**PROGRAMACIÓN ORIENTADA A OBJETOS**

**Excepciones**

**2025-1**

**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*](http://www.extremeprogramming.org/rules/simple.html)*.*

[Refactor](http://www.extremeprogramming.org/rules/refactor.html)  whenever and wherever possible.



**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.

**Unidades**

# EN BLUEJ

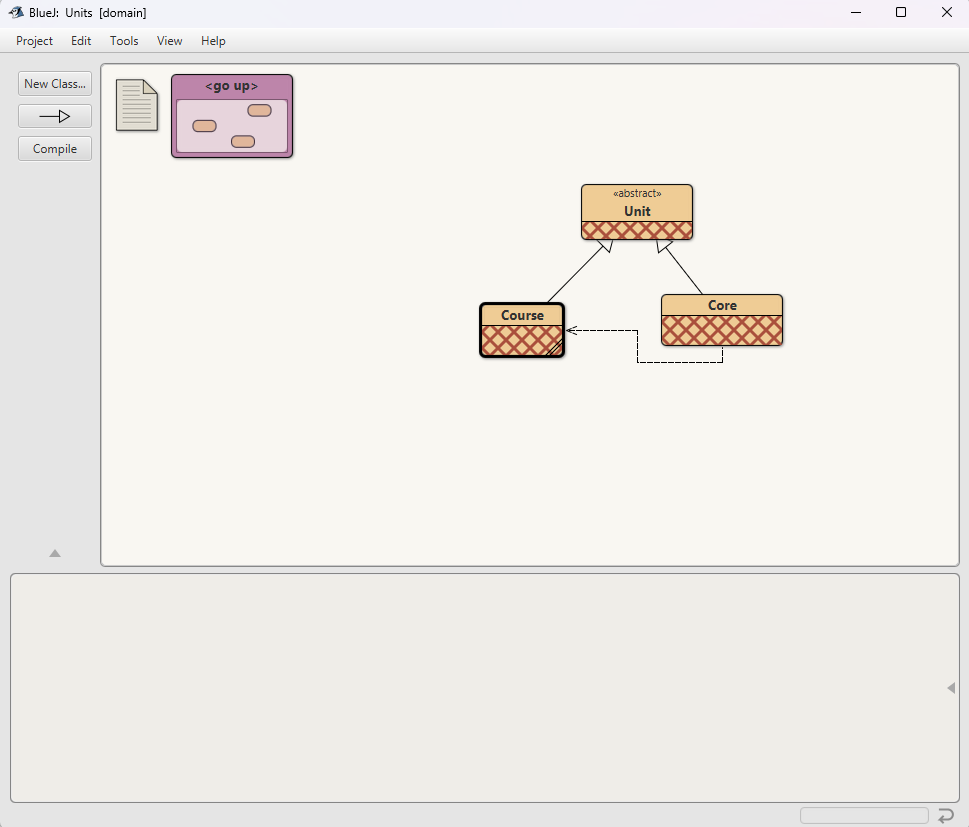
**PRACTICANDO MDD y BDD con EXCEPCIONES**

**[En lab04.doc, Plan15.asta y BlueJ units]**

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

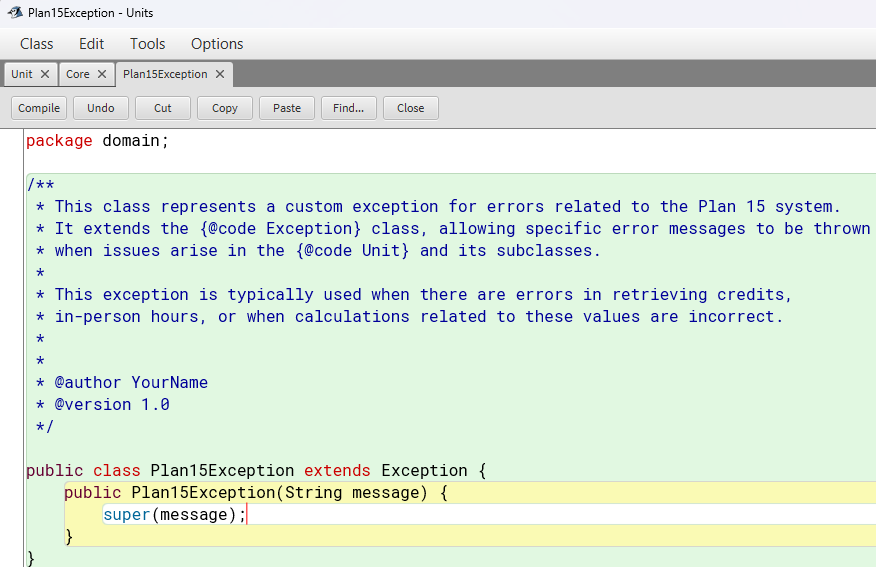
1. En su directorio descarguen los archivos contenidos en units.zip revisen el contenido y estudien el diseño estructural de la aplicación (únicamente la zona en azul).

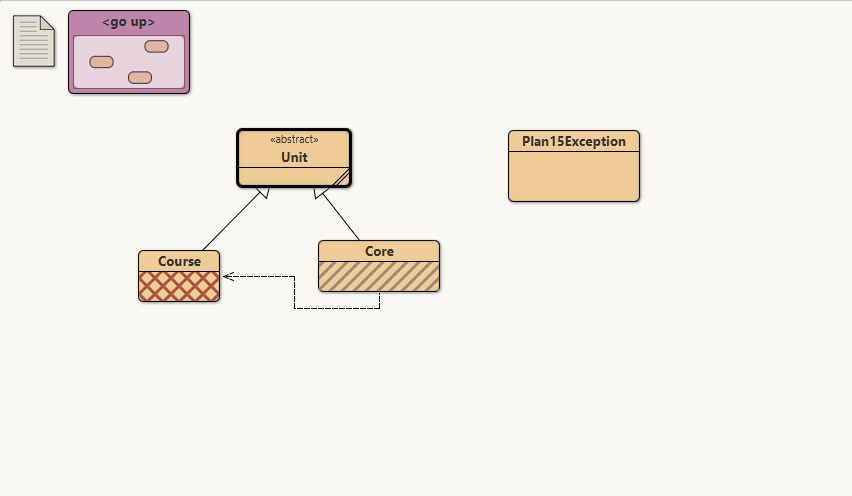
* Dentro del directorio de units.zip podemos encontrar package (BlueJ), doc y domain. Ya en sí, cuando abrimos el BlueJ podemos observar que se tiene una clase abstracta llamada Unit, y dos clases llamadas Core y Course que son heredadas de Unit, según lo que se puede ver en BlueJ.



1. Expliquen por qué el proyecto no compila. Realicen las adiciones necesarias para lograrlo.

* El proyecto no compila porque no se encuentra definida la acción de “Plan15Exception” que es lo que nos genera el error. Para esto simplemente creamos una clase llamada Plan15Exception y hacemos la acción necesaria para evitar esto.

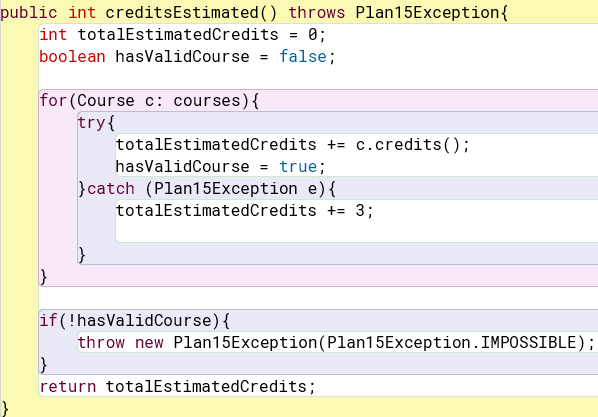




1. Dado el diseño y las pruebas documenten y codifiquen el método credits().



1. Dada la documentación y el diseño codifiquen y prueben el método creditsEstimated().



1. Documenten, diseñen, codifiquen y prueben el método inPersonEstimated().

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

**Plan15**

# EN CONSOLA

El objetivo de esta aplicación es mantener un catálogo de las unidades de aprendizaje correspondientes al Plan15. En este plan se ofrecen diferentes tipos de disfraces: básicos y completos.

**Conociendo el proyecto Plan15 [En lab04.doc]**

**No olviden respetar los directorios bin docs src**

1. En su directorio descarguen los archivos contenidos en plan15.zip, revisen el contenido.

¿Cuántos archivos se tienen? ¿Cómo están organizados? ¿Cómo deberían estar organizados?

* Al momento de descargar el archivo de plan15.zip dentro de este se encuentran tres archivos los cuales son:

Tabla

El contenido generado por IA puede ser incorrecto.

* Estos deben estar organizados como src,bin y docs. Además de esto, se deben crear los paquetes de “presentation” y “test”.

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?

* En este momento del laboratorio se tiene un total de solo un paquete nada más que se llama “domain”.
* Se tiene un total de 4 clases, Plan15, Unit, Core y course.
* En cuanto a la organización: La clase Plan15 tiene una relación de asociación con Unit, La clase Unit está relacionada con la clase Course, y la clase Core también está relacionada con la clase Course, La clase Core también tiene una relación de agregación con Course.
* Creemos que la **clase ejecutiva** es Plan15 ya que contiene métodos que sugieren la gestión y manipulación de datos del plan, como addSome(), consult(), addCourse(), addCore(), select(), data(), y search().

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

¿qué clases deben tener? ¿dónde están esas clases? ¿qué instrucciones debe dar para ejecutarlo?

* Se debe tener la estructura de src, bin, docs. Las 4 clases que mencionamos anteriormente se deberían encontrar en el paquete de domain que se encuentra en src, donde se encuentra todos los .java.

1. Ejecute el proyecto, ¿qué funcionalidades ofrece? ¿cuáles funcionan?

* Existen dos formas de ejecutar el programa, la primera es de forma manual desde BlueJ, entrando a Plan15GUI y se le da como si crearamos un objeto, con la pequeña diferencia que le damos es al void main(String[] args).
* La otra forma de poder ejecutar el programa es desde la consola, para esto se ejecuta el siguiente comando en la consola:

**java -cp bin src.presentation.Plan15GUI**

* Ofrece las funcionalidades de Listar, adicionar y buscar, funcionan las opciones de Listar y Limpiar.

Interfaz de usuario gráfica, Aplicación, Word

El contenido generado por IA puede ser incorrecto. Texto

El contenido generado por IA puede ser incorrecto.

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

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

* Las unidades iniciales se crean e insertan directamente en el método addSome() de la clase Plan15.
* La clase que "pide" o, mejor dicho, la clase que contiene la lógica para agregar las unidades iniciales es la clase Plan15. Específicamente, el método addSome() dentro de Plan15 es el responsable de crear e insertar los cursos y el núcleo inicial.
* La clase que realmente "adiciona" o agrega las unidades a las estructuras de datos (ArrayList y TreeMap) es también la clase Plan15, pero a través de sus métodos addCourse() y addCore().
  + El método addCourse() crea instancias de la clase Course y las agrega tanto al ArrayList<Unit> units como al TreeMap<String, Course> courses.
  + Texto, Aplicación

    El contenido generado por IA puede ser incorrecto.El método addCore() crea instancias de la clase Core y las agrega al ArrayList<Unit> units. Adicionalmente este método agrega objetos de la clase Course a la clase Core.

**Adicionar y listar. Todo OK.**  [En lab04.doc, Plan15.astay \*.java]

(NO OLVIDEN BDD - MDD)

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

1. Adicionen un nuevo curso y un nuevo núcleo Curso

DOSW Desarrollo y Operaciones Software 4 y 4 *Core*

NFPE Nucleo de formación específica 100

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

* Para hacer ingeniería reversa se hace lo siguiente:

Texto

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

* El comportamiento es adecuado como nos podemos dar cuenta ya que nos da DOSW, el número de créditos y además de esto de donde pertenece.

1. Revisen el código asociado a **adicionar** en la capa de presentación y la capa de dominio. ¿Qué método es responsable en la capa de presentación? ¿Qué método en la capa de dominio?

* El método responsable en la capa de presentación es el de **actionAdd()** de la clase Plan15GUI.
* En el caso del método de la capa de dominio encontramos dos, que son para courses y para core. Para courses es **addCourse()** y para core es **addCore().**

1. Realicen ingeniería reversa para la capa de dominio para **adicionar**.

Texto

El contenido generado por IA puede ser incorrecto.Capturen los resultados de las pruebas de unidad.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

1. Revisen el código asociado a **listar** en la capa de presentación y la capa de dominio. ¿Qué método es responsable en la capa de presentación? ¿Qué método en la capa de dominio?

* En la capa de presentación el método responsable es **actionList().**
* En la capa de dominio el método responsable es: **toString().**

1. Realicen ingeniería reversa para la capa de dominio para **listar**.

Capturen los resultados de las pruebas de unidad.

Texto

El contenido generado por IA puede ser incorrecto.Tabla

El contenido generado por IA puede ser incorrecto.

1. Propongan y ejecuten una prueba de aceptación.

* Nosotros proponemos que al momento de estar adicionando, en el caso de que se encuentre un espacio en blanco salga un aviso que diga que “Todos los campos son obligatorios”.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

**Adicionar una unidad. Funcionalidad robusto**  [En lab04.doc, Plan15.astay \*.java]

(NO OLVIDEN BDD – MDD)

El objetivo es perfeccionar la funcionalidad de adicionar un curso para hacerla más robusta.

**Para cada uno de los siguientes casos realice los pasos del 1 al 4.**

* 1. **¿Y si el nombre de la unidad no existe?**
  2. **¿Y si los valores enteros no son enteros?**
  3. **¿Y si el portage no está entre 0 y 100?**
  4. **Proponga una nueva condición**

1. Propongan una prueba de aceptación que genere el fallo.
2. Analicen el diseño realizado. Para hacer el software robusto: ¿Qué método debería lanzar la excepción? ¿Qué métodos deberían propagarla? ¿Qué método debería atenderla? Explique claramente.
3. Construya la solución propuesta. Capture los resultados de las pruebas de unidad.
4. Ejecuten nuevamente la aplicación con el caso de aceptación propuesto en 1. ¿Qué sucede ahora? Capture la pantalla.

Apartado a) **Y si el nombre de la unidad no existe**

1. Lo que sucede en este momento es lo siguiente, se añade correctamente, sin embargo, esto es incorrecto ya que no verifica si es lo que el usuario quiere en caso de un error.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

1. El método addCore es el responsable de validar si los cursos asociados al núcleo existen en el sistema. Si alguno de los cursos no se encuentra registrado, debe lanzar una excepción con el mensaje correspondiente, como se hizo en el código con la verificación del curso inexistente. Esta excepción debe ser propagada hacia los métodos que gestionan la lógica de la aplicación, como los controladores o la interfaz gráfica.
2. Nuestra soluci´´on:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

4.Ahora en BlueJ:

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Apartado b) **¿Y si los valores enteros no son enteros?**

1. Simplemente basta con ingresar en un campo donde esperamos un numero entero una letra.

Texto

El contenido generado por IA puede ser incorrecto.

1. El método addCourse debe ser el encargado de lanzar una excepción Plan15Exception cuando no se puedan convertir los textos a números, ya que es ahí donde ocurre la conversión crítica. Métodos como addSome, que llaman a addCourse, deben capturar o propagar esa excepción según el diseño. En este caso, addSome ya la atiende usando un bloque try-catch, por lo que está correctamente diseñado.
2. La solución propuesta:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

4.Ahora en BlueJ, aparece lo que nosotros definimos anteriormente:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Apartado c) **¿Y si el portage no está entre 0 y 100?**

**1.**En este momento ocurre esto, basta con poner 101 y el programa falla:

Texto

El contenido generado por IA puede ser incorrecto.

2. Para que el software sea robusto frente a porcentajes inválidos en los núcleos (Core), **el constructor de la clase Core** debe ser quien **lance la excepción** (Plan15Exception) cuando el porcentaje no esté entre 0 y 100, ya que es el responsable directo de validar ese atributo. Luego, el método addCore de la clase Plan15 debe **propagar la excepción**, permitiendo que cualquier error en la construcción de núcleos se notifique correctamente.

Texto

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.3.La solución propuesta:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.4.Con nuestro ajuste en BlueJ:

Apartado d) **Proponga una nueva condición**

**1.**Lo que proponemos es que al momento de completar las siglas estas tengan que estar estrictamente en mayúsculas, dado que es mejor para evitar mayores riesgos, además que es más fácil identificar las materias en este momento acepta en minisculas y mayusculas.

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

[En Plan15.asta, Plan15.log, lab04.java y \*.java]

(NO OLVIDEN BDD - MDD)

1. Consulten una unidad completo que inicie con I. ¿Qué sucede? ¿Qué creen que pasó?

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

* El método search(String prefix) busca unidades que comiencen con ese prefijo, pero si no encuentra ninguna, simplemente no muestra nada. **El usuario** debe saber que no hay unidades con ese prefijo, de forma clara (por ejemplo, con un mensaje). **La clase Log** debe registrar el intento fallido como evento.
* **La clase Plan15** NO necesita mostrar detalles del error, solo delegar el mensaje y el registro.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

1. Exploren el método record de la clase Log ¿Qué servicio presta?

* El método record de la clase log presta el servicio de **registrar eventos o errores importantes** en un archivo de texto llamado Plan15.log. Este presta varios servicios como:
  + Crear o reutilizar un archivo de log (Plan15.log).
  + Formatear los mensajes para que sean legibles (con SimpleFormatter).
  + Guardar información detallada sobre las excepciones: tipo, mensaje.

1. 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.

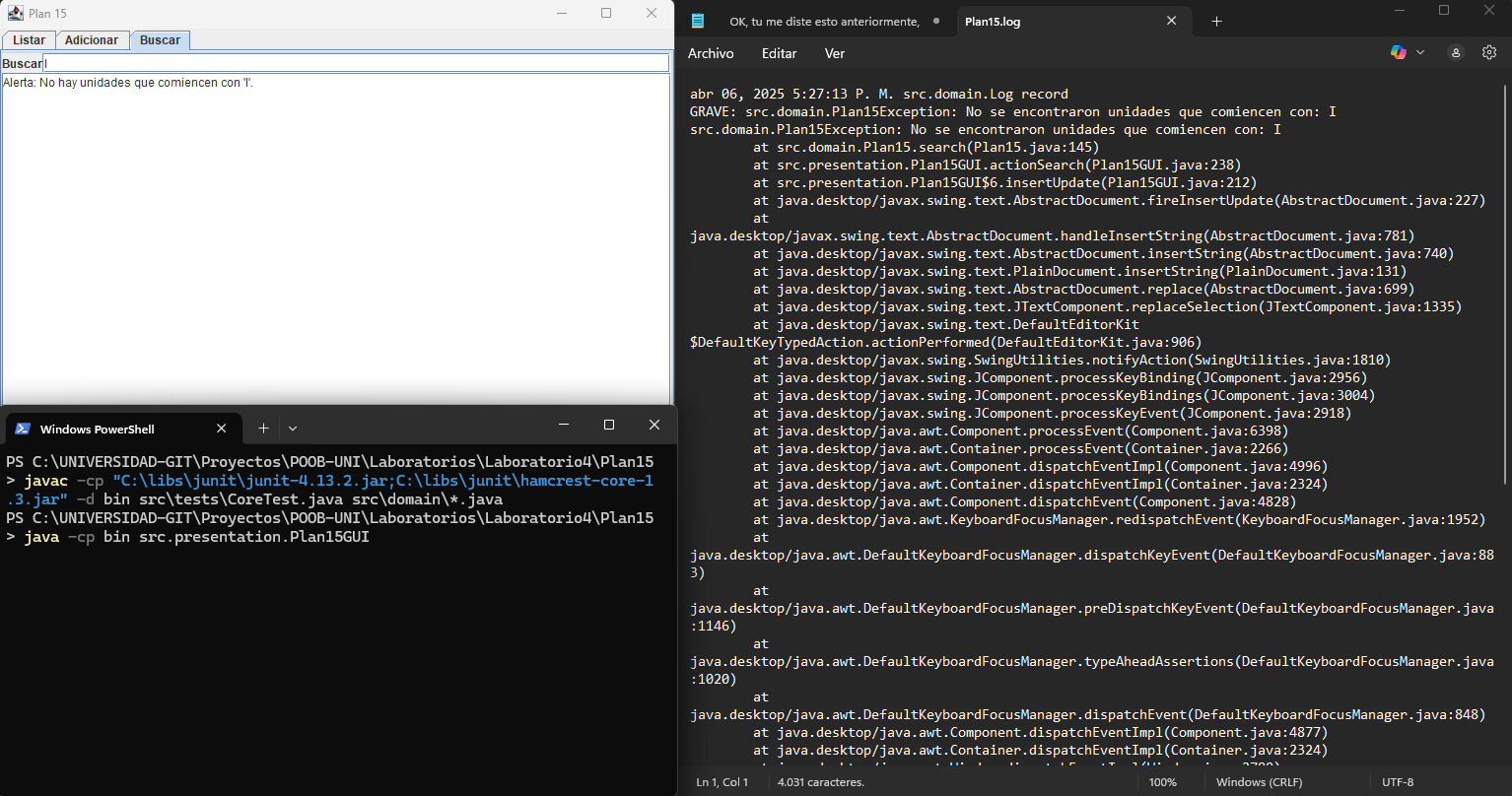
* Para manejar el caso en que no se encuentren unidades al consultar por un patrón, se modificó el método search para que lance una excepción Plan15Exception con un mensaje claro. Esta excepción se captura y se registra en el archivo Plan15.log usando la clase Log, y además se muestra un mensaje amigable al usuario sin detener la ejecución del programa.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

1. 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?

* Al momento de ejecutar de nuevo la aplicación con el caso propuesto en 1 nos damos cuenta de que lo que sucede es que aparece un pequeño error, sin interrumpir el programa.
* En pantalla no sale el mensaje: “Alerta: No hay unidades que comiencen con I”.
* El archivo de errores contiene la fecha de cuando se genero el error, el mismo mensaje que dijimos anteriormente, y un montón de errores que son normales.



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

* En este ejercicio sí es adecuado continuar, porque el error de no encontrar una unidad con cierto prefijo (como "I") no afecta el funcionamiento general del sistema ni corrompe los datos. Solo significa que no hay resultados para esa búsqueda. La decisión de continuar o detener dependería de si el error impide seguir usando el sistema con normalidad o si daña datos importantes, lo cual aquí no ocurre.

1. Modifiquen la aplicación para garantizar que **SIEMPRE** que haya un error se maneje de forma adecuada. ¿Cuál fue la solución implementada?

Para garantizar que todos los errores se manejen de forma adecuada, se implementó un sistema que combina el uso de excepciones personalizadas (Plan15Exception), ventanas emergentes para notificar al usuario (JOptionPane) y un registro centralizado en archivo mediante la clase Log. De esta forma, cada vez que ocurre un error, se informa al usuario de manera clara y, al mismo tiempo, se guarda el detalle en un log para facilitar el diagnóstico y mantenimiento del sistema. Esta solución asegura una experiencia más robusta y controlada ante cualquier fallo.

Interfaz de usuario gráfica, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

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

[En Plan15.asta, Plan15.log, lab04.java y \*.java]

(NO OLVIDEN BDD - MDD)

1. Revisen el código asociado a **buscar** en la capa de presentación y la capa de dominio. ¿Qué método es responsable en la capa de presentación? ¿Qué método es responsable en la capa de dominio?

* En la capa de presentación el método responsable es actionSearch().
* En la capa de dominio el método responsable es search(String prefix)

1. Realicen ingeniería reversa para la capa de dominio para **buscar**. Capturen los resultados de las pruebas. Deben fallar.

Texto

El contenido generado por IA puede ser incorrecto.

* Fallan, no sabemos la razón, pero no aparece el error clásico en consola, se intento con las 4 pruebas definidas al inicio, aun así falla.

1. ¿Cuál es el error? Soluciónenlo. Capturen los resultados de las pruebas.

* El error es el siguiente:

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

* Este error lo que nos generaría es un IndexOutOfBoundsException. Para solucionarlo simplemente le quitamos el =.

Solucionado:

Texto

El contenido generado por IA puede ser incorrecto.

1. Ejecuten la aplicación nuevamente con el caso propuesto. ¿Qué tenemos en pantalla? ¿Qué información tiene el archivo de errores?

* Con nuestro caso propuesto lo que sale en pantalla es que no se encontraron las unidades I, que fue el caso que añadimos para hacer ingeniería inversa, además de esto, sale la fecha y la hora del error.

1. Refactorice la funcionalidad para que sea más amable con el usuario. ¿Cuál es la propuesta? ¿Cómo la implementa?

* Nuestra se centra en mejorar la funcionalidad de búsqueda para que sea más intuitiva y proporcione retroalimentación clara al usuario, sin interrumpir su experiencia con mensajes de error intrusivos.
* La implementaríamos de la siguiente manera: Mejora de la funcionalidad de búsqueda, añadimos una etiqueta de estado que proporciona retroalimentación visual inmediata al usuario: muestra mensajes en azul cuando hay instrucciones o resultados positivos, y en rojo cuando no se encuentran coincidencias. Eliminé los molestos diálogos de error, reemplazándolos por mensajes integrados directamente en la interfaz que informan claramente al usuario sobre el estado de su búsqueda sin interrumpir su flujo de trabajo.
* Para usarla eliminar los /\* \*/ en el Plan15GUI.

Interfaz de usuario gráfica, Aplicación, Word

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Aplicación, Word

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

## RETROSPECTIVA

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

* Sánchez/20
* Ruiz/20

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

* Terminado, lo hicimos a tiempo.

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

* La de simplicidad ya que procuramos en mantener todo muy simple y sutil.

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

* Poder usar la consola correctamente sin necesidad del BlueJ, solo en caso de que se necesitara lo usábamos.

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

* Tal vez los errores que salían en consola ya que eran extraños para nosotros.

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

* Consideramos que hicimos todo bien, tal vez, a la próxima leer un poco más sobre consola.

1. ¿Qué referencias usaron? ¿Cuál fue la más útil? Incluyan citas con estándares adecuados.

* Más que nada usamos la API de Java para poder guiarnos:

Oracle. (s.f.). *Documentación de la API de Java*. Oracle. Recuperado de [https://docs.oracle.com/en/java/javase/17/docs/api/](<https://docs.oracle.com/en/java/javase/17/docs/api/>)