**Tabla Kanban para el Proyecto: Árbol Binario**

| **Pendiente** | **En Proceso** | **Por Revisar** | **Completado** |
| --- | --- | --- | --- |
|  |  |  | Crear la clase Nodo. |
|  |  |  | Crear la clase ArbolBinario. |
|  |  |  | Implementar el método de inserción en ArbolBinario. |
|  |  |  | Implementar el método de eliminación en ArbolBinario. |
|  |  |  | Implementar el método de búsqueda en ArbolBinario. |
|  |  |  | Pruebas de inserción de nodos. |
|  |  |  | Pruebas de eliminación de nodos. |
|  |  |  | Pruebas de búsqueda de nodos. |
|  |  |  | Documentar las clases Nodo y ArbolBinario. |
|  |  |  | Validar funcionamiento de inserción, búsqueda y eliminación. |
|  |  |  | Documentación final completa. |

**1.Descripción del Reto**

Título del Reto: Implementación de un Árbol Binario

Semana: Semana 1

Objetivo: El objetivo general de este reto es reforzar el entendimiento de estructuras de datos específicas, particularmente los árboles binarios, y aplicar algoritmos para gestionar colecciones de datos de manera eficiente. Se busca que los estudiantes comprendan cómo funcionan las operaciones de inserción, eliminación y búsqueda en un árbol binario de búsqueda, así como la importancia del recorrido en inorden.

Descripción del Problema: El reto consiste en desarrollar un programa que implemente un árbol binario de búsqueda (BST) para gestionar una lista de productos en una tienda. El sistema permitirá a los usuarios añadir, eliminar y buscar productos de manera eficiente. Los productos se almacenarán en el árbol binario, lo que proporcionará un acceso rápido y ordenado. Utilizaremos el recorrido en inorden para mostrar los productos en orden alfabético, lo que facilitará la visualización de la lista de productos disponibles.

**2. Enfoque de la Solución**

Para implementar la solución, se seguirá el siguiente enfoque:

Definición de Clases:

* Clase Nodo: Esta clase representará cada nodo del árbol. Cada nodo contendrá un valor (que en este caso será el identificador del producto), un puntero al hijo izquierdo y un puntero al hijo derecho.
* Clase ÁrbolBinario: Esta clase contendrá la lógica principal para manejar el árbol binario. Incluirá métodos para insertar, eliminar y buscar nodos, así como un método para mostrar los nodos en inorden.

Operaciones Básicas:

* Inserción: Se implementará un método que permitirá insertar un nuevo nodo en el árbol. Este método verificará la posición correcta del nuevo nodo para mantener las propiedades del árbol binario de búsqueda.
* Eliminación: Se implementará un método para eliminar nodos del árbol. Este método manejará diferentes casos de eliminación, como la eliminación de un nodo sin hijos, con un solo hijo, y con dos hijos.
* Búsqueda: Se implementará un método que permitirá buscar un producto en el árbol, devolviendo un valor booleano que indique si el producto está presente o no.

Recorrido en Inorden:

Se implementará un método que recorrerá el árbol en inorden y mostrará los valores de los nodos. Este recorrido es fundamental para mostrar los productos en orden alfabético, ya que se visitan primero los nodos del subárbol izquierdo, luego el nodo actual y finalmente el subárbol derecho.

Pruebas y Validación:

Se realizarán pruebas unitarias para validar que todas las operaciones (inserción, eliminación y búsqueda) funcionan correctamente. Se comprobará que el árbol mantiene su estructura después de cada operación.

