Universidad Nacional Autónoma de México

|  |  |
| --- | --- |
| Profesor: | M. en C. Alicia Montserrat Alvarado González |
| Asignatura: | Estructura y algoritmo de datos |
| Grupo: | 2 |
| No de Práctica(s): | 6 |
| Integrante(s): | Cortés García Jaime Rafael |
|  |  |
|  |  |
| Semestre: | 2017 - 1 |
| Fecha de entrega: | 26 de Septiembre del 2016 |
| Observaciones: |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

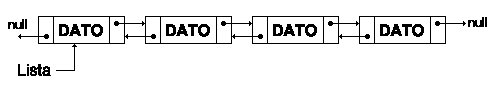
Introducción:

Empecemos con la cola circular, el único cambio que se realiza a una cola normal, es que, en la circular, el ultimo nodo que será apuntado por el apuntador cola, no apuntara a NULL, como lo hace una cola normal, sino que este último nodo apuntara a el primer nodo, de ahí el nombre de cola “circular”, porque podemos trabajarlo análogamente como un circulo, en el cual inevitablemente el final está ligado con el principio. Así, por ejemplo, si recorriéramos la cola, al llegar al último nodo, el siguiente nodo, sería el primer nodo.

Ahora hablemos un poco de la cola doblemente ligada, el único cambio que se le realiza es que nuestro nodo ahora va a contener dos apuntadores, en lugar de uno, como en la cola sencilla, de estos dos apuntadores, uno apuntara al nodo siguiente, y el otro nodo apuntara al nodo anterior. Veamos que nos puede contar un libro acerca de esta temática. Como tal la siguiente definición se refiere a las listas, sin embargo, solo habría que considerar un apuntador a la cola en esta lista, para tener nuestra cola doblemente ligada.

“En una lista doblemente enlazada, a diferencia de una lista simplemente enlazada, cada elemento tiene información de dónde se encuentra el elemento posterior y el elemento anterior. Esto permite leer la lista en ambas direcciones. Este tipo de listas es útil cuando la inserción, borrado y movimiento de los elementos son operaciones frecuentes. Una aplicación típica es un procesador de textos, donde el acceso a cada línea individual se hace a través de una lista doblemente enlazada. Las operaciones sobre una lista doblemente enlazada normalmente se realizan sin ninguna dificultad. Sin embargo, casi siempre es mucho más fácil la manipulación de las mismas cuando existe un doble enlace entre el último elemento y el primero, estructura que recibe el nombre de lista circular doblemente enlazada. Para moverse sobre una lista circular, es necesario almacenar de alguna manera un punto de referencia; por ejemplo, mediante un puntero al último elemento de la lista.” (Fco. Javier Ceballos, C/C++ Curso de Programación, Ra-Ma, 3° edición, 2007)

En una imagen:



Como podemos observar en esta imagen, que hace referencia a una lista doblemente ligada, la única implementación que habría que hacerle para convertirla en una cola doblemente ligada, sería crear un apuntador que apunte a la cola de esta lista, y el apuntador lista fungiría como el apuntador cabeza. Es importante recalcar como tanto la cabeza como la cola de esta estructura de datos dinámicos, deben apuntar a NULL para poder así saber cuál es la cabeza y cuál es la cola, para poder manejarla bien.

Objetivo:

Revisaras las definiciones, características, procedimientos y ejemplos de las estructuras lineales Cola circular y Cola doble, con la finalidad de que comprendas sus estructuras y puedas implementarlas.

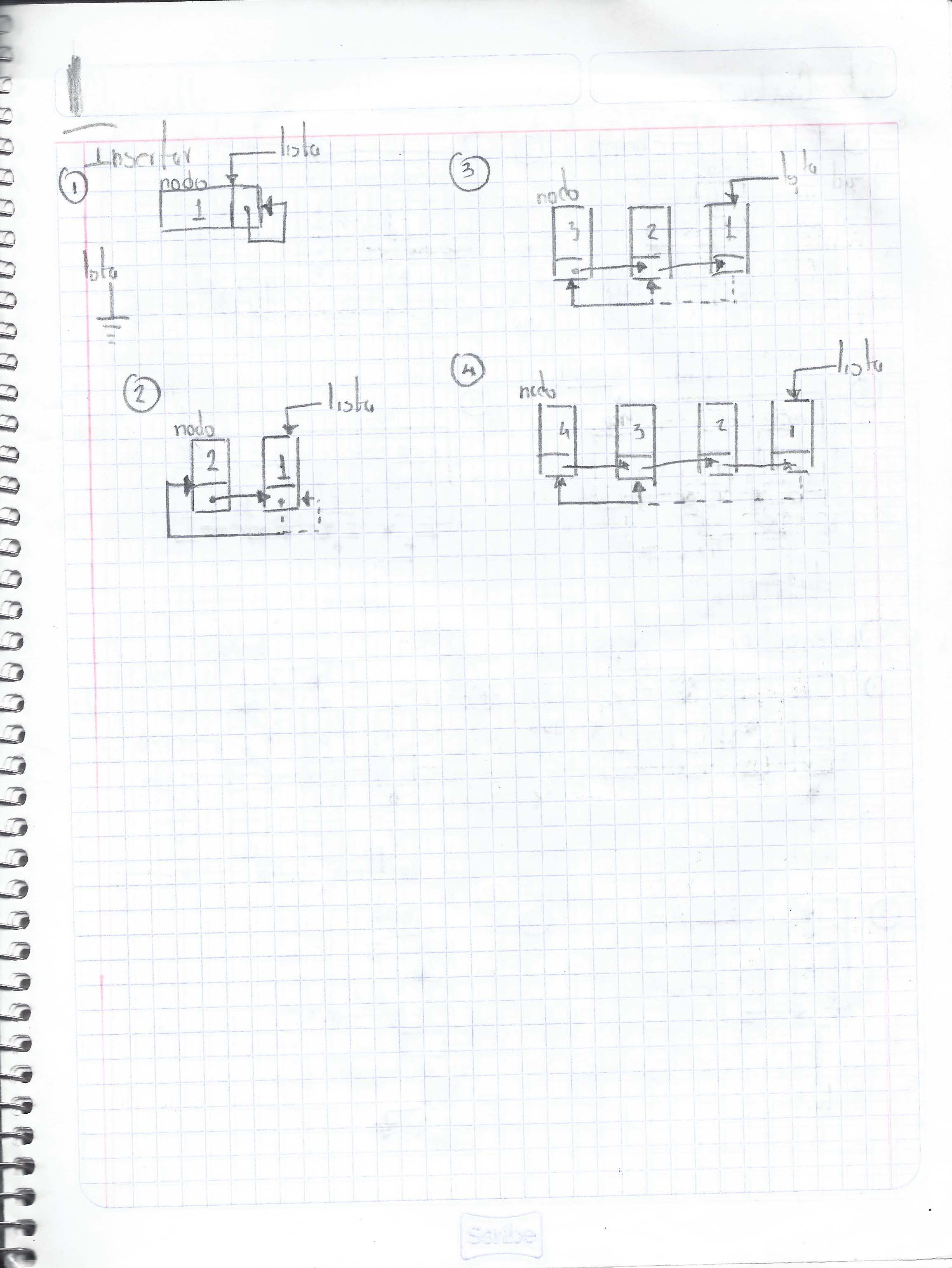
Materiales:

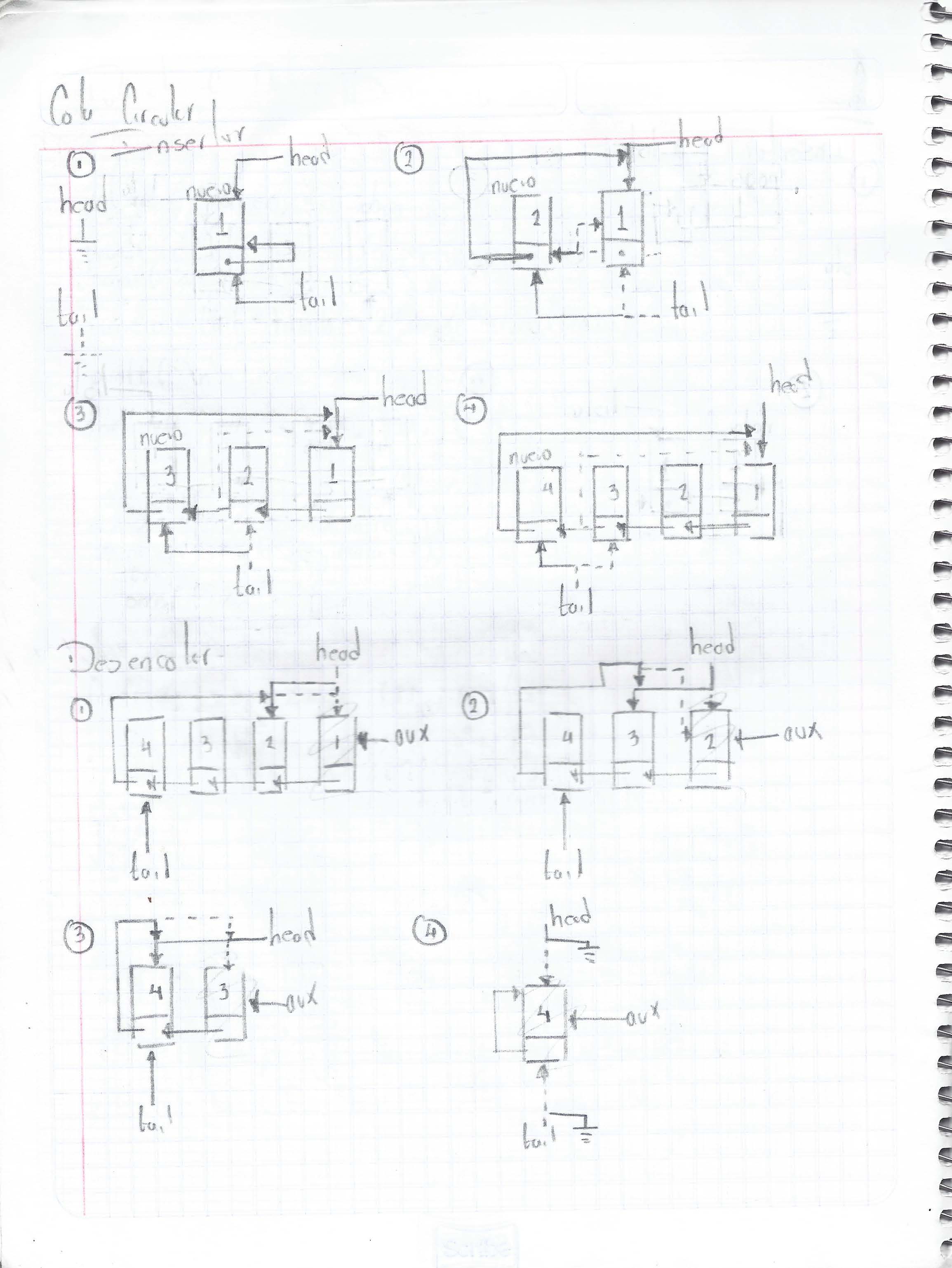
* Edición: Windows 10 Home Single Language
* Versión: 1607
* Compilación del sistema operativo: 14393.222
* Id. Del producto: 00327-30364-13497-AAOEM
* Procesador: AMD A10-8700P Radeon R6, 10 Compute Cores 4C+6G 1.80 GHz
* RAM instalado: 16.0 GB (15.4 GB usable)
* Tipo de Sistema: Sistema operativo de 64 bits, procesador x64
* Programa de Programación: Code Blocks

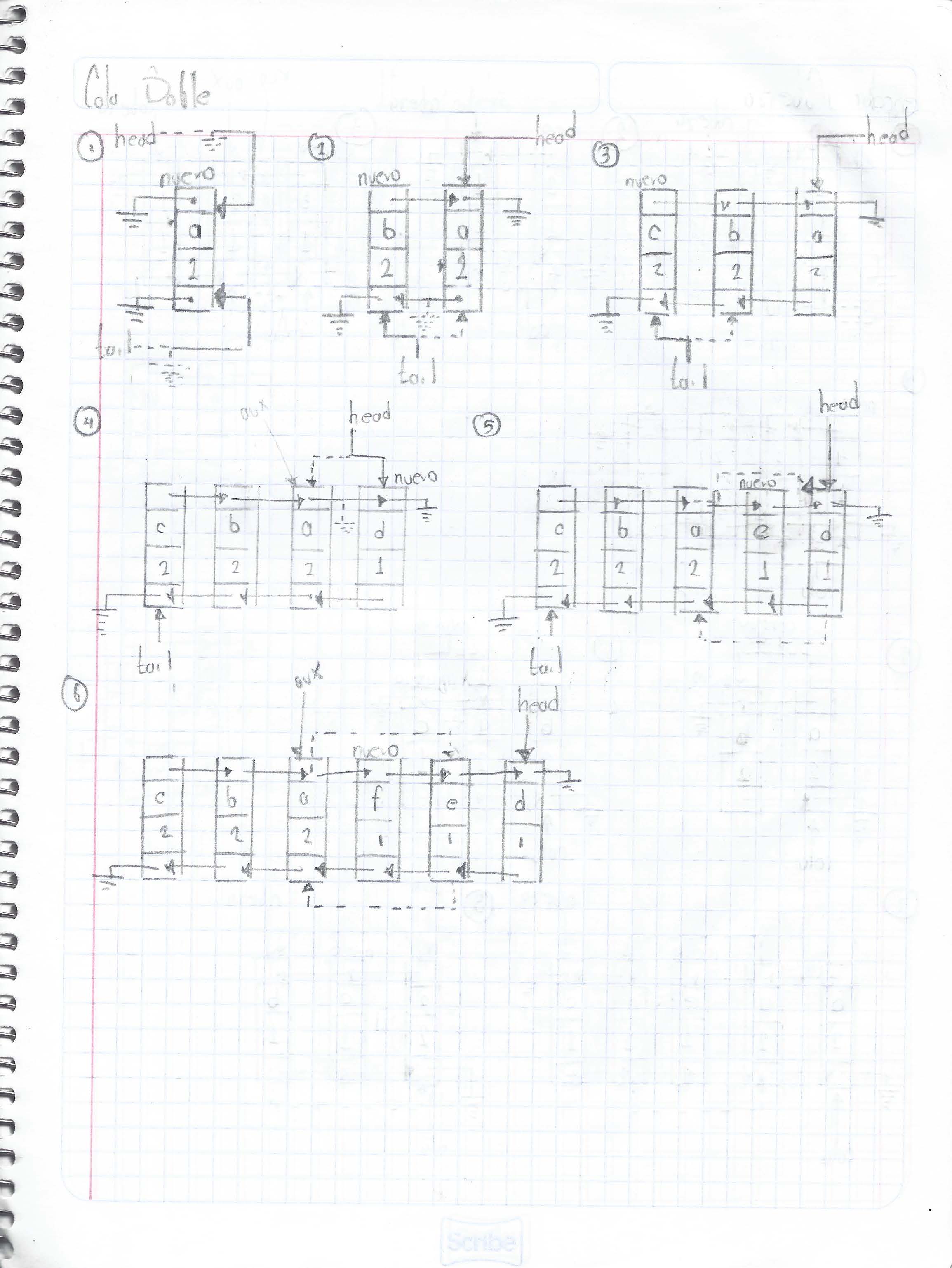
Método:

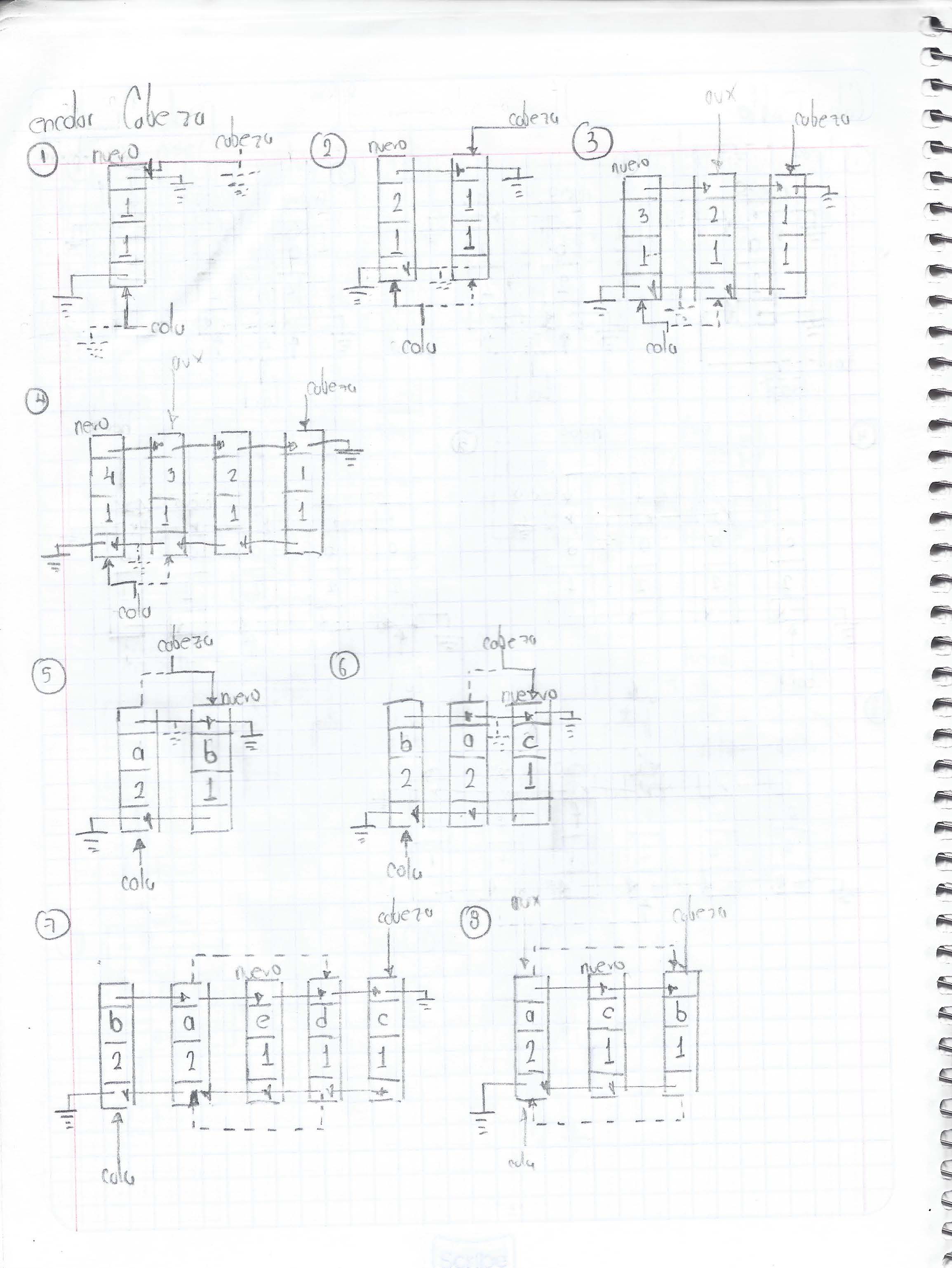
El método que elegí para poder llevar acabo con existo todos estos programas, puedo decir con alegría y satisfacción que fue haciéndolos desde el principio, al tener ahora una noción clara de cómo funcionaban estas estructuras de datos dinámicos, comprendí que podía diseñarlos desde el principio, así que a partir de esquemas que hacía en mi cuaderno, finalmente fui capaz de reunir todos los casos posibles en estas colas, para implementarlas en un mismo código.

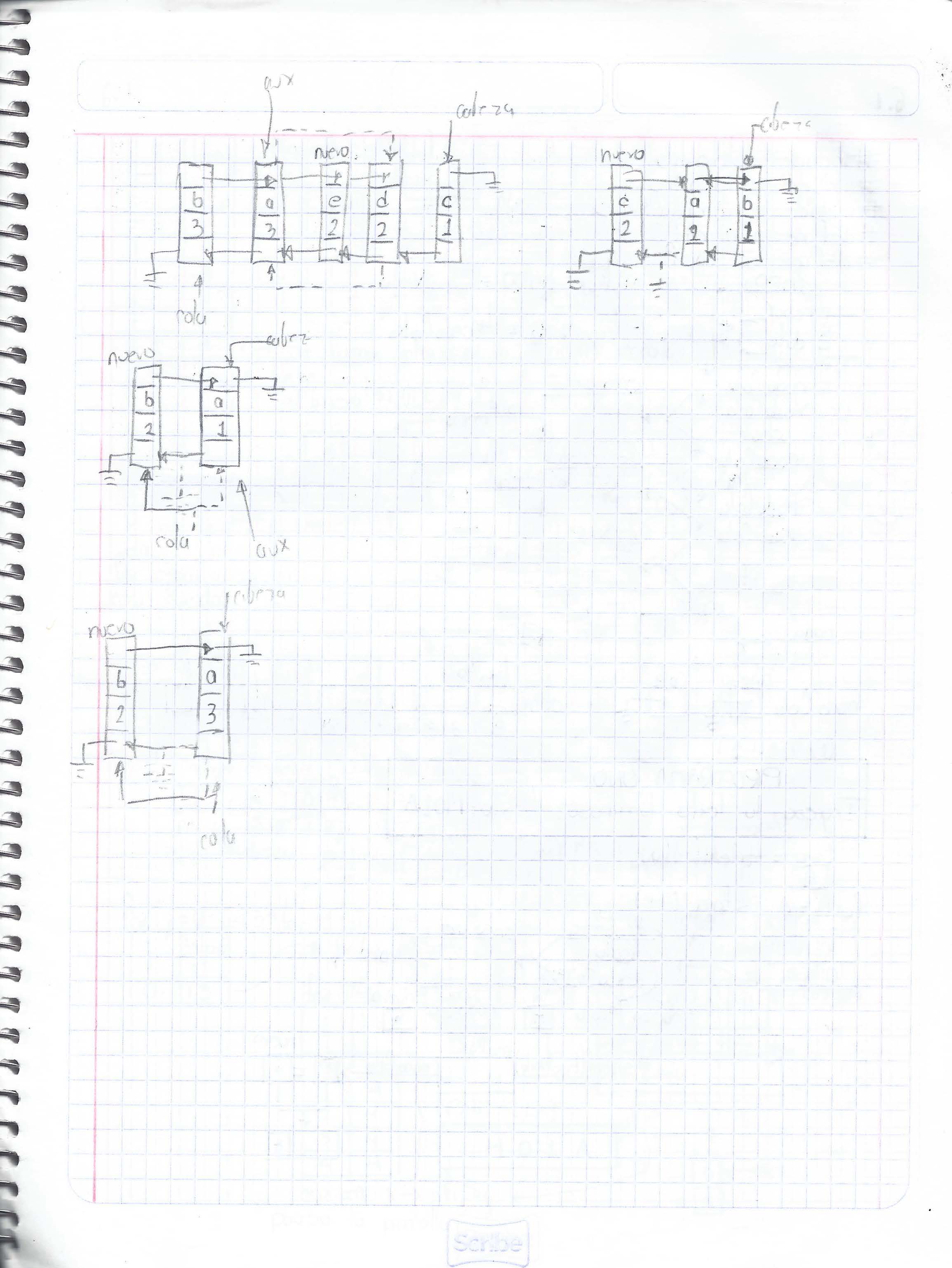
Así que he aquí mis esquemas preliminares con los que pude visualizar de mejor manera lo que ocurría en la computadora.

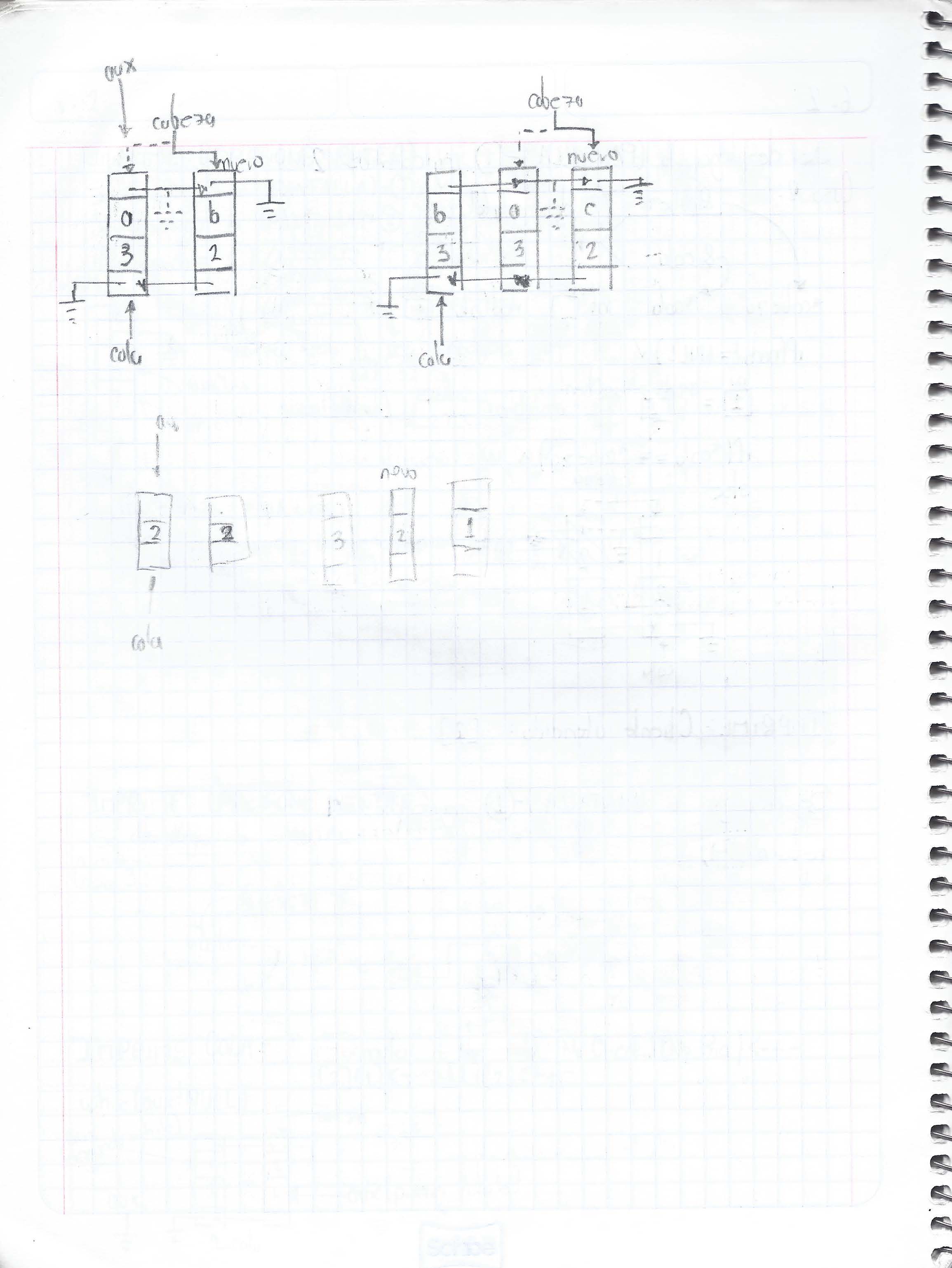




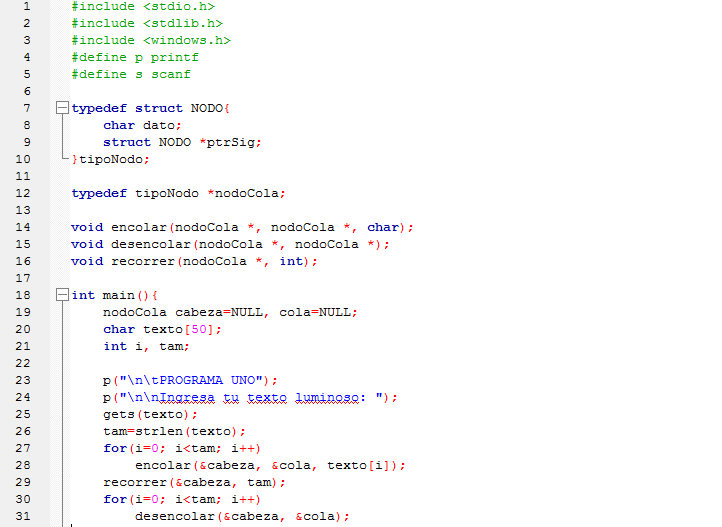


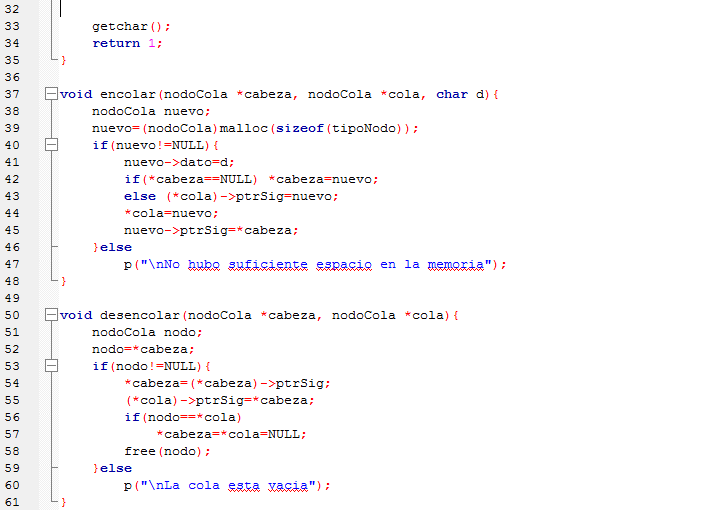


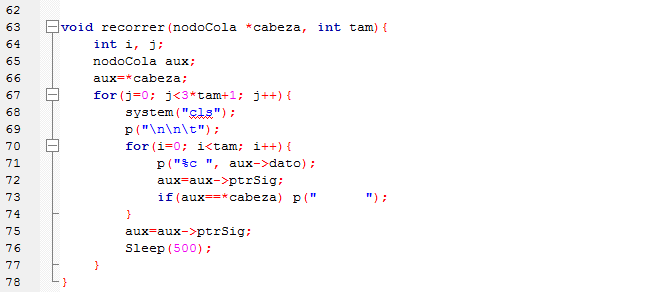


****

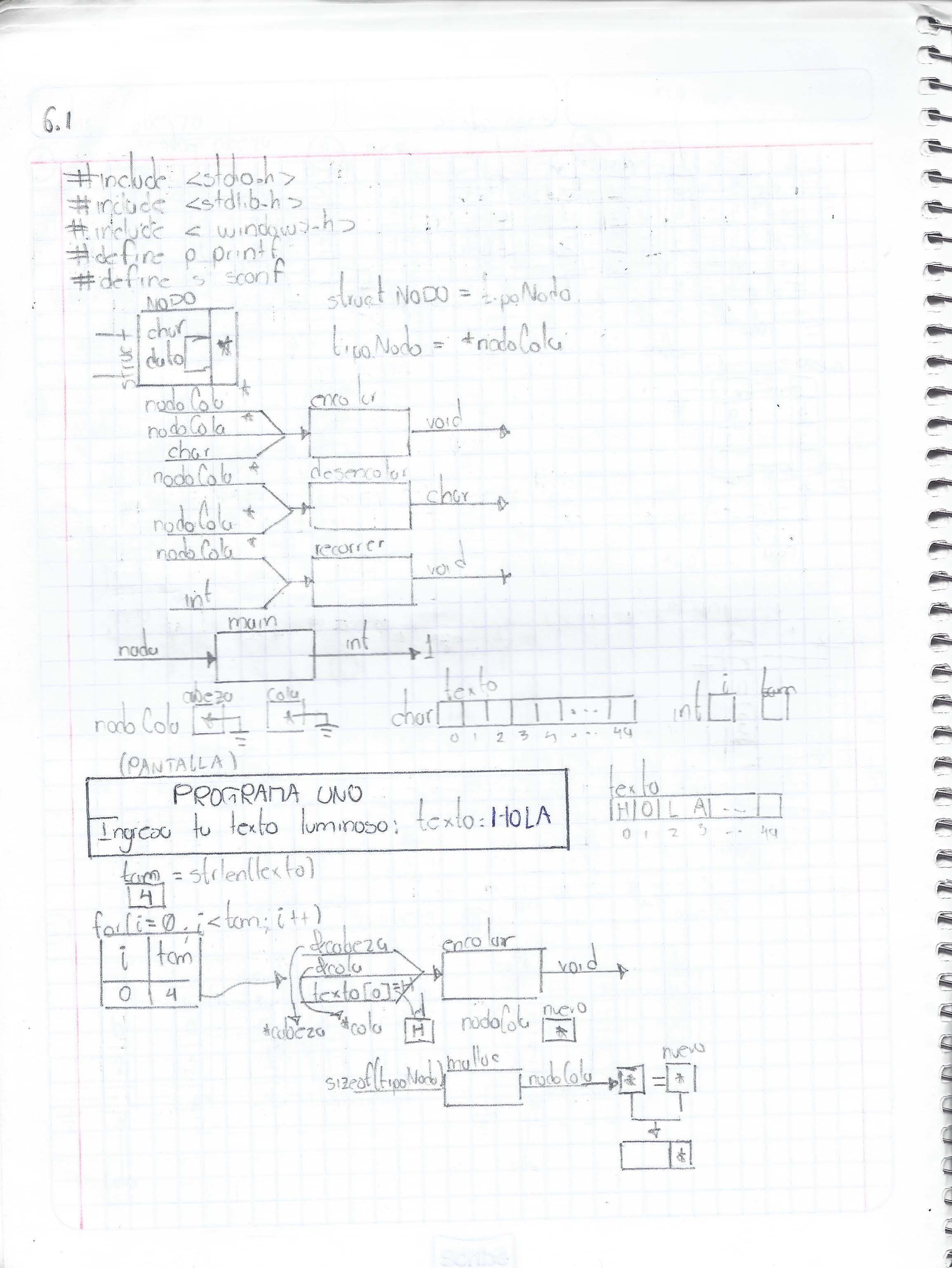
**Ejercicio 1:** Aviso de texto luminoso.

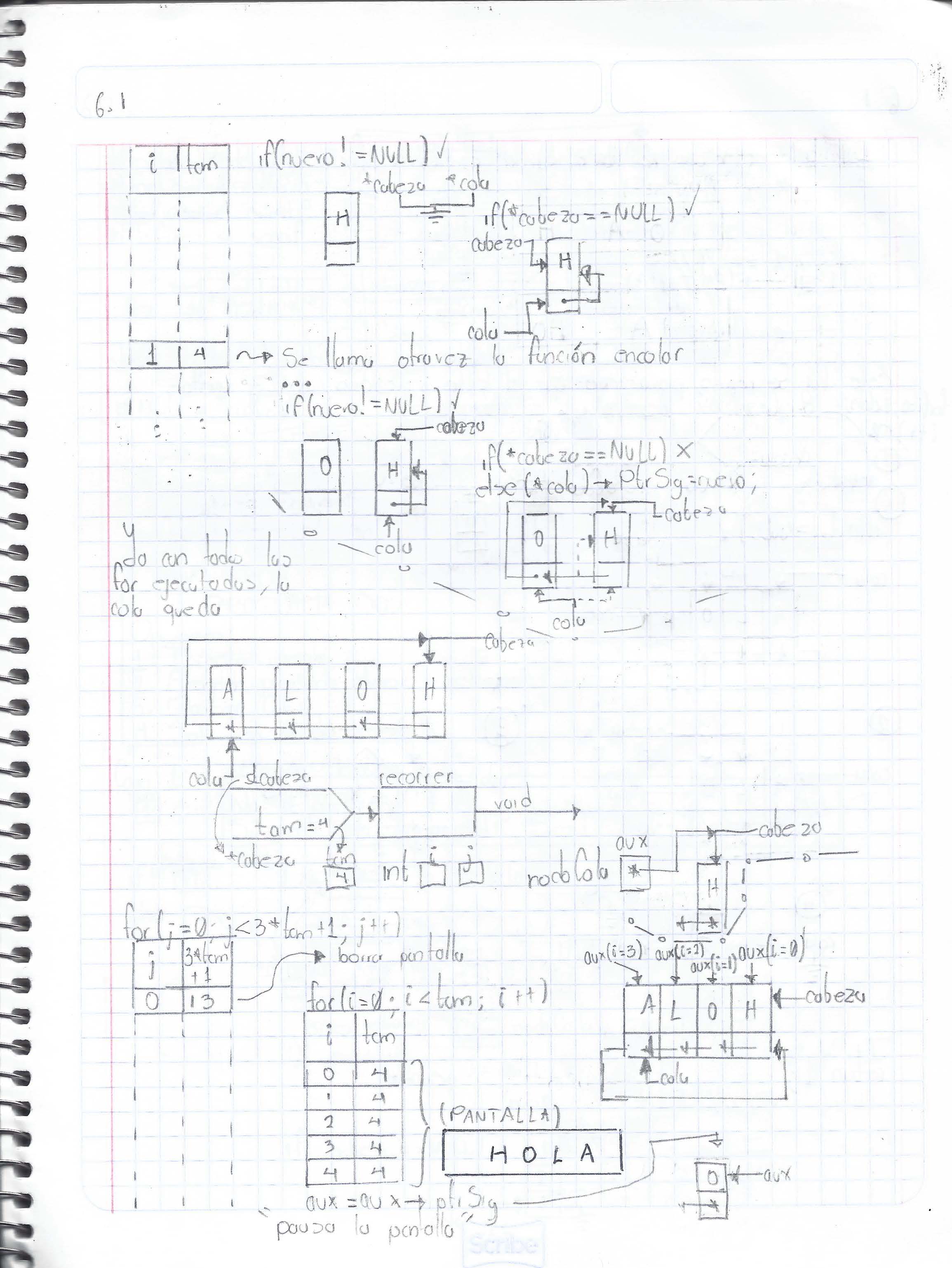






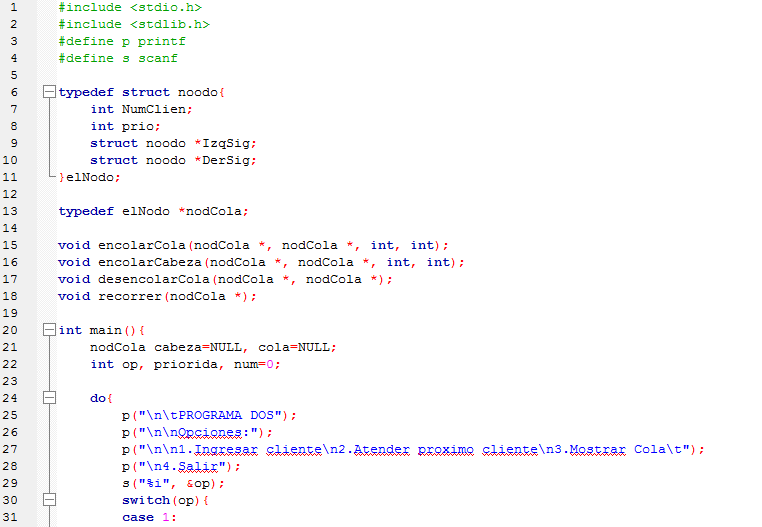
**Prueba de escritorio:**

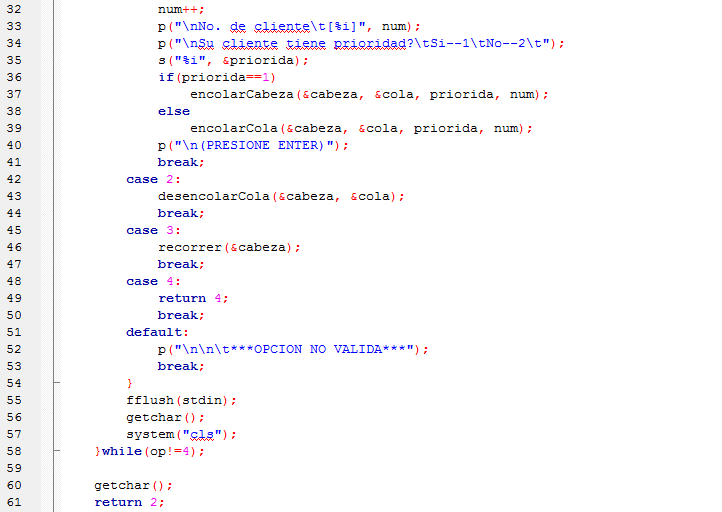
****

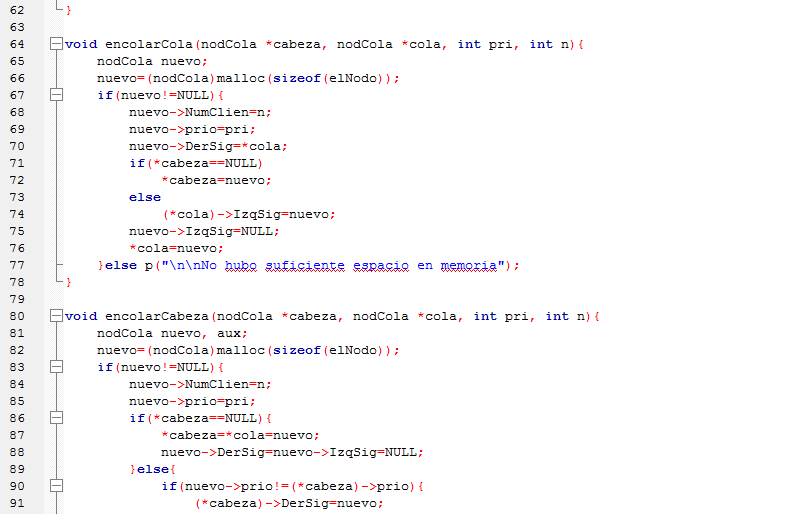
****

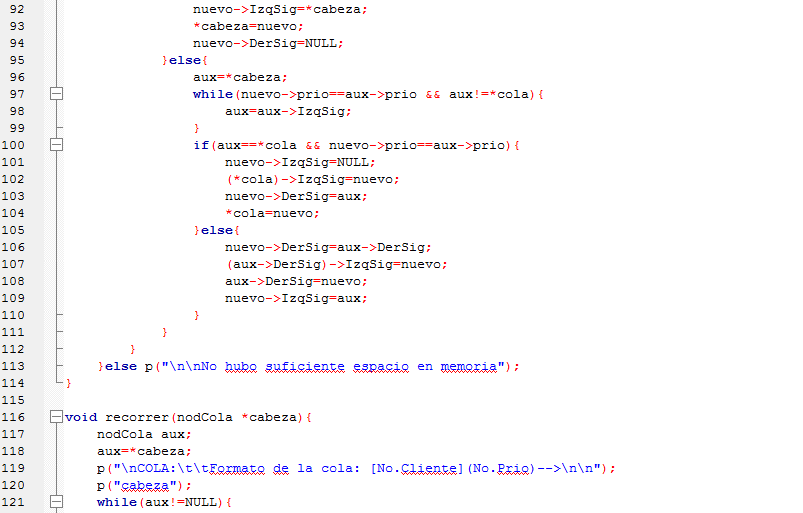
****

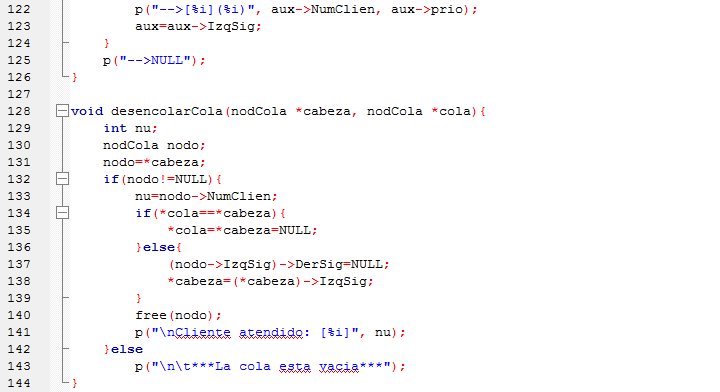
**Ejercicio 2:** Simular la forma de atención al cliente donde existen dos filas, una normal y otra con prioridad.



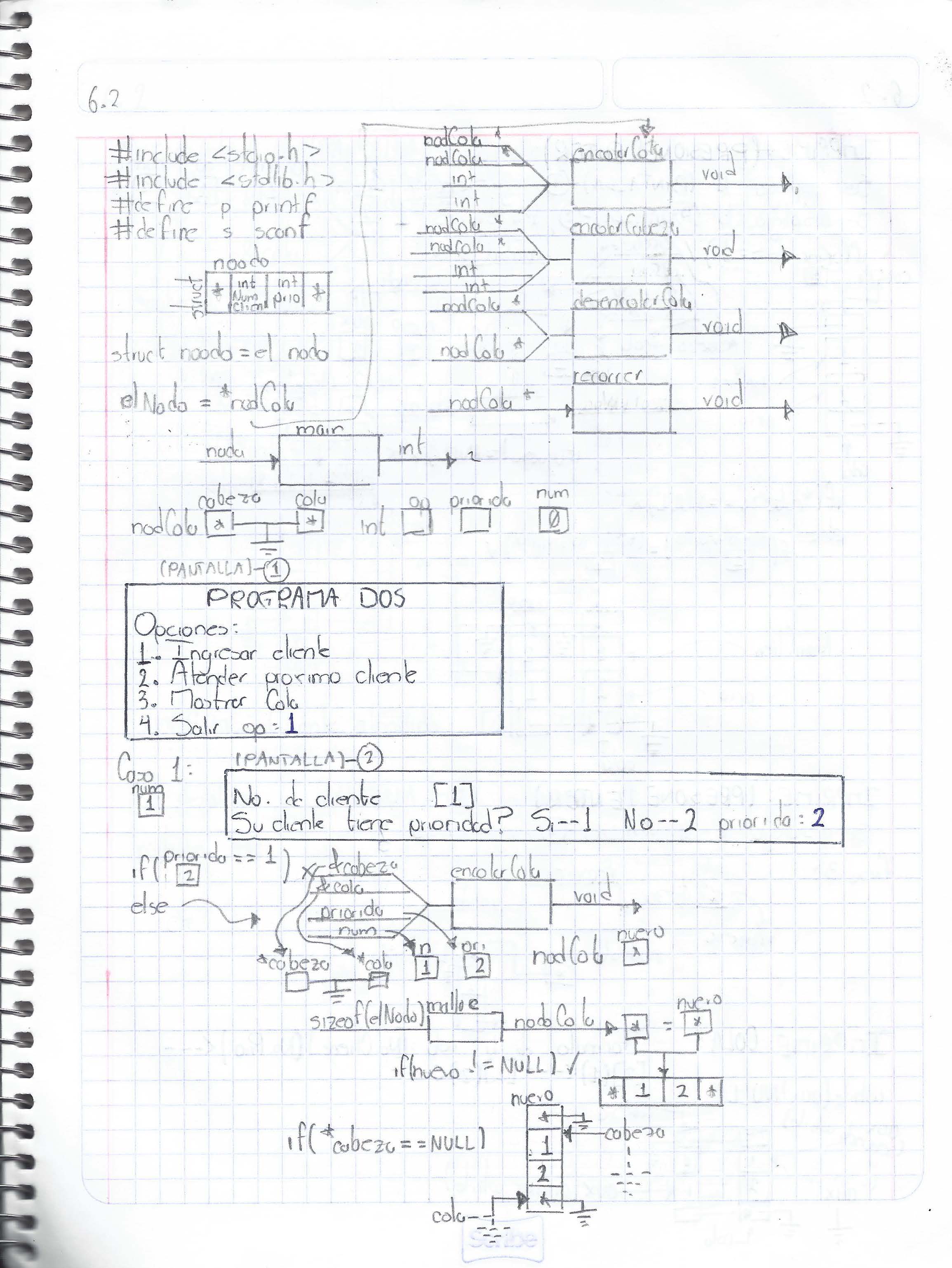


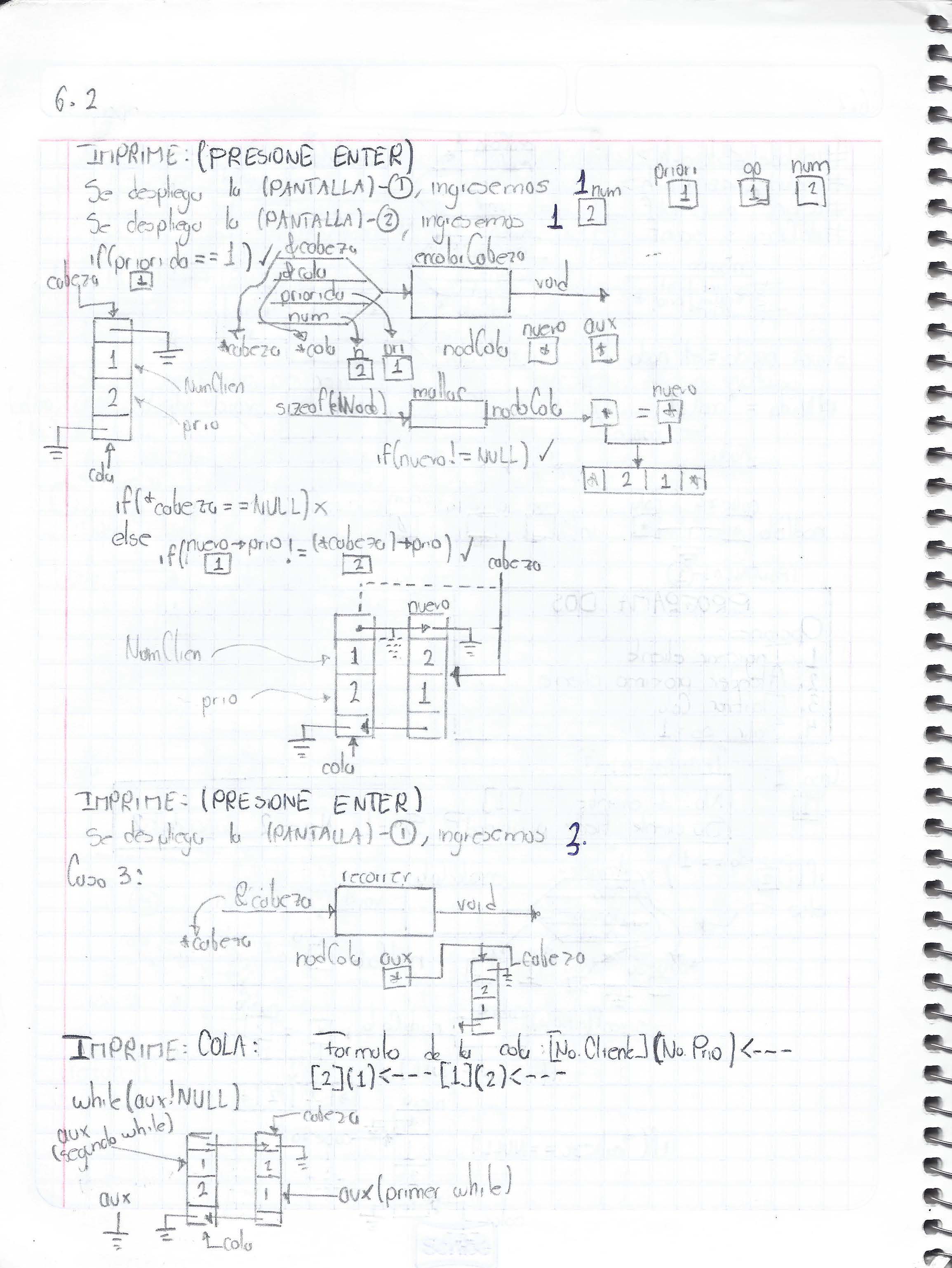


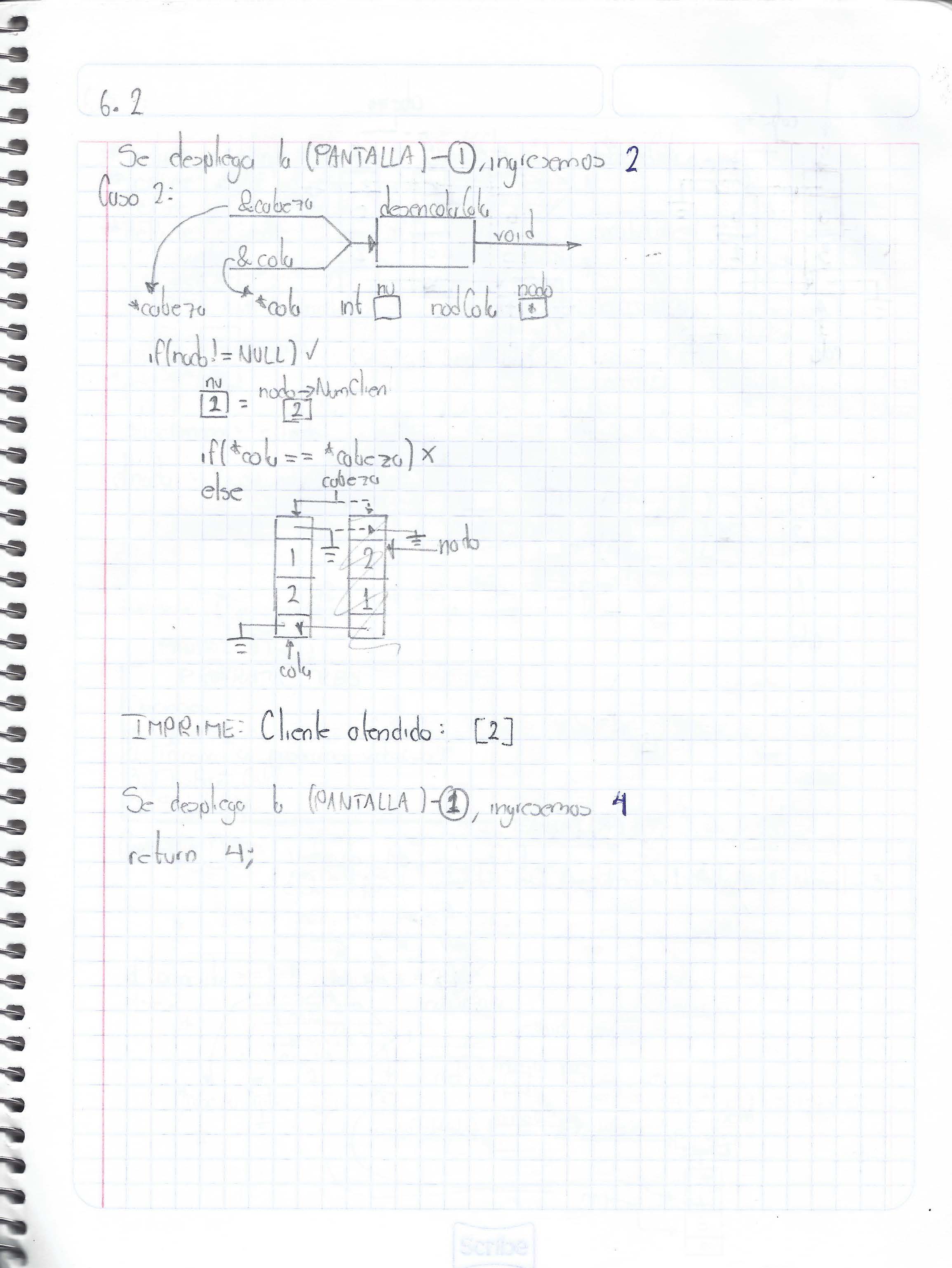




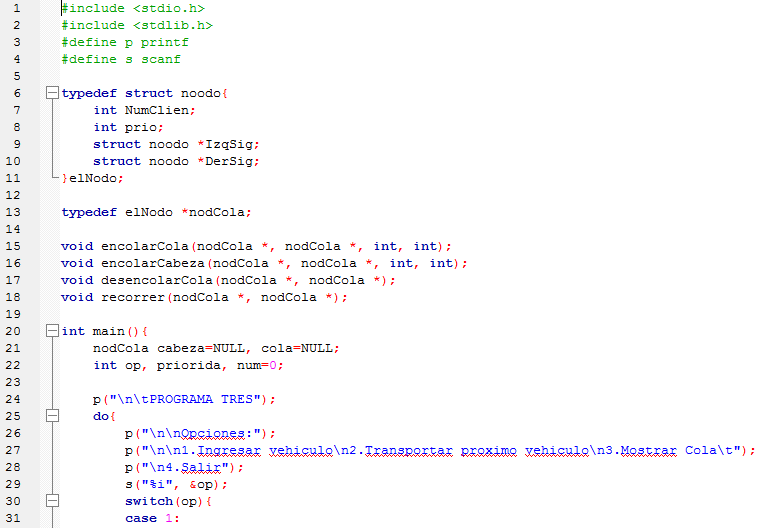
**Prueba de escritorio:**

****

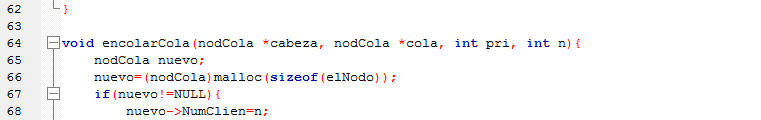
****

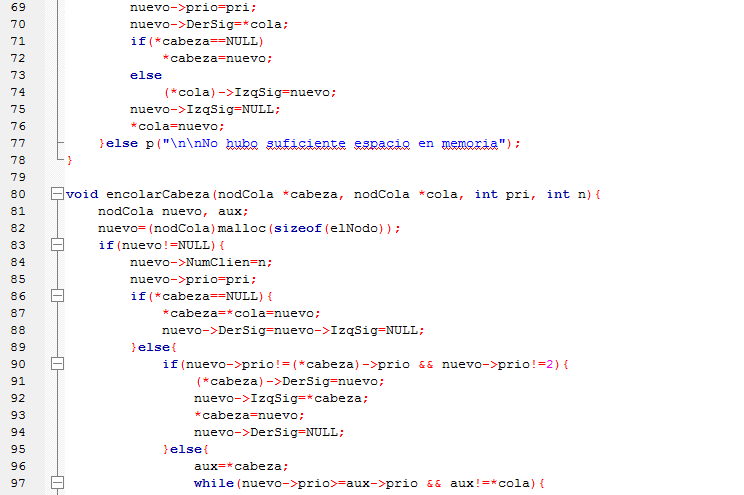
****

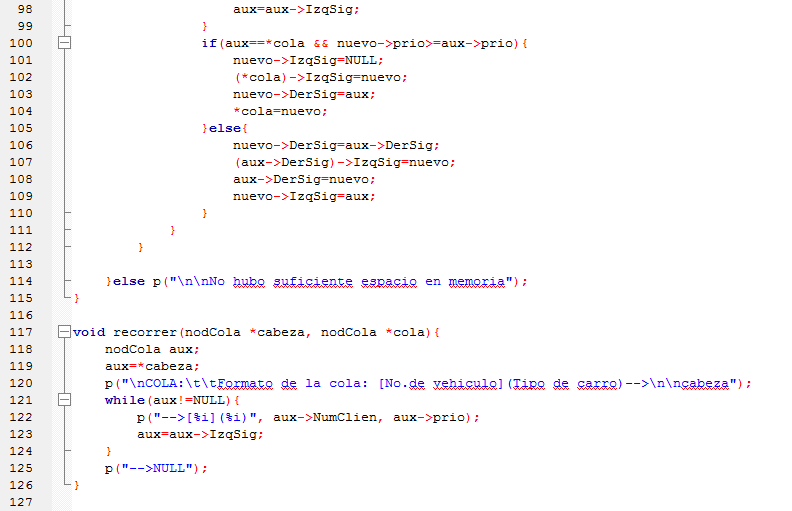
**Ejercicio 3:** Se desea realizar un algoritmo que simule un transbordador de vehículos, para que esos sean transportados entre las dos orillas de un rio.

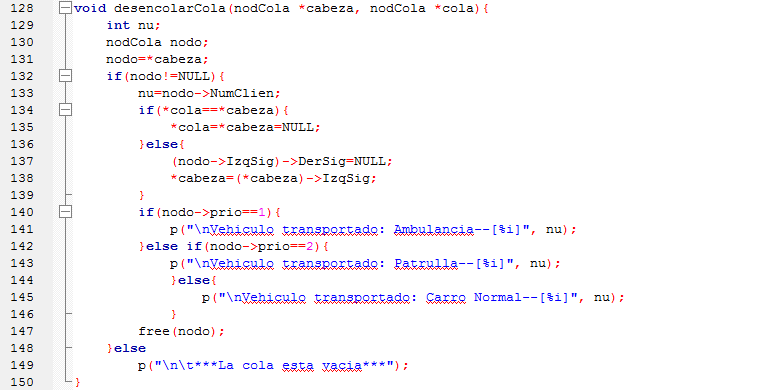




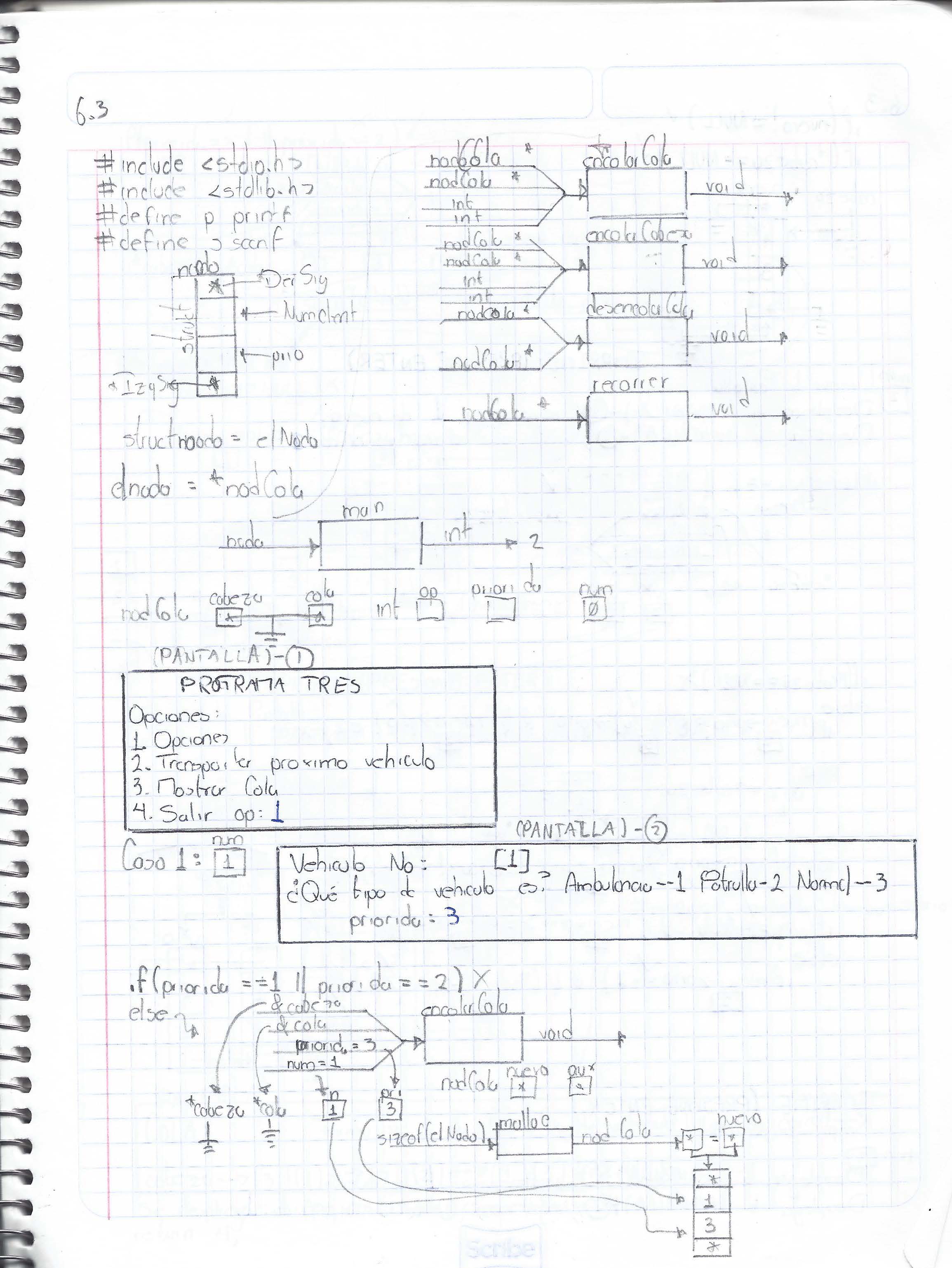


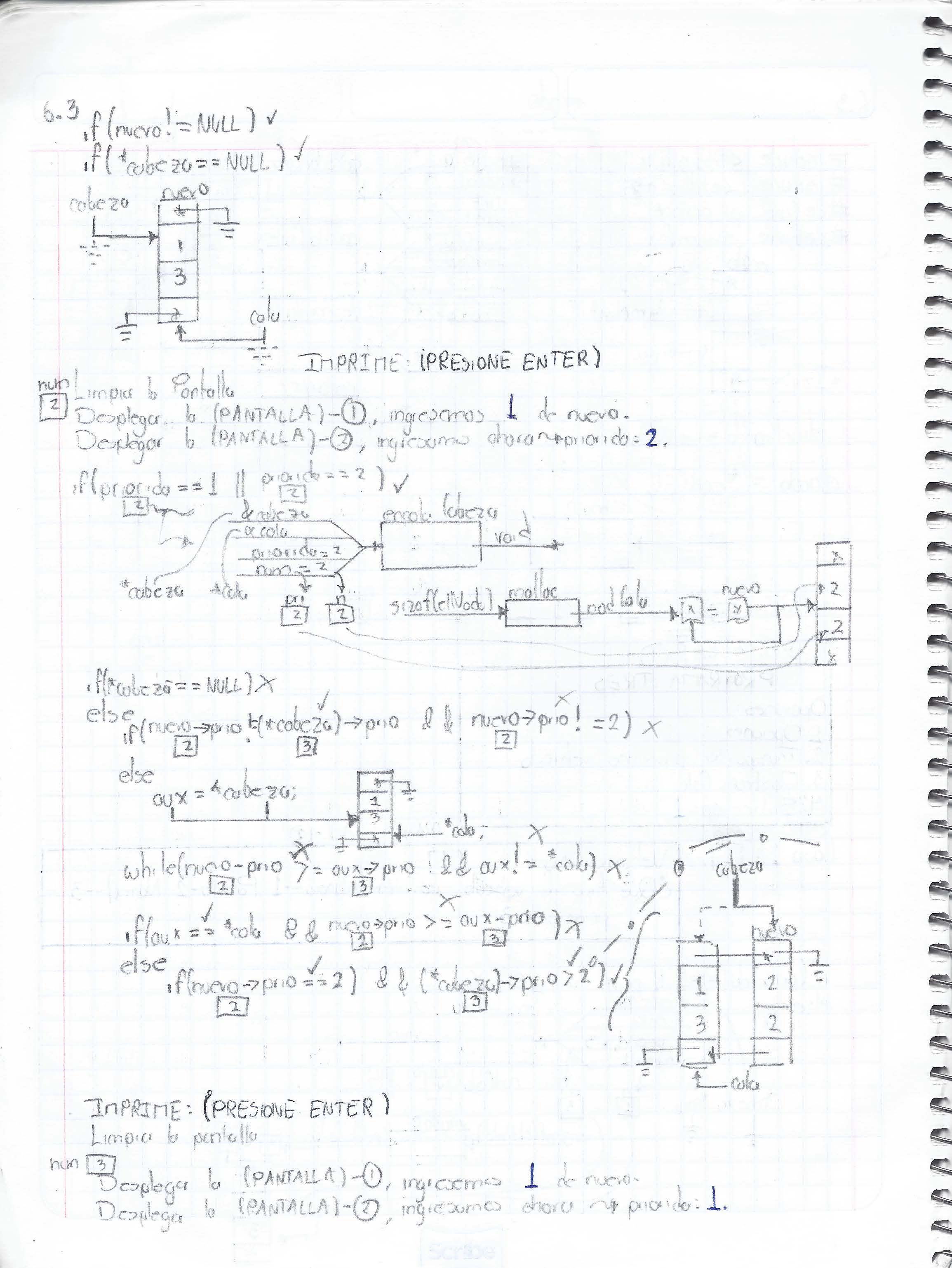


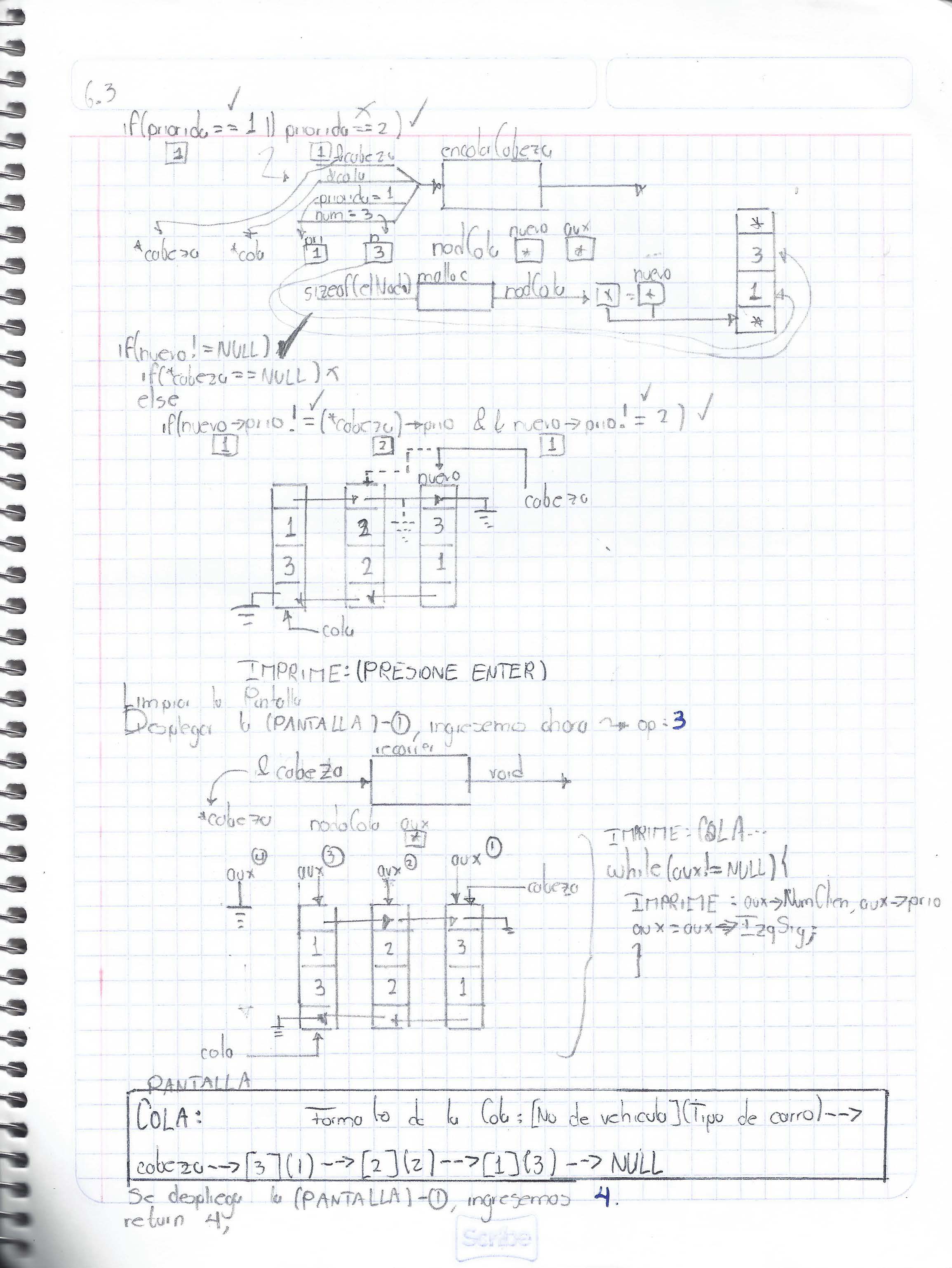




**Prueba de escritorio:**

****

****

****

Conclusión:

Vaya que esta práctica fue complicada. Pero finalmente logre hacer un código que cumpliera con las exigencias del ejercicio.

Realmente estoy muy satisfecho con mi desempeño, pues como comenté en el método, prácticamente no consulte ningún libro para llevar a cabo esta práctica, a partir de la idea general de cómo se realizan las estructuras dinámicas, yo contando con una hoja en blanco y un lápiz, me dispuse a fabricar mi propio código que hiciera lo que quisiese. Por lo mismo, tengo que aceptar que, si examinamos mi código, tal vez puede espantar un poco, pues confieso que me fue muy difícil optimizarlo, porque lo más seguro es que en algún buen libro, ya te presentan la estructura del código de forma optimizada. Sin embargo, como ya comenté, al realizarlo desde cero, no fue tan fácil para mi encontrar las cosas posibles de optimizar.

Ya ahora mi visión es más profunda hacia cómo funcionan los programas computacionales y cuál es el algoritmo que usan y realmente puedo ver grandes aplicaciones a estas estructuras de datos dinámicos, ahora con estas nuevas mejoras acerca de volverlas circulares y doblemente ligadas.

Bibliografía:

[1] Fco. Javier Ceballos, C/C++ CURSO DE PROGRAMACIÓN, Ra-Ma, 3° edición, 2007.

[2] Harvery M. Deitel, CÓMO PROGRAMAR EN C/C++ Y JAVA, México: PEARSON EDUCACIÓN, 2004.

[3] L. J. Aguilar, FUNDAMENTOS DE PROGRAMACION: Libro de problemas, España: McGRAW-HILL, 1996.

[4] E. M. O. Regino, LÓGICA DE PROGRAMACIÓN, Bogotá: Ecoe Ediciones, 2004.