**LABORATORIO NRO. 4**

**JEISSON CASALLAS Y CAMILO MURCIA**

**Disfaces**

EN BLUEJ

**PRACTICANDO MDD y BDD con EXCEPCIONES**

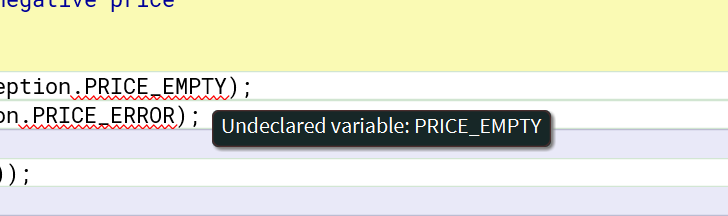
[En lab04.doc, CostumeShop.asta y BlueJ costumes]

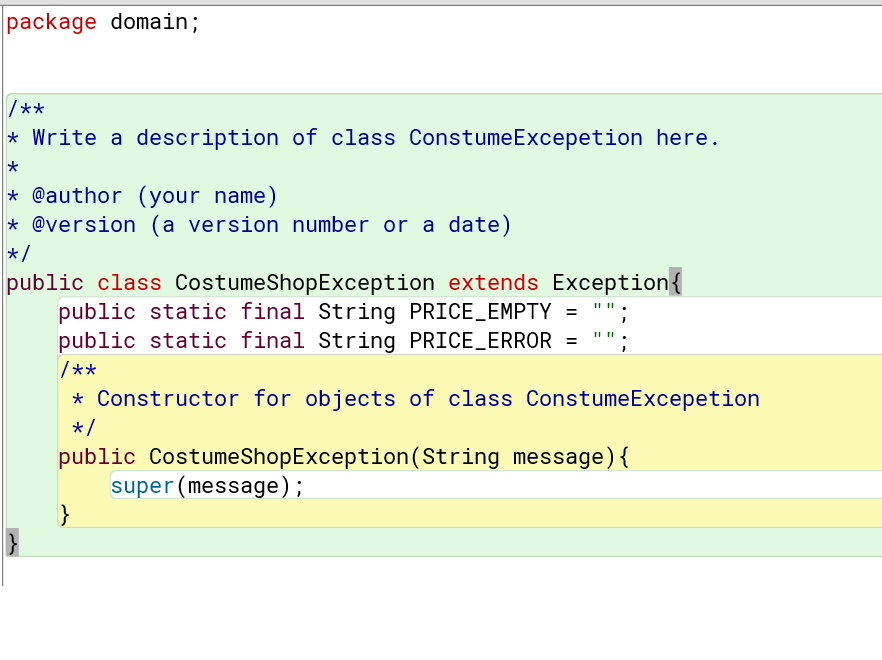
En este punto vamos a aprender a diseñar, codificar y probar usando excepciones. Para esto se van a trabajar algunos métodos de la clase Costume

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

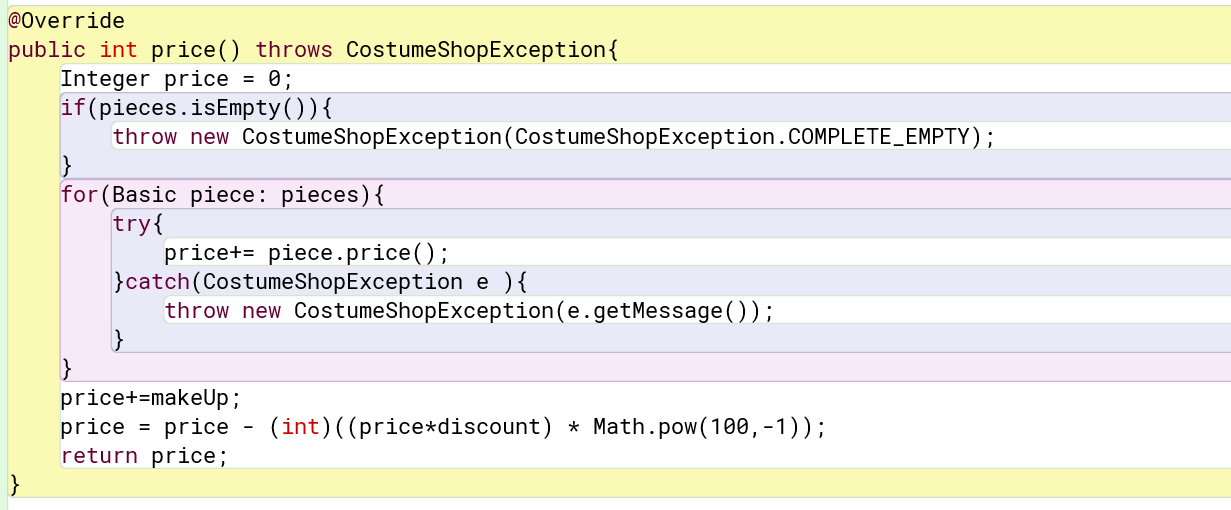
**2. Expliquen por qué el proyecto no compila. Realicen las adiciones necesarias para lograrlo.**

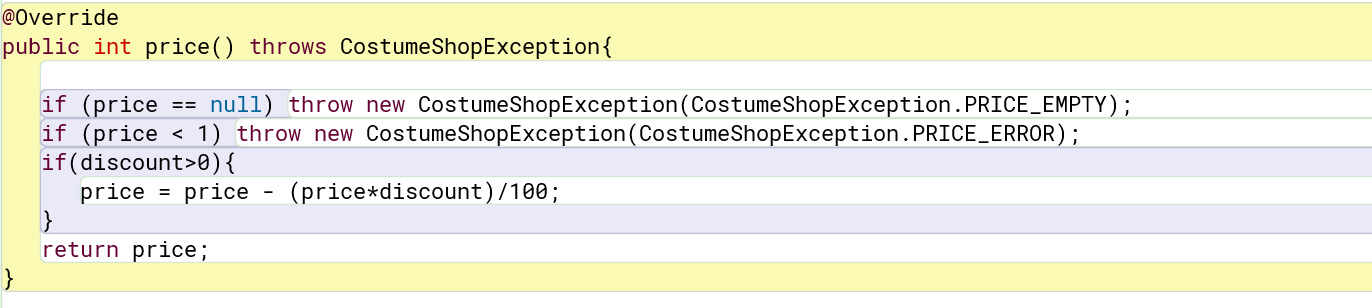
El proyecto no compila debido a que no hay aun una clase de excepción, y para que funcione debemos crear esta clase para así darle seguimiento a los errores que arrojan las clases en el proyecto.

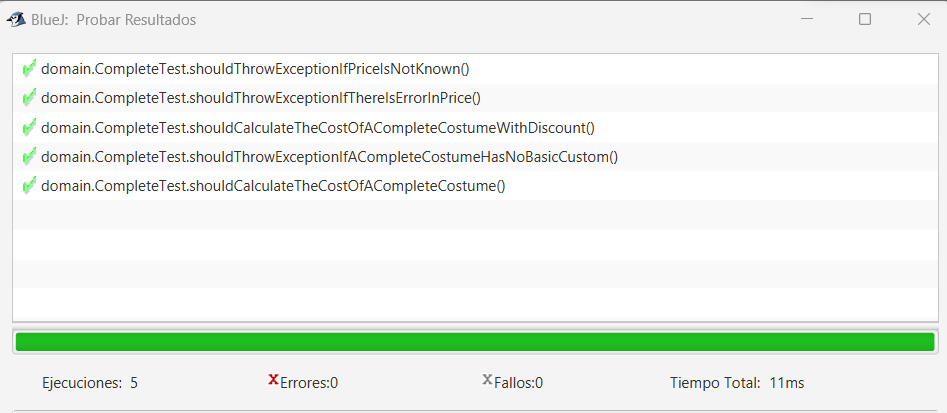




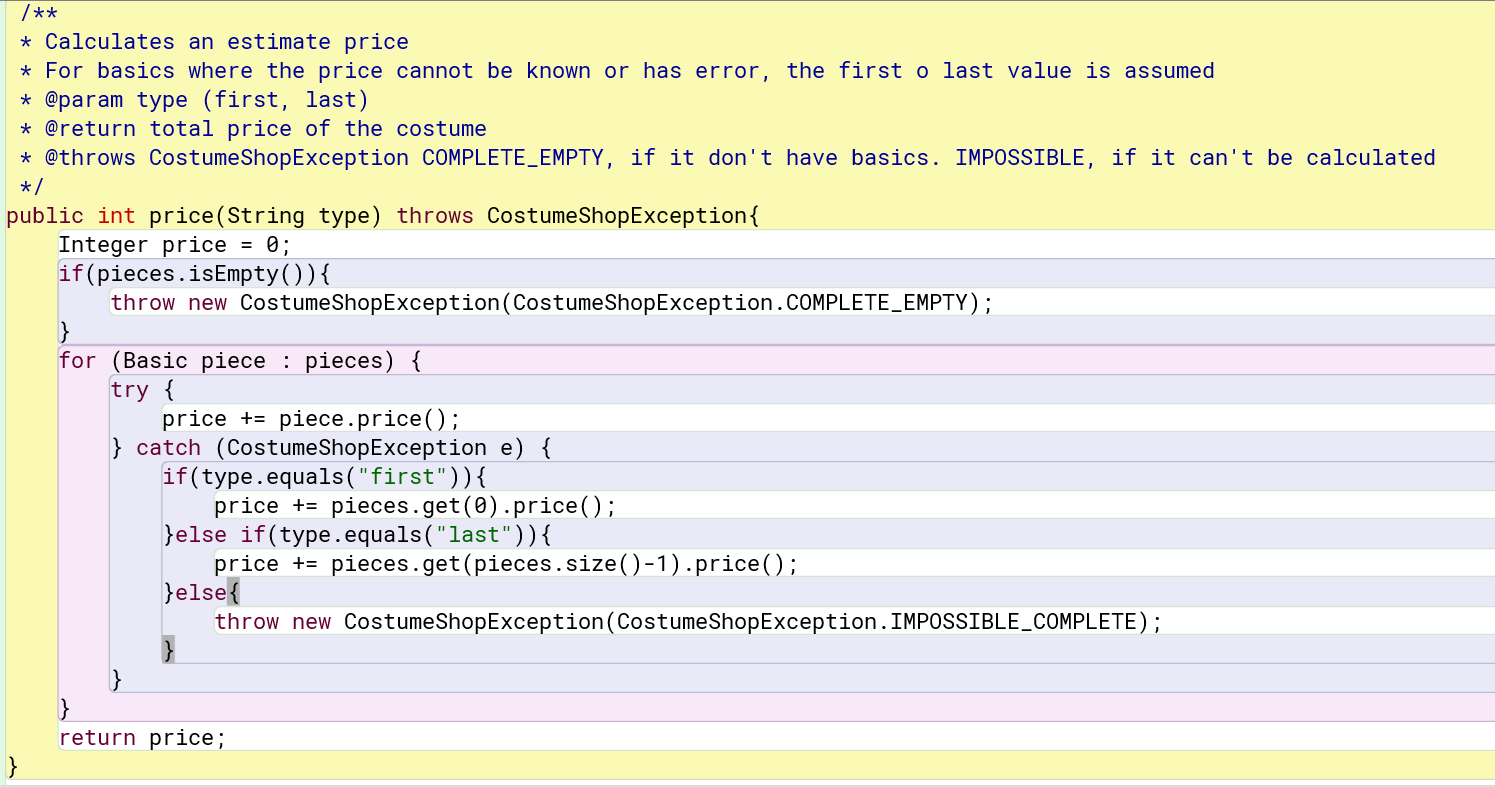
**3. Dado el diseño y las pruebas documenten y codifiquen el método price().**

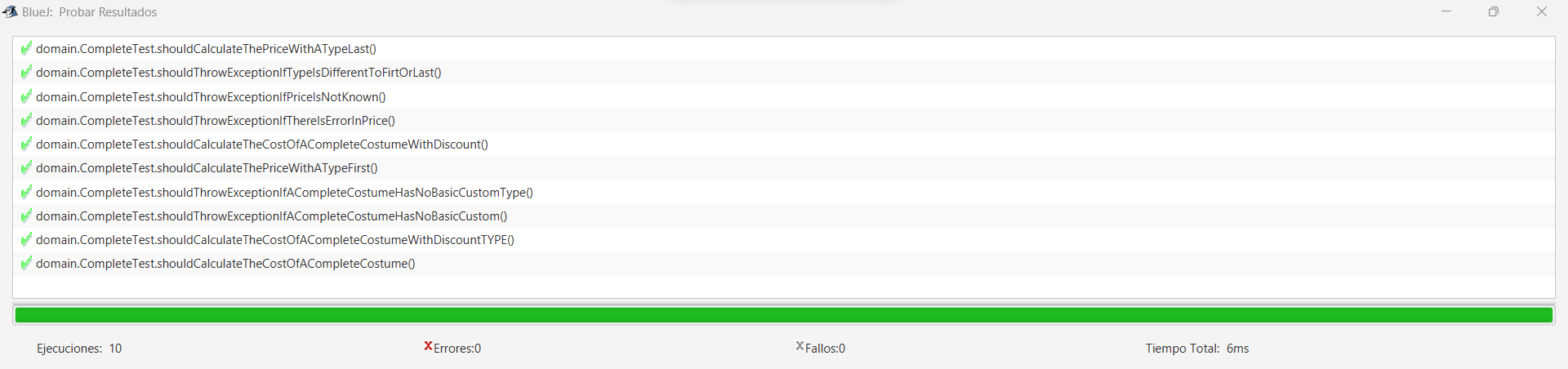




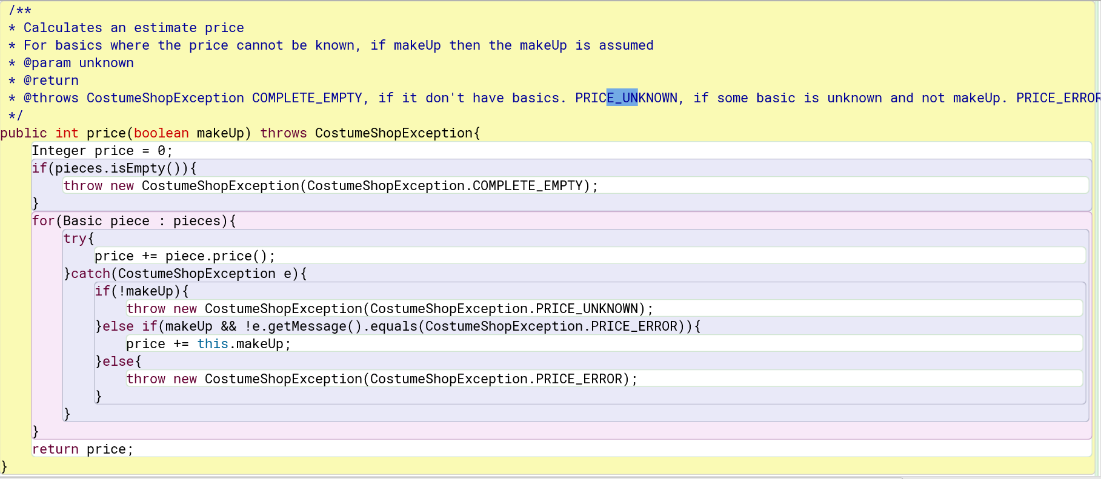


**4. Dada la documentación y el diseño codifiquen y prueben el método price(type).**





**5. Documenten, diseñen, codifiquen y prueben el método price(makeUp).**





**CostumeShop**

EN CONSOLA

El objetivo de esta aplicación es mantener un catálogo de los disfraces ofrecidos por una tienda en Halloween en el proyecto CostumeShop. En este proyecto se ofrecen diferentes tipos de disfraces:básicos y completos.

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

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

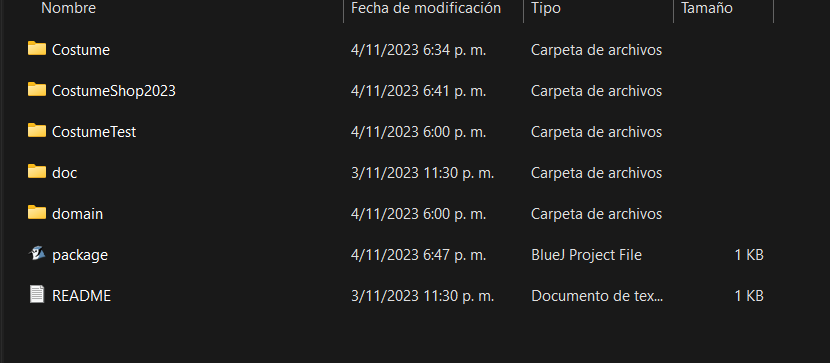
**1. En su directorio descarguen los archivos contenidos en costumeShop.zip, revisen el contenido. ¿Cuántos archivos se tienen? ¿Cómo están organizados? ¿Cómo deberían estar organizados?**

* Se tienen tres archivos, customShop, customShopGUI, Log, no estan organizados, solo se encuentran los .java en la carpeta, deberian estar estar en una carpeta de src.

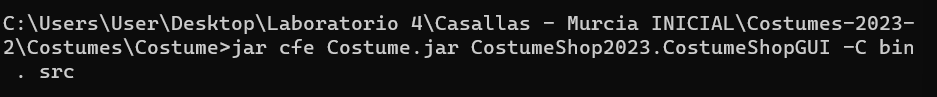
**2. Estudien el diseño del programa: diagramas de paquetes y de clases. ¿cuántos paquetes tenemos? ¿cuántas clases tiene el sistema? ¿cómo están organizadas? ¿cuál es la clase ejecutiva?**

* Solo tenemos un paquete, y tenemos 4 clases(Costume, Basic, Complete, CustomeShopExeption) y una de pruebas, estan organizadas mediante la documentacion(en la carpeta doc), y domain, en el que tenemos el resto de archivos

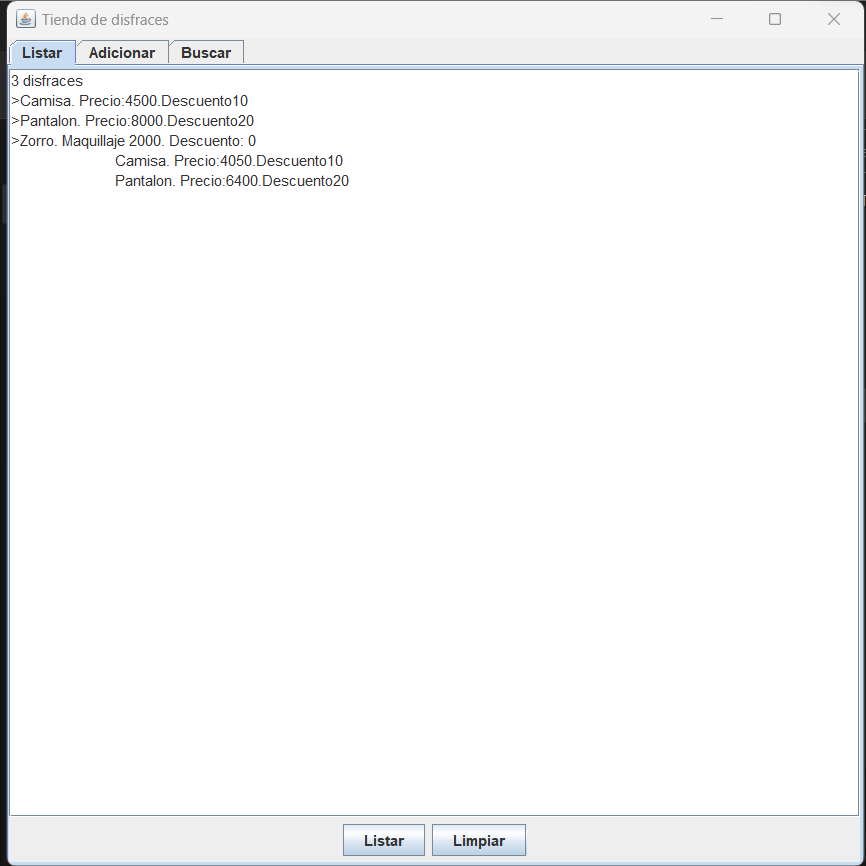
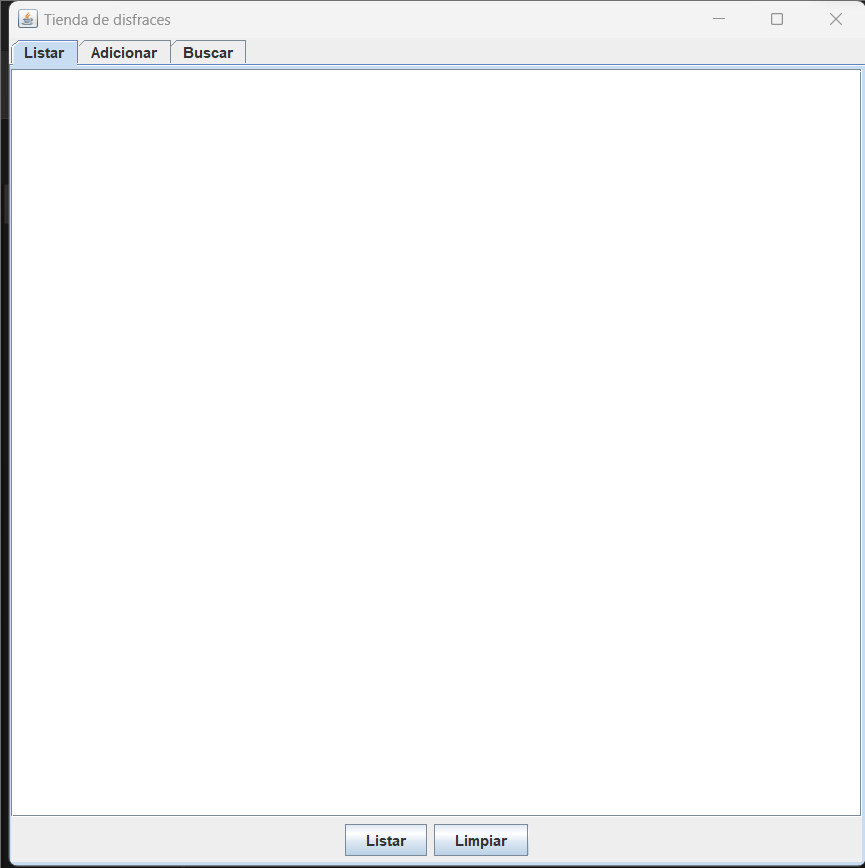
**3. Prepare los directorios necesarios para ejecutar el proyecto. ¿qué estructura debe tener? ¿qué clases deben tener? ¿dónde están esas clases? ¿qué instrucciones debe dar para ejecutarlo?**

Debe estar dividido en paquetes, uno para domain que es donde esta toda la parte logica, otro para la parte grafica que es CostumeShop, otro para la clase de pruebas que es CostumeTest y para crear el ejecutable tenemos el directorio costume, organizado con las carpetas src, bin, docs,lib.  


Las instrucciones utilizadas fueron mkdir para crear las carpetas, el copy \*.java, copy \*.class, el javadoc –d, y el comando del ejecutable que es el siguiente:

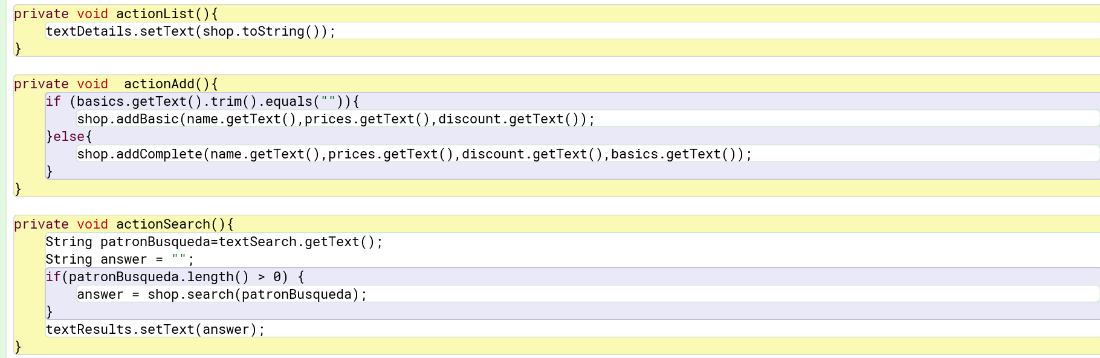


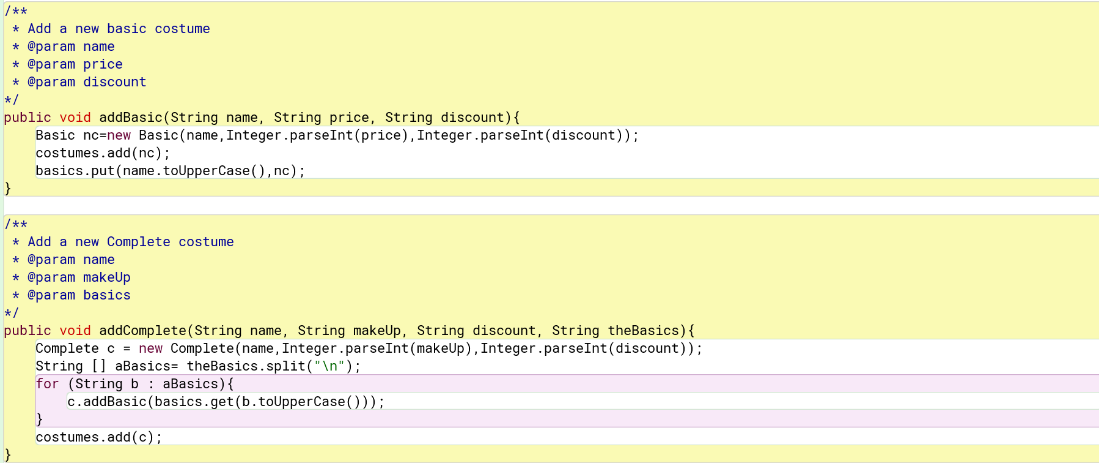
**4. Ejecute el proyecto, ¿qué funcionalidades ofrece? ¿cuáles funcionan?**



Tenemos la función de listar, que nos muestra los disfraces disponibles, junto con su precio y su descuento, también la función limpiar que hace invisible la lista. También tenemos una sección de Adicionar un disfraz y la de buscar uno ya existente.

**5. Revisen el código y la documentación del proyecto. ¿De dónde salen los disfraces iniciales? ¿Qué clase pide que se adicionen? ¿Qué clase los adiciona?**

En customeShopGUI podemos encontrar unas clases que es la que nos indica que disfraces queremos agregar al ejecutable.  


En costumeShop tenemos dos metodos los cuales nos agregan los disfraces que se creen  
  


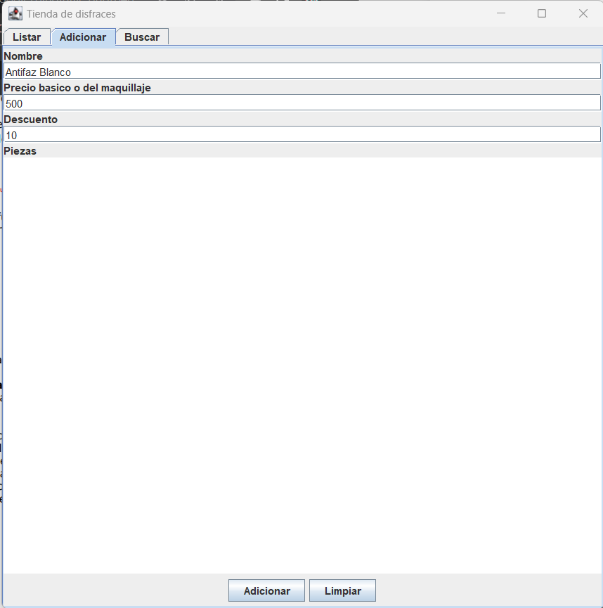
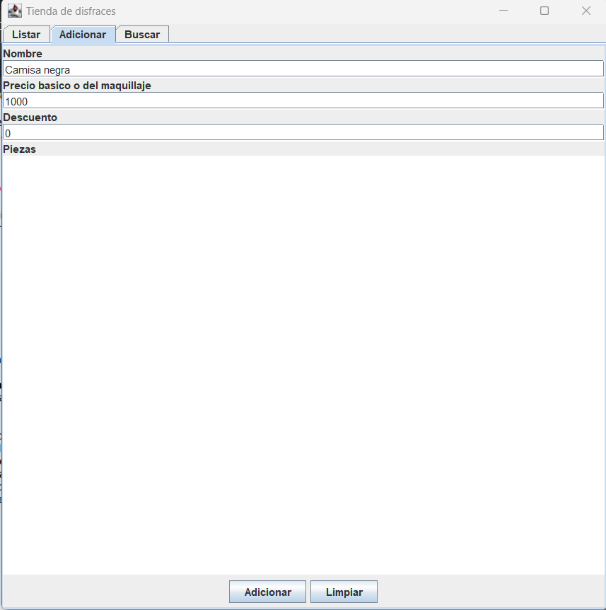
**Adicionar y listar. Todo OK.**

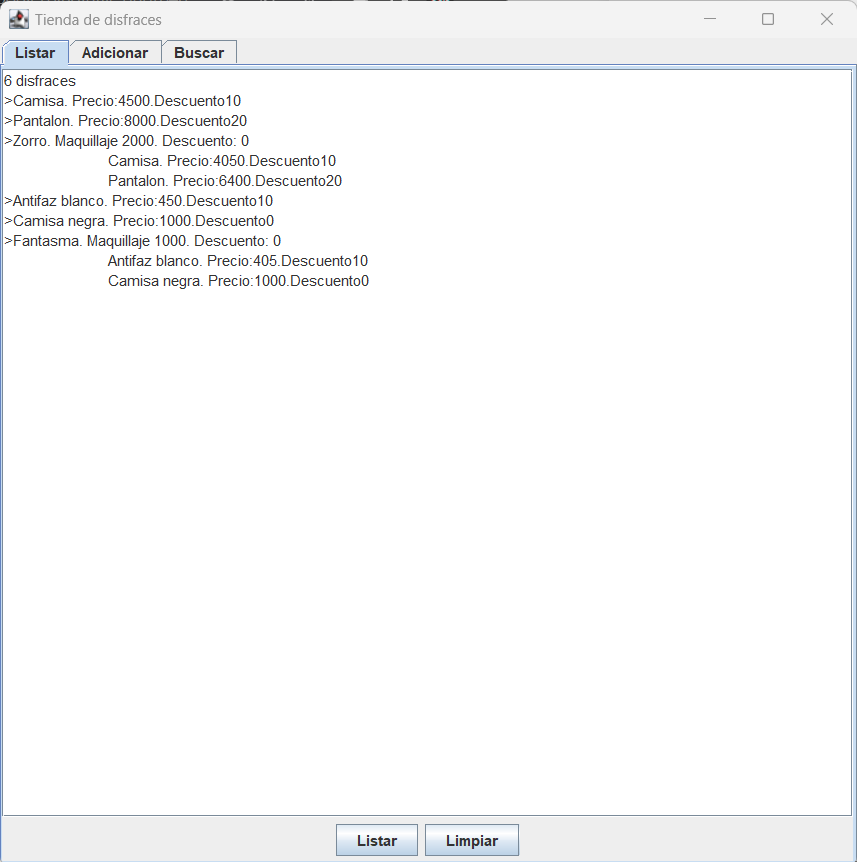
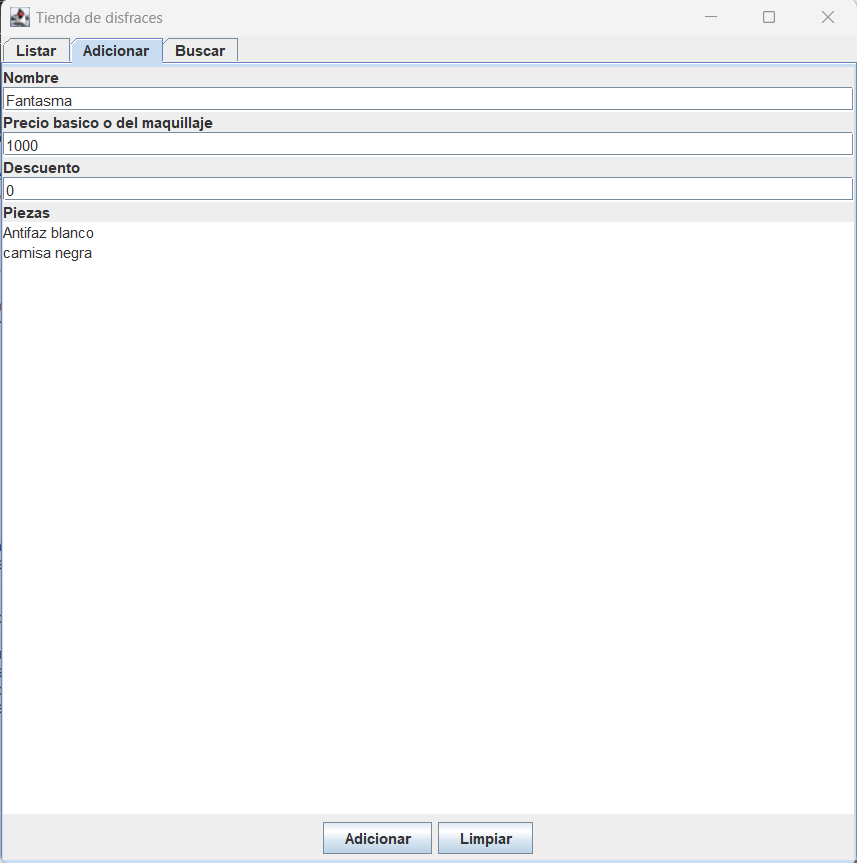
**[En lab04.doc, CostumeShop.astay \*.java] (NO OLVIDEN BDD - MDD)**

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

1**. Adicionen un nuevo disfraz básico y un nuevo disfraz completo Básico Antifaz blanco 500 , 10 Completo Fantasma 1000, 0 Camisa blanca Antifaz blanco ¿Qué ocurre? ¿Cómo lo comprueban? Capturen la pantalla. ¿Es adecuado este comportamiento?**

Primero se deben adicionar los disfraces basicos, una vez tenemos estos creados, procedemos a crear el disfraz completo usando los disfraces basicos que tenemos, el paso a paso se ve en las capturas:

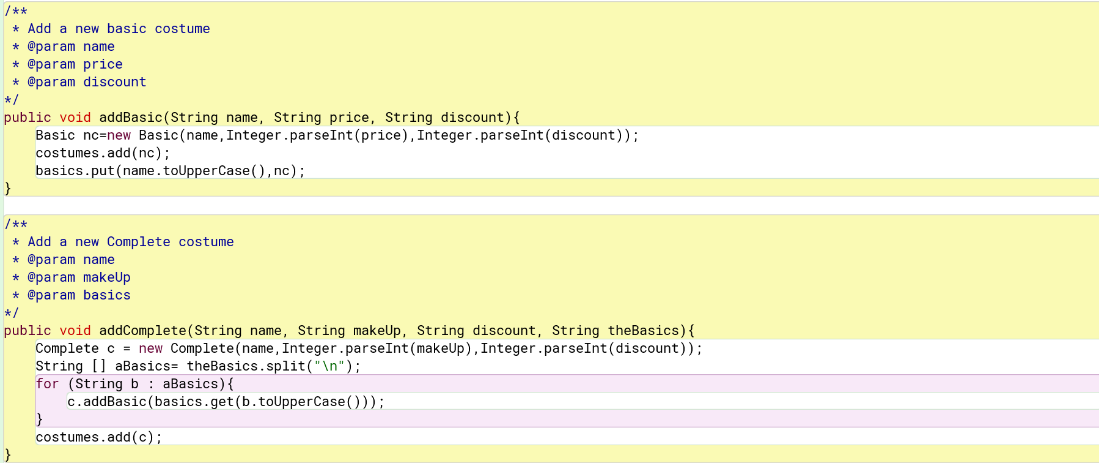


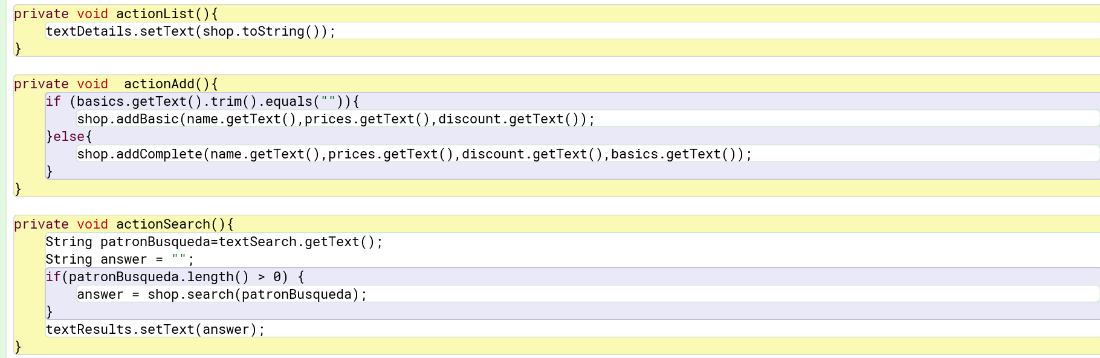


Al darle a listar, podemos ver que ya se ha adicionado el disfraz de fantasma correctamente.

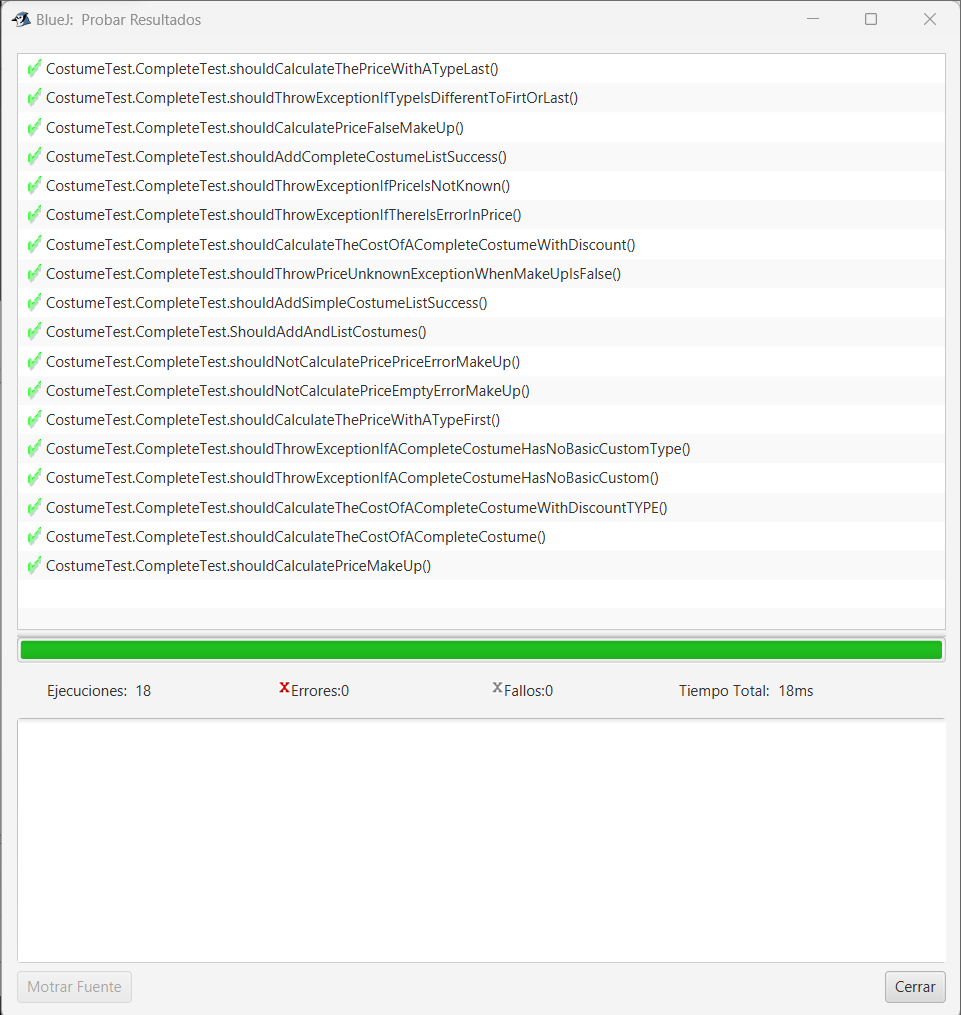
**2. Revisen el código asociado a adicionar en la capa de presentación y la capa de dominio. ¿Qué método es responsable en la capa de presentación? ¿Qué método en la capa de dominio?**

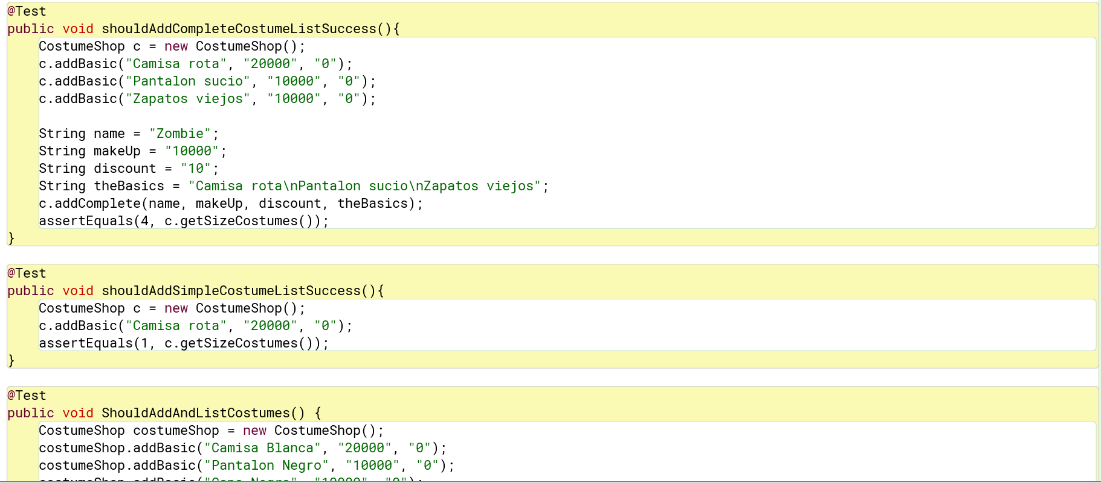
ActionAdd es el responsable de la capa de presentación, mientras que la capa de presentación los los métodos add con los cuales adicionamos los disfraces.





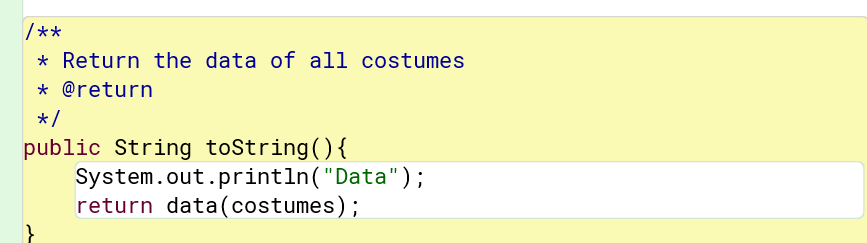
**3. Realicen ingeniería reversa para la capa de dominio para adicionar. Capturen los resultados de las pruebas de unidad.**

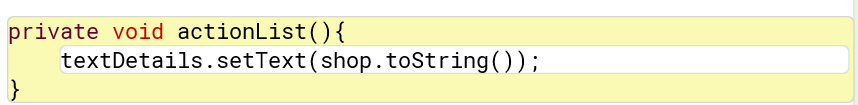




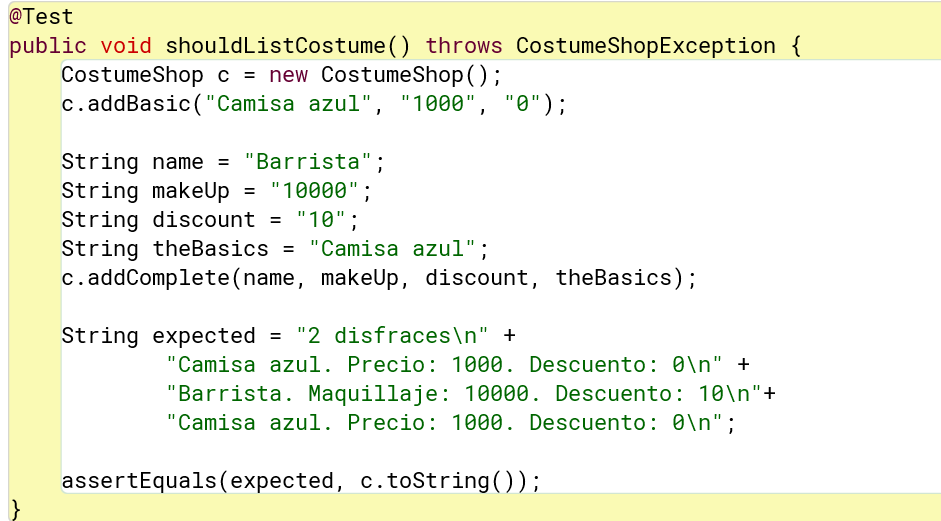
**4. Revisen el código asociado a listar en la capa de presentación y la capa de dominio. ¿Qué método es responsable en la capa de presentación? ¿Qué método en la capa de dominio?**

Tanto el metodo de CustomShop y CustopShopGUI son los responsables, donde el primero es el que se encarga de la capa de dominio y el segundo de la capa presentación, en su respectivo orden.

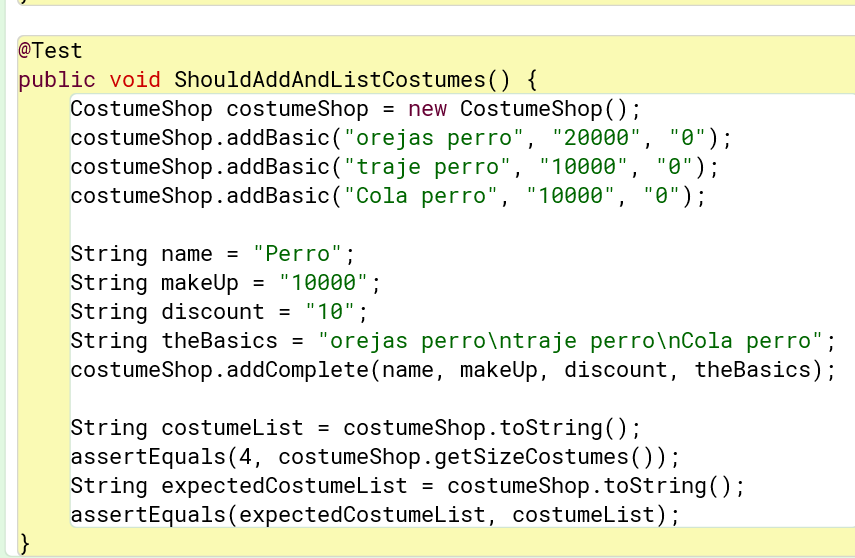


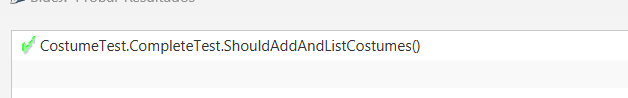


**5. Realicen ingeniería reversa para la capa de dominio para listar. Capturen los resultados de las pruebas de unidad.**



**6. Propongan y ejecuten una prueba de aceptación.**





**Adicionar un disfraz. Funcionalidad robusta**

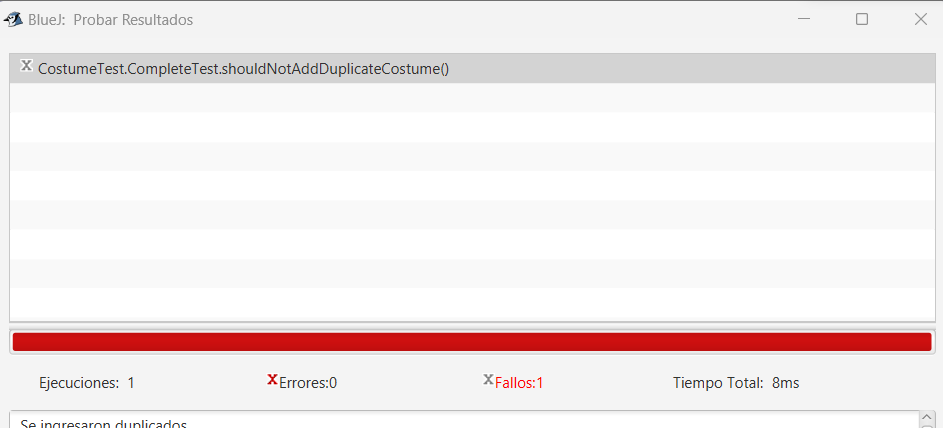
[En lab04.doc, CostumeShop.astay \*.java] (NO OLVIDEN BDD – MDD)

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.

**a.** ¿Y si el nombre del disfraz ya existe?

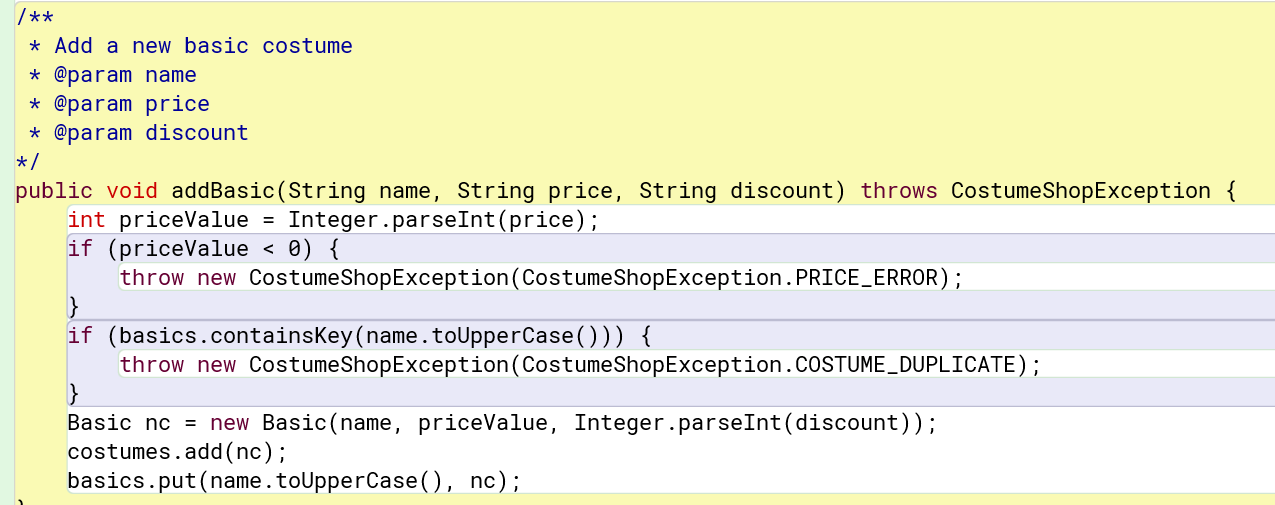
**1. Propongan una prueba de aceptación que genere el fallo.**

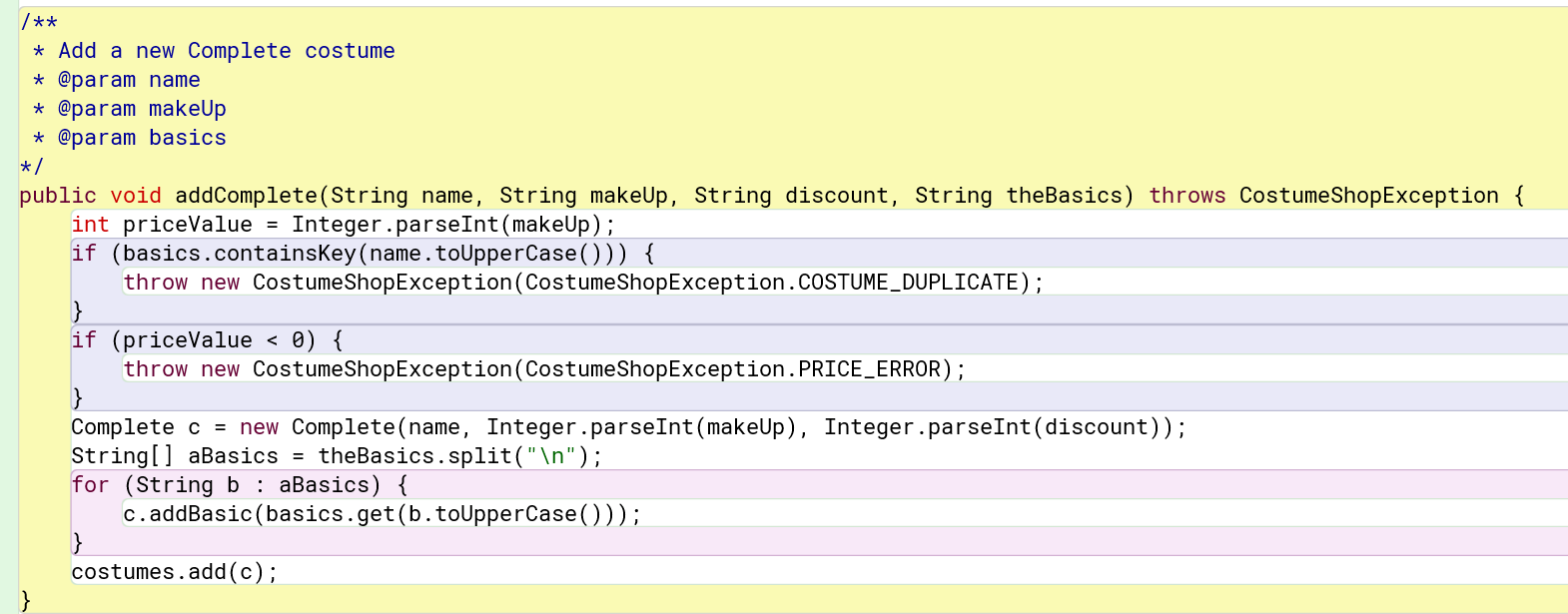




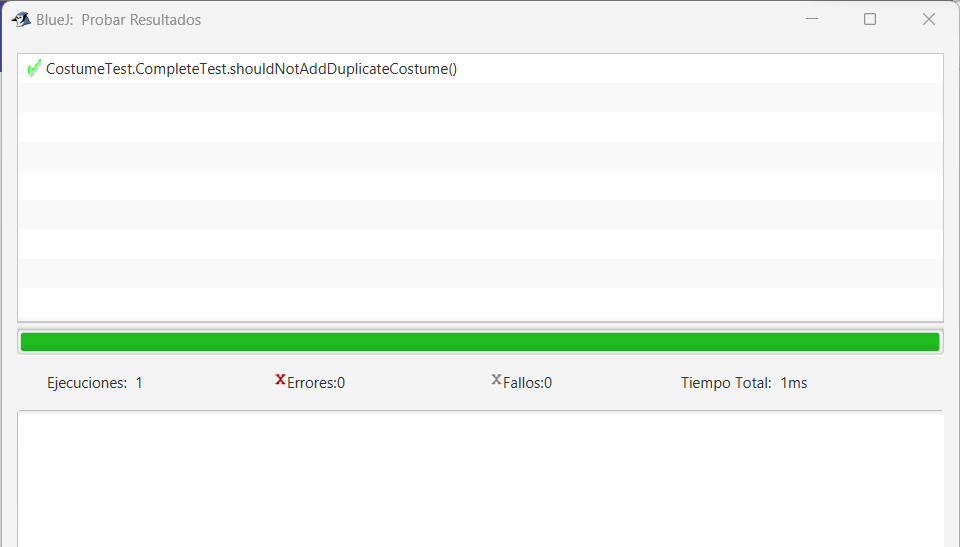
**2. Analicen el diseño realizado. Para hacer el software robusto: ¿Qué método debería lanzar la excepción? ¿Qué métodos deberían propagarla? ¿Qué método debería atenderla? Explique claramente.**

Deberia lanzarse y propagarse en el metodo addBasic y según sea la prueba tambien en addComplete, debido que aquí es donde se agregan al diccionario y se debe hacer la verificacion antes de que sean ingresados.



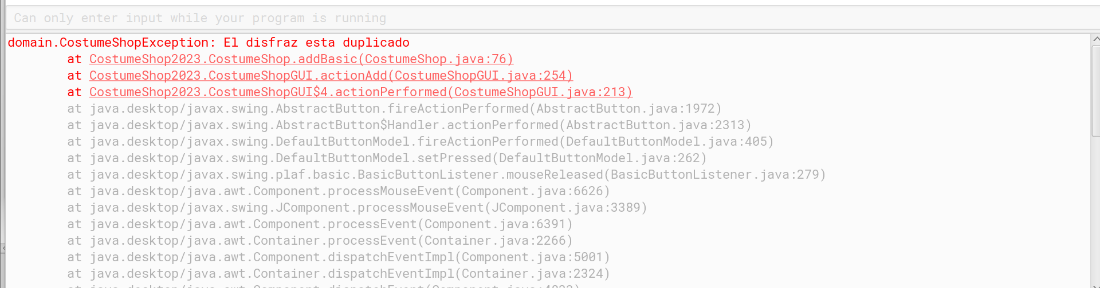


**3. Construya la solución propuesta. Capture los resultados de las pruebas de unidad.**



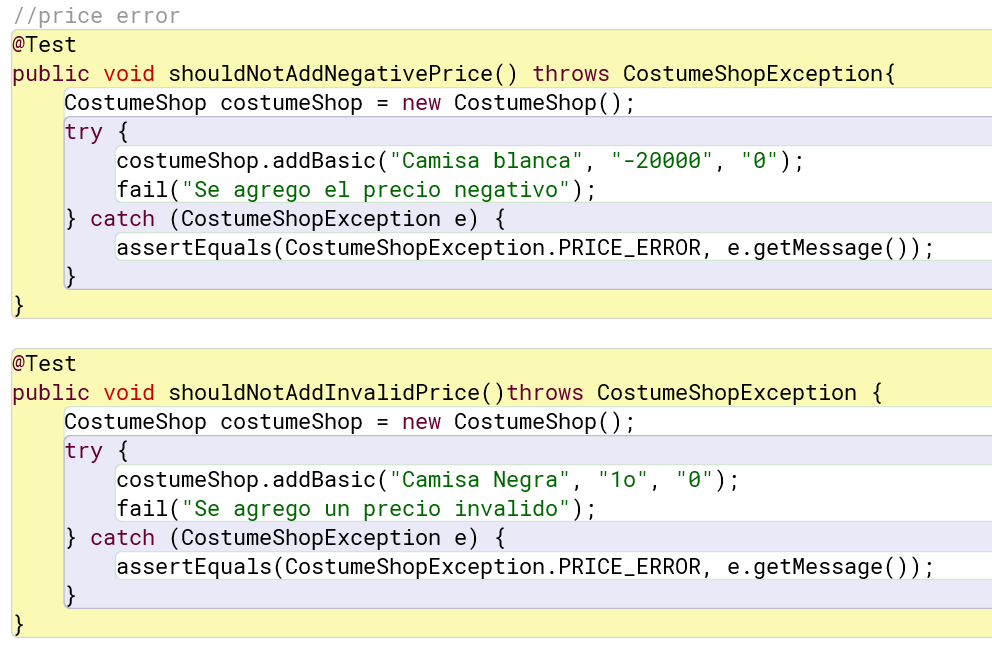
**4. Ejecuten nuevamente la aplicación con el caso de aceptación propuesto en 1. ¿Qué sucede ahora? Capture la pantalla.**

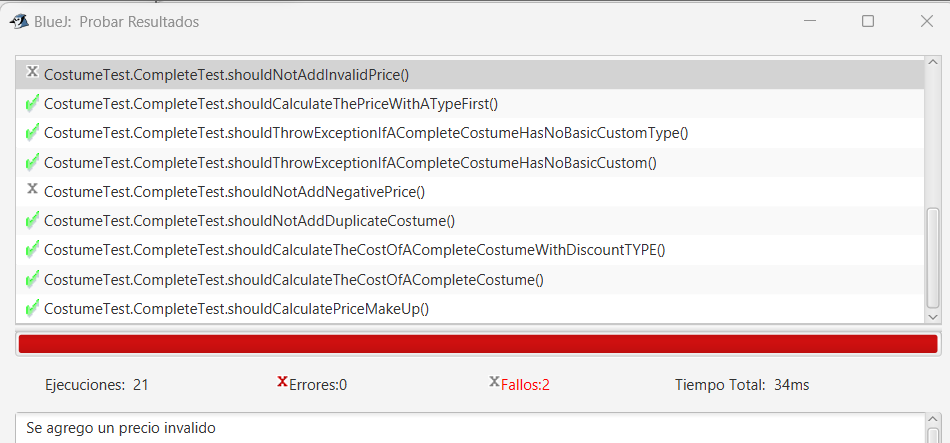
Arroja la Excepción al intentar agregar datos duplicados.



**b.** ¿Y si en precio no da un número? ¿o no da un número negativo?

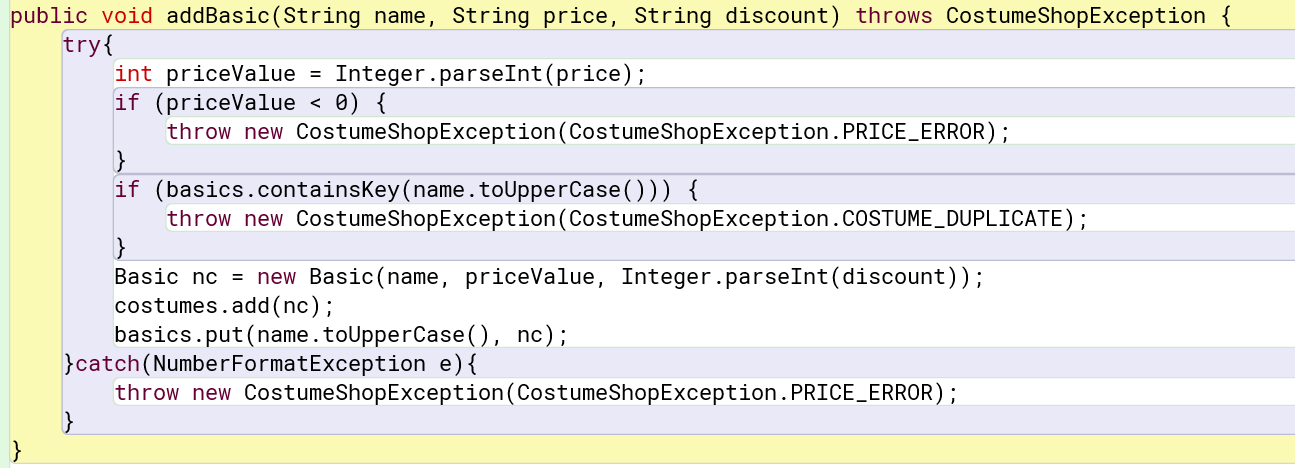
**1. Propongan una prueba de aceptación que genere el fallo.**

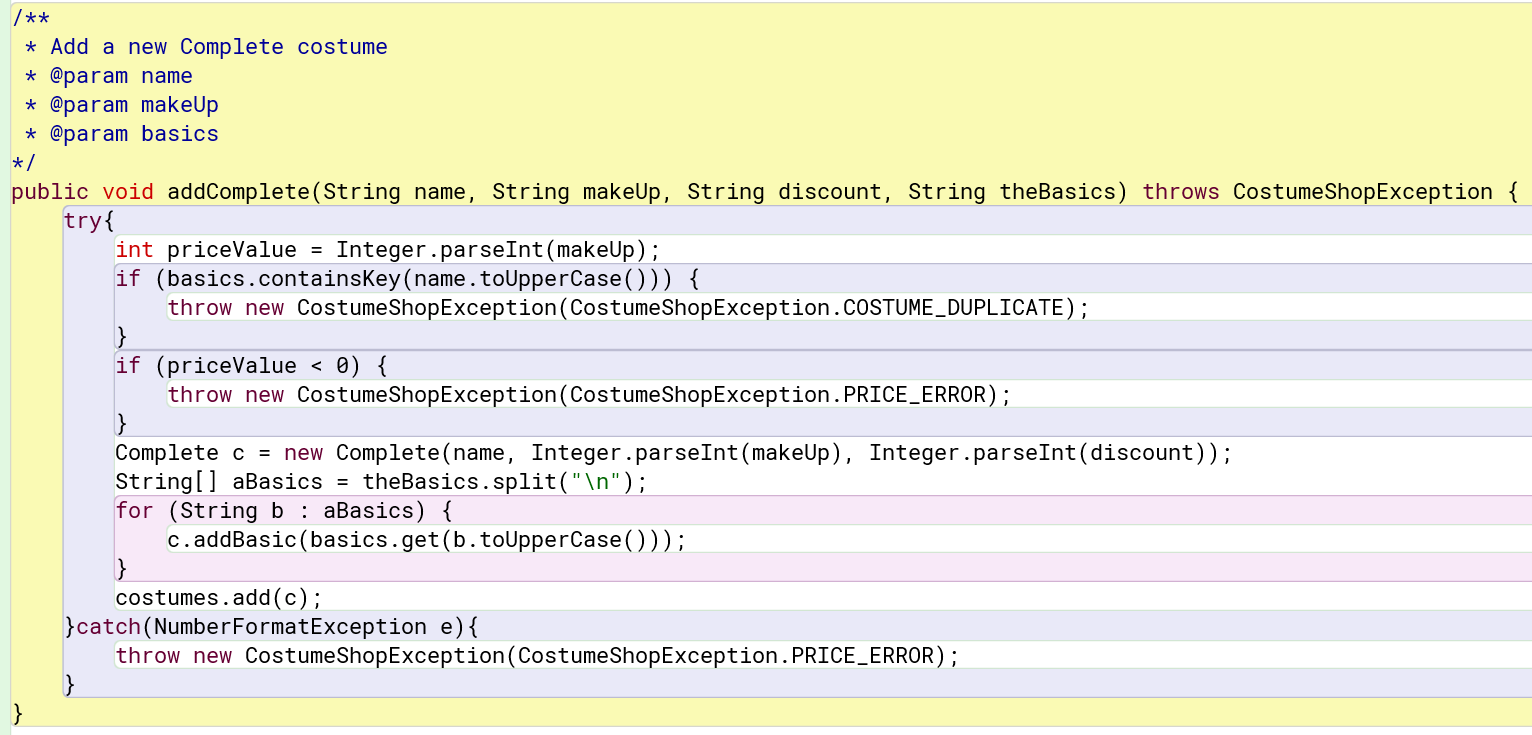




**2. Analicen el diseño realizado. Para hacer el software robusto: ¿Qué método debería lanzar la excepción? ¿Qué métodos deberían propagarla? ¿Qué método debería atenderla? Explique claramente.**

Al igual que en la anterior, tenemos que se deben lanzarse en las pruebas y en los metodos que implementen el addBasic y addComplete ,y progagarse y atenderse en addBasic y addComplete respectivamente.



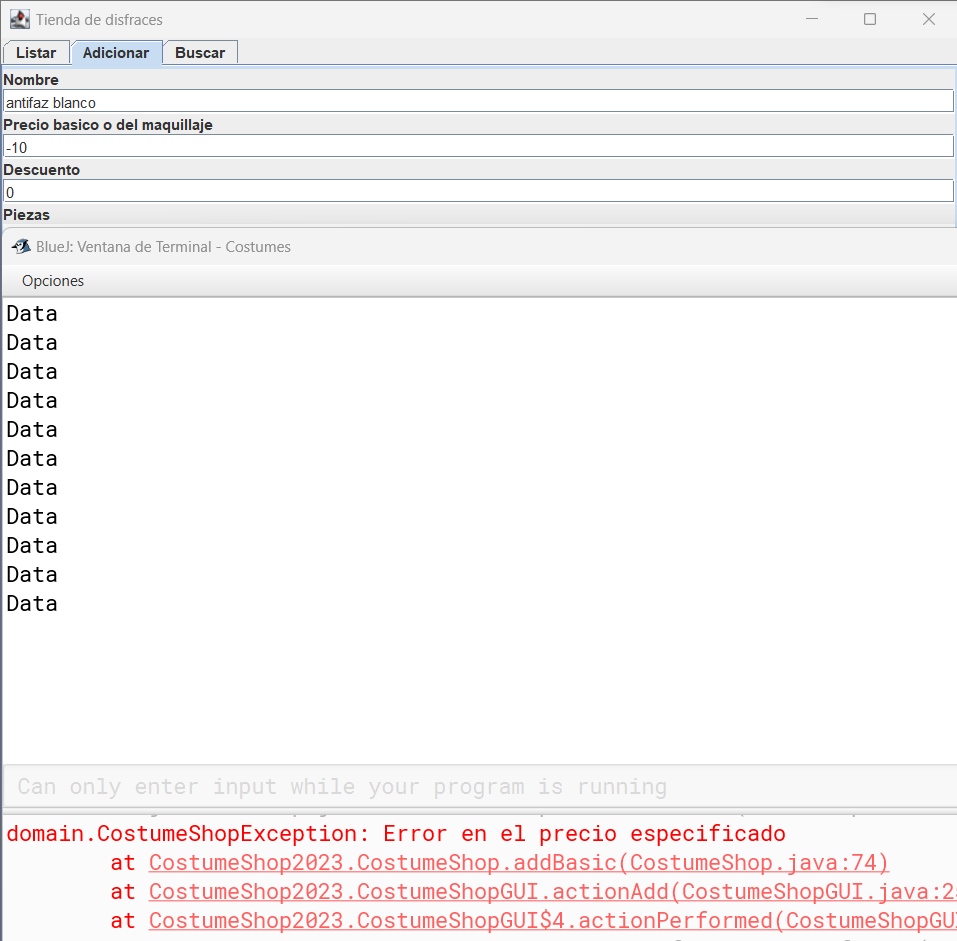


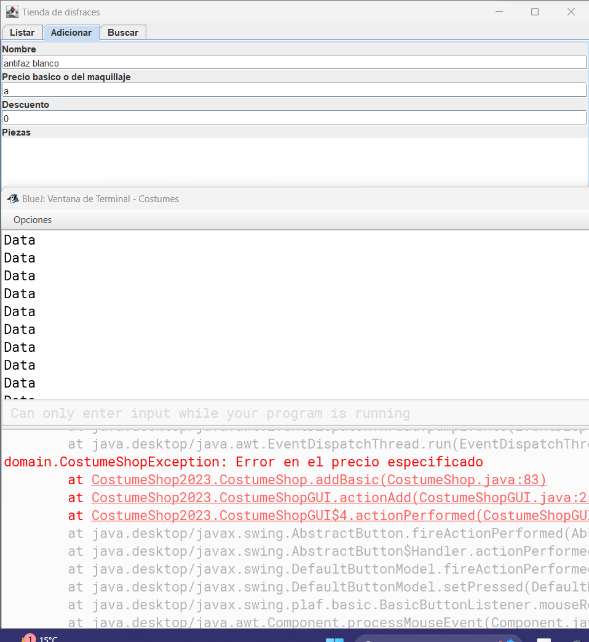
**3. Construya la solución propuesta. Capture los resultados de las pruebas de unidad.**





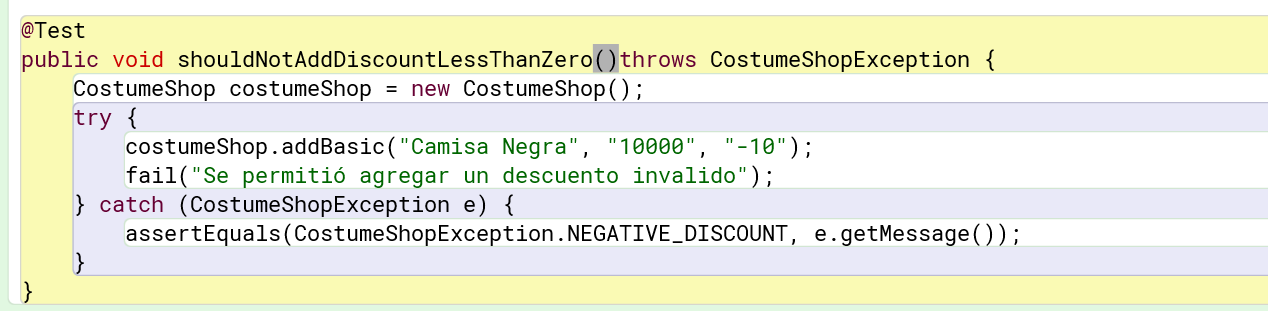
**4. Ejecuten nuevamente la aplicación con el caso de aceptación propuesto en 1. ¿Qué sucede ahora? Capture la pantalla.**

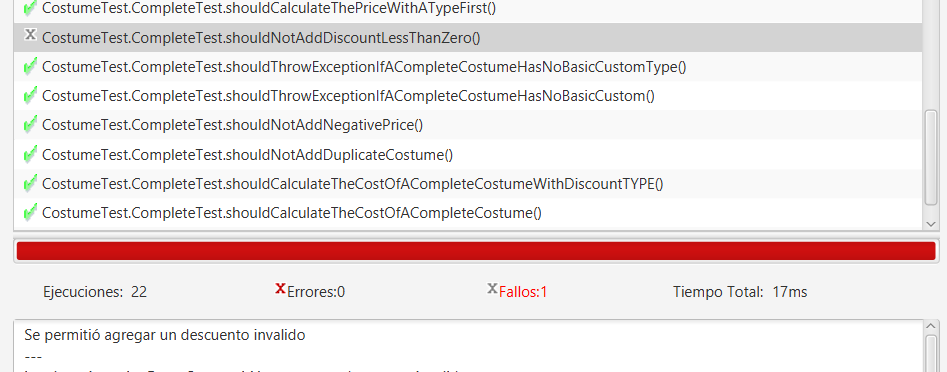




**c**. Proponga una nueva condición

**1. Propongan una prueba de aceptación que genere el fallo.**

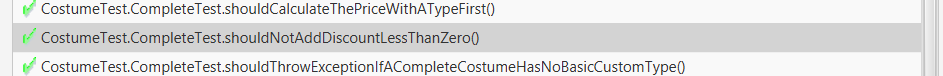




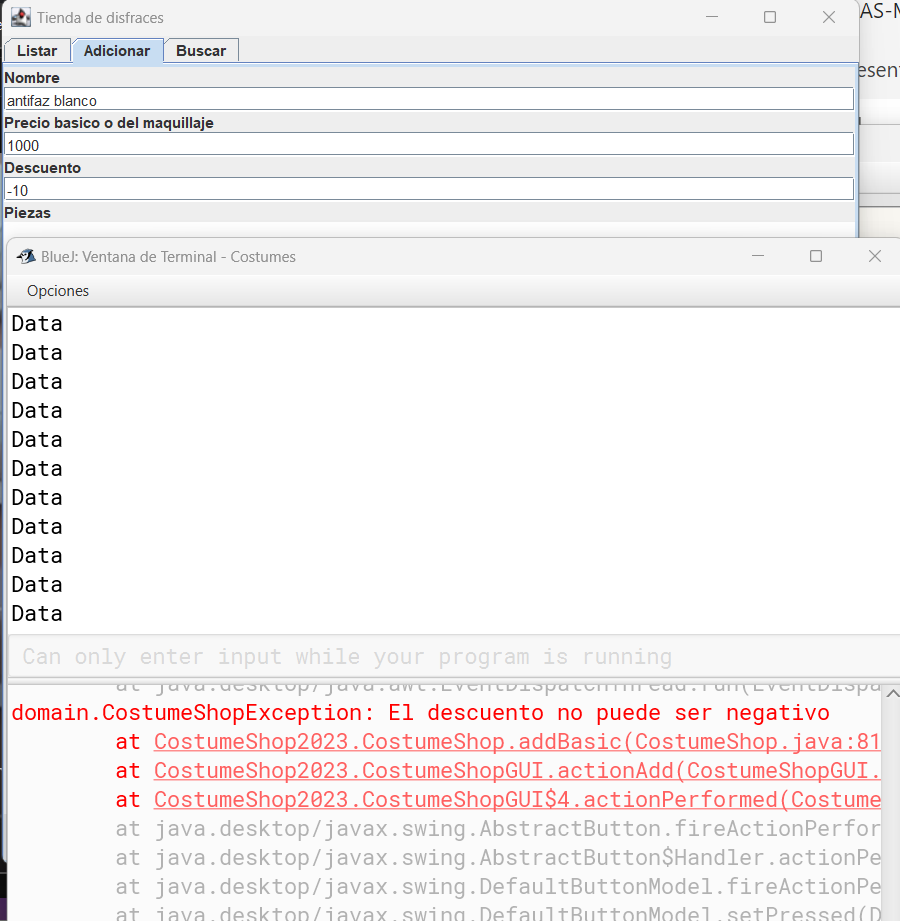
**2. Analicen el diseño realizado. Para hacer el software robusto: ¿Qué método debería lanzar la excepción? ¿Qué métodos deberían propagarla? ¿Qué método debería atenderla? Explique claramente.**

Al igual que en la anterior, tenemos que se deben lanzarse en las pruebas y en los metodos que implementen el addBasic y addComplete ,y progagarse y atenderse en addBasic y addComplete respectivamente

**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.**

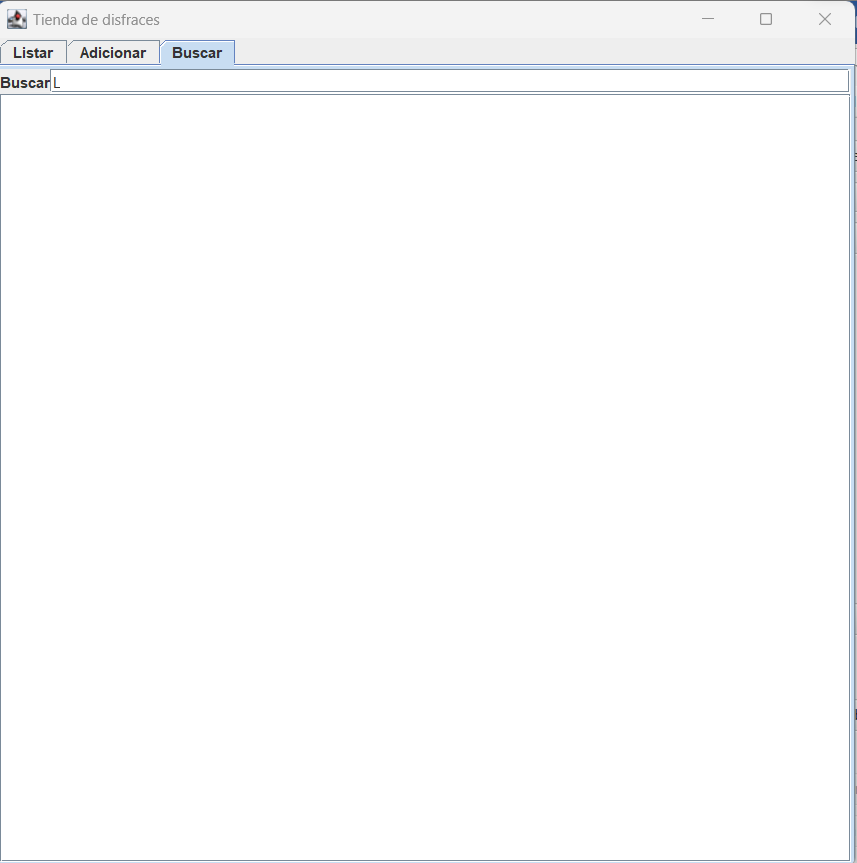


**Consultando por patrones. ¡ No funciona y queda sin funcionar!**

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

(NO OLVIDEN BDD - MDD)

**1. Consulten un disfraz completo que inicie con I. ¿Qué sucede? ¿Qué creen que pasó? Capturen el resultado. ¿Quién debe conocer y quien NO debe conocer esta información?**



Debido a que aun no sé encuentra funcionando el metodo de la busqueda y el que no se haya creado ninguno, no aparece en la busqueda, sin embargo, aunque se tuviera creado un difraz por L, no apareceria, pues falta la implementación.  
La información la debe conocer el Desarrollador debido que es quien debe entender cómo funciona, el usuario no debe estar al tanto de su funcionamiento interno.

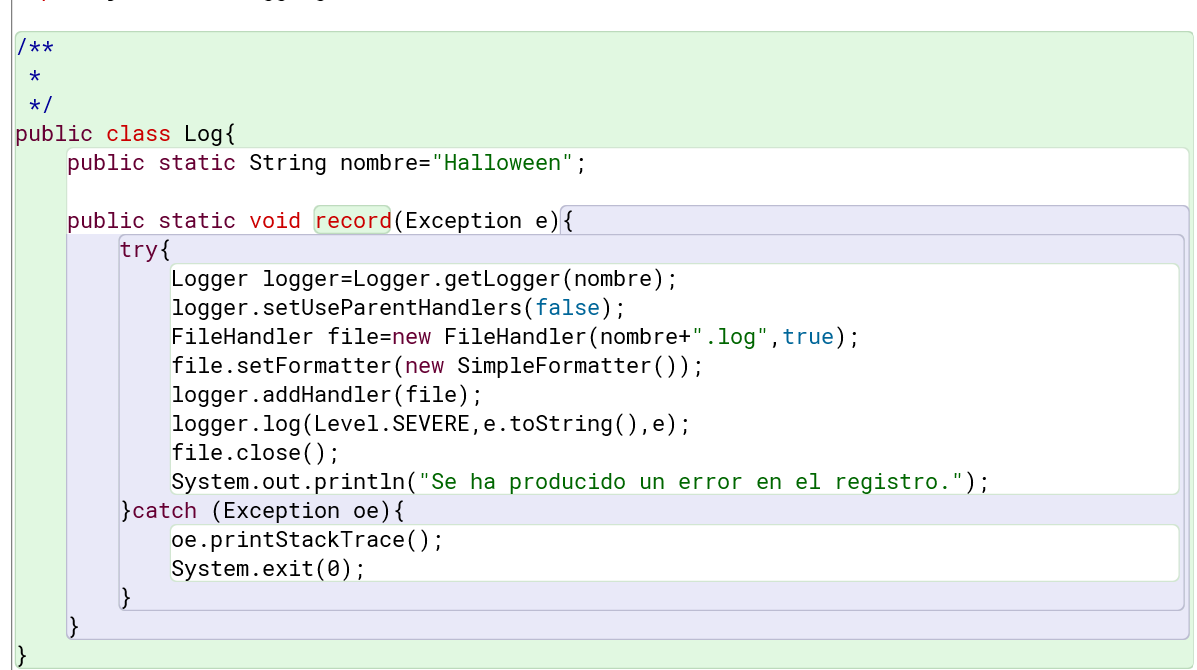
**2. Exploren el método record de la clase Log ¿Qué servicio presta?**



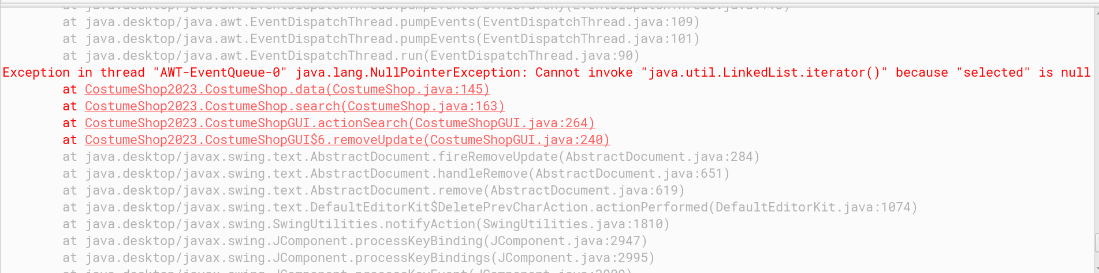
esta clase permite crear registros de errores en un archivo llamado "Halloween.log" y proporciona una forma de mantener un registro de excepciones y eventos importantes que pueden ocurrir en una aplicación. Estos registros son útiles para el seguimiento y la depuración de problemas en la aplicación.

**3. 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.**

Al mandarle el mensaje al final del try, este se va a mandar cada que se use el método récord de log, esto debe ser implementado en los demas metodos junto con JOptionPane para que pueda ser visto por el usuario



**4. 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?**

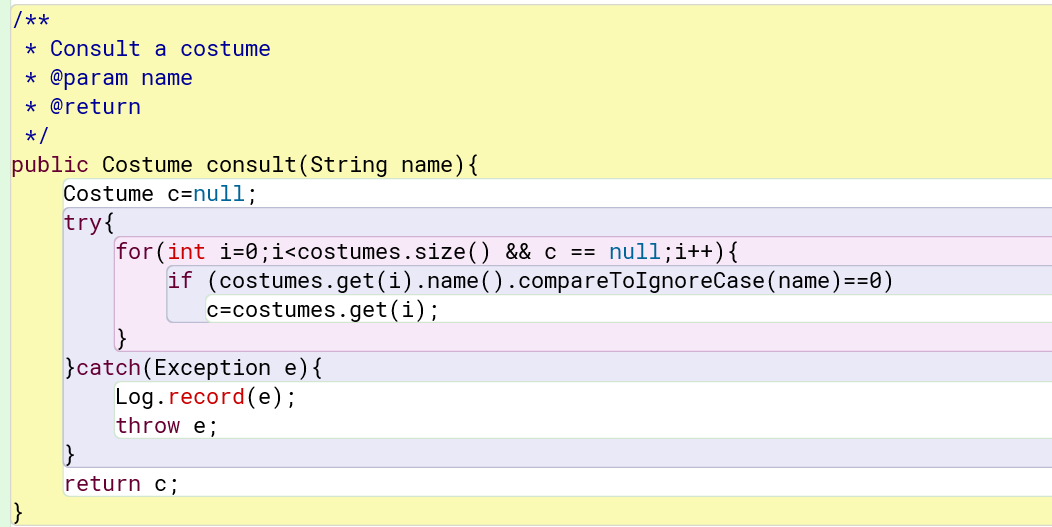


La aplicación no termina su ejecución, de igual forma permite seguir agregando disfraces, pero arroja error cuando trato de buscar en el apartado de búsqueda e ingreso un carácter.

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

La aplicación podría continuar con su ejecución normalmente mientras no sea algo que afecte en su totalidad al sistema.

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



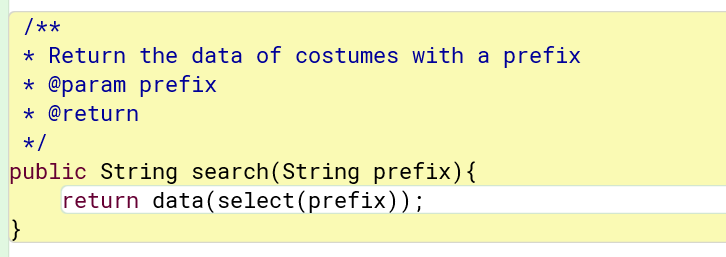
A cada uno de los métodos de CostumeShop se les agrega un Try-Catch de tal forma que ante cualquier error este pueda ser atendido por la clase de excepciones.

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

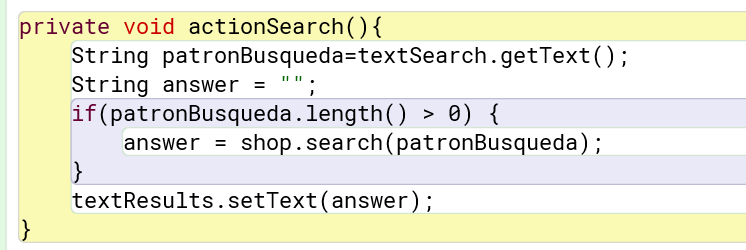
**[En CostumeShop.asta, CostumeShop.log, lab04.java y \*.java] (NO OLVIDEN BDD - MDD)**

**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?**

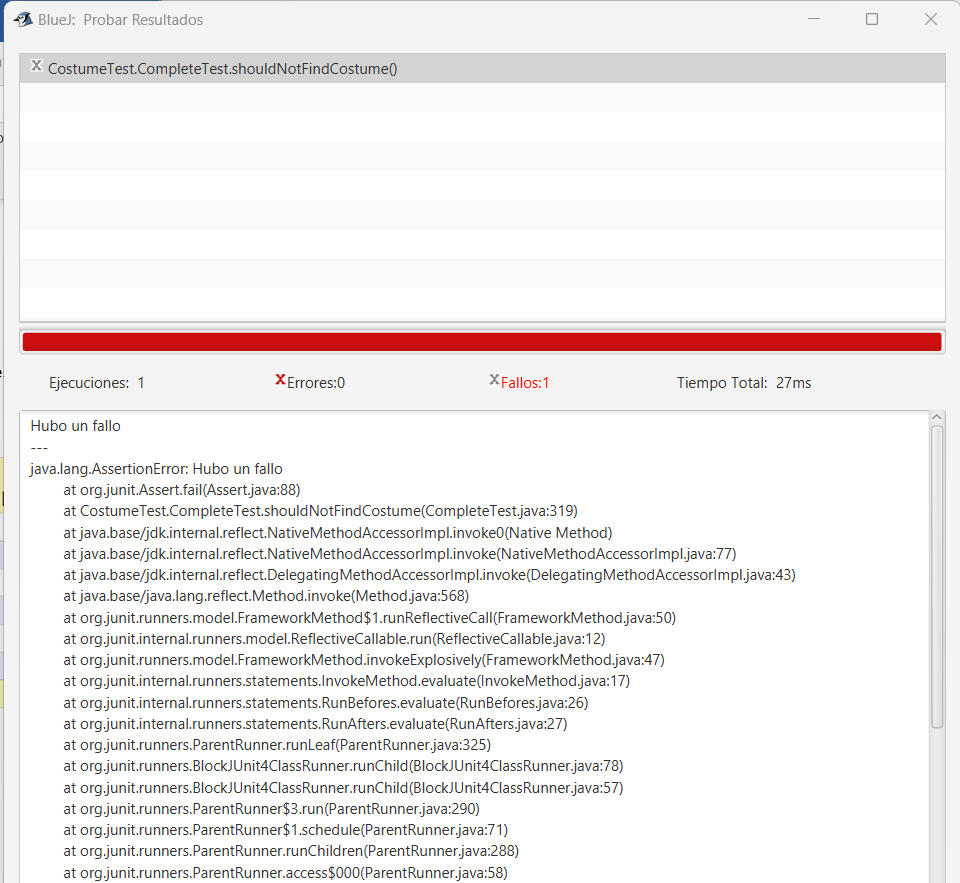
Capa de dominio Para Buscar



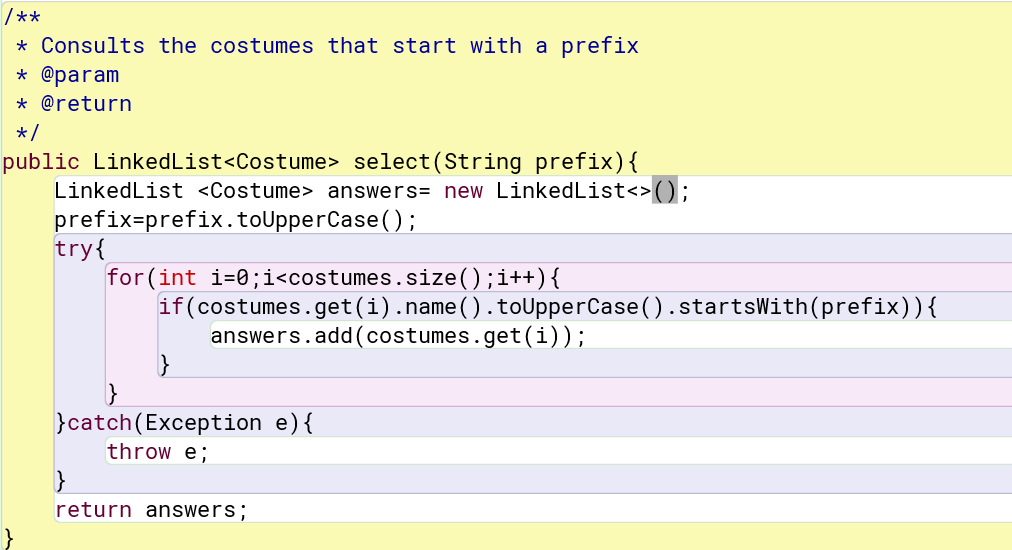
Capa de presentación para Buscar



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



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



Answers estaba definido como null, por lo cual nos arrojaba error.

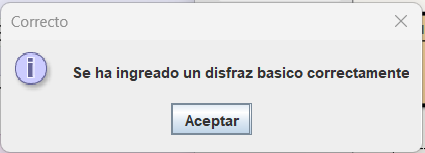
**4. Ejecuten la aplicación nuevamente con el caso propuesto. ¿Qué tenemos en pantalla? ¿Qué información tiene el archivo de errores?**

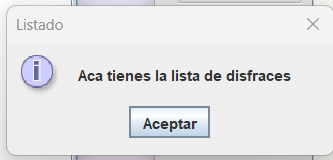


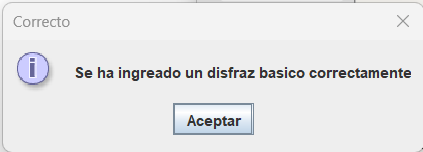
**5. Refactorice la funcionalidad para que sea más amable con el usuario. ¿Cuál es la propuesta? ¿Cómo la implementa?**

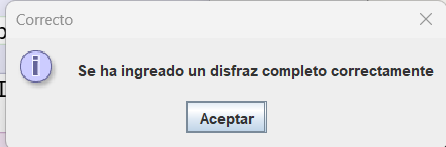
Debido a que, al realizar alguna acción dentro de la aplicación, desconocemos si se realizó o no, es bueno agregar un mensaje que le avise al usuario si por ejemplo se pudo agregar un disfraz, si se puede listar o en su defecto, un mensaje de bienvenida al usuario.

Esto se puede realizar por medio de un JoptionPane.









**RETROSPECTIVA**

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

**15 horas por persona**

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

**Se encuentra realizado en su totalidad, debido a que se le dedico el tiempo necesario**

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

**Simplicidad en el código, pues dejamos las cosas lo más funcional posible y con el menos código requerido.**

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

**El mayor logro fue lograr que todo el simulador funcionara, pues es bastante satisfactorio el ver como se logra darle funcionalidad.**

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

**No sé tuvo mayor problema tecnico en este laboratorio.**

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

**Cada uno compartió ideas y aporto lo necesario para el laboratorio. Nos comprometemos a seguir así.**