

Transcripción – Rubber Duck Debugging Método

insertarFinal()

Código original

```
public void insertarFinal(int dato){  
    Nodo nuevo = new Nodo(dato);  
    if(cabeza == null){  
        cabeza = nuevo;  
    }else{  
        Nodo actual = cabeza;  
        while(actual.siguiente != null){  
            actual = actual.siguiente;  
        }  
        actual.siguiente = nuevo;  
    }  
    tamaño++;  
}
```

Preguntas de la IA y mis respuestas

IA: ¿Puedes explicarme qué hace la primera línea de tu código y por qué es necesaria?

Yo: Creo un nuevo nodo con el dato que quiero insertar. Es necesario porque cada elemento de la lista debe guardarse en un nodo.

IA: ¿Qué pasa si la lista está vacía?

Yo: Si cabeza es null, hago que la cabeza apunte al nuevo nodo para que sea el primero.

IA: ¿Cómo sabes que llegaste al último nodo?

Yo: Porque el último nodo tiene siguiente = null. El ciclo termina cuando encuentro ese valor.

IA: ¿Qué pasaría si olvidas actualizar el tamaño?

Yo: El contador ya no reflejaría la cantidad real de nodos y podría causar errores en otros métodos.

Bugs o mejoras descubiertas

No estaba validando si el nodo nuevo debía apuntar a null (aunque normalmente el constructor lo hace).

El recorrido es O(n); si la lista fuera muy grande podría ser lento. Podría mejorar guardando una referencia al último nodo.

Código corregido (mejora opcional)

```
public void insertarFinal(int dato){  
    Nodo nuevo = new Nodo(dato);  
    nuevo.siguiente = null;  
  
    if(cabeza == null){  
        cabeza = nuevo;  
        cola = nuevo; // referencia al último nodo  
    }else{  
        cola.siguiente = nuevo; cola  
        = nuevo;  
    }  
    tamaño++;  
}
```

Método 2: eliminarInicio()

(1) Código original

```
public void eliminarInicio(){ if(cabeza  
!= null){  
    cabeza = cabeza.siguiente;  
    tamaño--;  
}  
}
```

Preguntas de la IA y mis respuestas

IA: ¿Qué hace esta línea cabeza = cabeza.siguiente?

Yo: Hace que la cabeza avance al siguiente nodo, eliminando el primero.

IA: ¿Qué pasa si la lista ya está vacía?

Yo: No entra al if, así que no ocurre nada.

IA: ¿Qué pasa si solo hay un nodo?

Yo: La cabeza se vuelve null, lo que significa que la lista queda vacía.

IA: ¿Estás seguro de que no hay otra referencia que debas actualizar?

Yo: Si uso una cola, también debería volverla null.

Bugs o mejoras descubiertas

Olvidé actualizar la cola cuando se elimina el único nodo.

No aviso si la lista está vacía (aunque no siempre es obligatorio).

(2) Código corregido public

```
void eliminarInicio(){  
    if(cabeza == null){ return;  
    }  
  
    cabeza = cabeza.siguiente;  
  
    if(cabeza == null){  
        cola = null;  
    }  
  
    tamaño--;  
}
```

Método 3: buscar(int dato)

(1) Código original

```
public boolean buscar(int dato){  
    Nodo actual = cabeza;  
  
    while(actual != null){  
        if(actual.dato == dato){  
            return true;  
        }  
        actual = actual.siguiente;  
    }  
}
```

```
    }  
  
    return false;  
}
```

Preguntas de la IA y mis respuestas

IA: ¿Por qué empiezas desde la cabeza?

Yo: Porque es el inicio de la lista; debo recorrerla completa.

IA: ¿Qué pasa si el dato está en el último nodo?

Yo: El ciclo lo revisa antes de que actual sea null, así que sí lo encuentra.

IA: ¿Qué ocurre si la lista está vacía?

Yo: actual sería null y el ciclo no se ejecuta, entonces retorna false.

IA: ¿Hay alguna forma de hacerlo más rápido?

Yo: No realmente, porque las listas enlazadas requieren recorrido secuencial ($O(n)$).

Bugs o mejoras descubiertas

No hay errores graves.

Podría devolver el nodo en lugar de boolean si necesitara manipularlo después.

Código corregido

```
public Nodo buscar(int dato){  
    Nodo actual = cabeza;  
  
    while(actual != null){  
        if(actual.dato == dato){  
            return actual;  
        }  
        actual = actual.siguiente;  
    }  
  
    return null;  
}
```

Conclusión de la sesión Rubber Duck

Explicar mi código línea por línea me ayudó a notar errores que no había visto, especialmente con referencias como la cola. También confirmé que entender un método significa poder explicarlo sin dudar.