

# Árboles de Decisión

## 1. Introducción

Un Árbol de Decisión es una herramienta que modela una secuencia de decisiones como un árbol. Dado un estado determinado, es posible tomar una decisión entre varias alternativas. Los estados o consecuencias de las decisiones son representadas como nodos, mientras que las ramas representan las decisiones que se pueden tomar en cada estado. La figura 1 muestra un ejemplo de un Árbol de Decisión para la selección de un empleado.

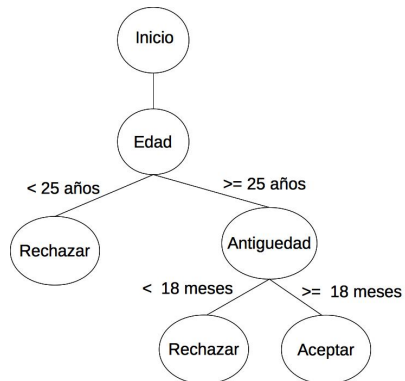


Figura 1: Árbol de Decisión para la admisión o rechazo de un empleado.

## 2. Código Morse

El Código Morse es un medio para enviar información de texto por medio de una serie de señales de diferente duración y transmisión intermitente. Consiste en que cada letra es transmitida de forma individual con un código que contiene puntos y rayas. La figura 2 muestra el código morse de las letras del alfabeto latino básico de la *International Organization for Standardization* (ISO).

A	.-	F	..-.	K	-.-	P	.-.-.	U	..-
B	-...	G	--.	L	.-...	Q	---.	V	...-
C	-.-.	H	....	M	--	R	.-.	W	.-.-
D	-..	I	..	N	-.	S	...	X	-.-.-
E	.	J	.-.-.-	O	---	T	-	Y	-.--
								Z	--..

Figura 2: Código Morse para las letras alfabeto latino básico.

Para decodificar una secuencia de letras o palabras codificadas en Código Morse, se puede modelar el Código Morse como un árbol de decisión como se muestra en la figura 3. Los nodos del árbol representan las letras del Código Morse, mientras los lados indican los

puntos y rayas que hay que escoger para la composición de las letras. La raíz del árbol es el carácter vacío y es así porque desde la raíz comienza la decodificación de una secuencia de puntos y rayas.

El procedimiento para decodificar una letra es como sigue. Se comienza a recorrer el árbol desde la raíz y se sigue la rama derecha o izquierda, dependiendo del símbolo actual de la secuencia que se este examinando. Si el símbolo es un punto, se recorre el árbol por hijo izquierdo, y si en una raya por el hijo derecho. Se sigue así para todos los símbolos. Si al final del recorrido se llega a una letra, entonces la misma corresponde al código, si por el contrario es NIL, entonces, el código está errado.

Por ejemplo, para decodificar el símbolo “- .-” comenzamos desde la raíz. Luego examinamos el primer símbolo y es una raya, por lo tanto seguimos por la rama derecha y alcanzamos al nodo “T”. Examinamos el siguiente símbolo y es otra raya, por lo tanto nos seleccionamos la rama derecha y llegamos al nodo con la letra “M”. Revisamos el último símbolo de la secuencia y es un punto, por lo que tenemos que tomar la rama izquierda, la cual nos hace llegar al nodo con la letra “G”. Debido a que este es el último nodo al cual arribamos, la secuencia corresponde a la letra “G”. En caso de que se intente decodificar una secuencia que es inválida vamos a terminar el recorrido del árbol en una hoja NIL.

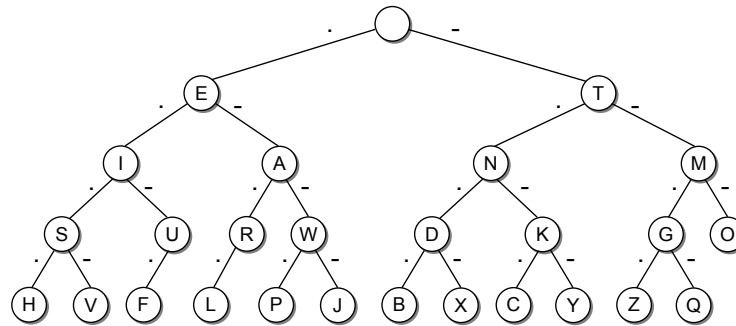


Figura 3: Código Morse modelado como Árbol de Decisión binario.

### 3. Actividad a realizar

Se quiere que implemente el TAD **Código Morse** usando la programación orientada a objetos. El TAD **Código Morse** posee, al menos, los siguientes métodos:

**Crear Código Morse:** Constructor de la clase. Crea un árbol de decisión con las letras del alfabeto latino básico (ISO).

**constructor** CrearCodigoMorse()

**Decodificar Letra:** Función que decodifica una secuencia en Código Morse para obtener la letra que corresponde. Si la secuencia es válida, se retorna la letra correspondiente, en otro caso se retorna Null.

**func** decodificarLetra( **secuencia** : String ) → String

**Decodificar Mensaje:** Función que decodifica una frase escrita con letras en Código Morse. Si todas las letras en Código Morse son válidas, retorna un string con la traducción de la frase. En caso contrario retorna Null. Como convención para separar las letras de una palabra se usará un espacio en blanco “ ” y para separar palabras se usará una barra “/”. Por ejemplo la frase:

.-... --- .../.- .-. -... --- .-... . .../... --- -./-.- ..... . ...- . .-. . ...

se traduce como “los arboles son cheveres”

```
func decodificarMensaje( frase : String ) → String
```

### 3.1. Detalles de la implementación

Debe implementar el TAD **Código Morse** como una clase pública de Kotlin, llamada **CodigoMorseAD**. Esta clase debe estar contenida en un archivo llamado **CodigoMorseAD.kt**. Se quiere que realice un programa cliente para probar el TAD **Código Morse**. Este cliente se llama **PruebaCodigoMorseAD.kt**, y se ejecuta por la línea de comandos, recibiendo como entrada un **String** en código Morse, y muestra por la salida estándar el código de entrada decodificado. Si el código Morse no es válido, entonces el cliente muestra la frase ‘**Error, codigo Morse no valido**’. Para ejecutar el cliente debe realizar un shell script, llamado **runCodigoMorse.sh**, que recibe como entrada el **String** a decodificar. Por último, debe entregar un archivo **Makefile** que compile todos los archivos del proyecto.

## 4. Condiciones de entrega

Los códigos del laboratorio, y la declaración de autenticidad firmada, deben estar contenidas en un archivo comprimido, con formato *tar.xz*, llamado *LabSem10\_X\_Y.tar.xz*, donde *X* y *Y*, son los números de carné de los estudiantes. La entrega del archivo *LabSem10\_X\_Y.tar.xz*, debe hacerse por la plataforma Classroom, antes de las 11:59 PM del día domingo 24 de marzo de 2024.

---

Guillermo Palma / gvpalma@usb.ve / Marzo 2024