**ÁRBOL DE DECISIONES:**

**Explicación funcionamiento del programa:**

Este programa es capaz de, a partir de la lectura de un fichero, generar un “árbol de decisiones” que permita en encontrar un elemento. El árbol está compuesto por lo que llamaremos nodos, entre los que podemos distinguir dos tipos. Los primeros son los nodos pregunta, que como su propio nombre indica, cada uno de ellos almacenará una pregunta que el programa hará al usuario. Según lo que responda, el programa avanzará por una rama u otra del árbol y ejecutará el siguiente nodo pregunta. Este proceso se repetirá sucesivamente hasta llegar al final de una de las ramas, donde nos encontramos el otro tipo de nodos, los nodos respuesta. El programa es capaz de detectar estos nodos como el final del árbol, y los ejecutará como respuesta. Sin embargo, te da la opción de cambiar la respuesta que te ha dado si no era la que esperabas. Para ello te pide que diferencies ambas respuestas a través de una nueva pregunta. Luego tenemos un nuevo nodo pregunta y un nuevo nodo respuesta que el programa se encargará de sobrescribir en el fichero de manera adecuada para que cuando se vuelva a crear el árbol, éste haya sido actualizado con los nuevos cambios.

En conclusión, este programa permite realizar la búsqueda de un elemento que ha sido clasificado en un árbol a partir de diferentes preguntas. Pero también permite clasificar nuevos elementos dentro del árbol mediante la inclusión también de una nueva pregunta.

Como implementaciones opcionales se han añadido además:

* La posibilidad de elegir previamente entre varios ficheros disponibles
* La opción de multirrespuesta generando así un árbol multicamino en vez de un árbol binario

**Descripción general de funciones:**

* *ready* no son más que explicaciones previas muy resumidas del funcionamiento del programa, por ello se ejecuta una sola vez al principio de todo.
* *choose\_file* es la que pide al usuario la elección del fichero a utilizar. Para añadir o cambiar los ficheros propuestos habría que modificar la función.
* *Deserialize* se encarga de construir el árbol a partir de la lectura del fichero, mientras que la de
* *Serialize* escribe nuevamente el fichero a partir del árbol, y es útil por tanto para actualizar el fichero cuando se ha modificado el árbol.
* *question* ejecuta las preguntas una a una avanzando por el árbol según lo que responda el usuario. Cuando llega al final del árbol llama a *finalquestion*, que pregunta si el elemento al que se ha llegado el lo que se estaba buscando o no. En caso de que no lo fuese, se accede a *updatetree* para modificar el árbol con el nuevo nodo pregunta y el nuevo nodo respuesta.
* El *main* simplemente se encarga de llamar a estas funciones y de comprobar la apertura de ficheros.

**Guía de uso del programa ejecutable:**

Para utilizar el programa simplemente deberá contestar a las preguntas que se vayan haciendo con una respuesta entre las que se propongan. Para responder utilice los números identificativos que aparecen delante de cada respuesta. Si se equivoca utilizando un número que no se encuentra entre las opciones disponibles, el programa volverá a ejecutar la misma pregunta esperando una respuesta válida.

La inclusión de los nuevos nodos es la parte mas compleja. Primero se le pedirá el nombre del elemento que estaba buscando y que el programa ha sido incapaz de encontrar. Seguidamente deberá escribir una nueva pregunta que diferencie el elemento que usted buscaba del que le propuso el programa. Por último, se le pide que diga el numero de respuestas que puede tener esa pregunta, asociando a cada respuesta un elemento. Obviamente, si selecciona un numero de respuestas k tal que k>2 (multirespuesta) se le solicitará que incluya (k-2) nuevos elementos para que todas las respuestas tengas un elemento asociado.

**Descripción del reparto de roles del equipo:**

* **Aléx Pérez Vicente:** búsqueda de información para resolver el problema de como debía escribirse el fichero para que el programa construyese el árbol correctamente.
* **Alex Pérez Vicente y Guillermo Moreno Vázquez:** escritura del programa completo como árbol binario con el respectivo diseño de algoritmos y funciones necesarias así como la escritura de algún fichero de prueba.
* **Guillermo Moreno Vázquez:** implementación opcional de una nueva función que permita la elección de un fichero entre varios disponibles en vez de utilizar un fichero único.
* **Jaime de Castro de Peñaranda:** optimización el programa e implementación opcional de multirespuesta convirtiendo el árbol en uno multicamino.
* **Todos:** Documentación del trabajo

Propuestas de mejora y valoración personal.