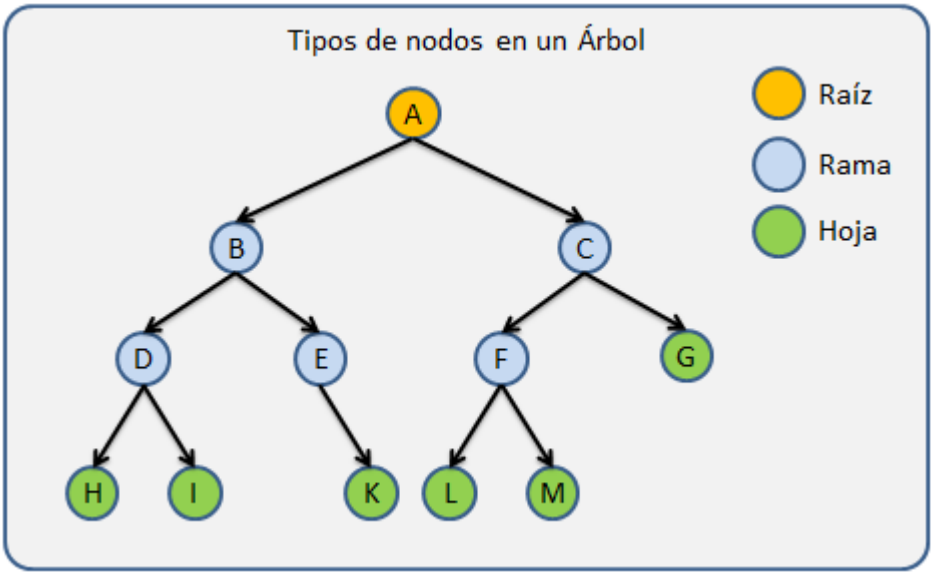
Los arboles de decisión son modelos de clasificación que permiten mediante un determinado criterio establecer que comportamiento toma un dato.

Para esto usaré de ejemplo el dataset de Iris de stklearn.

Pero previo a cualquier código o teoría debemos entender como esta compuesto un árbol.

Composición de un árbol

Podemos decir que un árbol es un grafo dirigido que solo va *"hacia abajo"* en una sola dirección.
Esta compuesto por una raíz, ramas y sus hojas, tal como se muestra continuación:



Estos pueden ser recorridos de diferentes maneras y a su vez escritos, para esto es ideal ver la carpeta de matemática discreta, en donde hay ejercicios realizados manualmente que también se pueden llevar a programación.

¿Qué es, informáticamente hablando, la entropía?

La *entropía* es la unidad que mide el nivel de incertidumbre de un conjunto de datos.

> **Entropía - Cover & Thomas 1991**
>
> *La entropía [...] es una medida de la incertidumbre de la variable aleatoria; es una medida de la cantidad de información requerida en promedio para describir la variable aleatoria*

La **entropía** (H(X)) se define como:

$$H(X) = -\sum_{x \in X} P(x) \log_2 P(x)$$

Ejemplo de árbol de decisión

Es nuestra responsabilidad como programadores relevar el dataset a estudiar y en base a este ver que características son verdaderamente relevantes para determinar la clasificación.
En el caso del dataset de iris queremos saber a que tipo de flor corresponden ciertas características o *features*.
En este caso el dataset no tiene muchas características, por lo que usaremos todas.
A su vez hay que establecer un criterio de decisión, en el caso del ejemplo actual es la entropía.

Pasos a seguir

Primero que nada debemos cargar el dataset de iris, llevarlo a un dataFrame e importar las librerías a utilizar.
Luego debemos separar los datos de entrenamiento.
Creamos el objeto:

```
python arbol_decision = tree.DecisionTreeClassifier(criterion="entropy")
```

Lo entrenamos con nuestros datos con *.fit(datos_entrenamiento, clase_entrenamiento)*.
Luego lo graficamos, acá si entraré más en detalle.
Debemos usar un *plot_tree* con las características que usamos para evaluar nuestros datos y asignar los atributos que queramos, al igual que con cualquier otro gráfico:

```
plt.figure(figsize=(25, 15))
tree.plot_tree(arbol,
               feature_names=["Longitud del sepalo", "Ancho del sepalo", "Longitud del petalo", "Ancho del petalo"],
               class_names=iris.target_names,
               filled=True,
               rounded=True,
               fontsize=12,
               precision=2,
               proportion=True)

plt.title("Árbol de Decisión - Clasificación de Iris", fontsize=16)

plt.show()
```

En este caso el árbol nos quedo bastante grande.
En caso de querer verlo ejecutar la siguiente google colab notebook: click aquí