**Procedimiento seguido para la implementación** (esta un poco caos porque es un copia pega de latex pero luego queda bonico)

En un primer lugar, resolvemos el problema que se da en el enunciado dela pr ́actica para poder utilizarlo de base.Para la lectura del archivo, creamos una claseReader, en la cual implemen-tamos la lectura de los archivos de entrada.Para la construcci ́on del ́arbol final, implementamos una claseNode, cuyoconstructor recibe unString, que ser ́a el valor seleccionado tras aplicar elID3, y otro nodo para seguir en una futura recursi ́on.En la claseID3es donde implentamos el algoritmo como tal. Tenemos doslistas, una deString attributes, donde almacenamos los atributos le ́ıdos, ydata, que es una de listas deStringque almacena los valores de los atributos.Tenemos un atributostring targetAttribute, el cual va a ser el valor del primernodo. Por defecto, elegimos el primer atributo, pero este puede cambiar en laprimera iteracci ́on de la recursi ́on. A partir de la primera recursi ́on, el valorde este atributo no cambia, ya que va a ser el nodo del cual cuelgue el restodel ́arbol. Guardamos el resto de atributos no elegidos en una lista de stringllamada noTargetAttributes, y a partir de aqu ́ı aplicamos elID3.Para empezar guardamos en un map llamadotargetValues, el n ́umero de ve-ces que aparece cada valor del atributo targetAttribute. Si el tama ̃no del mapes de1, significa que hemos llegado al final de ese sub ́arbol. Si por otro lado,est ́avac ́ıa, no tenemos informaci ́on por lo que no seguimos por esa rama, yvolvemos hacia atr ́as.Si todo va bien, a continuaci ́on, calculamos el m ́erito de cada atributo y nosquedamos con el mejor, es decir, nos quedamos con el de menor m ́erito.Paraello, iteramos sobre cada uno de los atributos y obtenemos el m ́erito a partirde los valores. Iteramos sobre cada atributo de la listarestoAtributos. Decada atributo iteramos en un bucle interior sobre los valores de ese atributo.Para el c ́alculo del m ́erito, usamos varias funciones:getPartitionFromData: Calcula el n ́umero de repeticiones que tiene ca-da uno de los posibles atributos del atributo dado. Por ejemplo: targe-3

tAttribute = Temperatura –¿caluroso(4), templado(6), fr ́ıo(4).getTargetValuesFromData: Obtiene los ejemplos a partir del atributodado. Ejemplo: Temperatura -¿Caluroso(4).getEntropy: Calculamos la entrop ́ıa siguiendo la f ́ormulainf or(p, n) =−p∗log2p−n∗log2nA continuaci ́on, calculamos elinforque es la suma de los m ́eritos de losvalores. Finalmente comparamos y nos quedamos con el menor.El siguiente paso es un punto de parada. Si no hemos obtenido m ́erito es-tamos al final de un sub ́arbol, por lo que obtenemos el valor del nodo y lodevolvemos.El siguiente paso es preparar el siguiente sub ́arbol, esta llamada recursiva setrata de la ampliaci ́on de la pr ́actica, por lo que la explicamos en el siguienteapartado.Al final de esta funci ́on, hemos obtenido el atributo elegido y sus atributos,que es el valor que se retorna en un objetoNode. Para construir este objeto,usamos la funci ́on auxiliargetNoRepeatedValuespara obtener los atributosdel mejor atributos

**Ampliaciones realizadas**

AQUÍ HAY QUE EXPLICAR LA RECURSIÓN.