

PROGRAMACION II – ALGORITMOS Y ESTRUCTURAS DE DATOS II

FINAL ADELANTADO 1/7/24

A saber:

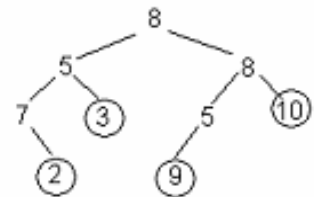
- El examen se implementa en las computadoras del aula (sin excepción). Se utiliza el entorno ECLIPSE.
- En los ejercicios que se pide “implementación”, se utiliza JAVA.
- No podrán tener material bibliográfico para consultar.
- Duración del examen: 3 hs.
- Para aprobar el examen debe estar bien un 50% de cada ejercicio.
- Los ejercicios “a mano” se adjuntan en un doc al .zip creado con apellido y nombre del alumno, que deberán ser enviado a la siguiente casilla de correo: lafelice@uade.edu.ar

1- Dado un **árbol binario** de enteros:

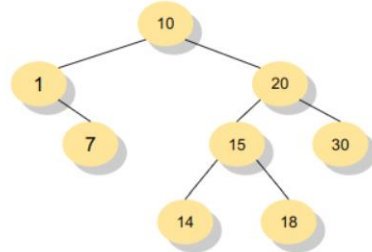
- implemente un método recursivo que recorra el árbol y almacene en una **lista enlazada** las hojas del árbol. Para el árbol del ejemplo debería almacenar <2, 3, 9, 10>.
- Muestre la salida en pantalla.
- Calcule la complejidad temporal del método.

Nota: Se dispone de la interfaz (básica) de la clase árbol binario. Implemente los métodos que necesita para resolver el ejercicio. Debe implementar el método “agregar a la lista”. Ambas son estructuras dinámicas.

```
public interface TDAArbol {  
    //Indica si el arbol está vacío  
    boolean EsVacio();  
    //Crea un arbol vacío  
    void InicializaArbol();  
    //Devuelve el sub-Arbol izquierdo  
    TDAArbol Hijolzquierdo();
```



2. Para el **ABB** de la figura, supongamos que se agrega el elemento (16).
- Muestre los estados intermedios por los que pasa el árbol hasta encontrarse nuevamente con la propiedad inicial luego de agregar el elemento dado.
 - Calcule las **alturas** y escríbalas sobre cada nodo.



3. Dada la interfaz de Diccionario múltiple que se da a continuación, se pide:
- Justifique la representación elegida para la estructura de datos (dinámica) de la clase.
 - Dé una implementación para los métodos Agregar y Eliminar.
 - Analice la complejidad temporal de los métodos.

```

public interface DiccionarioMultipleTDA {
    void InicializarDiccionario ();
    // siempre que el diccionario esté inicializado
    void Agregar( int clave , int valor);
    // siempre que el diccionario esté inicializado
    void Eliminar( int clave);
    // siempre que el diccionario esté inicializado
    void EliminarValor( int clave , int valor);
  }

```

4. Dibuje una representación con estructura dinámica para los siguientes grafos:

