

**MÉTODO DE LA INGENIERÍA**

**LUISA FERNANDA QUINTERO FERNÁNDEZ**

**JUAN DAVID PELÁEZ VALENCIA**

**NICOLAS ESTEBAN COLMENARES**

**KEVIN ANDRES FERNÁNDEZ**

**PROYECTO INTEGRADOR**

**UNIVERSIDAD ICESI**

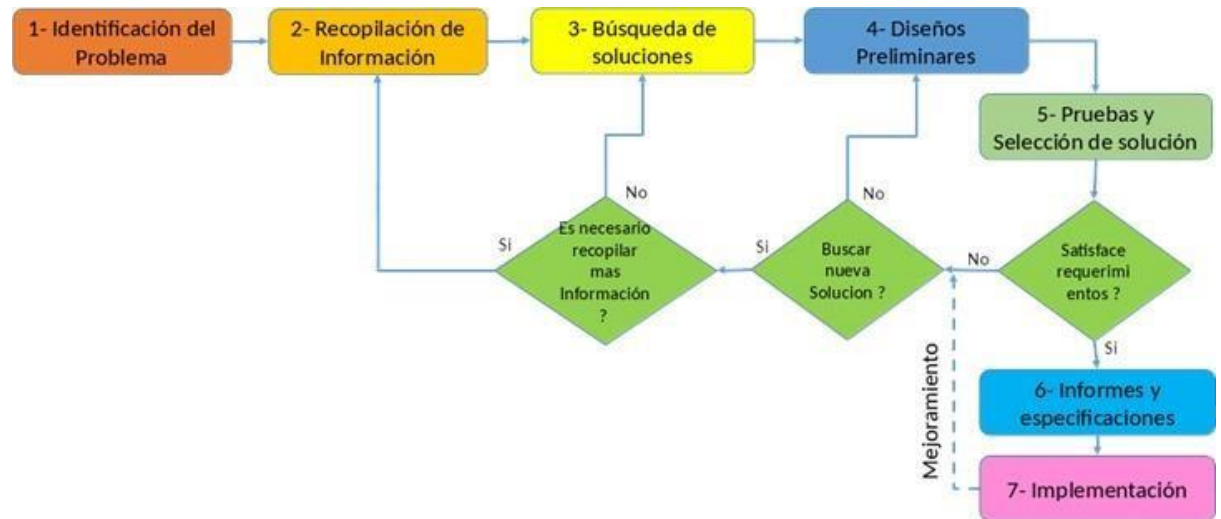
**Contexto problemático**

Un proyecto de investigación de una empresa desea implementar un software que permita generar reportes de la base de datos que poseen en la empresa, la cual maneja distintos tipos, modelos y precios de teléfonos.

Se requiere que, con base en los precios y tipos de teléfonos, se pueda manejar los datos, filtrarlos y a través de gráficos mostrar las estadísticas actuales de los teléfonos.

## Desarrollo de la Solución

Seguiremos estos pasos mostrados en el siguiente diagrama de flujo para llegar al desarrollo de la solución:



### Paso 1: Identificación del problema

#### Identificación de necesidades y síntomas

1. Se requiere un software capaz de realizar la lectura del archivo con los datos adquiridos de la empresa.
2. Se desea un mejor manejo de la información, tal como filtrar la información con base en un criterio específico, y el rango en el que quiere verse la información (si aplica).
3. Se evidencia una necesidad de poder visualizar los datos y sus resultados en forma de gráficos para una mejor comprensión.

## Requerimientos funcionales

**RF1:** Mostrar la base de datos de los dispositivos en una tabla, con un diseño de una tabla de base de datos de SQL. La base de datos será cargada a través de un archivo que el usuario ingrese al programa.

**RF2:** Generar diferentes gráficos sobre la base de datos de los dispositivos, cada gráfico será generado de acuerdo con cada uno de los criterios que contiene la base de datos. Además, serán de distintos tipos, es decir, habrá gráficos de barras, de tortas, de puntos, etc. Con el fin de tener una vista más comprendida por el usuario y poder diferenciar fácilmente cuál dato desea revisar y analizar.

**RF3:** Filtrar la base de datos, el usuario puede filtrar todos los datos de acuerdo con el criterio que desee ver, acorde con sus necesidades. En adición, también se tiene la posibilidad de que, aparte de escoger el criterio que desee, se pueda escoger un rango para los datos o una cantidad exacta de datos para ver. Así el usuario posee una filtración más personalizada.

**RF4:** Limpiar los filtros, el programa puede borrar todos los filtros seleccionados en la tabla para poder colocar unos distintos sin que se acumulen o los filtros se confundan entre sí.

**RF5:** Predecir el valor de un dispositivo, el usuario puede ingresar manualmente un nuevo dispositivo y de acuerdo con distintos tipos de preguntas que posee el programa internamente, se estima su precio y a su vez, predice la gama del teléfono, ya sea baja, alta o media.

**RF6:** Graficar el árbol de decisión, el programa permite que el árbol de decisión con el cual se está entrenando la base de datos, se pueda visualizar. Con el fin de poder verificar y revisar fácilmente el comportamiento de dicho árbol y los nodos que va obteniendo.

**RF7:** Generar clasificación, con base en los valores de atributos de la base de datos. La clasificación se puede realizar de dos maneras, a través del árbol de decisión implementado o, a través de una librería externa implementada.

**RF8:** Probar precisión, con el fin de poder revisar si el entrenamiento del machine Learning está correcto, el programa debe de poder calcular y analizar con una parte de la base de datos, la capacidad de precisión actual. Tanto con la librería externa como el árbol de decisión implementado.

#### Requerimientos no funcionales

**RNF1:** Tener el proveedor Microsoft.ACE.OLEDB.12.0 instalado.

**RNF2:** El archivo subido para leer en el programa, deberá ser un archivo tipo .xlsx

**RNF3:** Se debe utilizar un árbol de decisión para predecir el precio de un dispositivo móvil.

**RNF4:** Se utiliza una librería externa de C#, la librería es “ML.NET” de Microsoft, una librería para Machine Learning.

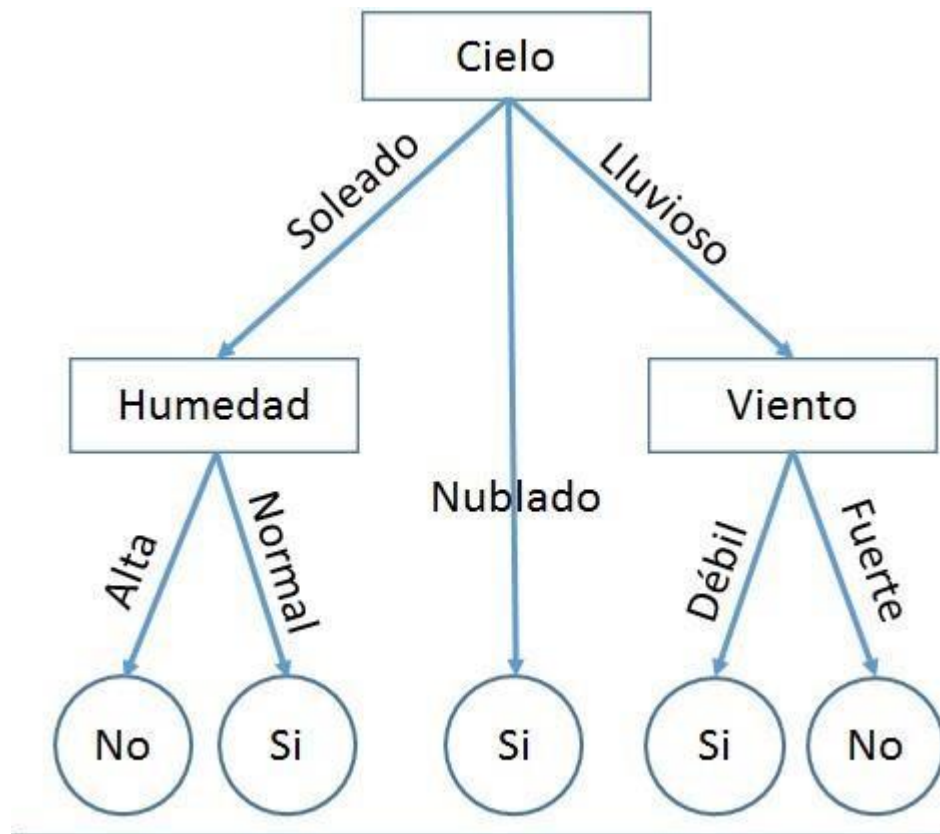
#### Definición del Problema

Una empresa requiere un software para tener un buen manejo de la información para poder generar un buen análisis. A su vez, requiere que el programa realice filtraciones de datos y generación de gráficas para facilitar sus investigaciones.

## Paso 2: Recopilación de la información

### Definiciones

1. **Árbol de decisión:** Es un mapa de los posibles resultados de una serie de decisiones relacionadas. Comienza con un único nodo y luego se ramifica en resultados posibles. Existen tres tipos de nodos: Nodos de probabilidad (círculo), nodos de decisión(cuadrado) y nodos terminales (triángulo)



2. **Microsoft.ACE.OLEDB.12.0:** Facilita la transferencia de datos entre archivos de Microsoft Office System y aplicaciones que no sean de

Microsoft Office, para así poder tratar la información que está en la base de datos.



3. **Lenguaje C#:** Es un lenguaje de programación multiparadigma, el cual toma varios de los funcionamientos de otros lenguajes como C++, C y Java.

Fue desarrollado por la empresa microsoft para su plataforma .NET



4. **Kaggle:** Es una página web de científicos que permite acceder y publicar distintos conjuntos de datos y de código. En el proyecto se utilizó el dataset de 'Mobile Price Classification'



5. **ML.NET:** Es una librería especificada para C# o F# la cual sirve para crear modelos de Machine Learning de acuerdo con la necesidad y personalización del proyecto y del desarrollador.

### **Paso 3: Búsqueda de soluciones creativas**

Para resolver el problema de cómo modelar información de una base de datos, con gráficas, mapa, y demás se obtuvieron las siguientes ideas a través del método de Scamper:

1. Utilizar la herramienta y/o lenguaje de programación java utilizando Scene Builder, para construir y modelar la información como se necesita, además de poder tener.
2. Utilizar Kaggle, como base de datos además de tomar el lenguaje C# para modelar la aplicación y utilizar Microsoft.ACE.OLEDB.12.0 para tratar la información, y poder utilizarla base de datos en nuestra aplicación.
3. Cuando se muestre todas las gráficas, mostrarlas en una misma ventana.
4. Crear una nueva ventana para mostrar todas las gráficas al mismo tiempo, para así poder mostrarlas en un mejor tamaño y con mayor espacio.
5. Utilizar el árbol de decisión para predecir el precio a través de distintas preguntas de clasificación.
6. Utilizar diferentes tipos de gráficos para mostrar las estadísticas.
7. De las distintas maneras de implementar un algoritmo de decisión (ID3, C4.5, C5.0, CART), se escoge la opción de CART.

### **Paso 4. Transición de las Ideas a los Diseños Preliminares**

Para los diseños preliminares se miró cuál de las dos ideas es mejor y elegimos la segunda que es utilizar a C# como herramienta para programar esta aplicación, ya que se facilita la creación de una aplicación, además no toca utilizar un agente externo como Scene Builder cómo sería el caso en java.

## **Paso 5. Evaluación y Selección de la Mejor Solución**

-Criterio A. Conocimiento. La alternativa ha sido estudiada:

- - [2] Sí (se prefiere una solución exacta)
- - [1] No

-Criterio B. Eficiencia temporal. Se prefiere una solución con mejor eficiencia que las otras consideradas. La eficiencia puede ser:

- - [4] Constante
- - [3] Mayor a constante
- - [2] Logarítmica
- - [1] Exponencial

-

-Criterio D. Facilidad en implementación algorítmica:

- - [2] Compatible con las operaciones aritméticas básicas de un equipo de cómputo moderno
- - [1] No compatible completamente con las operaciones aritméticas básicas de un equipo de cómputo moderno.

	<b>Criterio A</b>	<b>Criterio B</b>	<b>Criterio C</b>	<b>Total</b>



<b>CART</b>	2	2	2	6
<b>ID3</b>	1	2	2	5
<b>C4.5</b>	1	1	2	4

-Como todas las soluciones son similares entre sí para la implementación de un árbol de decisión, escogimos implementar el algoritmo de CART debido a que teníamos mayor conocimiento en este algoritmo.

### **Evaluación de librería externa**

CRITERIOS- CALIFICADOS DEL 1 AL 10

C1: Facilidad de uso

C2: Velocidad

C3: Precisión

<b>Alternativas</b>	<b>C1</b>	<b>C2</b>	<b>C3</b>	<b>total</b>
Accord.NET	3	8	8	19
ML.NET	10	10	9	29

Escogimos la librería de ML.NET debido a que tiene mayor facilidad de uso y supera en velocidad y precisión a Accord.NET

## **Bibliografía**

- Mobile Price Classification. (2018, 28 enero). Kaggle.

<https://www.kaggle.com/iabhishekoofficial/mobile-priceclassification?select=test.csv>

- Microsoft. (s. f.). Componente redistribuible del motor de base de

datos de Microsoft Access 2010. Microsoft Downloads. <https://www.microsoft.com/es-co/download/details.aspx?id=13255>

- Lucidchart. 2021. Qué es un diagrama de árbol de decisión. *Lucid Software Inc.* <https://www.lucidchart.com/pages/es/que-es-un-diagrama-de-arbolde-decision>

- Microsoft. ML.NET. Microsoft Machine Learning.

<https://dotnet.microsoft.com/apps/machinelearning-ai/ml-dotnet>

- Marketing. 2020. ¿Qué es C#? *BeSoftware*. <https://bsw.es/que-es-c/>