

Nombre y Apellido	DNI	Comisión

INSTRUCCIONES. El parcial debe resolverse de manera individual. El mismo debe ser resuelto de puño y letra. Se tendrá en cuenta la caligrafía y la claridad en la exposición.

## Programación OO y TADS

Dada la interface listas y una posible implementación de estas con listas enlazadas con la siguiente estructura:

```
public class ListaEnlazada<T> implements Lista<T>{
    private Nodo<T> comienzo;
    private int cantidad;

    public boolean isEmpty(){
        return (this.cantidad == 0)
    }
    ...
    // siguen mas metodos
}
```

En donde la clase Nodo es de la siguiente forma:

```
public class Nodo<T>{
    private T info;
    private Nodo<T> next;

    public Nodo<T> getNext(){
        return this.next;
    }
    public setNext(Nodo<T> next){
        this.next = next
    }
    ...
    // siguen mas metodos
}
```

(a) [2 ptos] Implementar el método `eliminarPrimero` en la clase `ListaEnlazada` que borra el primer elemento de la lista. Agregar el código faltante (anotado con `---`) en la siguiente implementación:

```
public void eliminarPrimero(){
    if( !this.isEmpty() ) {
        this.comienzo.setNext( this.comienzo.getNext().getNext( ) )
    }
    this.cantidad--;
}
```

(b) [3 ptos] Implementar el método `buscar` en la clase `ListaEnlazada` que retorna el *i*-ésimo elemento, empezando desde 0 . Agregar el código faltante en la siguiente implementación:

```
public T buscar(int i){
    if ( this.isEmpty() )
        throw new ListaException("Lista_Vacia");

    if ( i < 0 || i >= this.cantidad )
        throw new ListaException("Indice_fuera_de_Rango");

    Nodo<T> aux = this.comienzo.getNext();
    int k = 0;
    while(____aux.getNext() != null____){
        if ( ____i == k____ ){
            return ____aux.getInfo()____;
        }
        k++;
        ____aux = aux.getNext()____; // avanza aux
    }
}
```

## Programación Funcional

### Pregunta (c) [3 puntos]

Dadas las siguientes expresiones en Haskell, determine el tipo resultante:

1. `((:) 1 . tail) [0,2,3]` `[Int]`
2. `(odd . pred . succ) 10` `Bool`
3. `(fst . head) [(1,'a'), (2,'b')]` `Int`

### Pregunta (d) [2 puntos]

Se quiere hacer una función recursiva `sinDups` que elimine repetidos consecutivos de una lista. Por ejemplo:

```
> sinDups [1,2,2,3,10,10,5]
[1,2,3,10,5]
```

Completar el siguiente código (reemplazar `undefined` por el código correspondiente):

```
sinDups :: Num a => [a] -> [a] -- completar el perfil
sinDups [] = []
sinDups [x] = undefined
sinDups (x:(y:ys)) = undefined
```

**Pregunta BONUS (e) [2 puntos]**(sólo se suma si la nota de los ejercicios anteriores es  $\geq 5$  y  $< 10$ ) (satura en 10)

¿Qué política de pasaje de parámetros utiliza Java? Explicar con ejemplos.