

PROGRAMACIÓN ORIENTADA A OBJETOS

Excepciones

2021-1

Laboratorio 4/6

OBJETIVOS

1. Perfeccionar el diseño y código de un proyecto considerando casos especiales y errores.
2. Construir clases de excepción encapsulando mensajes.
3. Manejar excepciones considerando los diferentes tipos.
4. Registrar la información de errores que debe conocer el equipo de desarrollo de una aplicación en producción.
5. Vivenciar las prácticas Designing Simplicity.
Coding Code must be written to agreed standards

Acciones

EN BLUEJ

PRACTICANDO MDD y BDD con EXCEPCIONES

[En lab04.doc, acciones.asta y BlueJ acciones]

En este punto vamos a aprender a diseñar, codificar y probar usando excepciones. Para esto se van a trabajar algunos métodos de la clase Acción y la excepción AccionExcepcion

1. En su directorio descarguen los archivos contenidos en accion.zip, revisen el contenido y estudien el diseño estructural de la aplicación.
2. Expliquen por qué el proyecto no compila. Realicen las adiciones necesarias para lograrlo.

El proyecto no compila porque la clase AccionSimple y accionCompuesta heredan de la clase abstracta Accion y esta clase tiene tres métodos los cuales no están en las clases hijas y es necesario tener estos métodos elaborados ya que la clase padre es abstracta.

```
public int dias() throws AccionExcepcion{
    if(dias==null){
        throw new AccionExcepcion(AccionExcepcion.SIMPLE_SIN_DIAS);
    }
    else
        return dias.intValue();
}
```

```
public int dias() throws AccionExcepcion{
    int cont=0;
    for(int i=0;i<acciones.size();i++){
        cont+=acciones.get(i).dias();
    }
    if(cont==0){
        throw new AccionExcepcion(AccionExcepcion.COMPUESTA_VACIA);
    }
    return cont;
}
```

3. Dadas las pruebas, documenten, diseñen y codifiquen el método dias().
4. Dada la documentación, diseñen, codifiquen y prueben el método diasEstimados() .
5. Documenten, diseñen, codifiquen y prueben el método dias(String nombre).

En este método vamos a incluir dos nuevos casos especiales :

- 1) no existe una acción con ese nombre y
- 2) existe más de una acción con el mismo nombre

SINAP. Sistema Nacional de Áreas Protegidas

EN CONSOLA

Conociendo el proyecto SINAP [En lab04.doc]

No olviden respetar los directorios bin docs src

1. En su directorio descarguen los archivos contenidos en SINAP.zip, revisen el contenido.

¿Cuántos archivos se tienen?

Se tienen 5 archivos

¿Cómo están organizados?

Todos estan en la carpeta raiz

¿Cómo deberían estar organizados?

Por paquetes, los de aplicación en el paquete aplicación y los de presentación en el paquete de presentación.

2. Estudien el diseño del programa: diagramas de paquetes y de clases.

¿cuántos paquetes tenemos?

dos paquetes(aplicacion,presentacion)

¿cuántas clases tiene el sistema?

cuatro clases

¿cómo están organizadas?

En aplicacion esta:

Area,Sinap,registro

En presentacion esta:

SinapGUI

¿cuál es la clase ejecutiva?

SinapGUI

3. Prepare los directorios necesarios para ejecutar el proyecto.

¿qué estructura debe tener?

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Versión 10.0.19042.867]
(c) 2020 Microsoft Corporation. Todos los derechos reservados.

C:\Users\57350\Desktop\SINAP>MD src

C:\Users\57350\Desktop\SINAP>MD bin

C:\Users\57350\Desktop\SINAP>cd src

C:\Users\57350\Desktop\SINAP\src>MD aplicacion

C:\Users\57350\Desktop\SINAP\src>MD presentacion

C:\Users\57350\Desktop\SINAP\src>cd ..

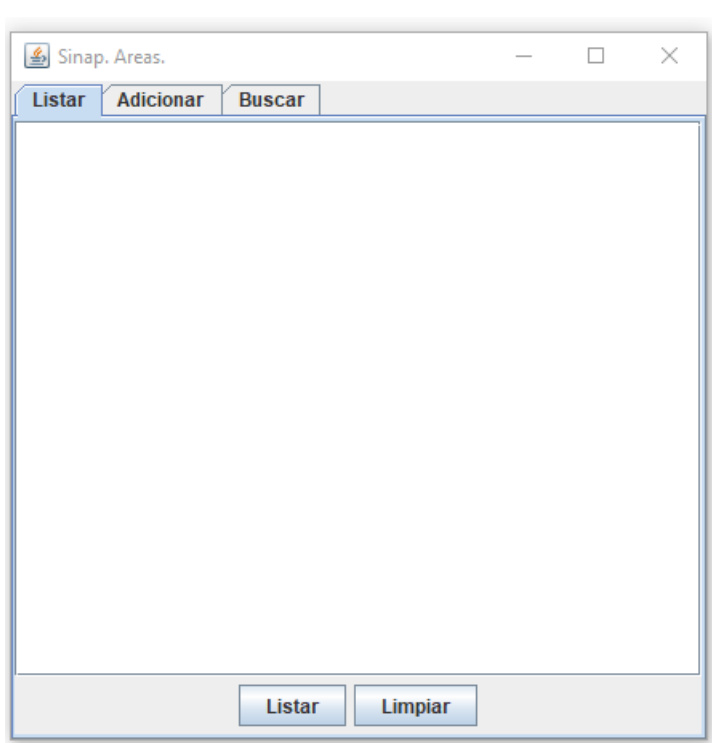
C:\Users\57350\Desktop\SINAP>tree
Listado de rutas de carpetas para el volumen Windows-SSD
El número de serie del volumen es B6E9-774E
C:.
|-- bin
|-- src
    |-- aplicacion
    |-- presentacion

C:\Users\57350\Desktop\SINAP>
```

¿qué instrucciones debe dar para ejecutarlo?

```
C:\Users\57350\Desktop\SINAP>javac -d bin src\aplicacion\*.java src\presentacion\*.java

C:\Users\57350\Desktop\SINAP>java -cp bin presentacion.SinapGUI
```



4. Ejecute el proyecto

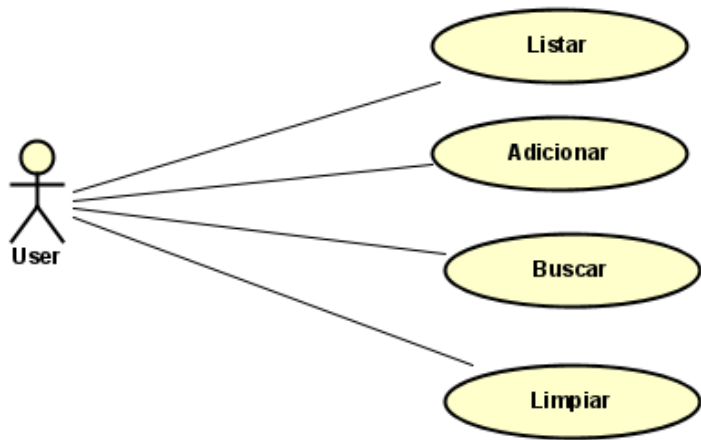
¿qué funcionalidades ofrece?

Listar,Limpiar,Adicionar y Buscar.

¿cuáles funcionan?

Listar,Adicionar

Realicen el diagrama de casos de uso correspondiente.



5. ¿De dónde salen las áreas iniciales?

De sinap

Revisen el código y la documentación del proyecto.

¿Qué clase pide que se adicionen?

La clase SinapGUI

¿Qué clase los adiciona?

Sinap

Adicionar y listar. Todo OK. [En lab04.doc, SINAP.asta y *.java]

(NO OLVIDEN BDD - MDD)

El objetivo es realizar ingeniería reversa a las funciones de adicionar y listar.

1. Adicionen una nueva área

Tuparro

Tuparro National Park

Vichada

548.000

Es una extensa sabana verde surcada por grandes ríos con potentes raudales y playas doradas, pequeños caños de aguas cristalinas, bosques de galería, morichales y saladillales, además de enormes rocas cristalinas en forma de cerros redondeados.

¿Qué ocurre? ¿Cómo lo comprueban?

Capturen la pantalla.

```
"alimentaria, entre otros factores."),
new Area("Tuparro","Tuparro National Park","Vichada","548.000","Es una extensa sabana verde surcada por grandes rios con potentes rauda:
"caños de aguas cristalinas, bosques de galería, morichales y saladillales, además de enormes rocas"+
"cristalinas en forma de cerros redondeados.")
;
or(Area detalles : ejemplos) {
    adicioneDetalles(detalles);
}
```

nsulta los detalles de una area

aram nombre El nombre de la area a buscar

aram name El nombre internacional de la area a buscar

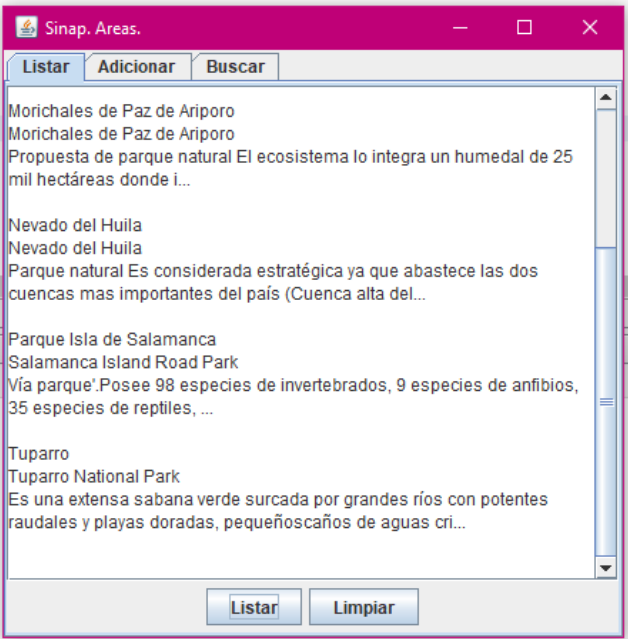
adicioneCinco >

lab4 (run) #2 x

lab4 (run) #3 x

lab4 (run) #4 x

lab4 (run) #5 x



Se agrega un área en la lista, se comprueba al ejecutar la aplicación y seleccionar el botón listar.

¿Es adecuado este comportamiento?

Si es adecuado ya que en esa lista se van añadiendo todas las areas que se adicionan en SINAP.

2. Revisen el código asociado a adicionar en la capa de presentación y la capa de aplicación.

¿Qué método es responsable en la capa de presentación?

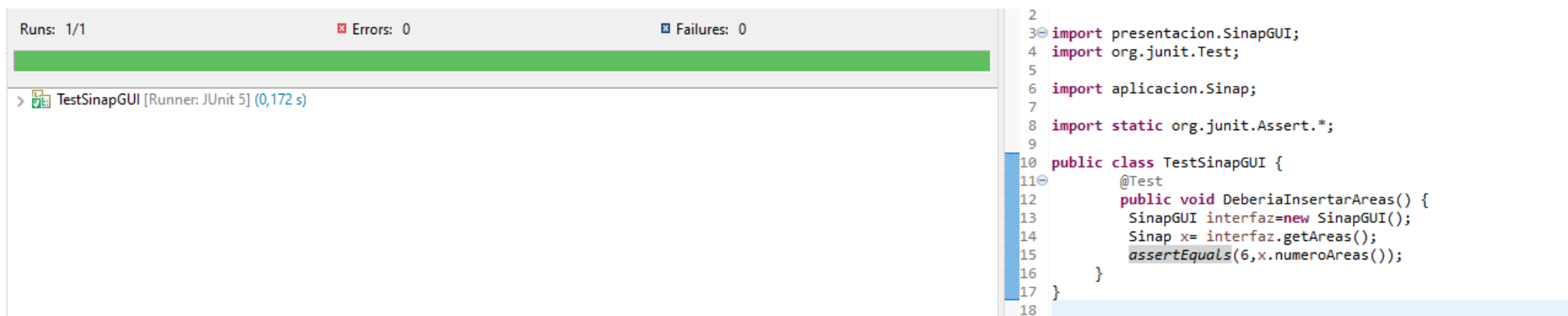
accionAdicionar()

¿Qué método en la capa de aplicación?

adicone(String nombre, String name,String ubicacion, String area, String descripcion)

3. Realicen ingeniería reversa para la capa de aplicación para adicionar.

Capturen los resultados de las pruebas de unidad.



4. Revisen el código asociado a listar en la capa de presentación y la capa de aplicación.

¿Qué método es responsable en la capa de presentación?

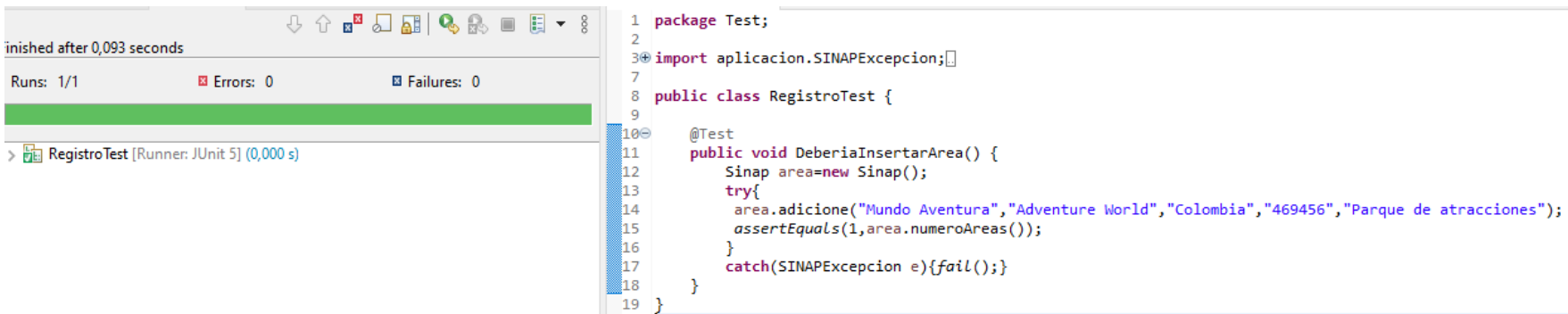
accionListar()

¿Qué método en la capa de aplicación?

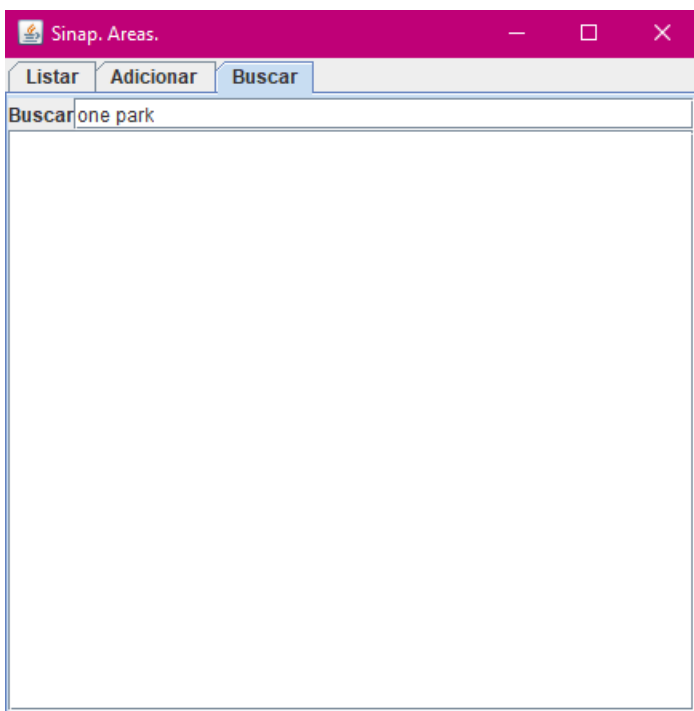
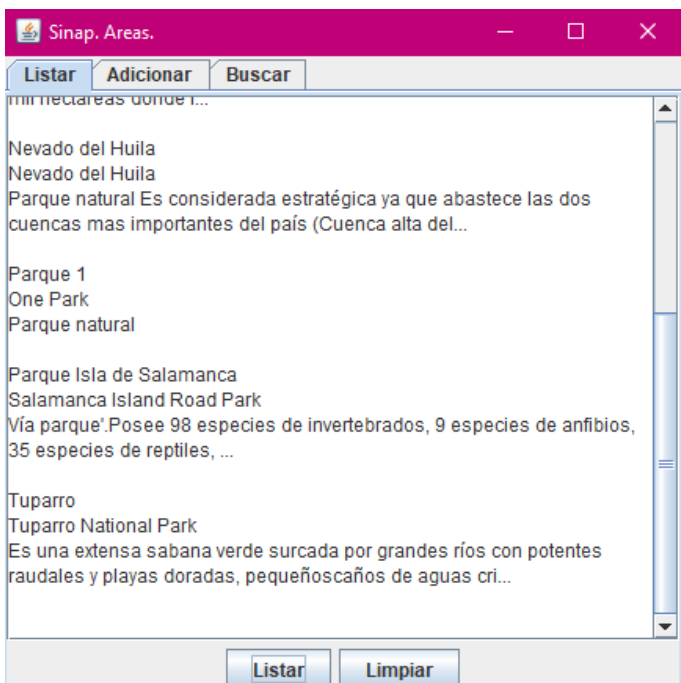
adiconeDetalles(Area detalles)

5. Realicen ingeniería reversa para la capa de aplicación para listar.

Capturen los resultados de las pruebas de unidad.



6. Propongan y ejecuten una prueba de aceptación.



```

at java.security.ProtectionDomain$JavaSecurityAccessImpl.doIntersectionPrivil
at java.awt.EventQueue$4.run(EventQueue.java:731)
at java.awt.EventQueue$4.run(EventQueue.java:729)
at java.security.AccessController.doPrivileged(Native Method)
at java.security.ProtectionDomain$JavaSecurityAccessImpl.doIntersectionPrivil
at java.awt.EventQueue.dispatchEvent(EventQueue.java:728)
at java.awt.EventDispatchThread.pumpOneEventForFilters(EventDispatchThread.ja
at java.awt.EventDispatchThread.pumpEventsForFilter(EventDispatchThread.java:
at java.awt.EventDispatchThread.pumpEventsForHierarchy(EventDispatchThread.ja
at java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:101)
at java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:93)
at java.awt.EventDispatchThread.run(EventDispatchThread.java:82)

```

Adicionar un área. ¿Y si no da un nombre? [En lab04.doc, SINAP.asta y *.java]
(NO OLVIDEN BDD – MDD)

El objetivo es perfeccionar la funcionalidad de adicionar un área.

1. Adicionen el área Tuparro sin nombre internacional.

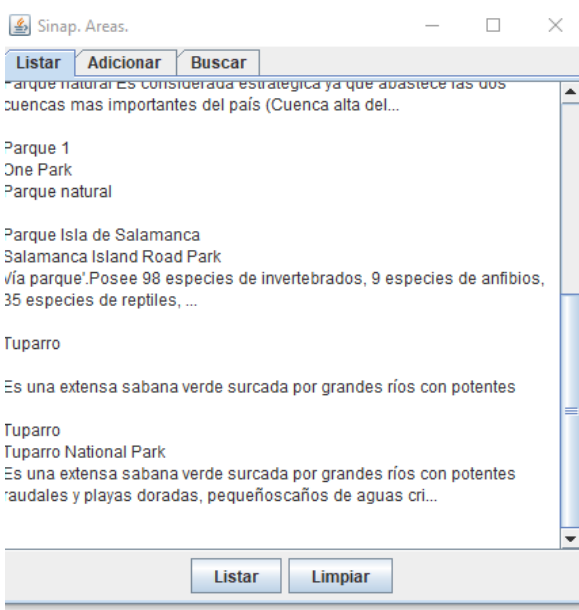
¿Qué ocurre?

Se adiciona el área dejando el espacio vacío.

¿Cómo lo comprueban?

Al ejecutar la aplicacion,adicionar y luego listar.

Capturen la pantalla.



¿Es adecuado este comportamiento?

No es correcto porque lo que se esperaría es que no se tenga ese espacio y nos pidiera el nombre.

2. Vamos a evitar la creación de áreas con un nombre internacional vacío manejando una excepción

SINAPExcepcion. Si el área no tiene nombre internacional, no la creamos y se lo comunicamos al usuario.Para esto lo primero que debemos hacer es crear la nueva clase SINAPExcepcion considerando este primer mensaje.

3. Analicen el diseño realizado.

¿Qué método debería lanzar la excepción?

adicione(String nombre, String name,String ubicacion, String area, String descripcion)

¿Qué métodos deberían propagarla?

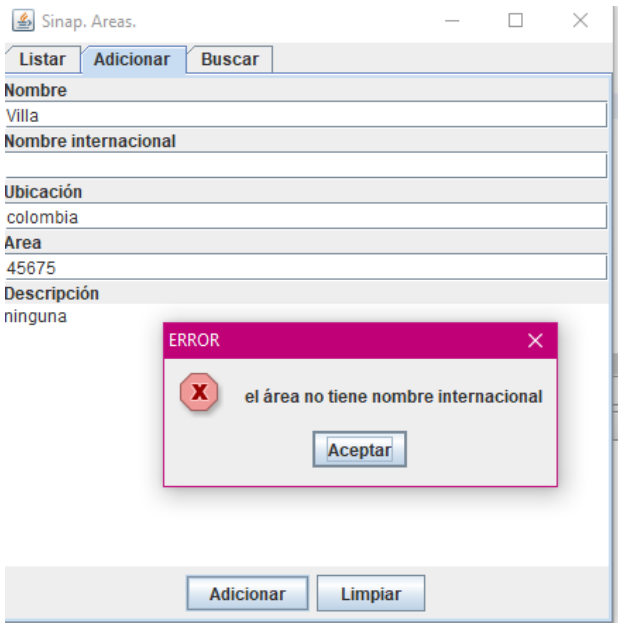
adicione(String nombre, String name,String ubicacion, String area, String descripcion)

¿Qué método debería atenderla? Explique claramente.

accionAdicionar()

4. Construya la solución propuesta.

Capturen los resultados de las pruebas.

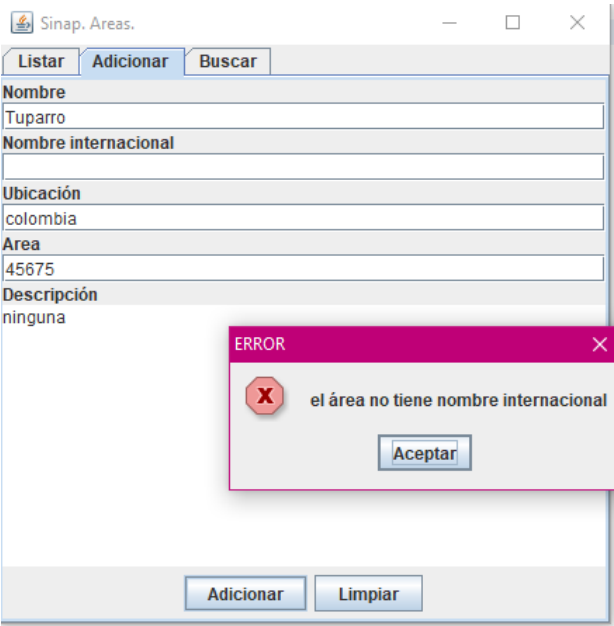


5. Ejecuten nuevamente la aplicación con el caso de prueba propuesto en 1.,

¿Qué sucede ahora?

Ya no permite registrar el área debido a que no tiene un nombre internacional.

Capture la pantalla.



Adicionar un área. ¿Y si ya se encuentra? [EnSINAP.asta, lab04.java y *.java] (NO OLVIDEN BDD - MDD)

El objetivo es perfeccionar la funcionalidad de adicionar un área.

1. Adicionen dos veces la nueva área Tuparro

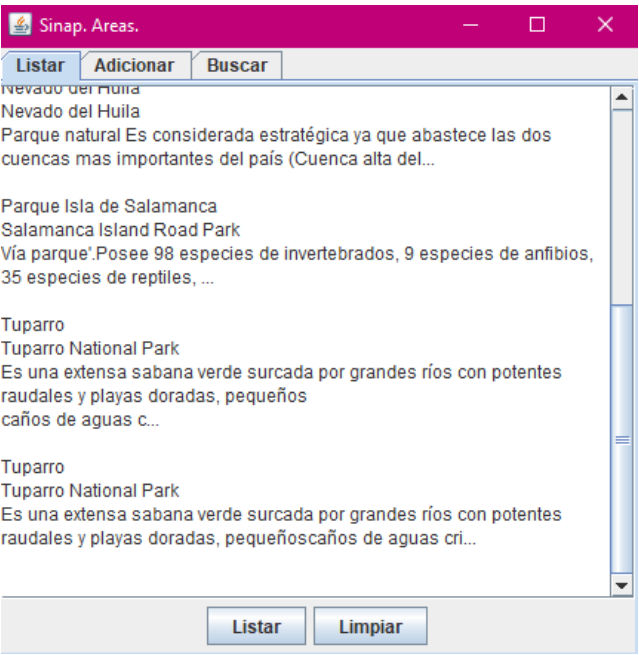
¿Qué ocurre?

Permite adicionar ambas áreas.

¿Cómo lo comprueban?

Al adicionar el área y luego listar las áreas.

Capturen la pantalla.



¿Es adecuado este comportamiento?

No debido a que ya existe un registro de esa área y no deberían estar repetidas solo debería existir un único área.

2. Analicen el diseño realizado. Explique claramente.

¿Qué método debería lanzar la excepción?

adiconeDetalles(Area detalles)

¿Qué métodos deberían propagarla?

adiconeDetalles(Area detalles)

¿Qué método debería atenderla?

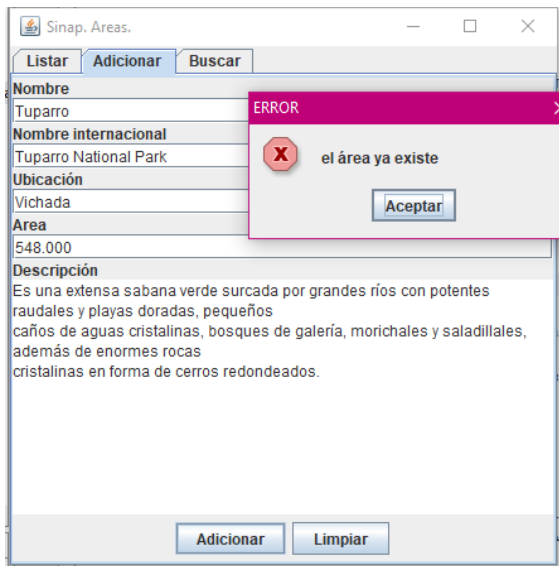
accionAdicionar()

3. Construya la solución propuesta. Capturen los resultados de las pruebas.

4. Ejecuten nuevamente la aplicación con el caso de prueba propuesto en 1.,

¿Qué sucede ahora? Capture la pantalla.

No se permite adicionar el área.



**Adicionar un área. ¿Otras condiciones? [En lab04.doc, SINAP.asta y *.java]
(NO OLVIDEN BDD – MDD)**

El objetivo es perfeccionar la funcionalidad de adicionar un área.

1. Propongan nuevas condiciones para que la adición de un área sea más robusta.

El área no puede ser igual o menor a 0.

2. Construya la solución propuesta. (diseño, prueba de unidad, código) Capturen los resultados de las pruebas.



**Consultando por patrones. ¡ No funciona y queda sin funcionar!
[EnSINAP.asta,SINAP.log, lab04.java y *.java]
(NO OLVIDEN BDD - MDD)**

1. Consulten un área especial que inicie con T.

¿Qué sucede?

La consola muestra un error

¿Qué creen que pasó? Capturen el resultado.

Que no se tiene bien guardadas las áreas para la clase buscar, en resultados solo se guardan valores nulos..

¿Quién debe conocer y quien NO debe conocer esta información?

Sinap debe conocer esa información

Registro no debe conocer esa información

2. Exploren el método registre de la clase Registro

¿Qué servicio presta?

Mostrar la lista de las áreas.

3. Analicen el punto adecuado para que SIEMPRE, al sufrir en cualquier punto el sistema un incidente como este, se presente un mensaje especial de alerta al usuario, se guarde la información del error en el registro de error y termine la ejecución. Expliquen y construyan la solución.

4. Ejecuten nuevamente la aplicación con el caso propuesto en 1.

¿Qué mensaje salió en pantalla?

Se genero un error

¿La aplicación termina?

Si se termina

¿Qué información tiene el archivo de errores?

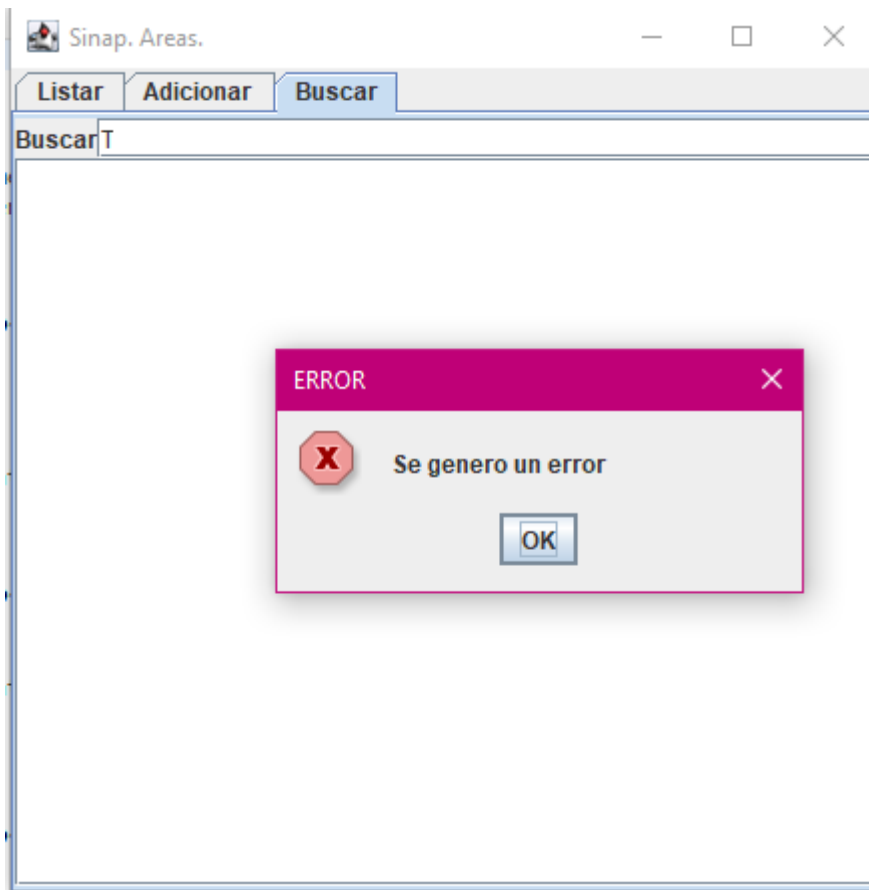
5. ¿Es adecuado que la aplicación continúe su ejecución después de sufrir un incidente como este?

Si es adecuado ya que solo debería informar

¿de qué dependería continuar o parar?

De lo critico que sea el error para el funcionamiento del programa.

6. Analicen el punto adecuado para que EN ESTE CASO se presente un mensaje especial de alerta al usuario, se guarde la información del error en el registro y continúe la ejecución. Expliquen y construyan la solución. No eliminen la solución de 3.



7. Ejecuten nuevamente la aplicación con el caso propuesto en 1.

¿Qué mensaje salió en pantalla?

Se genero un error

¿La aplicación termina?

si

¿Qué información tiene el archivo de errores?

Consultando por patrones. ¡Ahora si funciona!

(NO OLVIDEN BDD - MDD)

1. Revisen el código asociado a buscar en la capa de presentación y la capa de aplicación.

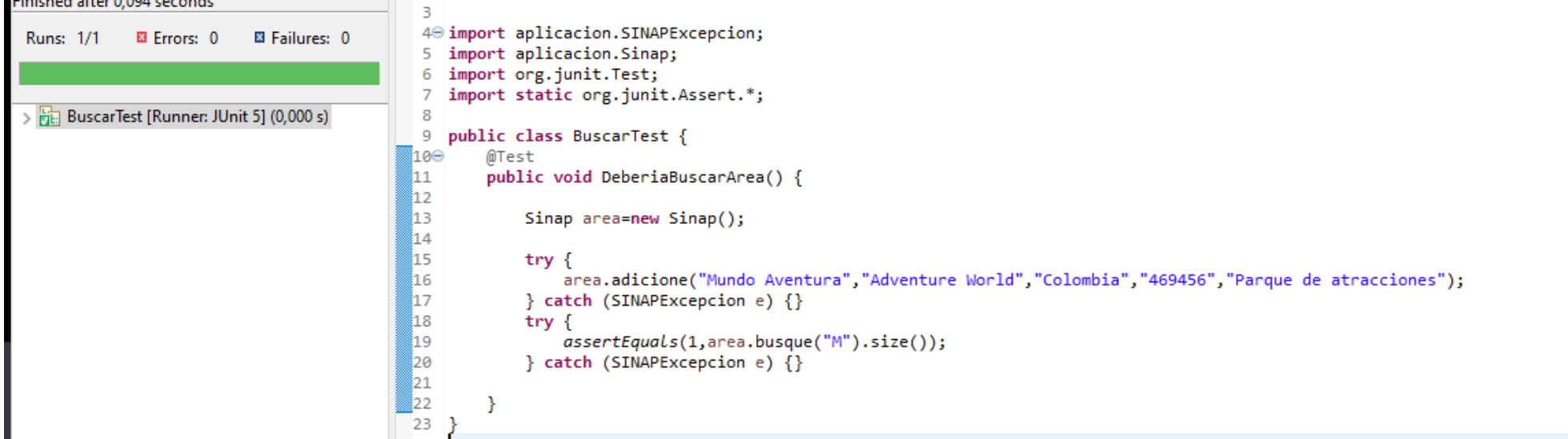
¿Qué método es responsable en la capa de presentación?

accionBuscar()

¿Qué método es responsable en la capa de aplicación?

busque(String prefijo)

2. Realicen ingeniería reversa para la capa de aplicación para buscar. Capturen los resultados de las pruebas. Deben fallar.



3. ¿Cuál es el error? Solúcionenlo. Capturen los resultados de las pruebas.

4. Ejecuten la aplicación nuevamente con el caso propuesto.

¿Qué tenemos en pantalla?

¿Qué información tiene el archivo de errores?

RETROSPECTIVA

1. ¿Cuál fue el tiempo total invertido en el laboratorio por cada uno de ustedes? (Horas/Hombre)

16 horas

2. ¿Cuál es el estado actual del laboratorio? ¿Por qué?

Están implementados la mayoría de los métodos.

3. Considerando las prácticas XP del laboratorio. ¿cuál fue la más útil? ¿por qué?

Simplicidad porque nos ayudaron en la elaboración de los diagramas y el código ya que al aplicarlo pudimos ahorrar mucho tiempo y obtener mejores resultados.

4. ¿Cuál consideran fue el mayor logro? ¿Por qué?

Lograr implementar todas las excepciones porque teníamos muchas dudas con este tema.

5. ¿Cuál consideran que fue el mayor problema técnico? ¿Qué hicieron para resolverlo?

La instalación del jdk, investigando en internet, mirando tutoriales.

6. ¿Qué hicieron bien como equipo? ¿Qué se comprometen a hacer para mejorar los resultados?

Socializar nuestras soluciones, ayudarnos mutuamente.