

A dark blue vertical bar on the left side of the slide. A blue arrow points to the right from the bar, containing the date.

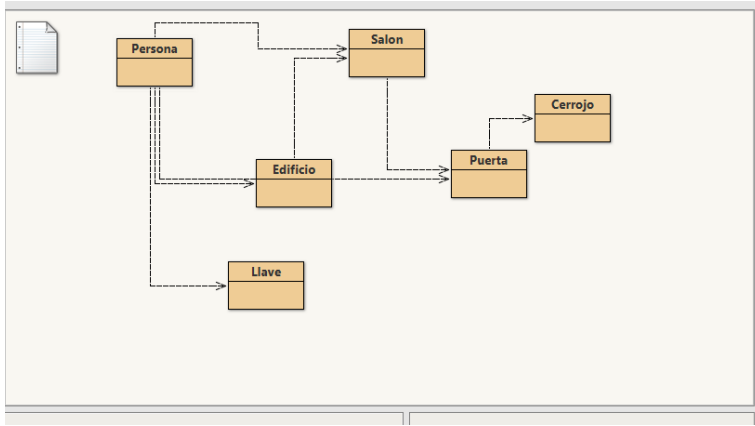
14-3-2018

PRUEBA DE LA CLASE PERSONA- EDIFICIO-SALON- PUERTA-CERROJO- LLAVE

PROGRAMACIÓN ORIENTADA A
OBJETOS (POO)

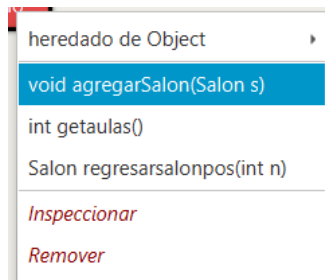
Several thin, curved lines in dark blue and light grey originate from the bottom left corner and curve upwards and to the right.

JOSÉ ÁNGEL GARCÍA GARCÍA

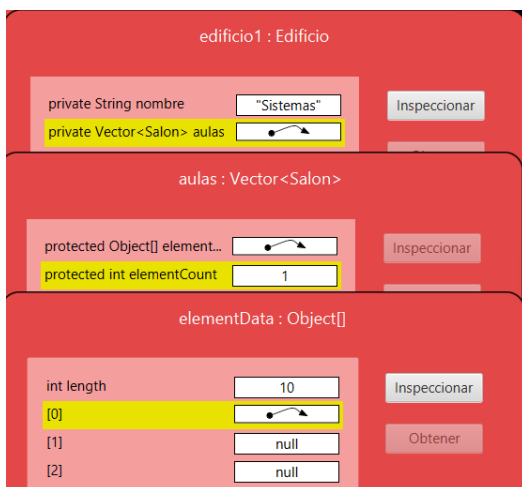


Partimos de que se tiene 6 clases, las que son persona, edificio, salón, puerta, cerrojo y llave. Procedemos a crear los objetos de su respectiva clase, para poder ejecutar los métodos correspondientes del programa.

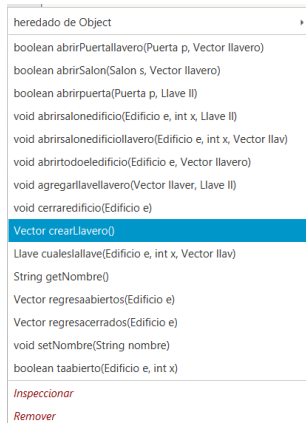
Aquí se muestran los objetos creados, para poder ejecutar los primeros métodos necesitaremos a la persona “Elmer” que es la responsable de ejecutar los dichos métodos. Se crea un edificio con el nombre de “Sistemas”, un salón “A1” que tiene una puerta “p” y esta a su vez el cerrojo de clave “123” y aparte una llave con código “123”.



Procedemos a ejecutar el método para agregar un salón al edificio “Sistemas” **agregarSalon(Salón s)**, que tendrá como función hacer que el salón pertenezca a la clase edificio.



Como podemos observar, el objeto de edificio ya tiene el salón agregado, ahora nos indica que tenemos 1 elemento en el vector de aulas. Ejecutándose de la manera correcta el método `agregarsalon(Salón s)`.



Ahora ejecutaremos el método **crearLlavero()**, con el cual crearemos nuestro llavero que estará compuesto por las llaves de las puertas, además es una herramienta necesaria para ejecutar métodos posteriores.

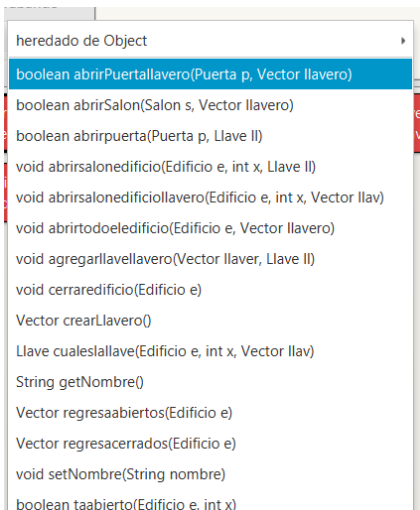
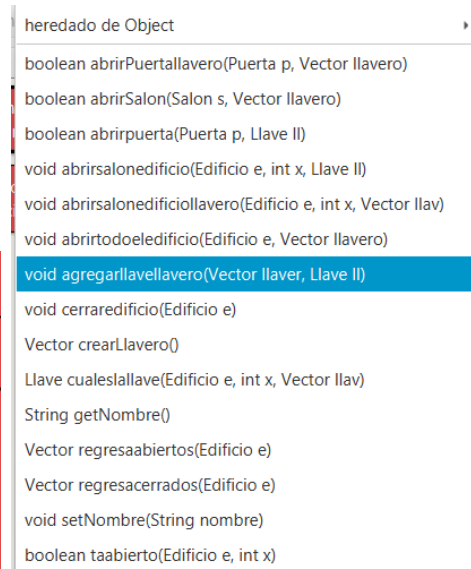
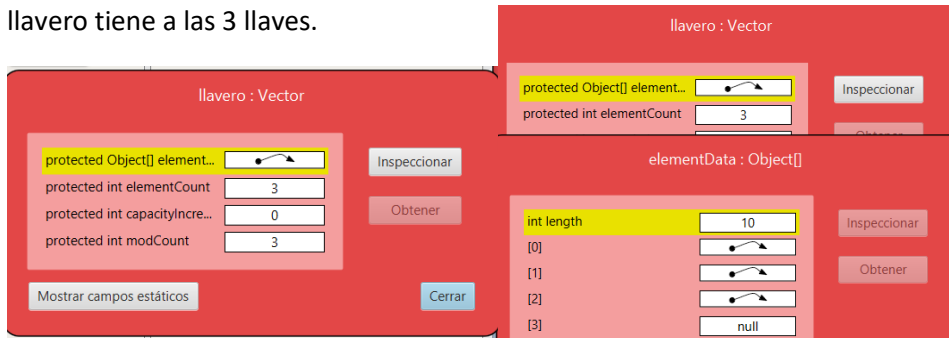
Podemos ver que el objeto llavero se ha creado de la forma correcta.



Ahora creamos otras llaves, que serán necesarias para métodos posteriores.

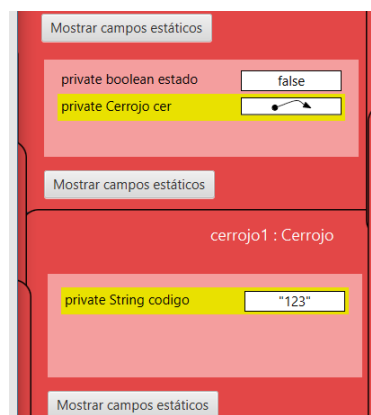
Procedemos a ejecutar el método **agregarllaverllavero(Vector llaver, Llave ll)**, que es el método que utilizaremos para agregar las llaves creadas anteriormente a nuestro llavero.

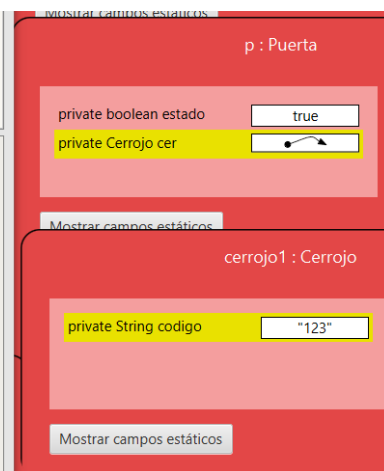
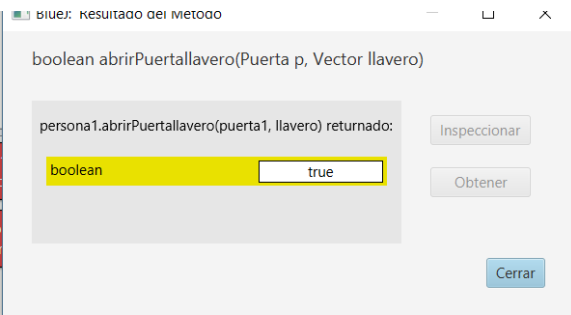
Revisando en la imagen de abajo, podemos comprobar que el método funciona correctamente, agregando las llaves al llavero, ahora nuestro vector llavero tiene a las 3 llaves.



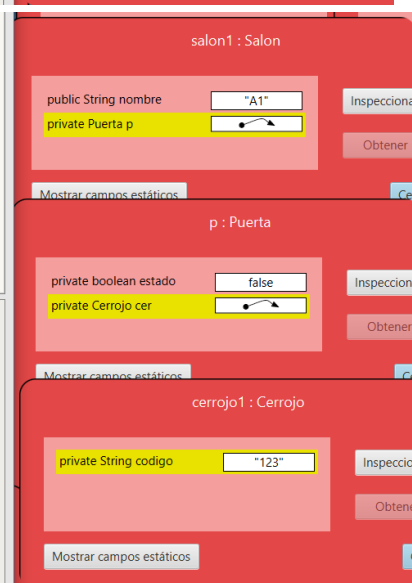
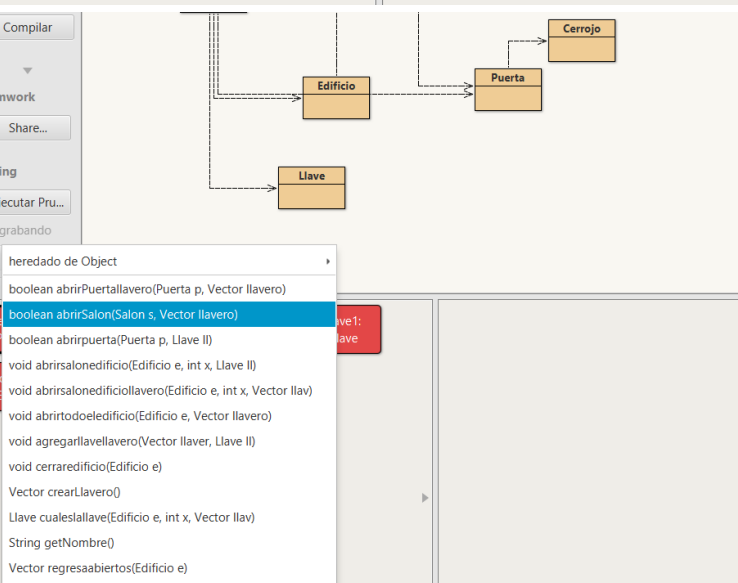
El siguiente método a ejecutar es el de **abrirPuertallavero(Puerta p, Vector llavero)**, con el cual podremos abrir una puerta utilizando un llavero.

Vamos a abrir esta puerta, que tiene como clave "123"

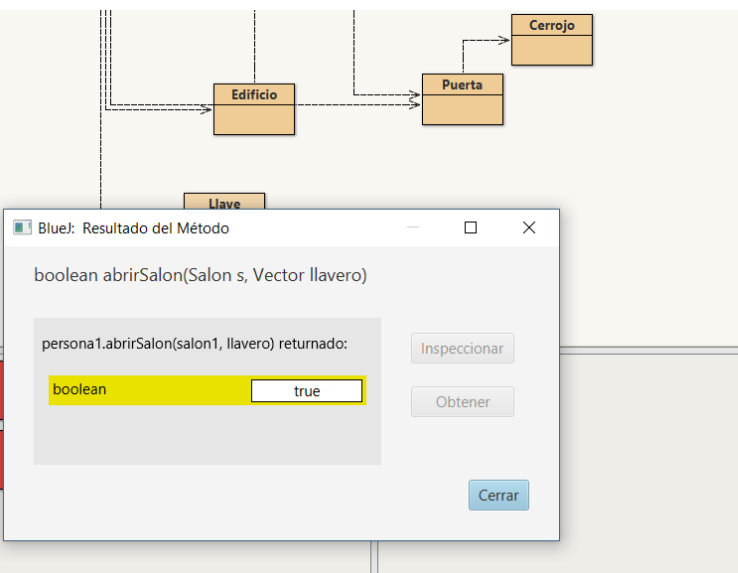




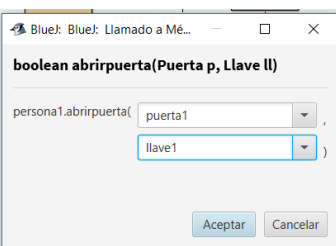
Observando la imagen podemos comprobar que el método se ejecuto de la forma correcta, hizo lo esperado, abrir la puerta utilizando un llavero, en el cual se encontraba la llave con la clave “123”.



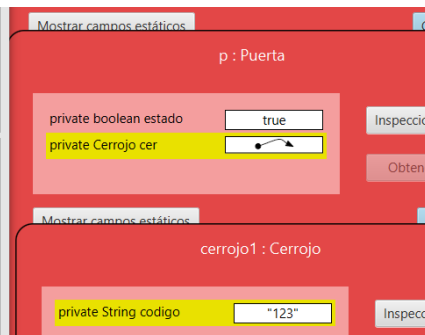
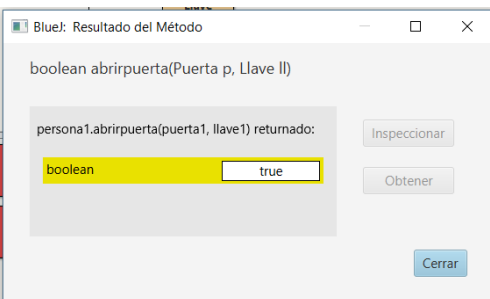
Ahora procedemo a ejecutar el método **abrirSalon(Salón s, Vector llavero)**, este método requiere del salón y del llavero. Como podemos ver nuestro salón “A1”, tiene una puerta que esta cerrada y la clave de esta es “123”.



Como podemos ver en la imagen, el método se ejecutó de la forma adecuada, se abrió la puerta del salón “A1”.



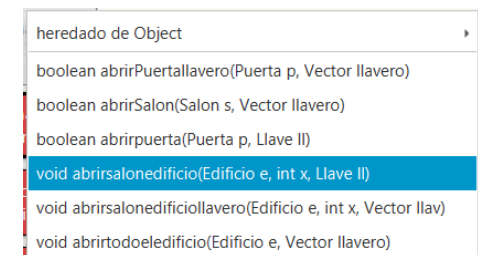
Ahora ejecutaremos el método **abrirpuerta(Puerta p, Llave ll)**, que es muy similar al método de abrir puerta con llavero, pero aquí la diferencia es que la abrimos con su llave correspondiente, no se hace uso del llavero, solamente de la llave.



Como se ve en la imagen, el método funciona correctamente, haciendo que la puerta se abra.

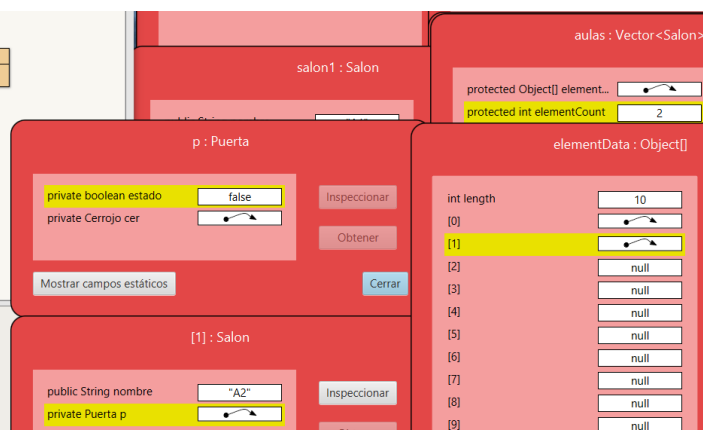
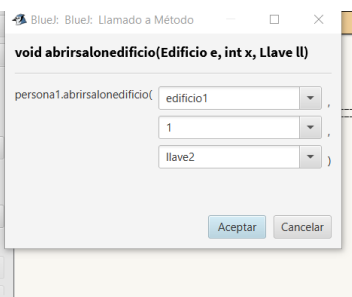


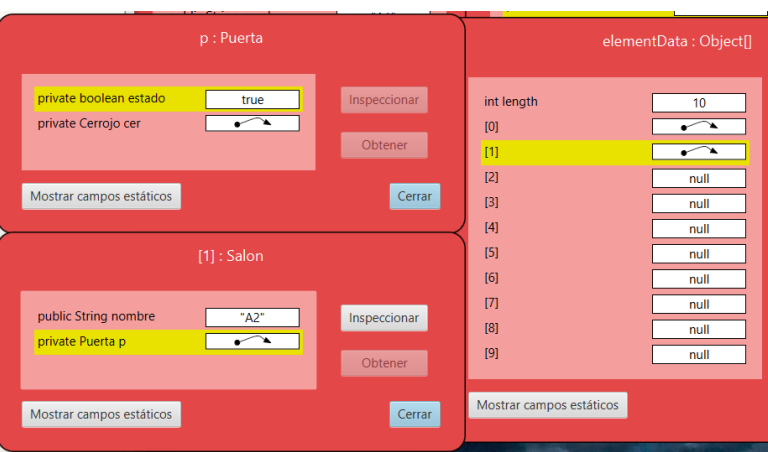
Para los otros métodos utilizaremos más salones, por lo tanto los debemos crear y de igual forma, mas puertas, mas cerrojos y mas llaves.



El método a ejecutar es **abrisalonedificio(Edificio e, int x, Llave ll)**, que funciona de igual forma que los otros métodos, solo que ahora solo necesitamos el edificio que tiene al salón y para ello necesitamos la posición del salón que se desea abrir que es "x" y la llave que pertenece a dicho salón.

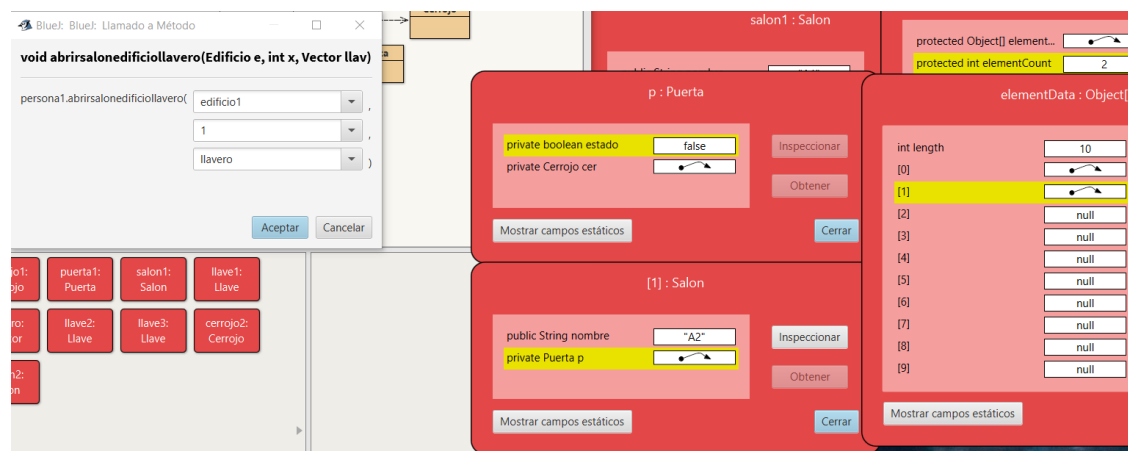
Para ejecutar el método, haremos la prueba con nuestro edificio "Sistemas" y abriremos al salón en la posición 1 que corresponde al salón "A2", para ello le damos de argumento su llave que es la "llave2".





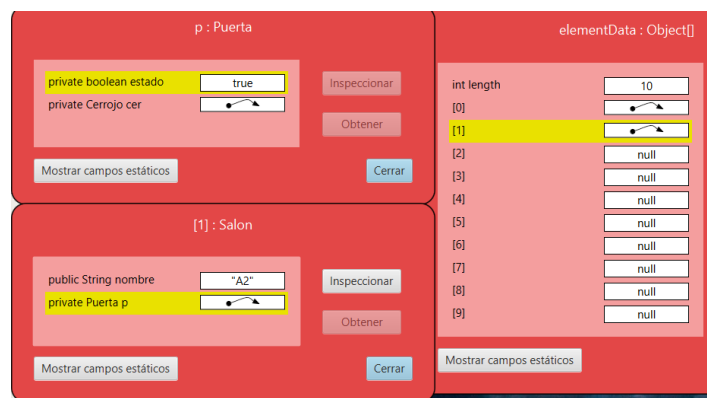
Listo, el método se ejecutó satisfactoriamente y la puerta del salón “A2” se abrió como se esperaba.

El otro método a ejecutar es **abrirsalonedificiollavero(Edificio e, int x, Vector llav)**, que funciona similar al método anterior y la única diferencia es que utilizaremos el llavero para abrir la puerta del salón y no la llave.



Para nuestro caso, le damos de

argumento nuestro edificio “Sistemas” que es donde se encuentra nuestro salón en posición 1 que corresponde al “A2” y el llavero que contiene las llaves de todos los salones.



En esta imagen se aprecia, que el método cumple con lo requerido, haciendo que la puerta del salón indicado se abrió satisfactoriamente.

Partiendo que tenemos nuestro edificio con 3 salones, y cada salón tiene su puerta cerrada, procederemos a ejecutar el método **abrirtodoeledificio(Edificio e, Vector llavero)**, con este método podremos abrir todas las puertas de los salones que se encuentran cerrada. Que, en nuestro caso, las puertas del salón “A1”, “A2”, “A3” se encuentran cerradas. Tendremos nuestro llavero con la llave de cada puerta de los salones.

Una vez ejecutado el método, las puertas de los salones se abren, por lo tanto, el método funciona correctamente.

Partiendo de esta situación, ahora procederemos a ejecutar el otro método, llamado **cerrareledificio(Edificio e)**, que hará lo inverso a este, cerrar las puertas, pero solo teniendo como parámetro el edificio.

```

heredado de Object
boolean abrirPuertallavero(Puerta p, Vector llavero)
boolean abrirSalon(Salon s, Vector llavero)
boolean abrirpuerta(Puerta p, Llave ll)
void abrisalonedificio(Edificio e, int x, Llave ll)
void abrisalonedificiollavero(Edificio e, int x, Vector llav)
void abritodoeledificio(Edificio e, Vector llavero)
void agregarllavellavero(Vector llavero, Llave ll)
void cerraredificio(Edificio e)
Vector crearLlavero()
Llave cualeslallave(Edificio e, int x, Vector llav)
String getNombre()
Vector regresaaabierto(Edificio e)
Vector regresacerrado(Edificio e)
void setNombre(String nombre)
boolean taabierto(Edificio e, int x)

```

Para nuestro caso, le damos de argumento el edificio1 que corresponde a el de “Sistemas”.

Como podemos observar, las puertas vuelven a estar cerradas, por lo tanto, el método funciona excelente.

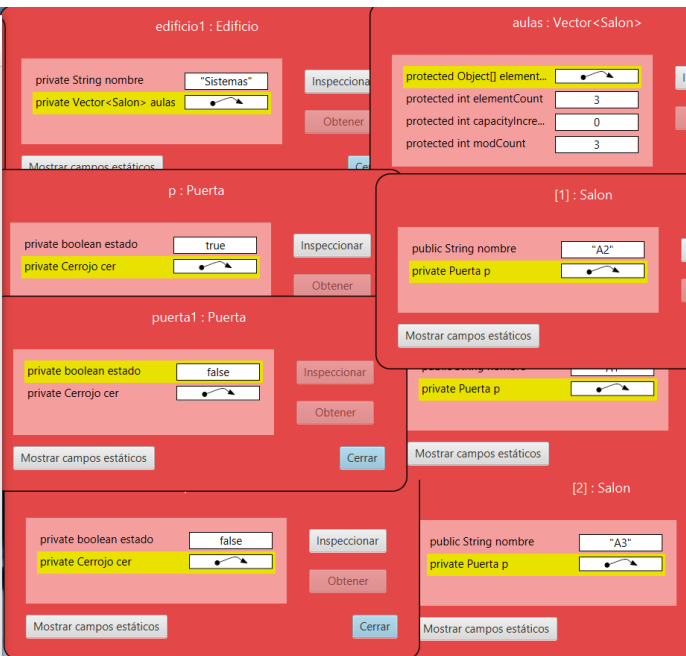
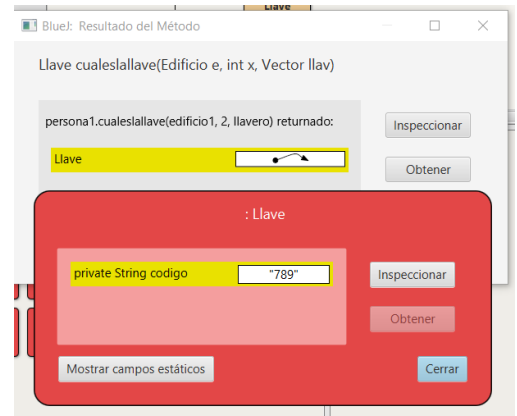
```

heredado de Object
boolean abrirPuertallavero(Puerta p, Vector llavero)
boolean abrirSalon(Salon s, Vector llavero)
boolean abrirpuerta(Puerta p, Llave ll)
void abrisalonedificio(Edificio e, int x, Llave ll)
void abrisalonedificiollavero(Edificio e, int x, Vector llav)
void abritodoedificio(Edificio e, Vector llavero)
void agregarllavellavero(Vector llavero, Llave ll)
void cerraredificio(Edificio e)
Vector crearLlavero()
Llave cualeslallave(Edificio e, int x, Vector llav)
String getNombre()
Vector regresaabiertos(Edificio e)
Vector regresacerrados(Edificio e)
void setNombre(String nombre)
boolean taabierto(Edificio e, int x)

```

El próximo método a ejecutar, es el de **cualeslallave(Edificio e, int x, Vector llav)**, que regresara una llave, que es la cual corresponde a la que abre la puerta del salón ubicado en el edificio.

En la imagen de la derecha se muestra la llave que regresa, si damos de argumento el edificio1, el salón "A3" y el llavero.

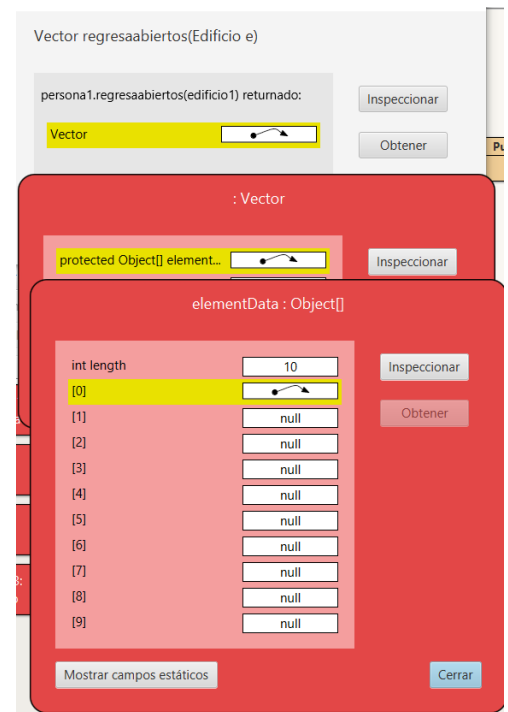


funcionando correctamente.

Partiendo de que tenemos 2 salones con su puerta cerrada y 1 con la puerta abierta, procedemos a ejecutar los otros métodos.

El otro método es **regresaabiertos(Edificio e)**, que regresa un vector donde tiene guardado al salón que tiene la puerta abierta.

Ejecutando el método podemos ver en la imagen de la derecha, que el método retorna la puerta que se encuentra abierta, que corresponde al salón "A2"



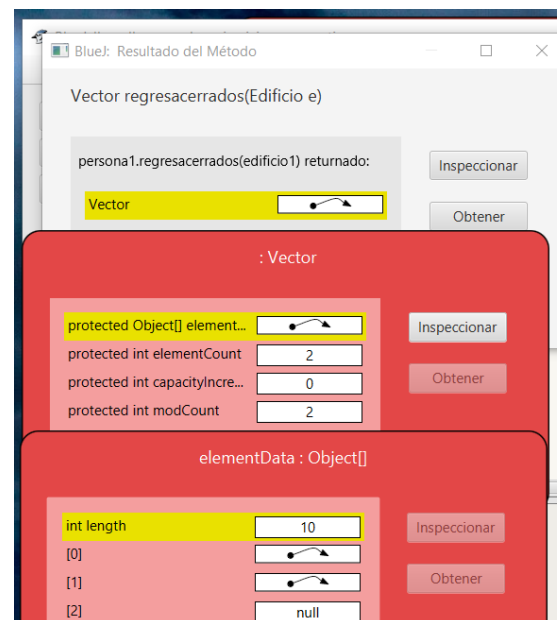
```

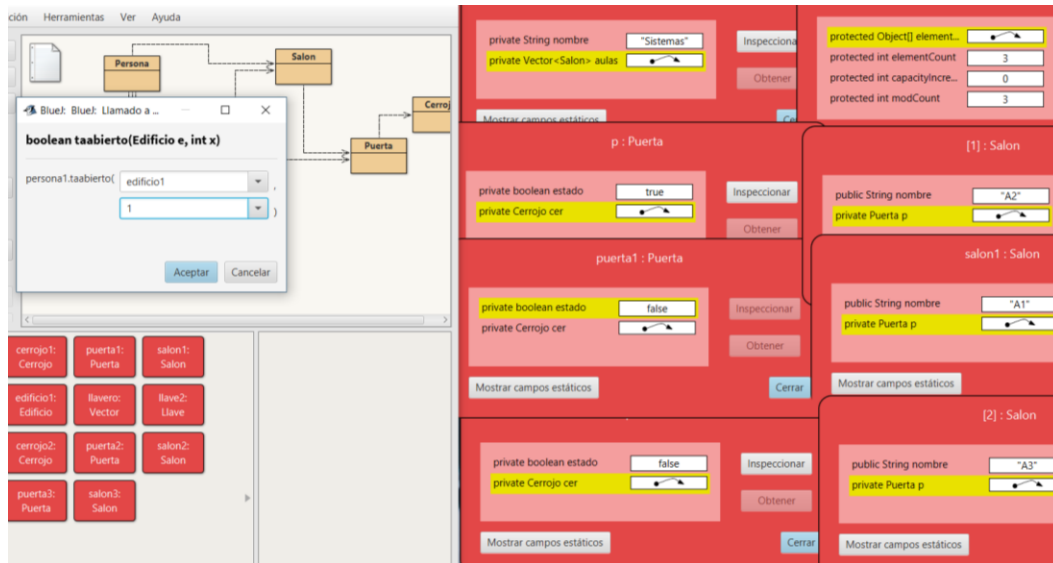
heredado de Object
boolean abrirPuertallavero(Puerta p, Vector llavero)
boolean abrirSalon(Salon s, Vector llavero)
boolean abrirpuerta(Puerta p, Llave ll)
void abrisalonedificio(Edificio e, int x, Llave ll)
void abrisalonedificiollavero(Edificio e, int x, Vector llav)
void abritodoedificio(Edificio e, Vector llavero)
void agregarllavellavero(Vector llavero, Llave ll)
void cerraredificio(Edificio e)
Vector crearLlavero()
Llave cualeslallave(Edificio e, int x, Vector llav)
String getNombre()
Vector regresaabiertos(Edificio e)
Vector regresacerrados(Edificio e)
void setNombre(String nombre)
boolean taabierto(Edificio e, int x)

```

Partiendo de la misma situación, ahora procedemos a ejecutar el método **regresacerrados(Edificio e)**, que en este caso nos regresara a los otros dos salones correspondientes.

Observando la imagen de la derecha, podemos concluir que el método funciona correctamente, ya que retorna a dos salones que tiene la puerta cerrada, que corresponden al salón "A1" y "A3"

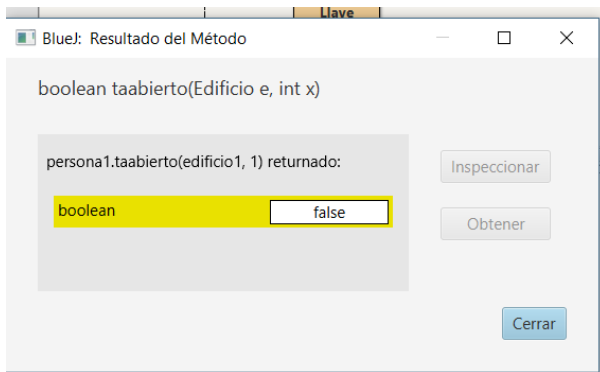




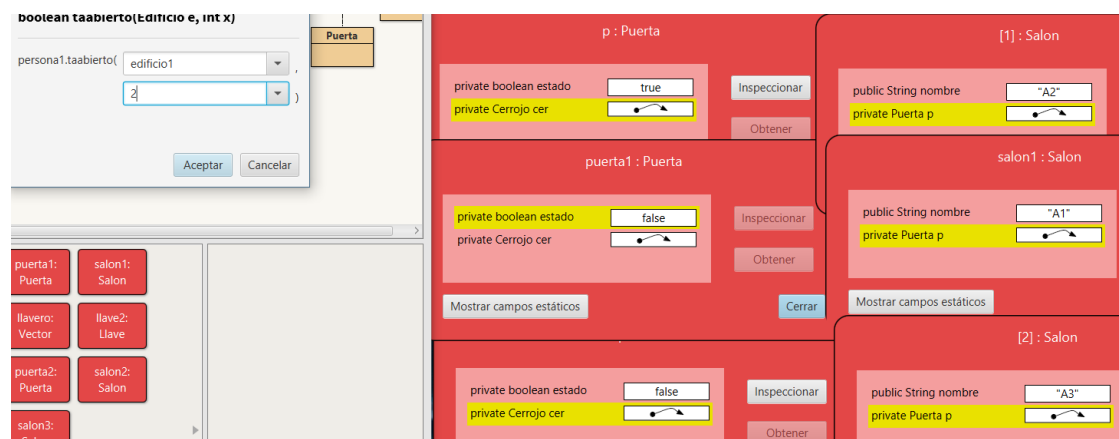
Ejecutamos el ultimo método que es **taabierto(Edificio e, int x)**, que este método regresara un true si el salón esta cerrado y un false si el salón esta abierto.

Lo comprobamos preguntando el estado del salón en la posición 1, que corresponde al salón "A2", que podemos ver que esta abierto por lo tanto el método nos regresaría

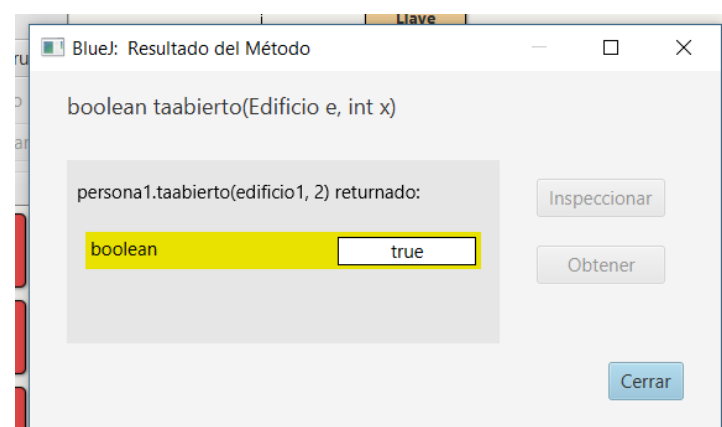
false.



Y justamente como se esperaba, el método retorno false, por lo tanto, podemos concluir que el método es correcto. De igual forma lo podemos probar con el salón "A3", que se encuentra cerrado y nuestro método nos retornaría un true.



Para probar con el salón "A3", introducimos el edificio al que pertenece y su posición que es 2.



Observamos que el método funciona correctamente.

Con este último método se comprueban los métodos que existen en el programa, concluyendo que todos funcionan de una forma correcta y como se solicitaba