es '.' significa que el árbol correspondiente debe estar vacío, se ha empleado el criterio Prefijo para el recorrido del árbol.

Se recomienda, una vez escrita la función, incluir en el programa la verificación de la correcta generación del árbol realizando un recorrido prefijo del árbol y comprobando que este recorrido es idéntico al contenido en el archivo “animales.arb”.

Actividad 2: Recorrido del árbol interaccionando con el usuario.

Se trataría de completar el programa de la actividad 1 de forma que fuera posible recorrer el árbol interaccionando con el usuario de la siguiente manera: comenzando por el nodo raíz, si el nodo es no terminal entonces interpretar la información del nodo como una pregunta a realizar al usuario, en función de la respuesta a dicha pregunta avanzar al siguiente nodo (se supone que los subárboles izquierdos corresponden a respuestas afirmativas y los derechos a respuestas negativas). Este proceso se repite hasta alcanzar un nodo terminal (árbol no vacío con dos subárboles vacíos), en cuyo caso se indicará el nombre del animal que cumple todas las características que ha indicado el usuario.

Una vez finalizado un recorrido le preguntaremos al usuario si quiere volver a buscar otro animal (s/n), si responde 's' volveremos a recorrer el árbol, si responde 'n' finalizara el programa.

5.- Después de asistir al laboratorio

Contesta a las cuestiones que aparecen en el Cuestionario Posterior al laboratorio (Lab6_CuestionarioPosterior).

6.- Ficheros a entregar

| | |
|-------------------------|--|
| Antes del laboratorio | Lab6_Prerrequisito claseAB.h, claseAB.cpp, prueba.cpp y prueba.dev |
| En el laboratorio | decision.cpp y decision.dev correspondiente a las Actividades 1 y 2 Lab6_HojaTrabajo |
| Después del laboratorio | Lab6_CuestionarioPosterior |