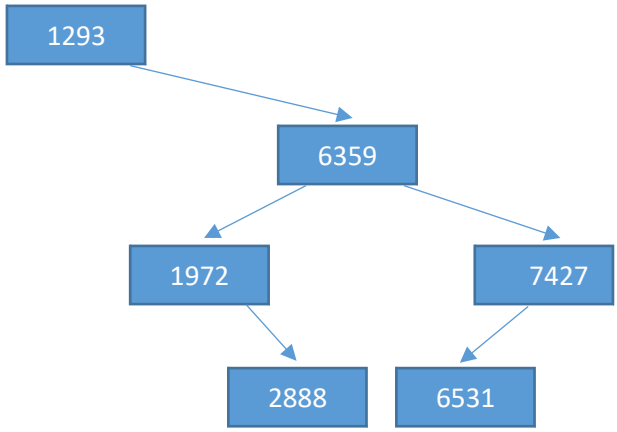
Hoja de trabajo

Parte 1: Conceptos de árboles

Material de referencia:

- Lectura árboles binarios.
  Libro: "*Data Structures and Program Design in C++*" de Kruse y Ryba.
  Páginas: 429-443.
- Lectura árboles binarios de búsqueda.
  Libro: "*Data Structures and Program Design in C++*" de Kruse y Ryba.
  Páginas: 444-462.

Se tiene el siguiente árbol, el cual guarda números de carné de pacientes de una clínica.



1. ¿Qué dato tiene el nodo raíz? 1293
2. ¿Qué dato tiene el nodo que es hijo izquierdo del nodo con carné 1293? null
3. ¿Qué dato tiene el nodo que tiene dos hijos? 6359
4. ¿Cuál es el número de carné más pequeño? 1293
5. ¿Cuál es el número de carné más grande? 7427
6. ¿Cuántos niveles tiene el árbol? 4
7. Se desea buscar el paciente con número de carné 6359. ¿Qué nodos es necesario visitar para encontrar este número de carné? 1293, 6359
8. Se desea buscar el paciente con número de carné 1972. ¿Qué nodos es necesario visitar para encontrar este número de carné? 1293, 6359, 1972
9. Se desea buscar el paciente con número de carné 6531. ¿Qué nodos es necesario visitar para encontrar este número de carné? 1293, 6359, 7427, 6531
10. En sus palabras, ¿cómo se debería recorrer el árbol para encontrar el dato más pequeño? Indique cuales nodos habría que visitar -¿en qué dirección?-. Dado que la naturaleza de un árbol binario es que cualquier subarbol izquierdo tendrá datos menores que su padre, así que me parece que el dato más pequeño debería ser el primer nodo producido por un Preorder Traversal. Esto se traduce a que si queremos el dato más pequeño del árbol, siempre debemos buscar los subarboles izquierdos.

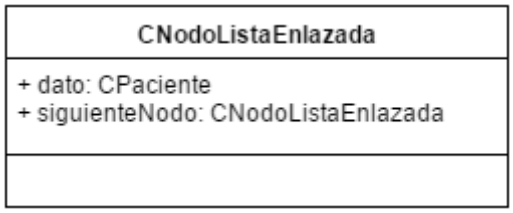
11. En sus palabras, ¿cómo se debería recorrer el árbol para encontrar el dato más grande? Indique cuales nodos habría que visitar -¿en qué dirección?-. Siempre buscar en los subárboles derechos.

Parte 2: Diseño de software del tipo de dato abstracto "Árbol binario de búsqueda"

Pasando a otro tema:

12. ¿Qué es un tipo de dato abstracto? Un tipo de dato que tiene bien definida la relación entre sus componentes y las operaciones que podemos realizar con ellos.

Un nodo de una **lista enlazada** y que guarda datos de un paciente se puede representar por el siguiente diagrama de clases de UML:



13. ¿Qué diagrama de clases propone para un nodo de árbol binario (que guardará datos de un paciente)? ¿Cuántos enlaces deberá tener? ¿Qué nombre le pondrá a cada enlace?

Una clase de nodo de lista enlazada, se puede programar de la siguiente manera.

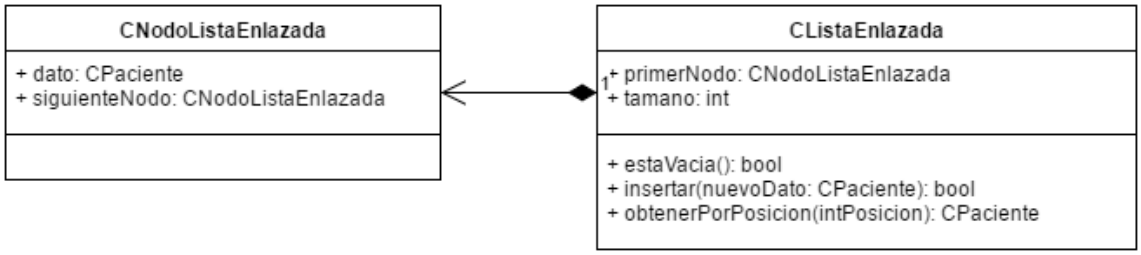
```
public class CNodoListaEnlazada
{
	public CPaciente dato;

	public CNodoListaEnlazada siguienteNodo;

	public CNodoListaEnlazada()
	{
		dato = null;
		siguienteNodo = null;
	}
```

```
        public CNodoListaEnlazada(CPaciente unDato)
        {
                dato = unDato;
                siguienteNodo = null;
        }
}
```

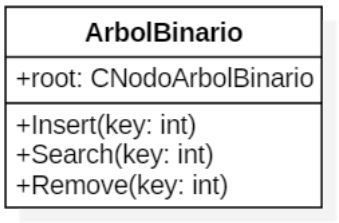
14. Según su diagrama de clases propuesto en la pregunta No. 13, coloque el código de C# para programar dicha clase.

```
public class CNodoArbolBinario<T>
{
    public T dato;
    protected CNodoArbolBinario parent;
    protected CNodoArbolBinario left;
    protected CNodoArbolBinario right;

    public CNodoArbolBinario(T dato, CNodoArbolBinario parent,
            CNodoArbolBinario left, CNodoArbolBinario right)
            {
                this.dato = dato;
                this.parent = parent;
                this.left = left;
                this.right = right;
            }

    public void SetParent(CNodoArbolBinario p) { parent = p; }
    public void SetLeft(CNodoArbolBinario l) { left = l; }
    public void SetRight(CNodoArbolBinario r) { right = r; }
    public CNodoArbolBinario GetParent() { return parent; }
    public CNodoArbolBinario GetLeft() { return left; }
    public CNodoArbolBinario GetRight() { return right; }
}
```

Una lista enlazada se puede representar con el siguiente diagrama de clases.



15. Si el árbol binario se le pueden aplicar las operaciones de consultar si está vacío, insertar y buscar llave, proponga un diagrama de clases para un árbol binario. En el árbol binario, ¿hay un nodo equivalente al primerNodo de la lista enlazada? Suponga que la llave del paciente es su número de carné.



El constructor de una lista enlazada es similar a este:

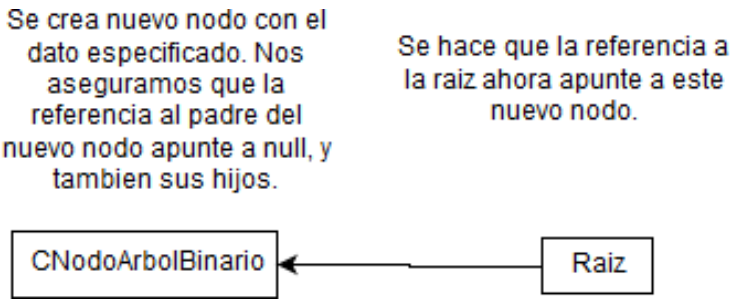
```
public CListaEnlazada()
{
	// Inicialmente la lista está vacía
	primerNodo = null;
	tamano = 0;
}
```

16. ¿Qué operaciones debería de tener el constructor de un árbol binario? Debería iniciar la raiz para que apunte a null, y en caso de manejar un campo que nos indique la cantidad de nodos en el árbol, lo iniciamos a cero.
17. ¿Cómo se sabe si un árbol binario está vacío?

    Podemos rvisar si la referencia a la raíz apunta a null, en caso de ser verdadero, el árbol está vacío.

Parte 3: Inserción en árbol binario

18. Cuando el árbol está vacío, ¿qué operaciones deben realizarse para insertar el primer nodo? Realice una ilustración.



19. Escriba un algoritmo en pseudocódigo para la inserción en árbol binario del primer nodo (el árbol está vacío antes de la inserción).

```
VARIABLES
  CNodoArbolBinario nuevoNodo
INPUT CPaciente p
INICIO
  nuevoNodo = CrearNuevoNodo(p, null, null, null)
  raiz = nuevoNodo;
FIN
```

20. Escriba un método en C# para la inserción en árbol binario del primer nodo (el árbol está vacío antes de la inserción).

```
public void Insertar(CPaciente p)
{
    if(raiz == null)
    {
        CNodoArbolBinario nuevoNodo = new CNodoArbolBinario(p, null, null, null);
        raiz = nuevoNodo;
    }
    else
    {
        // ...
    }
}
```

21. La clínica se inaugura y el primer paciente que se inscribe es el carné No. 1293. Este es el único nodo del árbol. Si ahora se inserta el segundo paciente con número de carné 1344, ¿qué operaciones deben realizarse?
    - Debemos empezar por la raíz de nuestro árbol y comparar claves para encontrar el lugar apropiado para el nuevo paciente.
    - En este caso, 1393 > 1293, así que el lugar apropiado es el hijo derecho de la raíz
    - Creamos un nuevo nodo y hacemos que el hijo derecho de la raíz apunte a este nuevo nodo, y que el padre del nuevo nodo apunte a la raíz.

22. Escriba un algoritmo en pseudocódigo.

```
ALGORITMO Insertar

INPUT
  CPaciente paciente, CNodoArbolBinario nodoActual
VARIABLES
  CNodoArbolBinario nuevoNodo, padre

INICIO
  nuevoNodo = CrearNodo(paciente, null, null, null);
  SI(raiz == null) ENTONCES
  INICIO
    raiz = nuevoNodo
  FIN
  SINO
  INICIO
    SI(paciente.clave > nodoActual.dato.clave)
    INICIO
      SI(nodoActual.derecho == null) ENTONCES
      INICIO
        nodoActual.derecho = nuevoNodo
        nuevoNodo.padre = nodoActual
      FIN
      SINO
      INICIO
        Insertar(paciente, nodoActual.derecho)
      FIN
    FIN
    SINO
    SI(nodoActual.izquierdo == null) ENTONCES
      INICIO
        nodoActual.izquierdo = nuevoNodo
        nuevoNodo.padre = nodoActual
      FIN
      SINO
      INICIO
        Insertar(paciente, nodoActual.izquierdo)
      FIN
    INICIO
      // buscar en el arbol izquierdo
    FIN
  FIN
FIN
```

23. Agregue este caso en el método anterior, codificando en C#.

```
public void Insertar(CPaciente p, CNodoArbolBinario nodoActual)
{
    if(raiz == null)
    {
        CNodoArbolBinario nuevoNodo = new CNodoArbolBinario(p, null, null, null);
        raiz = nuevoNodo;
    }
    else
    {
        if(p.clave > nodoActual.clave)
        {
          if(nodoActual.derecho == null)
          {
            nodoActual.derecho = nuevoNodo;
            nuevoNodo.padre = nodoActual;
          }
          else
          {
            Insertar(p, nodoActual.derecho);
          }
        }
        else
        {
          if(nodoActual.izquierdo == null)
          {
            nodoActual.izquierdo = nuevoNodo;
            nuevoNodo.padre = nodoActual;
          }
          else
          {
            Insertar(p, nodoActual.izquierdo);
          }
        }
    }
}
```

24. Si ahora se inscribe el tercer paciente con número de carné 1258, ¿qué operaciones deben realizarse?
    - 1258 < 1293, entonces se inserta como hijo izquierdo de la raiz.
25. Escriba un algoritmo en pseudocódigo.
    - Es el mismo de arriba
26. Agregue este caso en el método anterior, codificando en C#.

- es el mismo de arriba

27. Si se agrega un cuarto paciente con número de carné 1387, ¿qué operaciones deben realizarse?
    - 1387 > 1293, entonces buscamos por el subarbol derecho de la raiz. 1387 < 1393, y insertamos 1387 como hijo izquierdo de 1393.
28. Escriba un algoritmo en pseudocódigo.
    - Es el mismo de arriba.
29. Agregue este caso en el método anterior, codificando en C#.
    - es el mismo de arriba.
30. Si se agrega un quinto paciente con número de carné 1174, ¿qué operaciones deben realizarse?
    - 1174 < 1293, buscamos en el subarbol izquierdo de la raiz.
    - 1174 < 1258, insertamos 1174 como hijo izquierdo de 1258.
31. Escriba un algoritmo en pseudocódigo.
    - Es el mismo de arriba.
32. Agregue este caso en el método anterior, codificando en C#.
    - es el mismo de arriba.
33. Su método creado en la pregunta anterior, ¿funciona para insertar el nodo con carné No. 1361?

    si.

¡Ya hay una cola de pacientes dispuestos a inscribirse en la clínica! Y eso que es sólo la inauguración.

34. ¿El método está preparado para seguir registrando pacientes sin importar el tamaño del árbol? Si no es así, modifique el método y coloque el código de C#.

- Ya funciona para cualquier caso.