## PROGRAMACIÓN ORIENTADA A OBJETOS

**Excepciones Octubre 2017 Laboratorio 4/6**

**OBJETIVOS**

 Perfeccionar el diseño y código de un proyecto considerando casos especiales y errores.

 Construir clases de excepción encapsulando mensajes.

 Manejar excepciones considerando los diferentes tipos.

 Registrar la información de errores que debe conocer el equipo de desarrollo de una aplicación en producción.

 Vivenciar la prácticas ***Designing*** [*Simplicity*](http://www.extremeprogramming.org/rules/simple.html)*.*

**Coding** Code must be written to agreed [standards](http://www.extremeprogramming.org/rules/standards.html)

## ENTREGA

 Incluyan en un archivo .zip los archivos correspondientes al laboratorio. El nombre debe ser los dos apellidos de los miembros del equipo ordenados alfabéticamente.

 Deben publicar el avance al final de la sesión y la versión definitiva en la fecha indicada,

en los espacios preparados para tal fin.

# EQUIPOS EN BLUEJ

**PRACTICANDO MDD y BDD con EXCEPCIONES [En lab04.doc, equipos.asta y BlueJ equipos**

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

1. En su directorio descarguen los archivos contenidos en equipo.zip, revisen el contenido y estudien el diseño estructural de la aplicación.
2. ¿De qué clase no pueden consultar las fuentes? ¿En qué formato está? ¿Por qué? Estudien su documentación en el diseño o en el código (/doc).

**R:/** De la clase persona, esta en formato html,

1. Dadas las pruebas, diseñen y codifiquen el método valorHora.
2. Dada la especificación, diseñen, codifiquen y prueben el método valorHoraEstimado.

PARA LAS PRUEBAS



### 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? ¿Cómo están organizados? ¿Cómo deberían estar organizados?

**R:/**se tienen 5 archivos 4 de java y el de astah, están organizados por las clases que existen en el paquete, se debe organizar de acuerdo a su documentación y a su compilación(.java,.class,.ctxt) con sus respectivas carpetas bin y doc.

1. Estudien el diseño del programa: diagramas de paquetes y de clases. ¿cuántos paquetes tenemos? ¿cuántas clases tiene el sistema? ¿cómo están organizadas? ¿cuál es la clase ejecutiva?

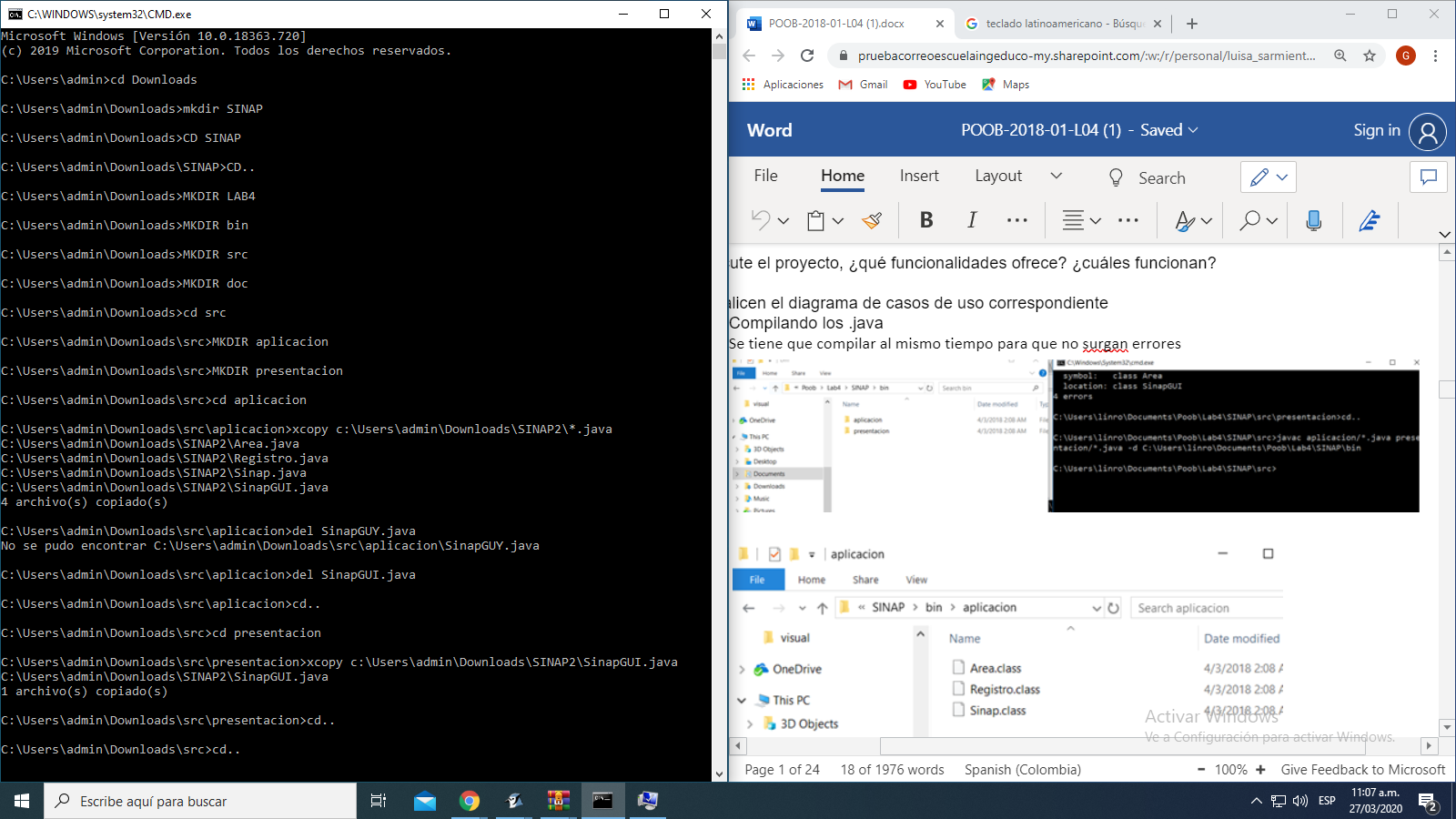
**R:/**se tienen 2 paquetes el de aplicación y el de presentación,posee 4 clases Sinap, SinapGUI, Registro, Área; Además de las heredadas para el control de botones. están organizadas por .java en la carpeta y la clase ejecutiva es SinapGUI

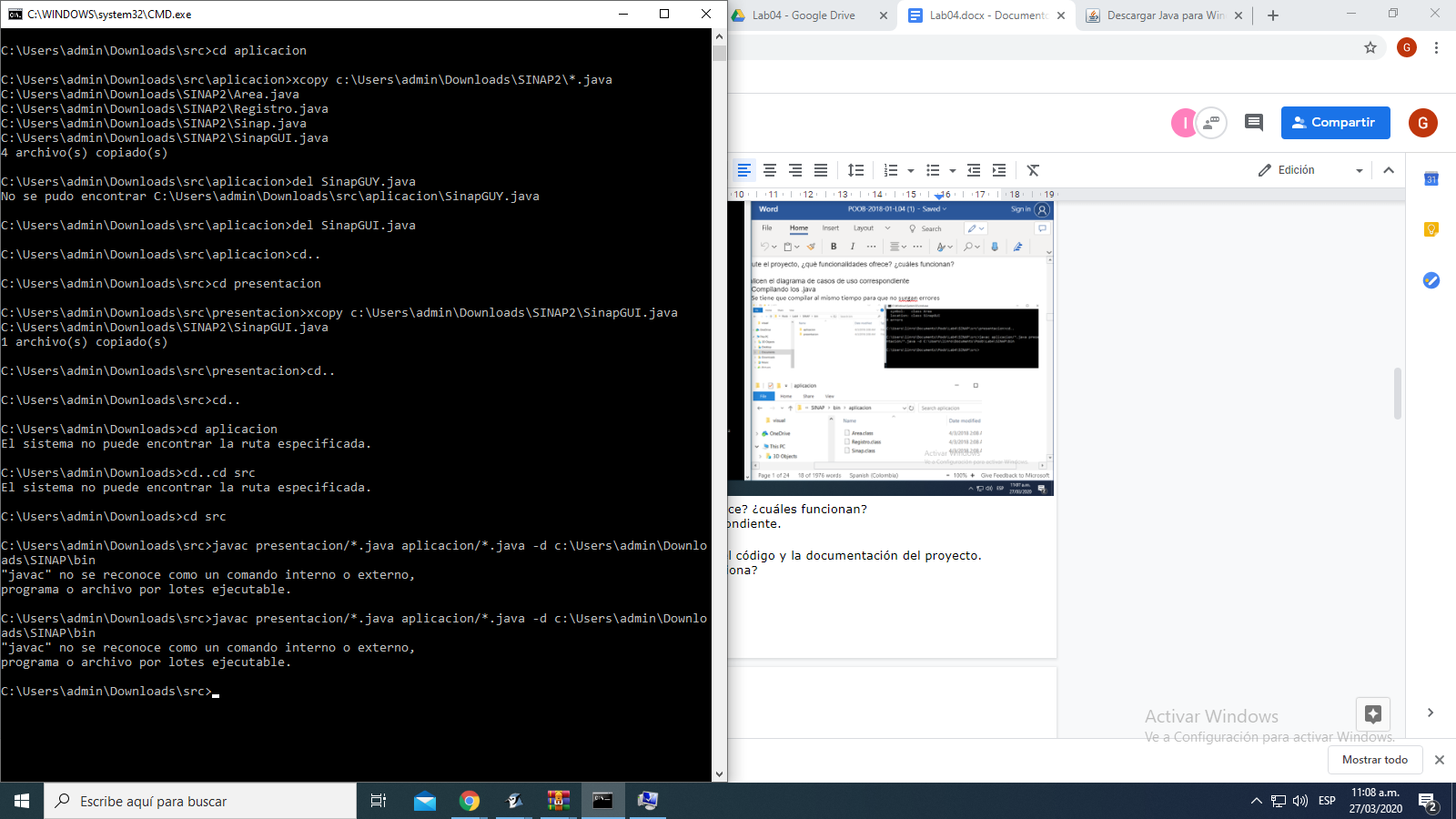
1. Prepare los directorios necesarios para ejecutar el proyecto. ¿qué estructura debe tener?

¿qué instrucciones debe dar para ejecutarlo?

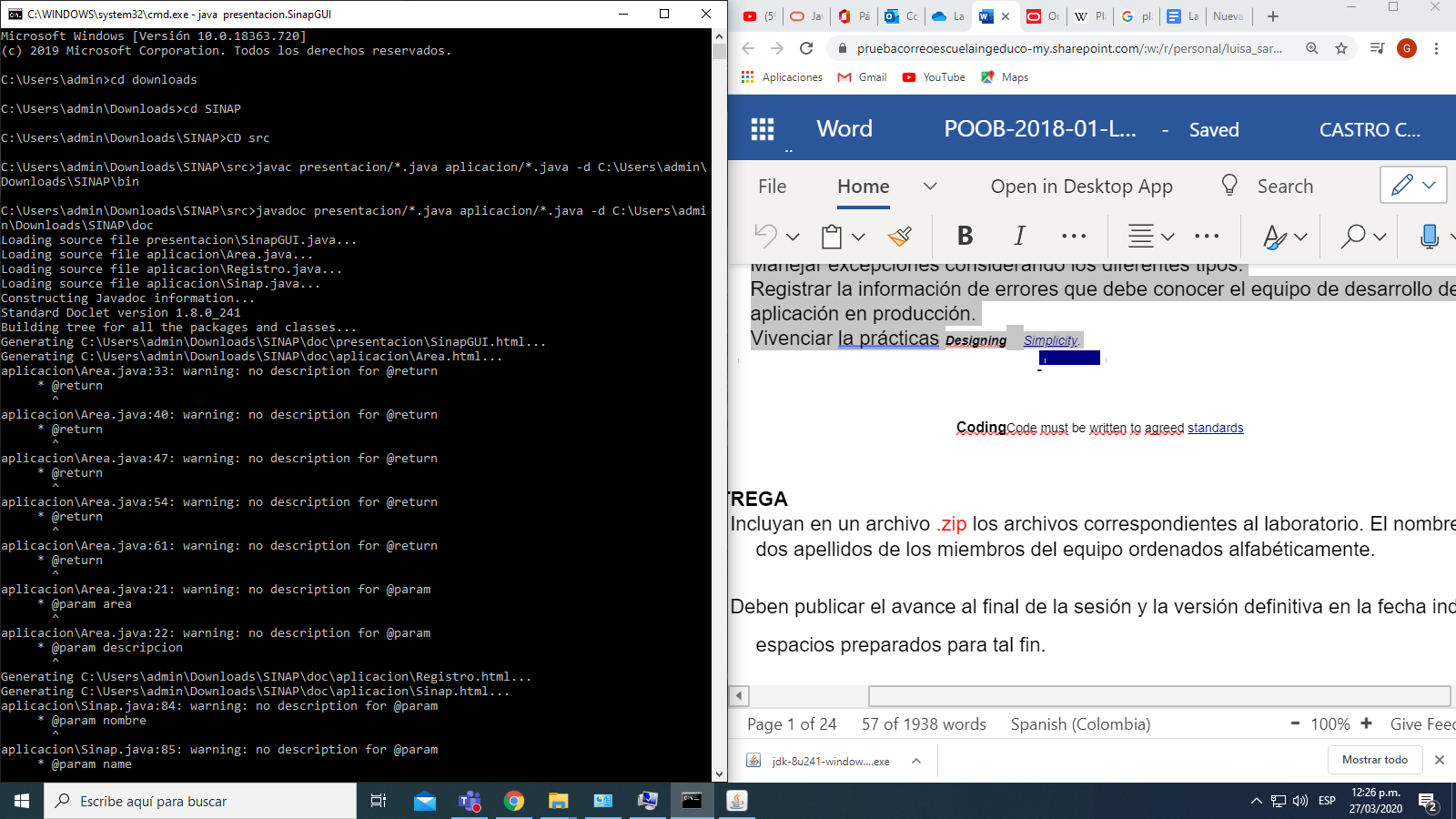
**R:/**Se debe tener las dos carpetas de cada respectivo paquete, en las respectivas carpetas bin y sus carpetas doc y src.

INSTRUCCIONES

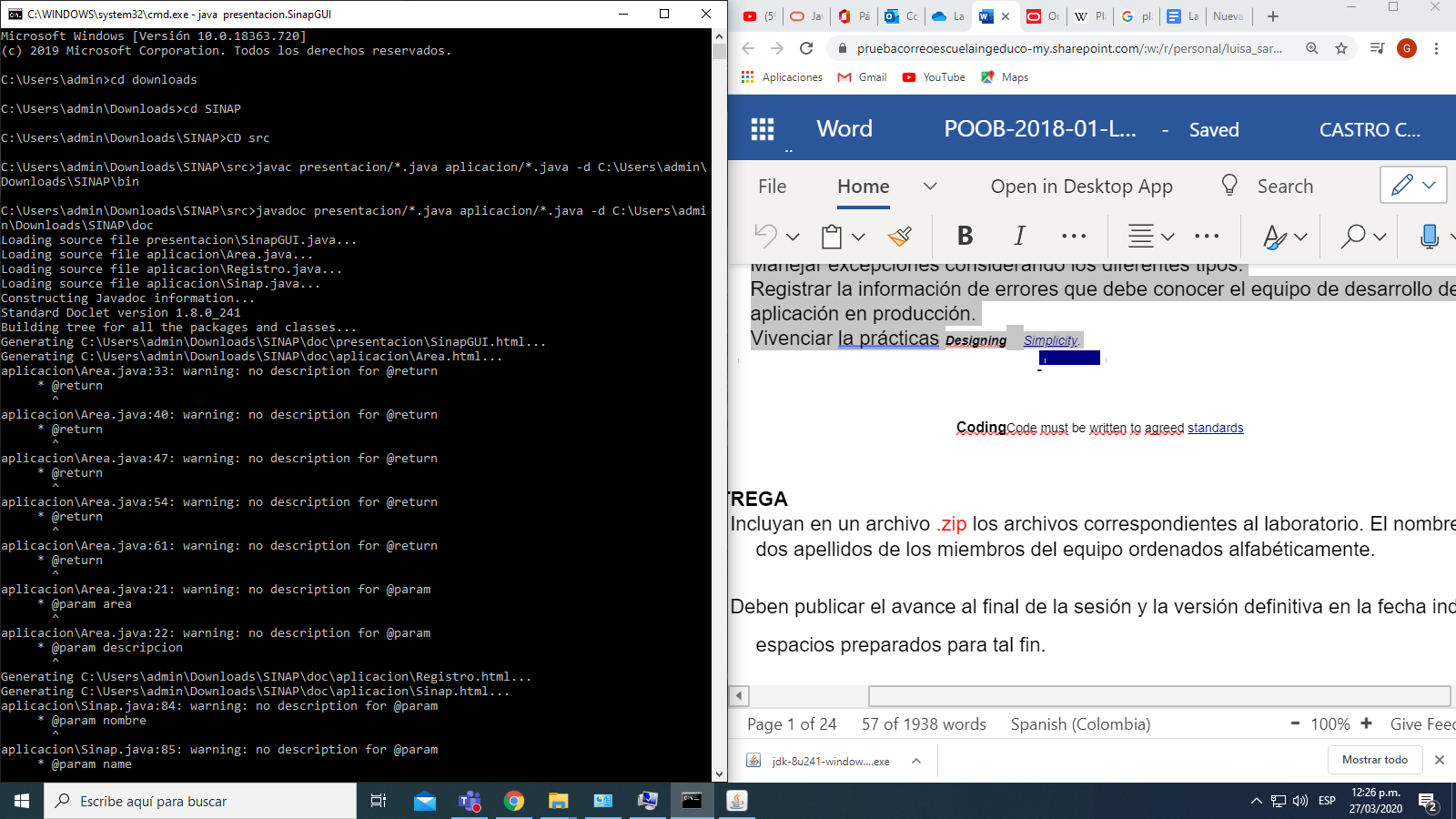
1. debemos organizar por carpetas src bin y doc el documento, en src se deben copiar los archivos java
2. luego debemos compilar el programa en la carpeta bin.



1. luego se tiene que documentar los archivos java

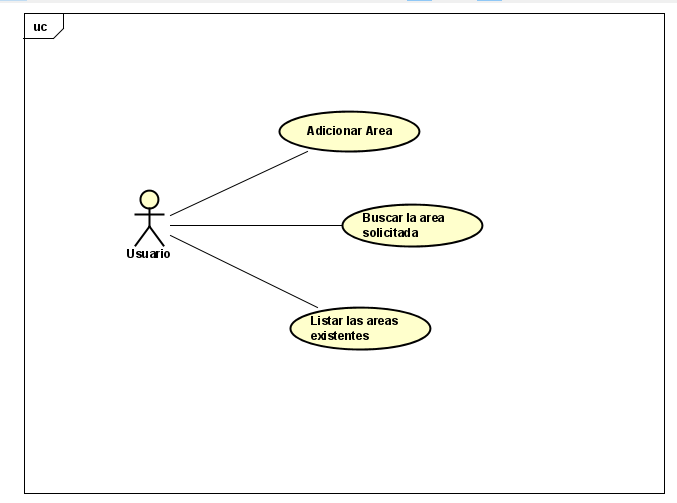


y por último debemos ejecutar el programa para mirar cómo funciona



1. Ejecute el proyecto, ¿qué funcionalidades ofrece? ¿cuáles funcionan? Realicen el diagrama de casos de uso correspondiente.

**R:/**ofrece agregar,listar y buscar un área, funcionan listar y agregar.



1. ¿De dónde salen las áreas iniciales? Revisen el código y la documentación del proyecto.

¿Qué clase pide que se adicionen? ¿Qué clase los adiciona?

**R:/**Las áreas iniciales se encuentra del método adicionarCinco en la clase Sinap, este método lo pide la clase SinapGUI en el constructor y la clase Sinap adiciona las nuevas zonas

**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

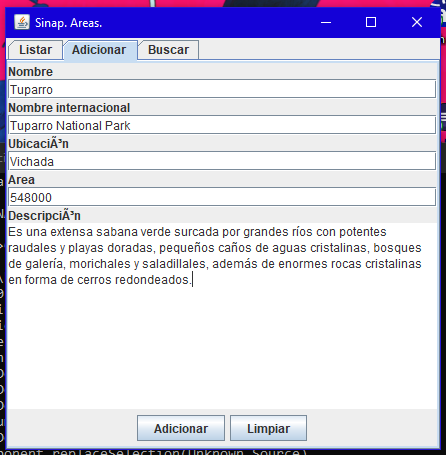
Tuparro

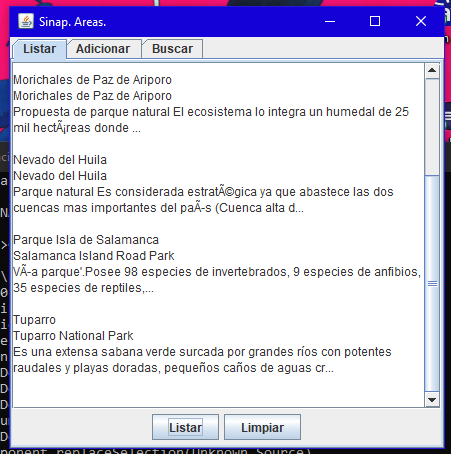
Tuparro National Park Vichada

548.000

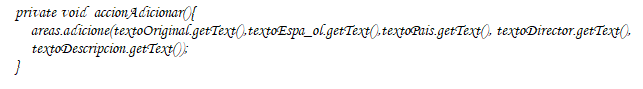
Es una extensa sabana verde surcada por grandes ríos con potentes raudales y playas doradas, pequeños caños de aguas cristalinas, bosques de galería, morichales y saladillales, además de enormes rocas cristalinas en forma de cerros redondeados.

¿Qué ocurre? ¿Cómo lo comprueban? Capturen la pantalla. ¿Es adecuado este comportamiento?





1. 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? ¿Qué método en la capa de aplicación?

R/: 

1. Realicen ingeniería reversa para la capa de aplicación para **adicionar**. Capturen los resultados de las pruebas de unidad.
2. 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? ¿Qué método en la capa de aplicación?

R/:   


1. Realicen ingeniería reversa para la capa de aplicación para **listar**. Capturen los resultados de las pruebas de unidad.
2. Propongan y ejecuten una prueba de aceptación.

**Adicionar un área. ¿Y si no da un nombre?** [En lab04.doc, SINAP.asta y \*.java]

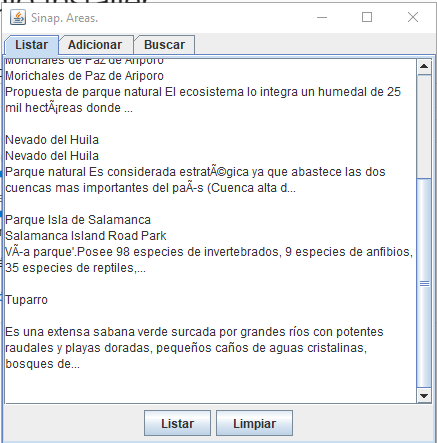
(NO OLVIDEN BDD – MDD)

El objetivo es perfeccionar la funcionalidad de adicionar un área.

1. Adicionen el área Tuparro sin nombre internacional. ¿Qué ocurre? ¿Cómo lo comprueban? Capturen la pantalla. ¿Es adecuado este comportamiento?

RTA// deja agregar el área sin ese item, listando las areas

no ya que no debería dejarme agregar un area en el programa porque falta esto y queda incompleto.



1. 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[1](#_heading=h.gjdgxs). Para esto lo primero que debemos hacer es crear la nueva clase SINAPExcepcion considerando este primer mensaje.
2. Analicen el diseño realizado. ¿Qué método debería lanzar la excepción? ¿Qué métodos deberían propagarla? ¿Qué método debería atenderla? Explique claramente.

RTA// ¿Qué método debería lanzar la excepción?

Lo debería lanzar adicione() en Sinap

* ¿Qué métodos deberían propagarla?

Lo debería propagar adicione() en Sinap

* ¿Qué método debería atenderla? Explique claramente.

accionAdicionar() de SinapGUI

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

RTA//

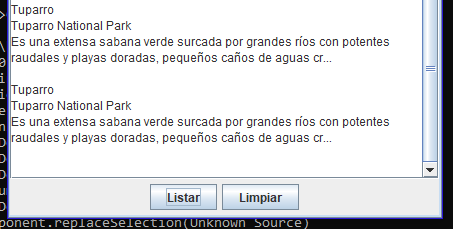
1. Ejecuten nuevamente la aplicación con el caso de prueba propuesto en 1., ¿Qué sucede ahora? Capture la pantalla.

**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? ¿Cómo lo comprueban? Capturen la pantalla. ¿Es adecuado este comportamiento?

R/:  
 

1. Analicen el diseño realizado. ¿Qué método debería lanzar la excepción? ¿Qué métodos deberían propagarla? ¿Qué método debería atenderla? Explique claramente.
2. Construya la solución propuesta. Capturen los resultados de las pruebas.
3. Ejecuten nuevamente la aplicación con el caso de prueba propuesto en 1., ¿Qué sucede ahora? Capture la pantalla.

**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.[2](#_heading=h.30j0zll)
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)

 Consulten un área especial que inicie con T. ¿Qué sucede? ¿Qué creen que pasó?

Capturen el resultado. ¿Quién debe conocer y quien NO debe conocer esta información?

 Exploren el método registre de la clase Registro ¿Qué servicio presta?

 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.

 Ejecuten nuevamente la aplicación con el caso propuesto en 1. ¿Qué mensaje salió en pantalla? ¿La aplicación termina? ¿Qué información tiene el archivo de errores?

 ¿Es adecuado que la aplicación continúe su ejecución después de sufrir un incidente como este? ¿de qué dependería continuar o parar?



1 Para presentar los mensajes de error al usuario use el méodo de clase de JOptionPane public static void **showMessageDialog**(Component parentComponent,

Object message, String title, int messageType)

throws HeadlessException

Con componente padre:este mensaje: la cadena correspondiente al mensaje de error de la excepcion correspondiente, titulo: ERROR y tipo de mensaje: JOptionPane.ERROR\_MESSAGE

2Robustez o solidez. Se refiere a la capacidad del software de defenderse de las acciones anormales que llevan al sistema a un estado no deseado o por lo menos no previsto, causando un comportamiento inesperado, indeseado y posiblemente erróneo

 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.

 Ejecuten nuevamente la aplicación con el caso propuesto en 1. ¿Qué mensaje salió en pantalla? ¿La aplicación termina? ¿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? ¿Qué método es responsable en la capa de aplicación?

1. Realicen ingeniería reversa para la capa de aplicación para **buscar**. Capturen los resultados de las pruebas. Deben fallar.
2. ¿Cuál es el error? Soluciónenlo. Capturen los resultados de las pruebas.
3. 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)

RTA// 15 horas por hombre

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

RTA// está incompleta ya que algunos métodos no entendemos porque al correrlo no sirve y no funciona el programa

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

RTA// la de pares ya que si no hubieramos trabajado en pares no hubieramos logrado hacer nada con el tema de trabajo en casa.

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

RTA//obtener las pruebas correctamente en el primer punto, porque aprendimos a usar las excepciones.

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

RTA// el usar la consola ya que no corren los programas que hicimos, no lo logramos solucionar.

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

RTA// trabajar en pareja a pesar de todo lo sucedido, dedicar más tiempo para el desarrollo de esto.