**Iemois. Innovación Educativa con MOOC en el programa de Ingeniería de Sistemas**

**Conociendo el Proyecto**

1. Se tienen 3 archivos .java, dentro de la carpeta iemois, estos deberían estar dentro de la carpeta src
2. El proyecto debe tener la siguiente estructura:

Iemois

src

bin

docs

Empleando el comando java, desde el directorio raíz se debe utilizar la siguiente instrucción:

* + javac -d bin src/aplicacion/Mooc.java src/aplicacion/Iemois.java src/presentacion/IemoisGUI.java
  + javadoc -d docs src/aplicacion/Mooc.java src/aplicacion/Iemois.java src/presentacion/IemoisGUI.java
  + java -cp bin presentacion/IemoisGUI

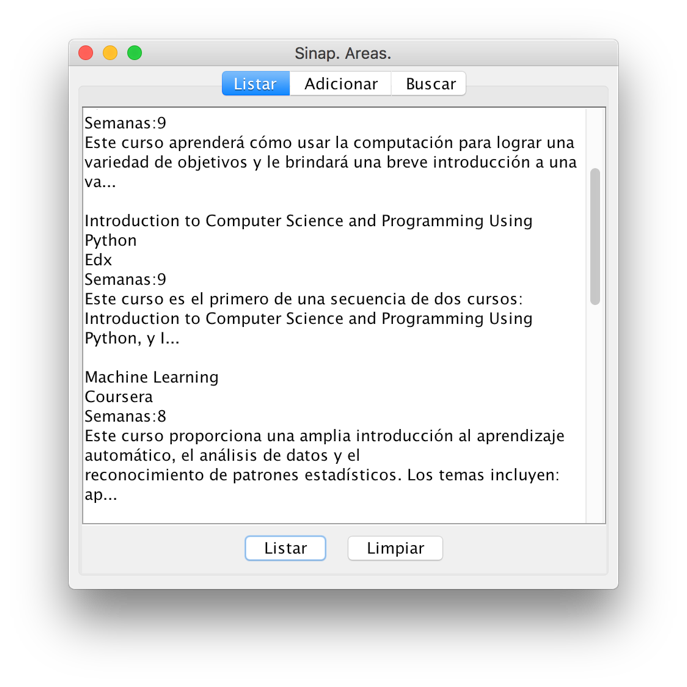
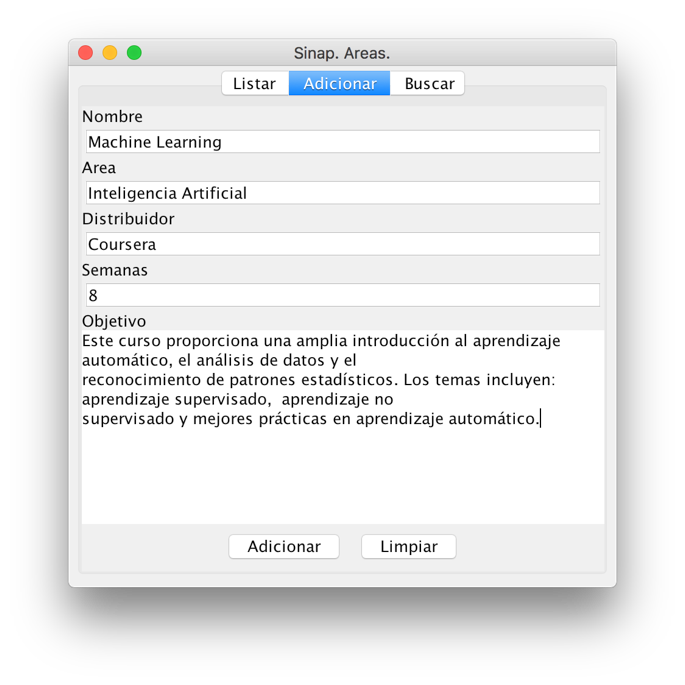
1. Ofrece Listar, adicionar, buscar y limpiar.

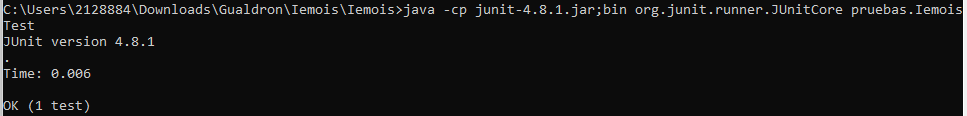
Solo funcionan 3, Listar, adicionar y buscar

1. Los cursos iniciales salen de la clase Iemois del método adicione, en donde se adicionan 3 cursos

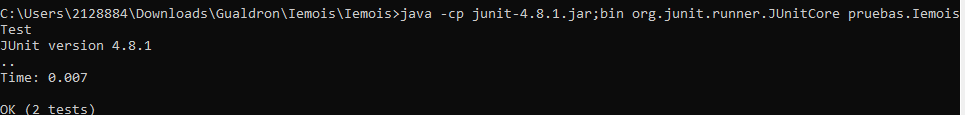
**ADICIONAR Y LISTAR. TODO OK**

1. Se adiciona un nuevo curso, para comprobarlo se utiliza el botón de Listar, y el comportamiento es adecuado debido a que se lista los cursos que hay.

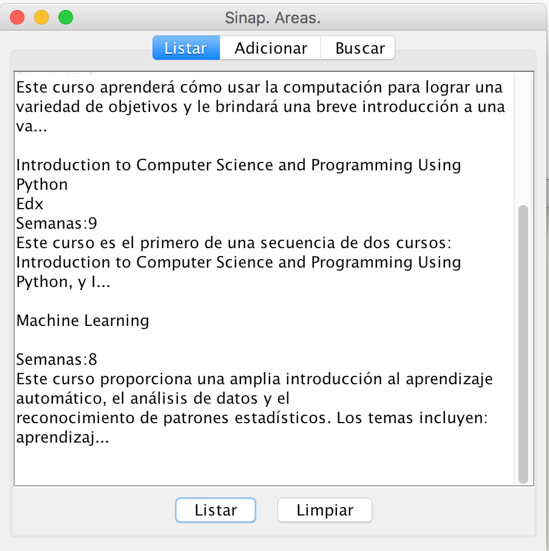
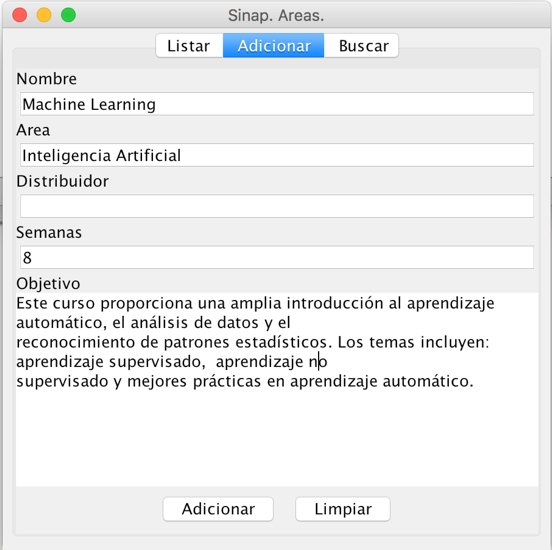


1. En aplicación el método adicione es el encargado de adicionar mientras que en presentación, el encargado de eso es accionAdicionar
2. . 
3. En la capa de presentación el método encargado es accionAdicionar

En la capa de aplicación el método encargado es toString

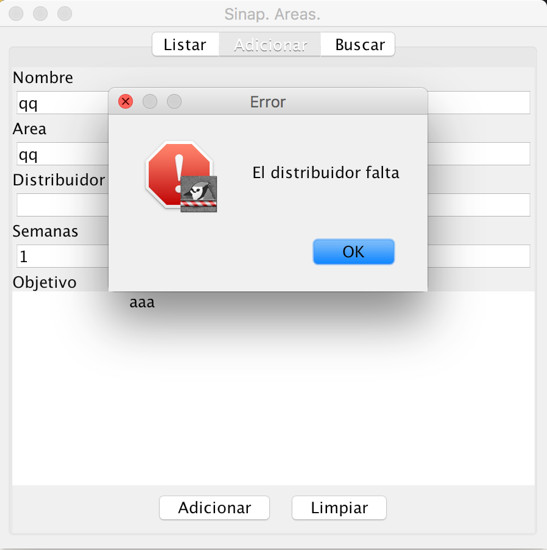
1. **.**
2. 

**Adicionar un curso. ¿Y si no da un distribuidor?**

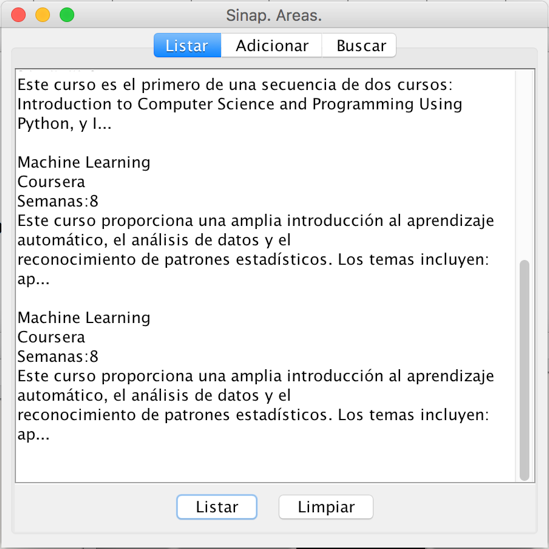


Se adiciona sin el distribuidor, se comprueba listando los cursos, el comportamiento es inadecuado, no debería permitir agregar el curso con información faltante.

1. .
2. El método adicionar de Iemois debería propagarla, y el método accionAdicionar de IemoisGUI debería atenderla
3. .

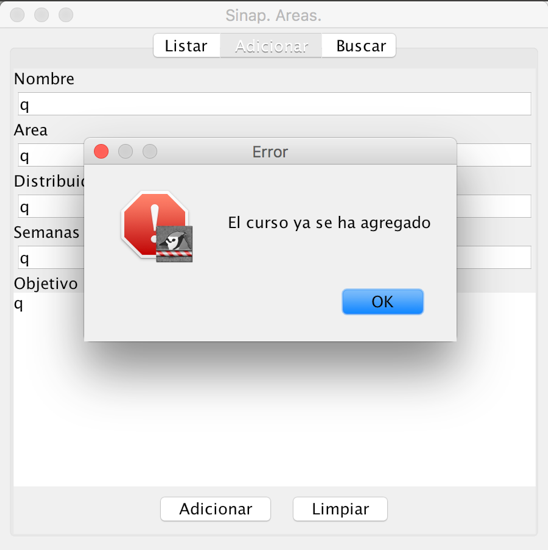
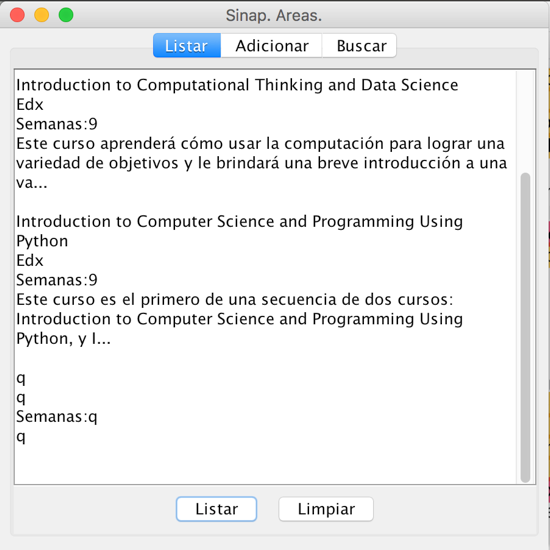


**Adicionar un curso. ¿Y si ya se encuentra?**

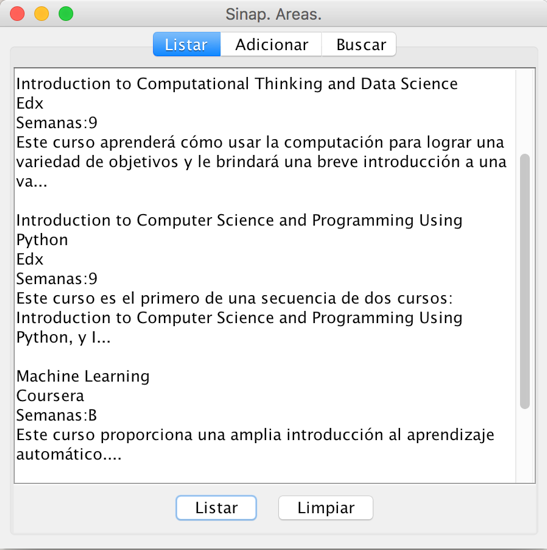
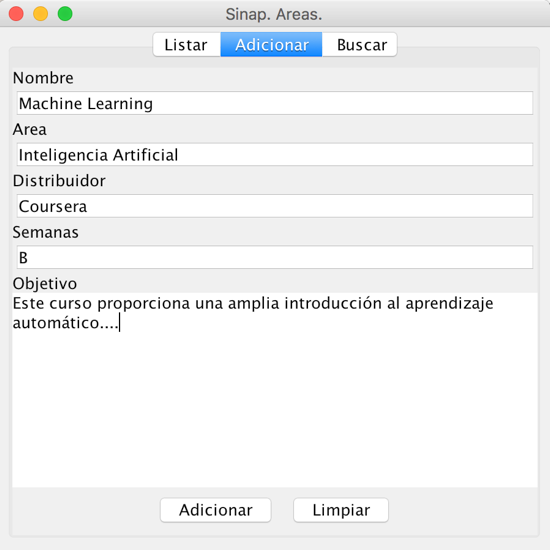
****

Se adicionan dos cursos pero dos veces, se comprueba listando los cursos, el comportamiento no es adecuado, solamente se debería poder agregar un curso sin repetición

1. La excepción debería ser lanzada por adicione de igual manera debe ser propagada por este método y finalmente el método accionAdicionar en la clase IemoisGUI debe ateneder esta excepción.
2. .

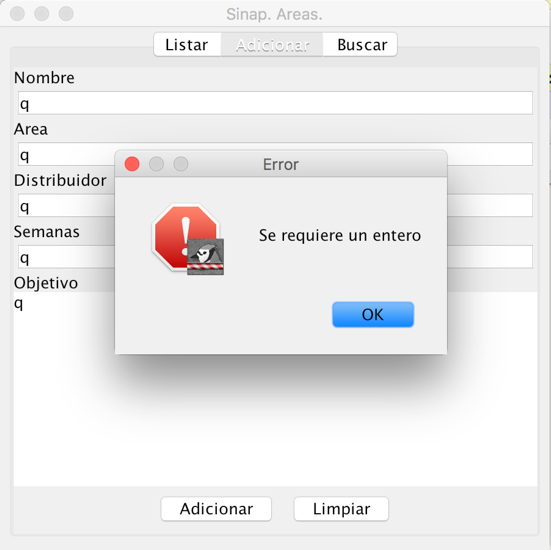


**Adicionar un curso. ¿Y si dan mal las semanas?**

1. 

Se agrega el curso sin revisar que sea una letra en vez de un numero, se comprueba listando los cursos agregados, el comportamiento no es adecuado

1. el método accionAdicionar en la clase IemoisGUI debe atender la excepción.
2. .

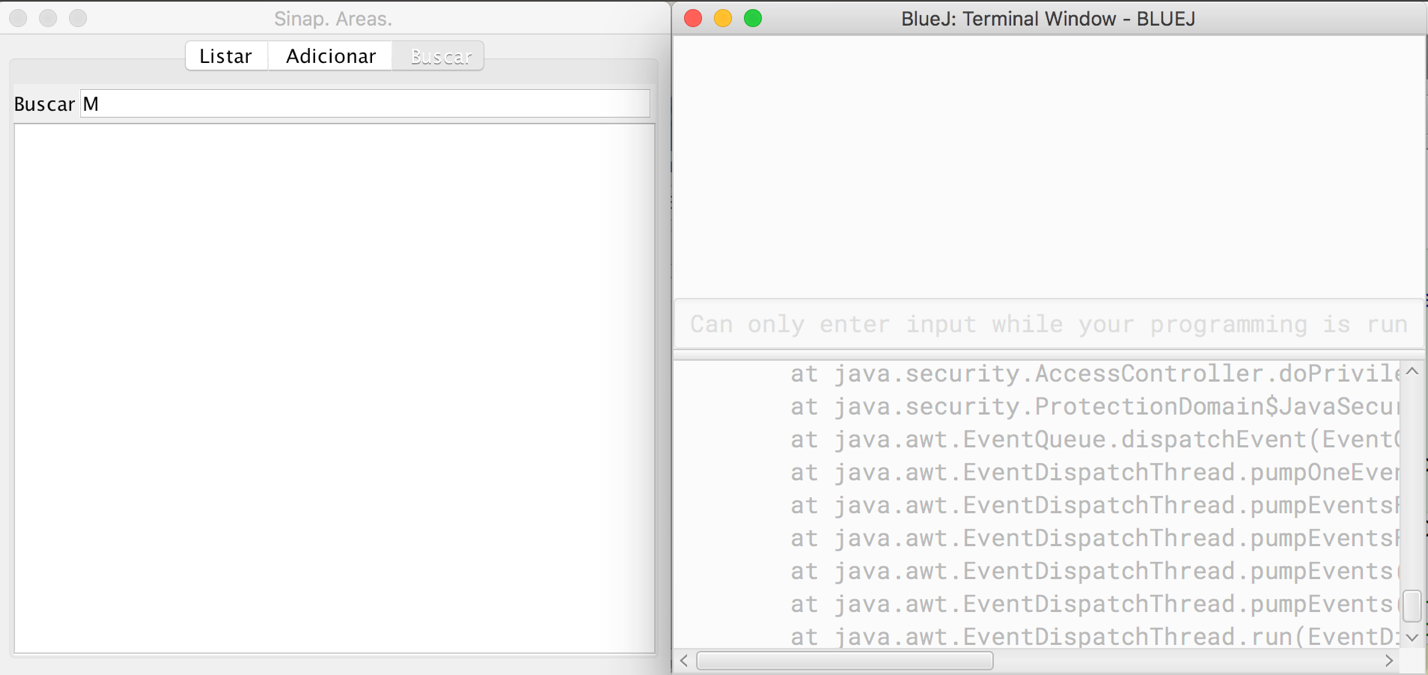


La excepción es atendida de la manera correcta

**Adicionar un curso. ¿Otras condiciones?**

1. Nada puede estar vacío; nombre, área, distribuidor y objetivo deben empezar por String pero pueden contener enteros y todo debe empezar por mayúscula
2. .

**Consultando por patrones. ¡ No funciona y queda sin funcionar!**



Sucede que hay un error al buscar los cursos, el programador debe conocer los errores de la aplicación, pero el usuario NO.

1. El método registre se encarga de llevar una bitácora, en donde en caso de haber alguna excepción, nos va a mostrar el rastro de esta.
2. .
3. .
4. .
5. Se realiza un try and catch de manera que si no es posible hacer una búsqueda, la aplicación sea capaz de poder tratar la excepción y poder continuar su ejecución, y para poder mostrar al usuario este error se muestra una alerta
6. Se muestra la alerta definida anteriormente, la aplicación continua su ejecución. El nuevo archivo llamado mooc.log contiene el rastro de todos las excepciones.

**Consultando por patrones. ¡Ahora si funciona!**

1. En la capa de presentación el encargado es el método accionBuscar y en la capa de aplicación, el método encargado es busque