**PRÁCTICA 27 | APLICACIÓN ÁRBOLES**

# Introducción

Estamos llegando al final del tema de árboles y por ende agregando las operaciones que faltaban por analizar y programar. Recordaba que para el proceso de eliminación que vi el semestre pasado en la materia de Matemáticas Discretas había varios casos y en uno de los casos varias opciones. Dichos casos y explicaciones son los mismos que se mencionaron en el video que fue compartido en la plataforma. Me imaginaba que el código para realizar una eliminación en un árbol sería extenso, y de primera vista se ve así. Espero, además del código para realizar dicho proceso, aprender algo más a base del análisis del pseudocódigo.

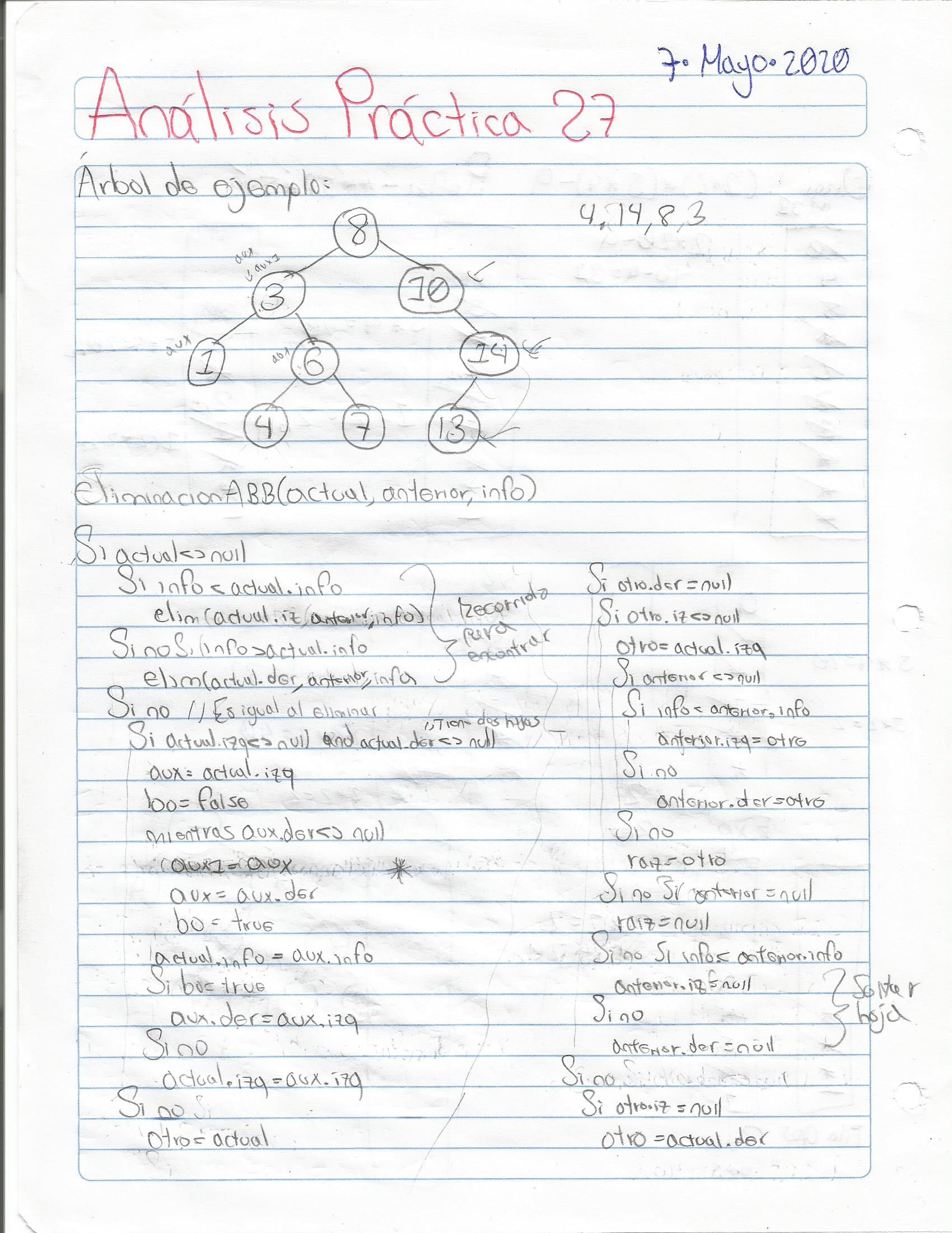
# Desarrollo

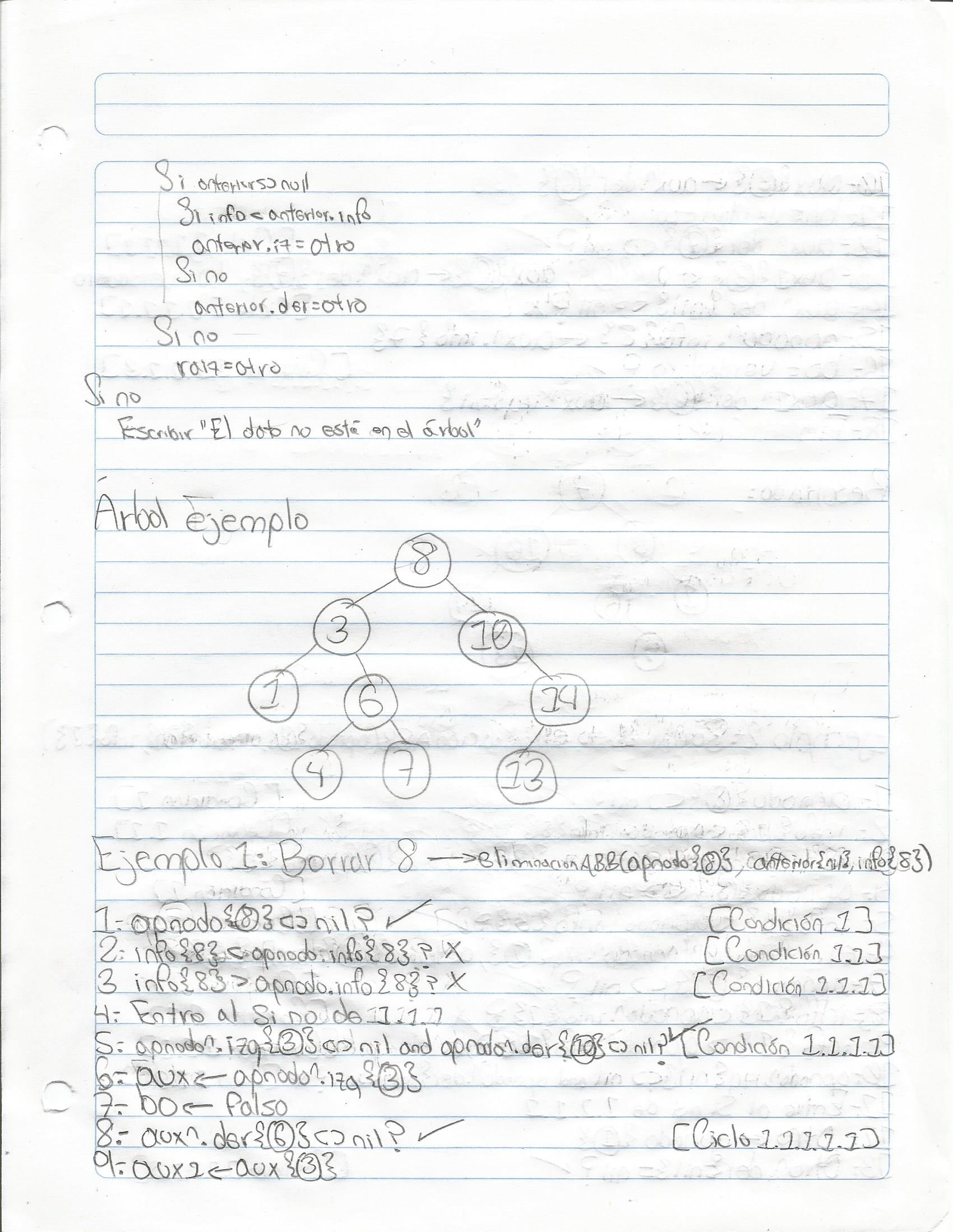
Lo primero fue ver los vídeos que estaban en la plataforma y agregar el código que se manejaba ahí: se añadió una nueva clase de nombre **ArbolBinarioBusqueda,** así como los métodos para insertar y buscar un nodo contenido en éste.

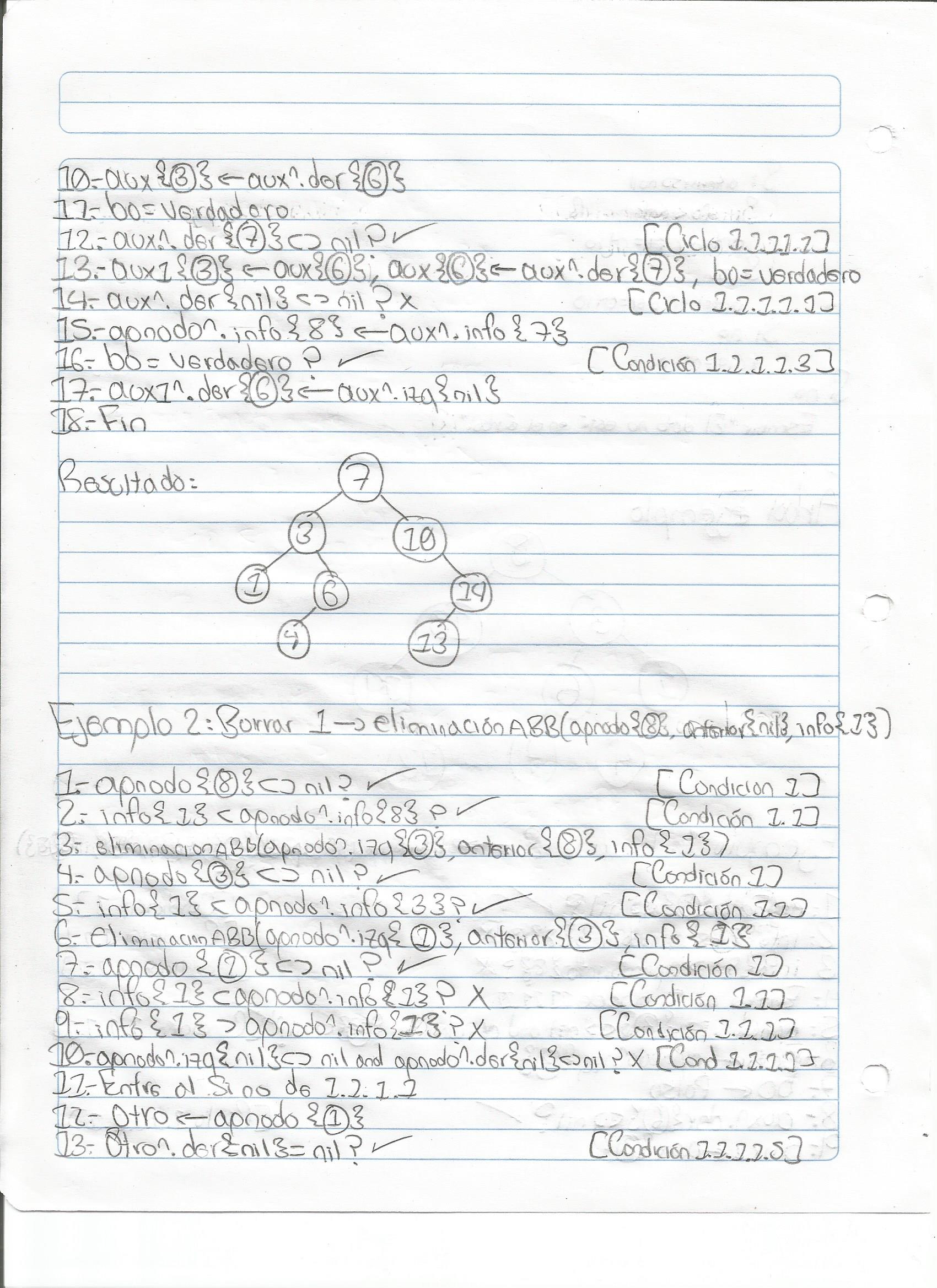
Al terminar lo anterior, estaba el planteamiento de agregar también la operación faltante entre los árboles binarios de búsqueda: el método para eliminar. Por tanto, en el vídeo compartido en la plataforma se explicó un poco sobre los casos que existen al momento de realizar dicho proceso. Finalmente, venía un pseudocódigo con los pasos para este procedimiento, que al último tenía que ser traducido a Java para incluirlo como en la misma clase anteriormente mencionada y así completar la operación faltante.

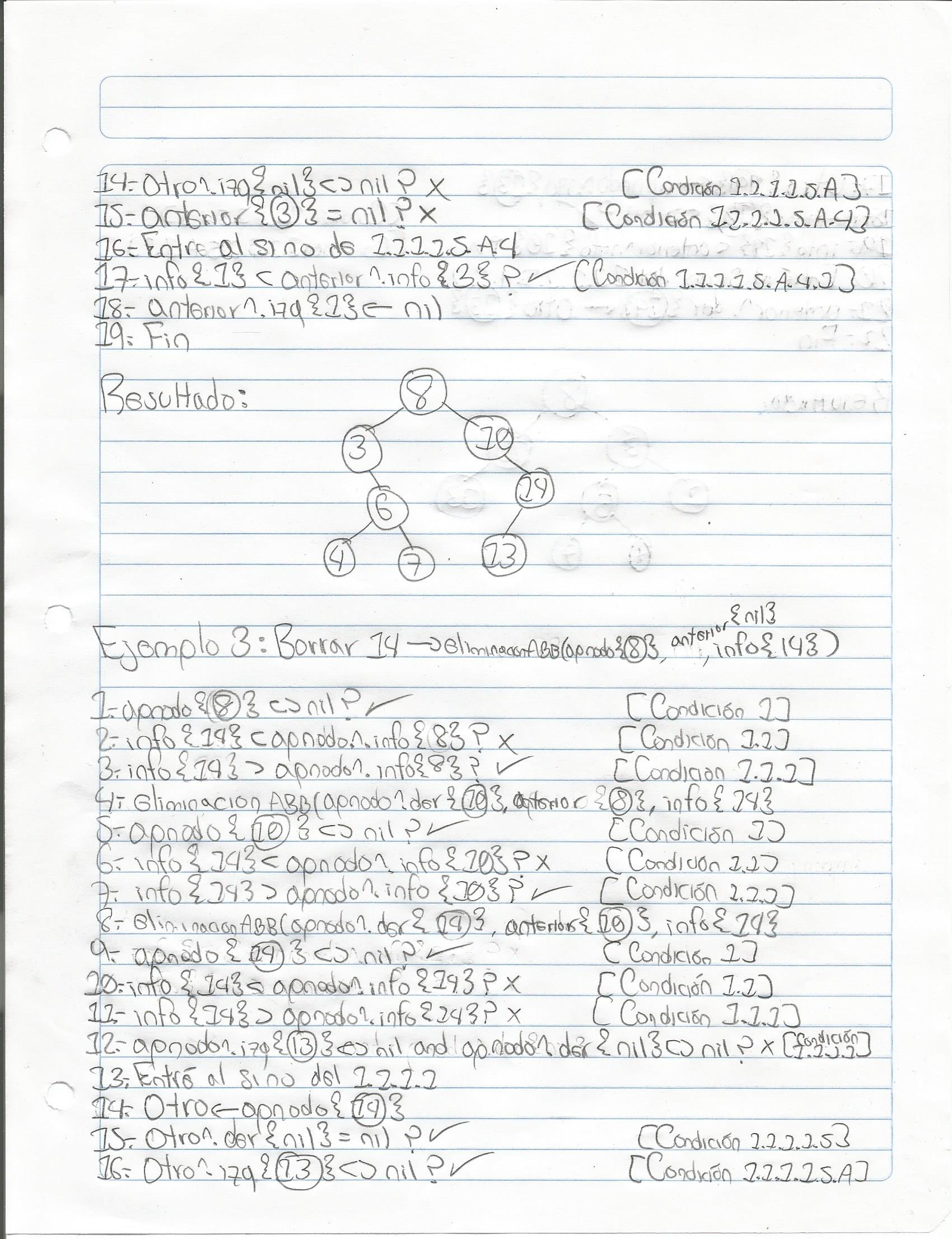
Traducir el pseudocódigo no era todo lo que se tenía que hacer. También era necesario analizar éste mismo en papel con cada uno de los tres casos para su respectiva comprensión. Lo primero que hice fue transcribir el pseudocódigo a mi cuaderno para tener una mejor visibilidad del mismo. Después, proseguí a indicar, a manera de pasos, el camino que tomaría la eliminación de un nodo en cada uno de los tres casos planteados. Use como referencia el mismo árbol que se puso de ejemplo en las diapositivas.

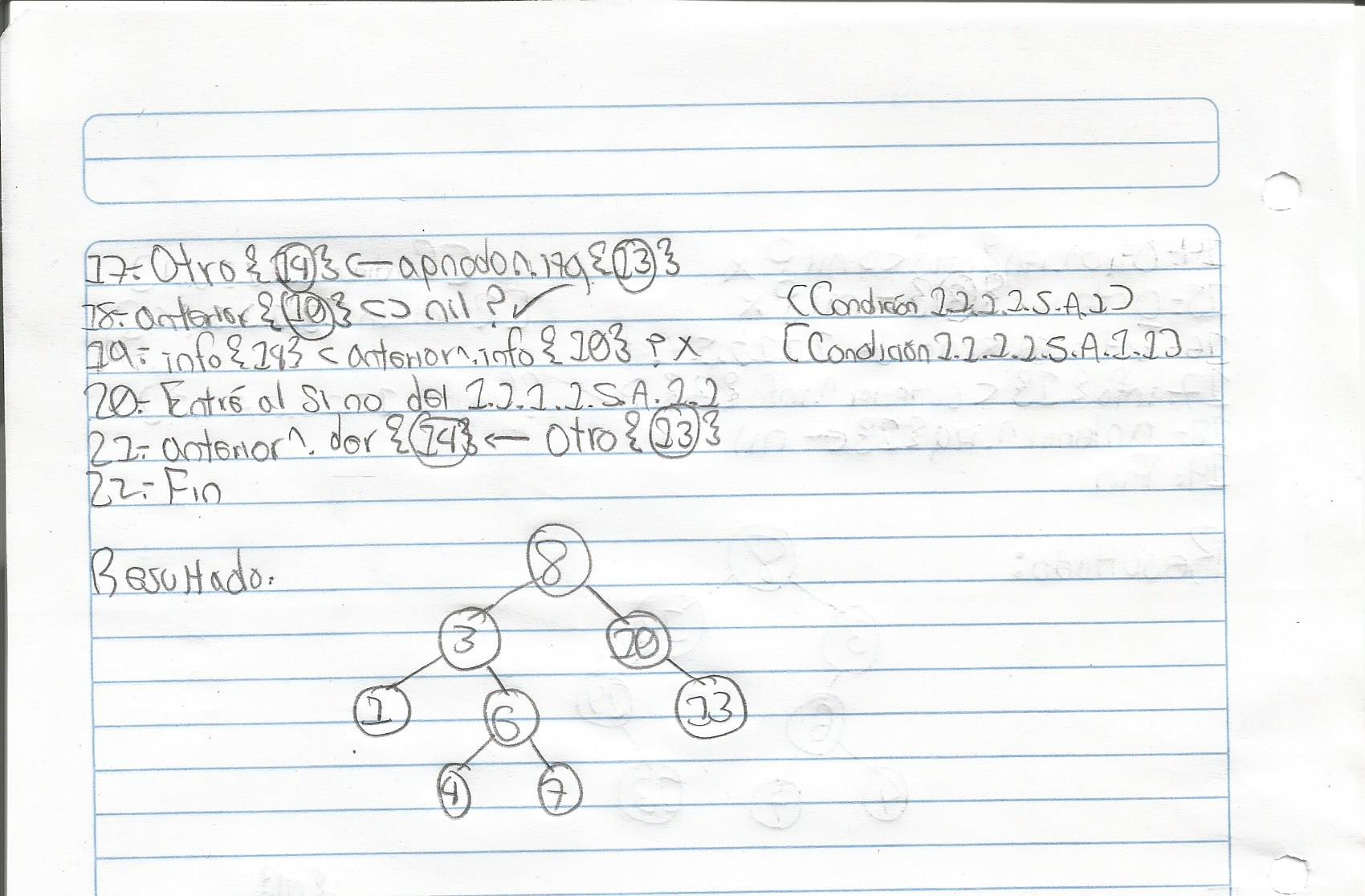
A continuación, anexo las fotografías del **análisis realizado**.











Terminando el análisis anteriormente ilustrado, proseguí a traducir el pseudocódigo al proyecto.

A continuación, se indican los archivos creados:

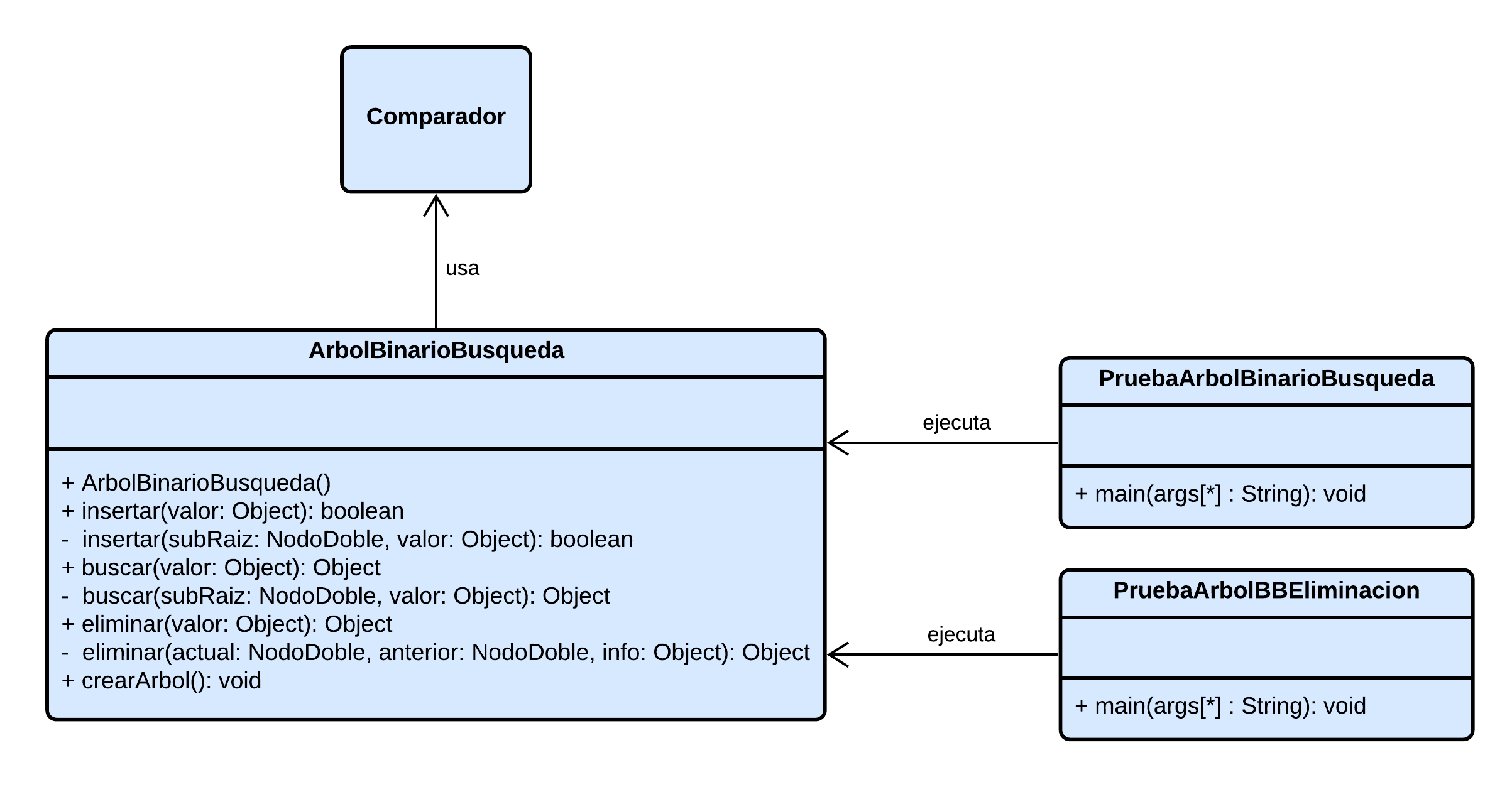
* ***ArbolBinarioBusqueda.java (paquete ednolineal).***
* ***PruebaArbolBinarioBusqueda.java (paquete interfaces).***
* ***PruebaArbolBBEliminacion.java (paquete interfaces).***

La penúltima clase contiene las pruebas que se indicaron en el vídeo de la plataforma.

En cuanto a la última, contiene las pruebas para la eliminación de nodos de un árbol tomando en cuenta los tres casos.

# Diagrama de clases

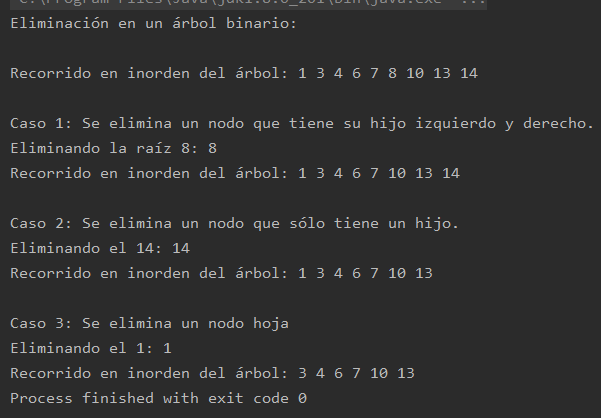
En el diagrama de clases se tiene una vista más cómoda de lo que se hizo en cada una de las clases mencionadas anteriormente. Cabe destacar que para las que ya existían en el proyecto solamente se ilustran los métodos o atributos que fueron agregados. En cambio, para las nuevas se muestra en su totalidad métodos y atributos que fueron incluidos.



**Si desea revisar la documentación de lo que fue agregado al proyecto vaya a las siguientes rutas:**

* **/docs/ednolineal/ArbolBinarioBusqueda.html**
* **/docs/interfaces/PruebaArbolBinarioBusqueda.html**
* **/docs/interfaces/PruebaArbolBBEliminacion.html**

# Captura del programa funcionando



# Conclusiones

Aunque esta práctica en cuanto código fue en su mayoría transcribir, el análisis sí me tomó bastante tiempo, porque por una parte me puse a pensar en cómo hacerlo, y por otra, cuando ya tenía la idea, me perdía a veces entre las condiciones, además de que cada procedimiento quedó con más de diez pasos como se pudo ver en las imágenes. A pesar de todo, ya tengo una clase que me imagino tiene muchas aplicaciones.