Daniel Walteros, Nicolás Aguilera

POOB-2

**SHAPES Conociendo el proyecto shapes**

2. Ofrece 4 clases : Triangle , Circle , Rectangle , Canvas

Entre ellas existe una relación de composición y de jerarquía

3. El paquete shapes posee 4 clases : canvas , triangle , circle y rectangle

En la clase circle : No se encuentra ningún atributo en la documentación

Metodos para que cambie la figura : public void **changeColor**([String](http://download.oracle.com/javase/6/docs/api/java/lang/String.html?is-external=true) newColor)

y public void **changeSize**(int newDiameter)

4. Tiene 5 atributos:

Diameter , xPosition , YPosition , color , isVisible

Todos son privados

Hay en total 14 metodos , dos de ellos son privados : erase() y draw(). Todos los metodos que hacen un cambio de posicion , tamaño o color , o que hacen visible o invisibale a la figura hacen uso de los metodos privados.

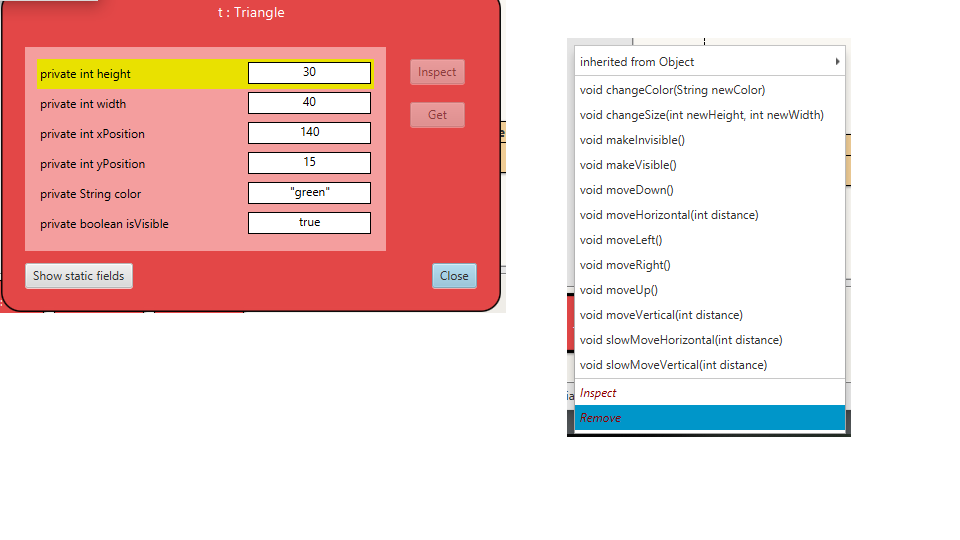
5.Porque en la documentación no se suelen poner metodos ni atributos privados.

6. Indica un valor aproximado del numero π. Además nos indica que es un valor constante

7. Entender el uso basico de bluej asi como el diseño basico de una clase y el manejo de metodos dentro de la programacion OO.

**Manipulando objetos. Usando opciones.**

1.Existen 4 clases : triangle , circle , rectangle , canvas

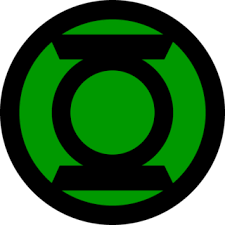
2 y 3. Valores de inicio de todos los atributos del objeto Triangle , además se muestra el comportamiento de dicho objeto

4. Hay 4 clases , creamos 4 objetos.

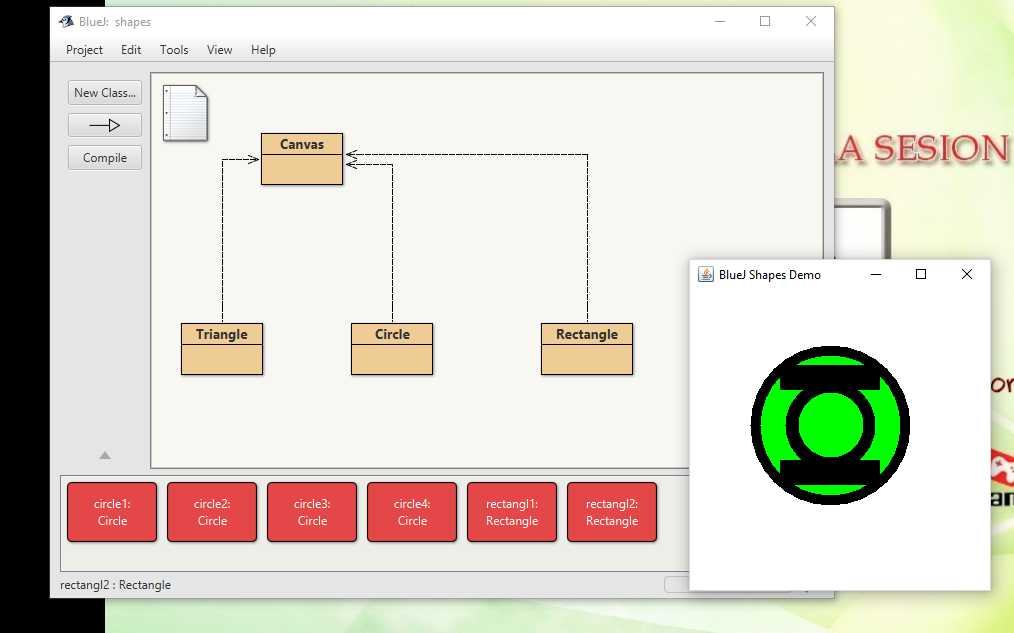
No aparecen los métodos draw (), erase () porque son privados y tampoco aparece el de la inicialización

Se hace uso de dos clases. Se utilizaron 4 círculos y 2 rectángulos

**Logo oficial :**



**Hecho en blueJ:**

****

**Manipulando objetos. Analizando y escribiendo código**

1.



2. Primer paso : Existen tres variables y 0 objetos

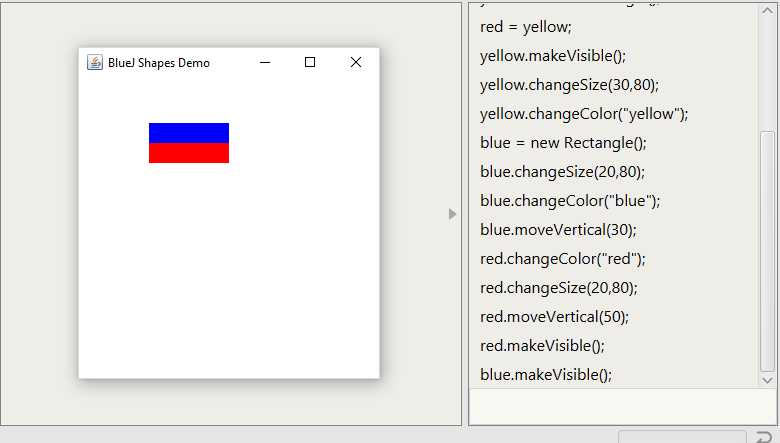
Segundo paso: Existen dos objetos , 3 variables y los dos objetos son de color magenta , que es el color predefinido y el primer objeto se puede ver

Tercer paso : Existen dos objetos , 3 variables , un objeto es de color magenta y otro de color amarillo y este último es visible

Cuarto paso : Hay 3 objetos y 3 variables. El objeto de color azul es invisible, el de color amarillo es visible y el tercer objeto se quedó sin asignación.

Quinto paso : Hay 3 objetos y 3 variables . El objeto color azul es invisible y el objeto coloir amarillo pasó a ser de color rojo y sigue siendo visible

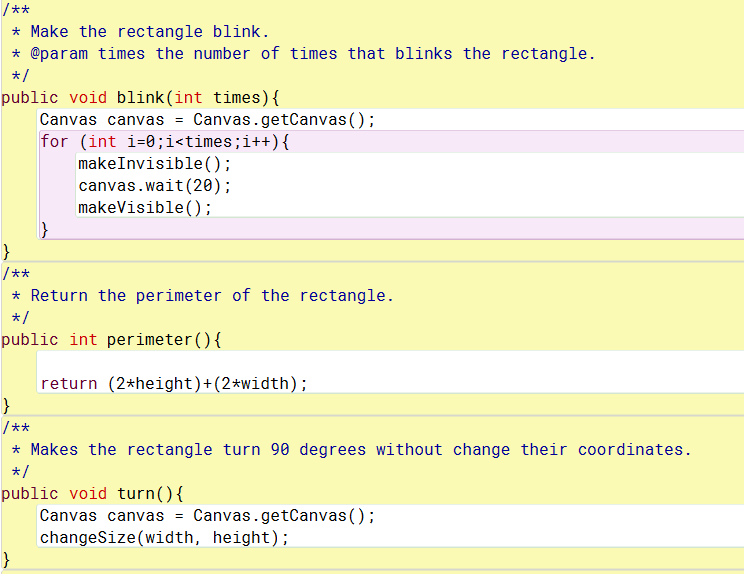
Sexto paso : Hay 3 objetos y 3 variables y el objeto de color azul se vuelve visible.



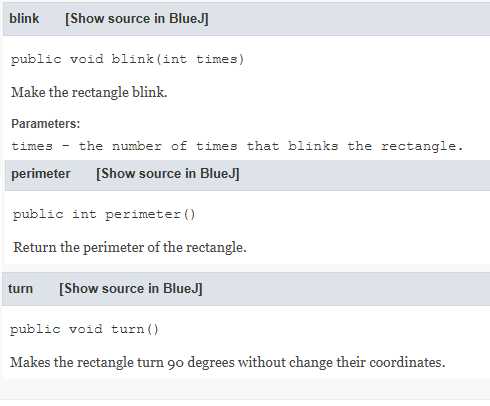
3.No, porque al asignarle a la variable red el objeto de la variable yellow las dos variables hacen referencia al mismo objeto; causando que el color amarillo de la imagen sea el color rojo.

**Manipulando objetos. Usando opciones.**

1,2,3. Codigos de los metodos :



4. Documentación



**Implementando una nueva clase. Ceiling.**

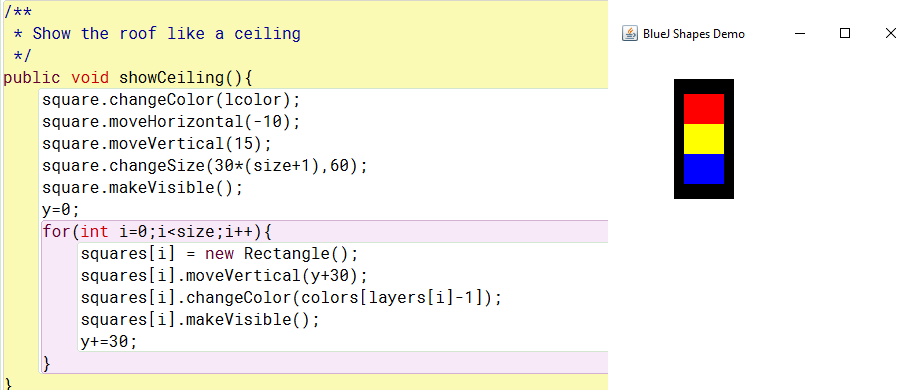
1. **Metodos constructores :** \_(layers: int[], color: String)

**Metodos analizadores:** showCeiling() , showTree()

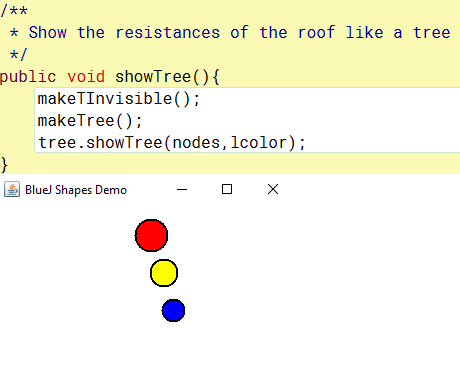
**Métodos modificadores:** add (layer), move (int x, int y), delete ()

2.

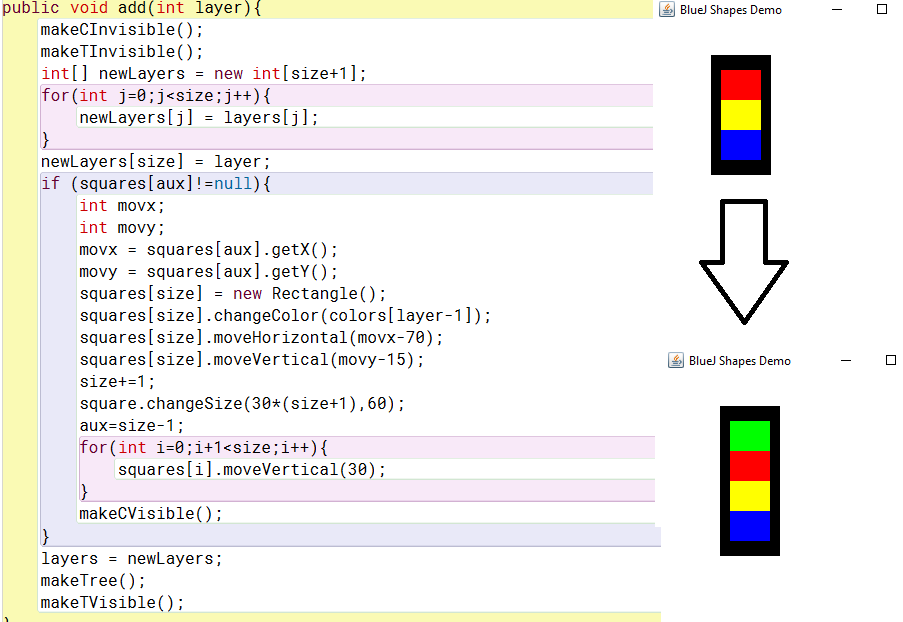
**showCeiling()**



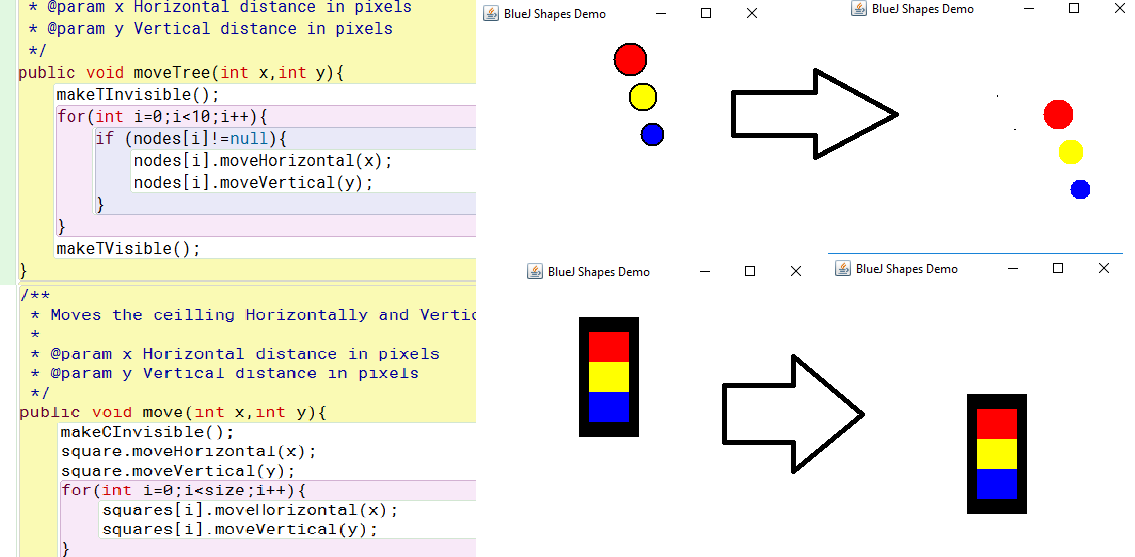
**showTree()**



**add(int layer)**

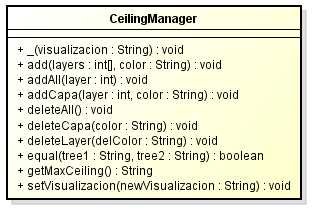


**movetree(intx,inty) , move(intx,inty)**



**Definiendo y creando una nueva clase. CeilingManager**

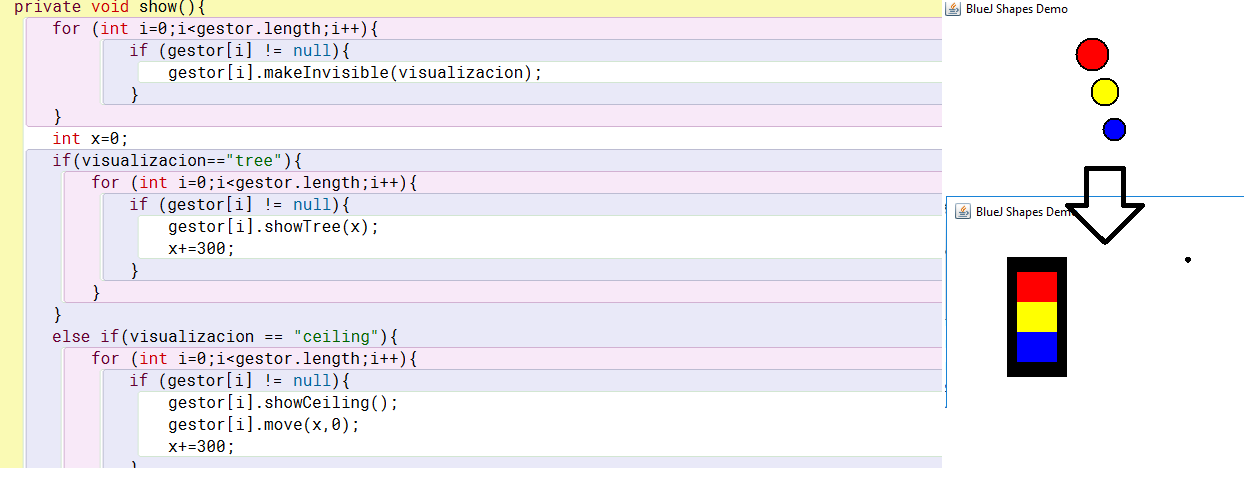
**Diseño**

****

**Imágenes relevantes**

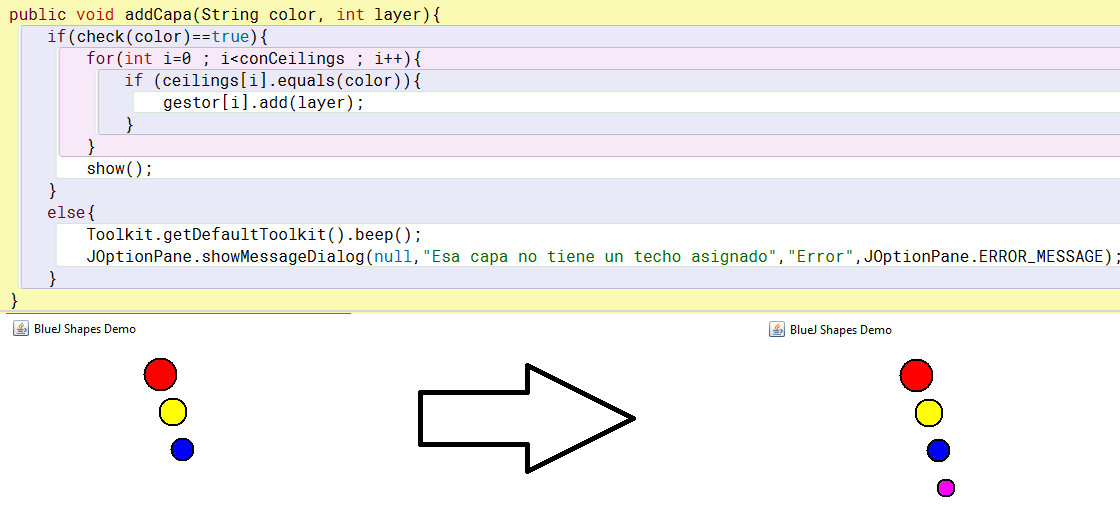
**show()**

Cambiar el modo de visualización del gestor: techo o árbol.

****

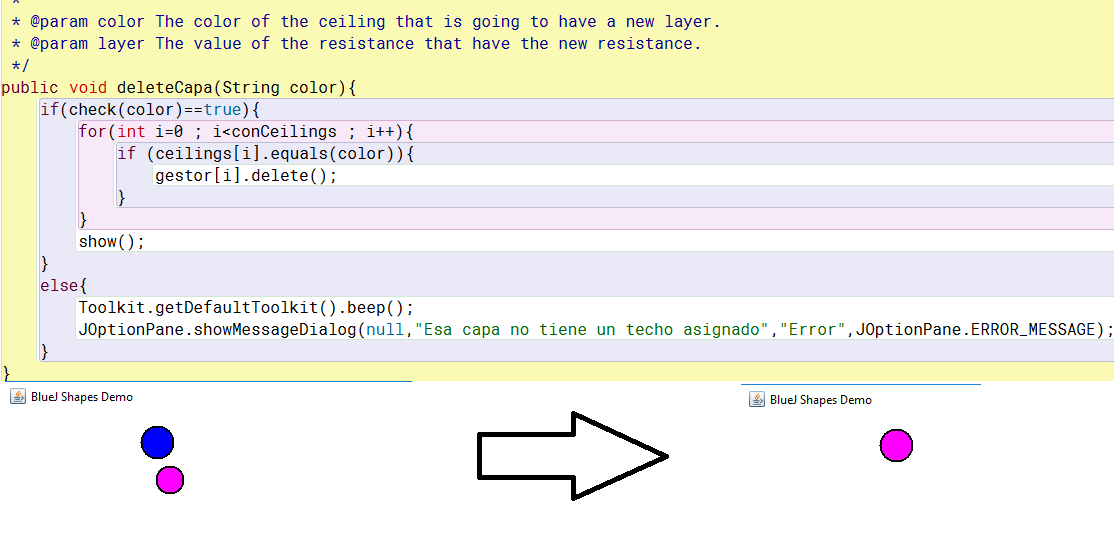
**addCapa()**

Adicionar un nuevo techo dadas sus capas y color. El color debe permitir identificar el techo. No se permiten adicionar techos con las mismas capas

****

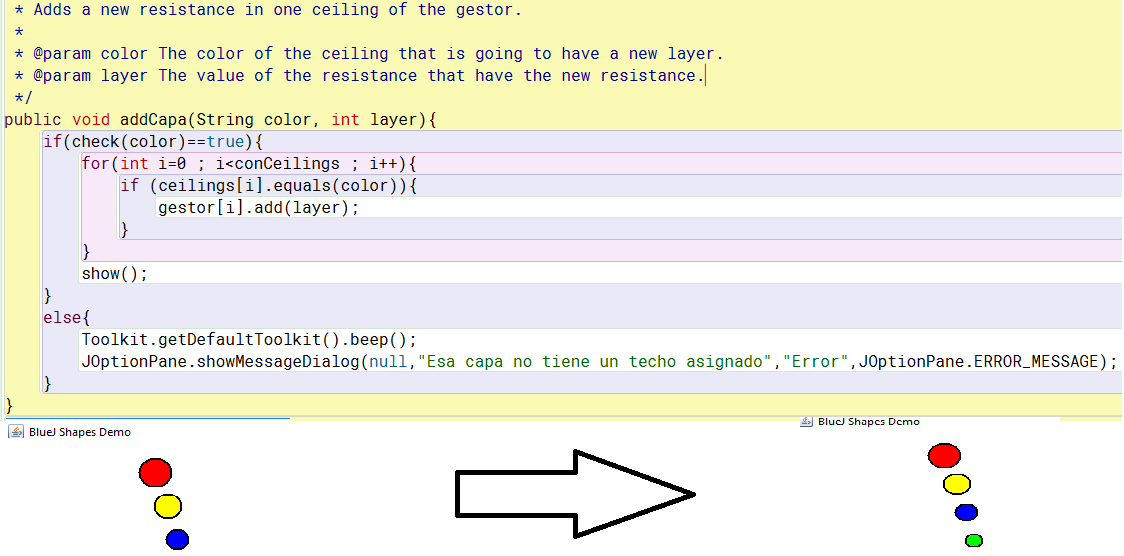
**deleteCapa(String color)**

Eliminar capas a un techo.

****

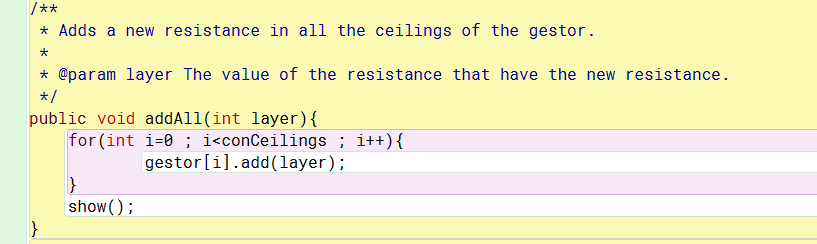
**addCapa(String color, int layer)**

Adicionar capas a un techo

****

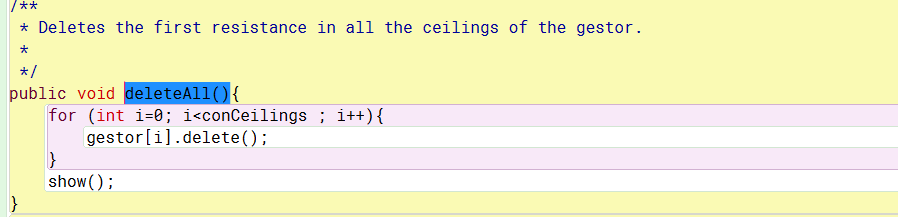
**addAll(int layer)**

Adicionar una capa a todos los techos

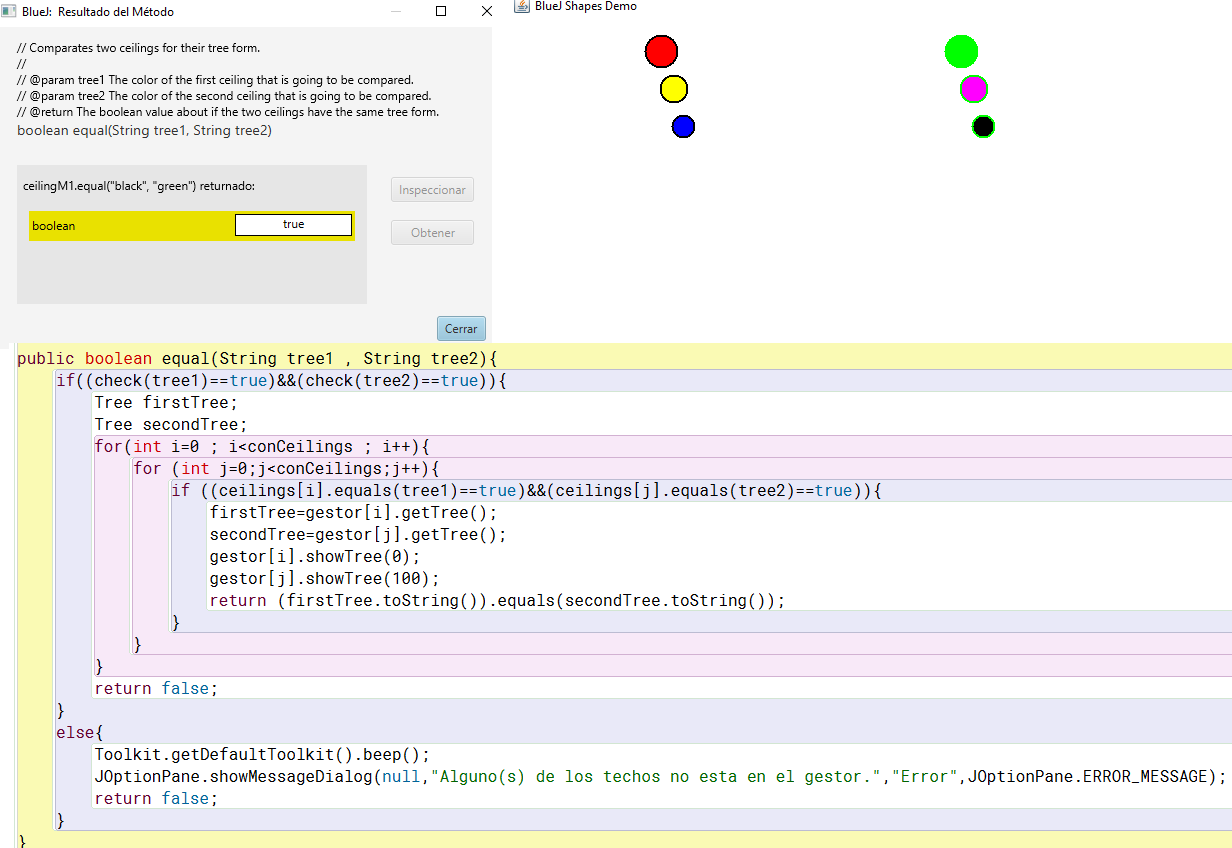


**deleteAll()**

Eliminar una capa de todos los techos



**equal(String tree1 , String tree2)**

****Decidir si dos techos son equivalentes (tienen la misma forma de árbol)

**Retrospectiva**

1. Cada uno trabajó 12 horas

2. Al laboratorio le hace falta algunos detalles del Ceiling Manager y ocurren pequeños errores al momento de añadir o eliminar capas ( se añaden correctamente pero el marco se mueve de posición). Se han obtenido grandes avances gracias al trabajo de los dos integrantes

3. El trabajo a parejas ha sido fundamental para el laboratorio. El complemento de ambos integrantes ha influido en el desarrollo eficaz del trabajo.

4. La realización del árbol porque era la parte más complicada del trabajo y se logró sacar adelante.

5. La elaboración del árbol fue el mayor problema. Recurrimos a conocimientos de materias anteriores para resolverlo

6. Unimos ideas y conceptos de ambos en pro del laboratorio. Además, trabajamos la mayoría del laboratorio de manera unida en un solo computador. Trataremos de mejorar el trabajo a distancia para la próxima vez