



Nombre y Apellido:

Legajo:

## Examen Parcial

Tiempo mínimo para el examen: 1 hora reloj.

Tiempo máximo para el examen: 2 horas reloj.

### Práctica

1. Sabiendo que la clase `Libro` tiene los métodos `obtener_autor` y `obtener_titulo` que devuelven cadenas de caracteres, escribir la clase `Biblioteca` con los métodos:

A `agregar_libro` que recibe un `Libro` y lo agrega a la colección.

B `sacar_libro` que recibe el nombre de un título y el de un autor y lo saca de la biblioteca, devolviéndolo o imprimiendo en caso de que los datos no correspondan con los de algún libro agregado.

C `contiene_libro` que recibe el nombre de un título y el de un autor y devuelve `True` o `False` de acuerdo a si está en la colección o no.

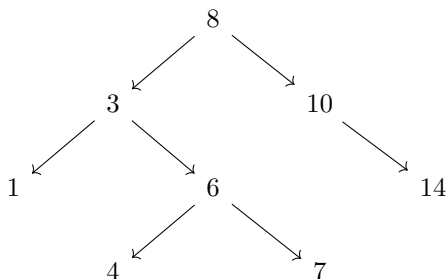
2. Implemente para la clase `Lista Enlazada`, un método `intercambiar(i, j)` que reciba dos índices válidos `i` y `j`, e intercambie los valores en las posiciones `i` y `j`.

**Nota:** Puede considerar que están definidos todos los métodos de la interfaz de `Lista` (`__str__`, `__len__`, `append`, `insert`, `remove`, `pop` e `index`).

```
class ListaEnlazada:
    ...

    def intercambiar(self, i: int, j: int) -> None:
        # COMPLETAR
        ...
```

3. Implemente una función `reflexión`, que recibe un árbol binario y devuelve un árbol donde, para cada nodo, su hijo izquierdo es reemplazado por su hijo derecho, y viceversa.
4. Considere el siguiente árbol



1. Decida si se trata o no de un árbol binario de búsqueda.
2. Si se trata de un árbol binario de búsqueda, inserte el valor 13 en el mismo. Si no se trata de un árbol binario de búsqueda, justifique porqué.

## Teoría

Multiple-Choice. Para cada elemento, marque la opción correcta y justifique.

5. ¿Como se llama la condición que ocasiona que una función recursiva deje de llamarse a si misma?

- ☐ Caso base.
- ☐ Peor caso.
- ☐ Mejor caso.
- ☐ Ninguna de las anteriores.

6. Considere la siguiente función recursiva. ¿Que imprime la llamada `funcion(5)`?

```
def funcion(contador):  
    if contador == 0:  
        return  
    funcion(contador -1)  
    print(contador)
```

`funcion (5)`

- ☐ Los números del 5 al 1, inclusive, en ese orden.
- ☐ Los números del 1 al 4, inclusive, en ese orden.
- ☐ Los números del 1 al 5, inclusive, en ese orden.
- ☐ Ninguno de los anteriores.

7. Considere la siguiente función recursiva. ¿Cual es el resultado de llamar a `g([1,2,4], 2)`?

```
def g(xs: list, n: int) -> bool:  
    if len(xs) == 0:  
        return False  
    if xs[0] == n:  
        return True  
    return g(xs[1:], n)
```

- ☐ True
- ☐ False
- ☐ La recursión no termina.
- ☐ El código da un error.