**FUNDACIÓN UNIVERSITARIA INTERNACIONAL DE LA RIOJA**

**Desarrollo de aplicaciones web (COLINSO) Octubre 2024 PER 12153**

##### **Actividad 1: Laboratorio: Desarrollo de aplicaciones en Eclipse y NetBeans**

**DANIEL ESTEBAN SANCHEZ ALVAREZ**

**DICIEMBRE 9 DE 2024**

**Introducción**

**Objetivos**

**Desarrollo**

**Conclusiones**

**Referencias**

**Introducción**

En el presente informe se planea la elaboración de un aplicativo en Java utilizando la librería visual Swing. La aplicación deberá realizar pedidos farmacéuticos a través de una interfaz visual, la cual presentará errores y modales de confirmación para realizar el proceso de pedido, además de un modal de confirmación que le avisará al usuario cuando su pedido ha sido enviado.

El aplicativo finalizado se puede encontrar tanto en GitHub ([link](https://github.com/DannyDev93/pharmacy-project.git)), como en un archivo comprimido (adjunto en el comprimido entregable), junto con el archivo JAR compilado.

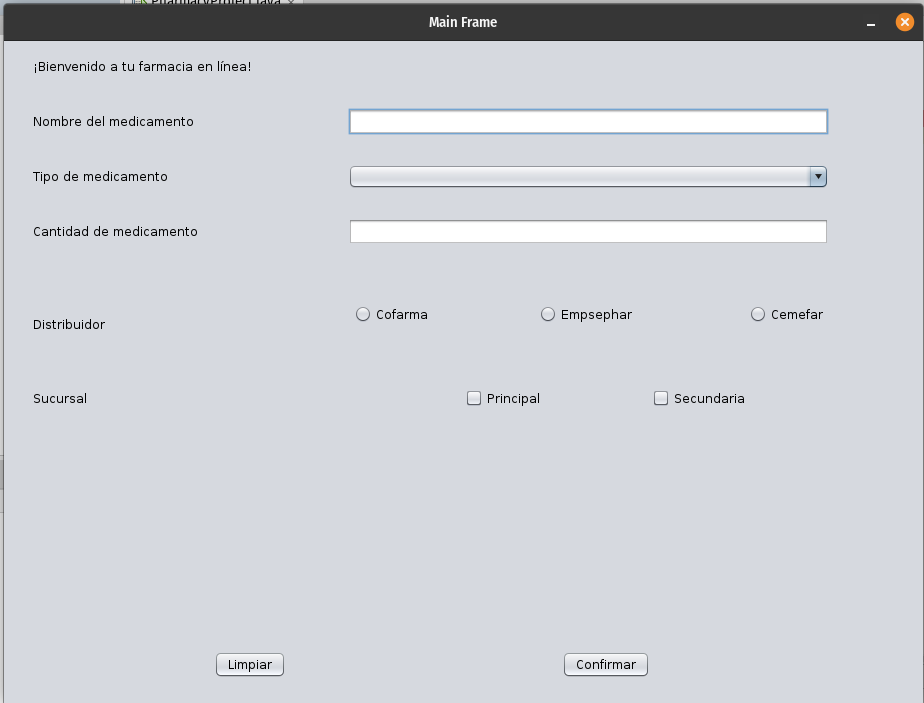
**Objetivos**

1. Realizar un aplicativo con Java SE 21 y Swing en NetBeans IDE.
   1. Construir una aplicación funcional.
      1. Construir una aplicación que le permita al usuario realizar pedido farmacéuticos y que le brinde la posibilidad de ver sus errores en el procedimiento o confirmarle que su pedido fue exitoso.
   2. Probar el funcionamiento de la aplicación a través de un proceso muestra que realice el procedimiento completo de pedido.
   3. Realizar pruebas de software a través de *breakpoints.*
2. Practicar los conocimientos recibidos de dicha lengua de programación.
3. Documentar el funcionamiento del aplicativo.

**Desarrollo**

El aplicativo recibe la siguiente información (*Figura 1*):

1. El nombre del medicamento.
   1. El nombre del medicamento es revisado a través de una expresión regular para verificar que sea alfanumérico.
2. El tipo de medicamento, seleccionado mediante un *dropdown.*
3. La cantidad del medicamento.
   1. La cantidad del medicamento es revisada en el proceso de confirmación para que sea un entero positivo y que no sea mayor a 10 dígitos.
4. El distribuidor del medicamento.
   1. Los *radio buttons* realizan una función para limpiar el valor de los demás cuando son seleccionados, ya que el distribuidor es único y no opción múltiple.
5. La o las sucursales que estarán enviando el medicamento.



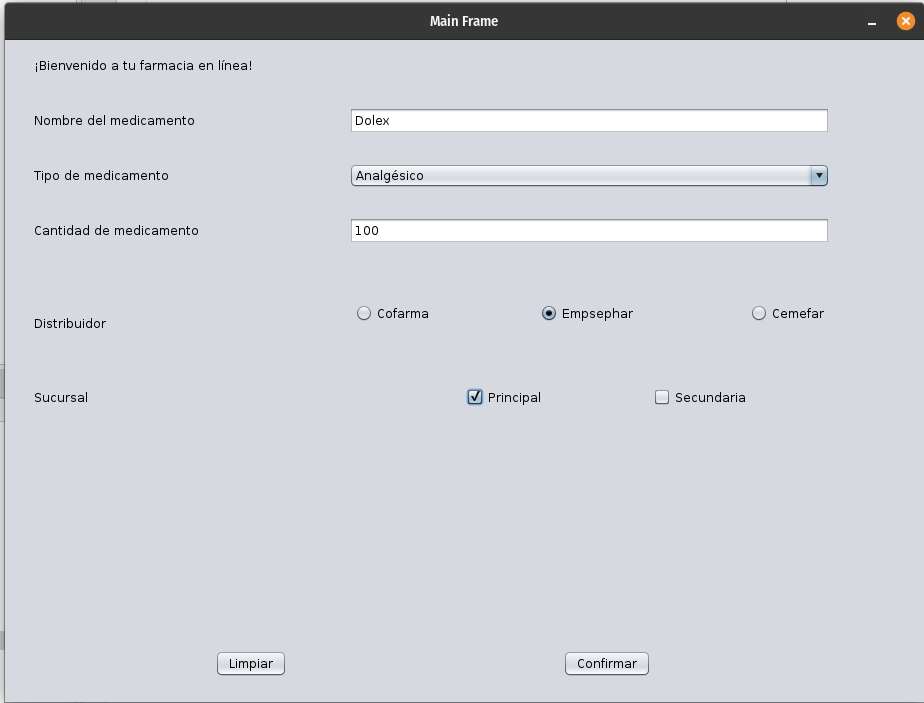
*Figura 1: pantalla principal del aplicativo*

El proceso de pedido del medicamento se realiza de la siguiente forma:

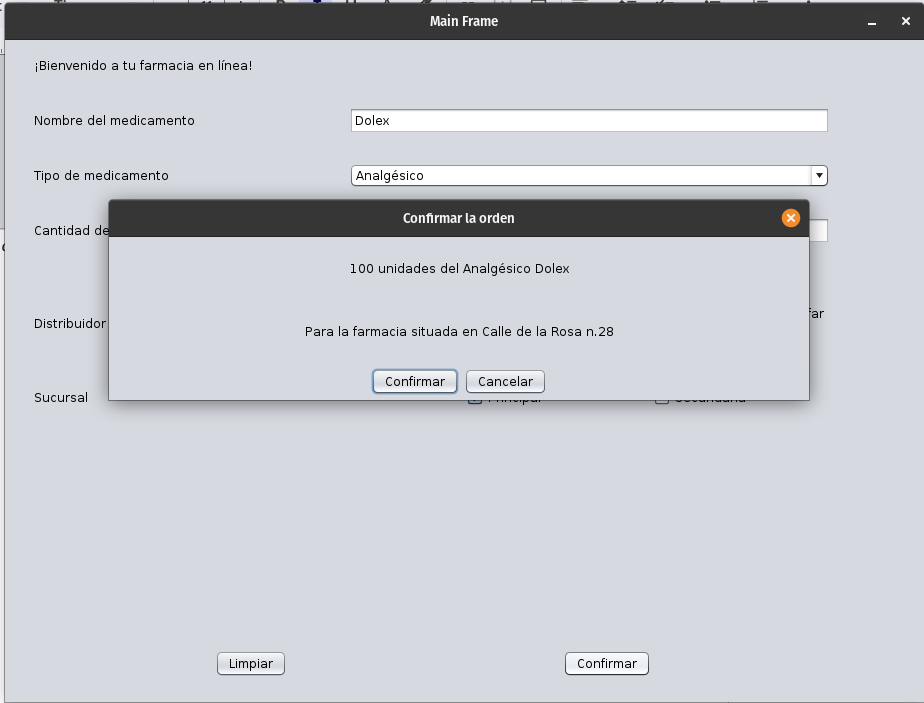
1. Se escriben los valores requeridos en el aplicativo (*Figura 2*).
2. Se confirma a través del botón “Confirmar”
   1. Se muestra una ventana de confirmación con el nombre, tipo y cantidad de medicamento, además de la dirección o direcciones de las cuales se enviará el medicamento (*Figura 3*).
   2. Se confirma a través del botón “Confirmar” en la ventana emergente.
   3. Se muestra una ventana emergente que confirma que el pedido ha sido enviado (Figura 4).
      1. A su vez, se limpian los campos del formulario para permitir un nuevo pedido.

Se tienen en cuenta dos casos extra:

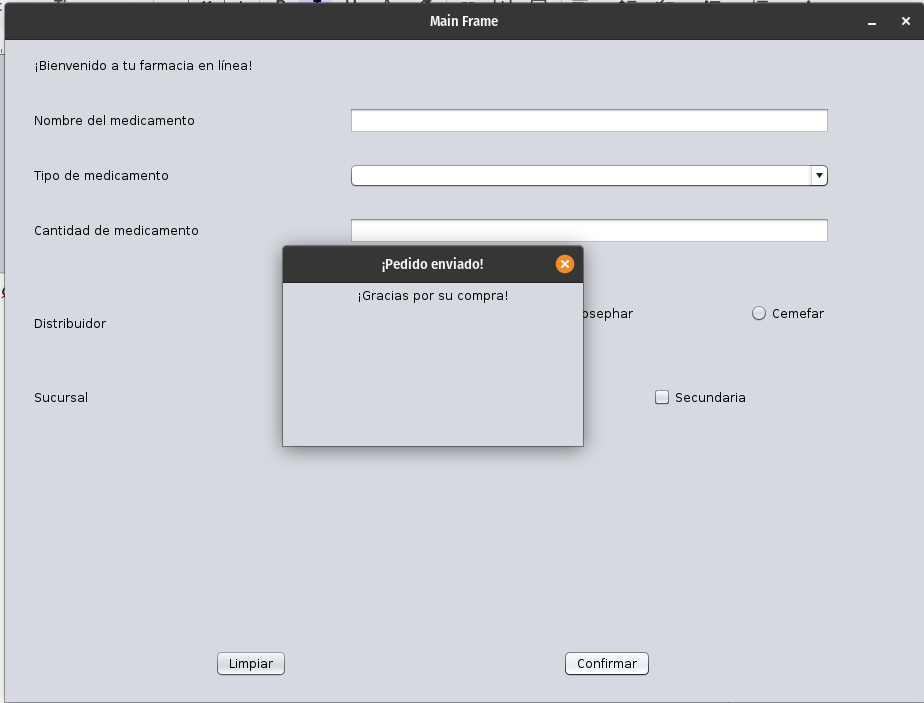
1. Se tiene un botón en el formulario para limpiar todos los campos, en caso de que el usuario lo quiera.
2. Se tiene un botón de “Cancelar” en la primera ventana emergente que regresa a la ventana anterior, en caso de que el usuario quiera hacer correcciones en la información.



*Figura 2: captura de información.*

**

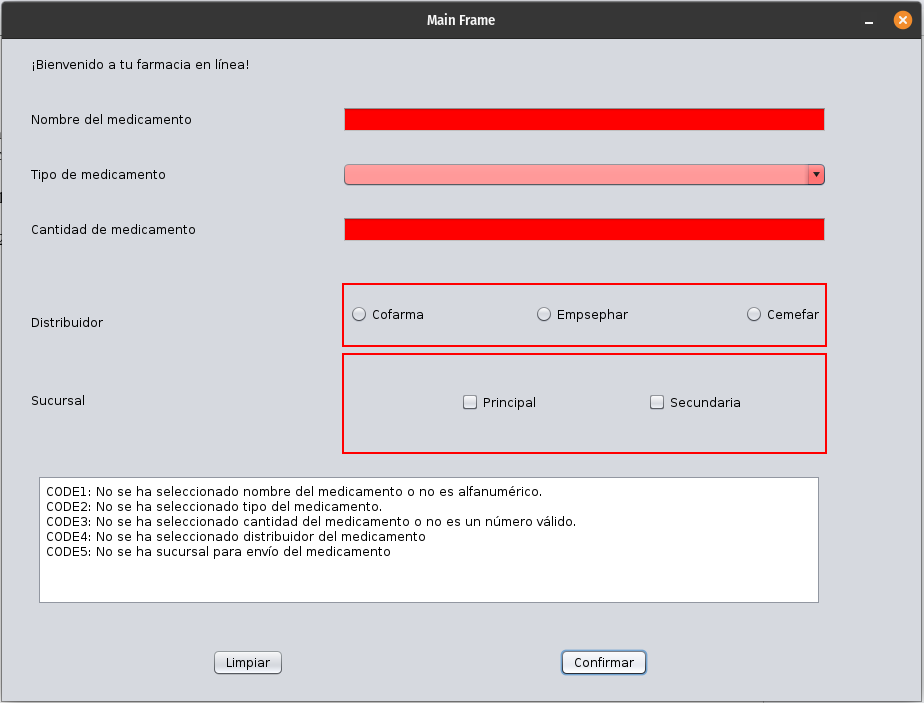
*Figura 3: confirmación de pedido.*

**

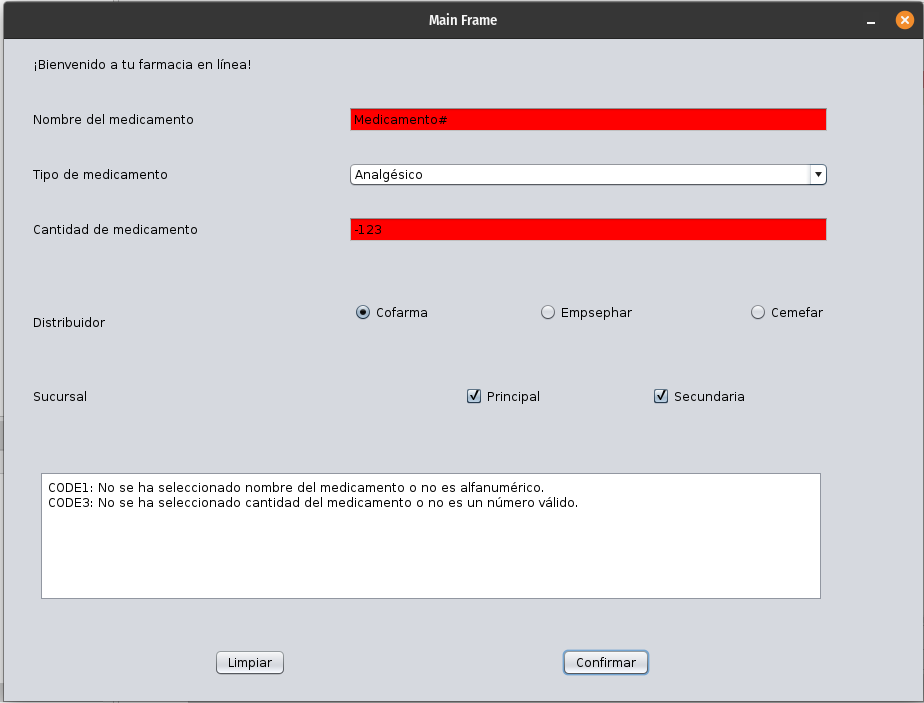
*Figura 4: mensaje de envío del pedido.*

El manejo de errores de usuario se realiza a través de la verificación de datos que se ingresan en el aplicativo al momento de confirmar el pedido:

1. Si los datos son nulos, se muestra en rojo los campos afectados, además de un mensaje en la parte inferior que explica qué campos son incorrectos (*Figura 5)*
2. Si los datos se han ingresado, pero son incorrectos, se muestra en rojo los campos afectados, además de un mensaje en la parte inferior que explica qué campos son incorrectos (*Figura 6)*

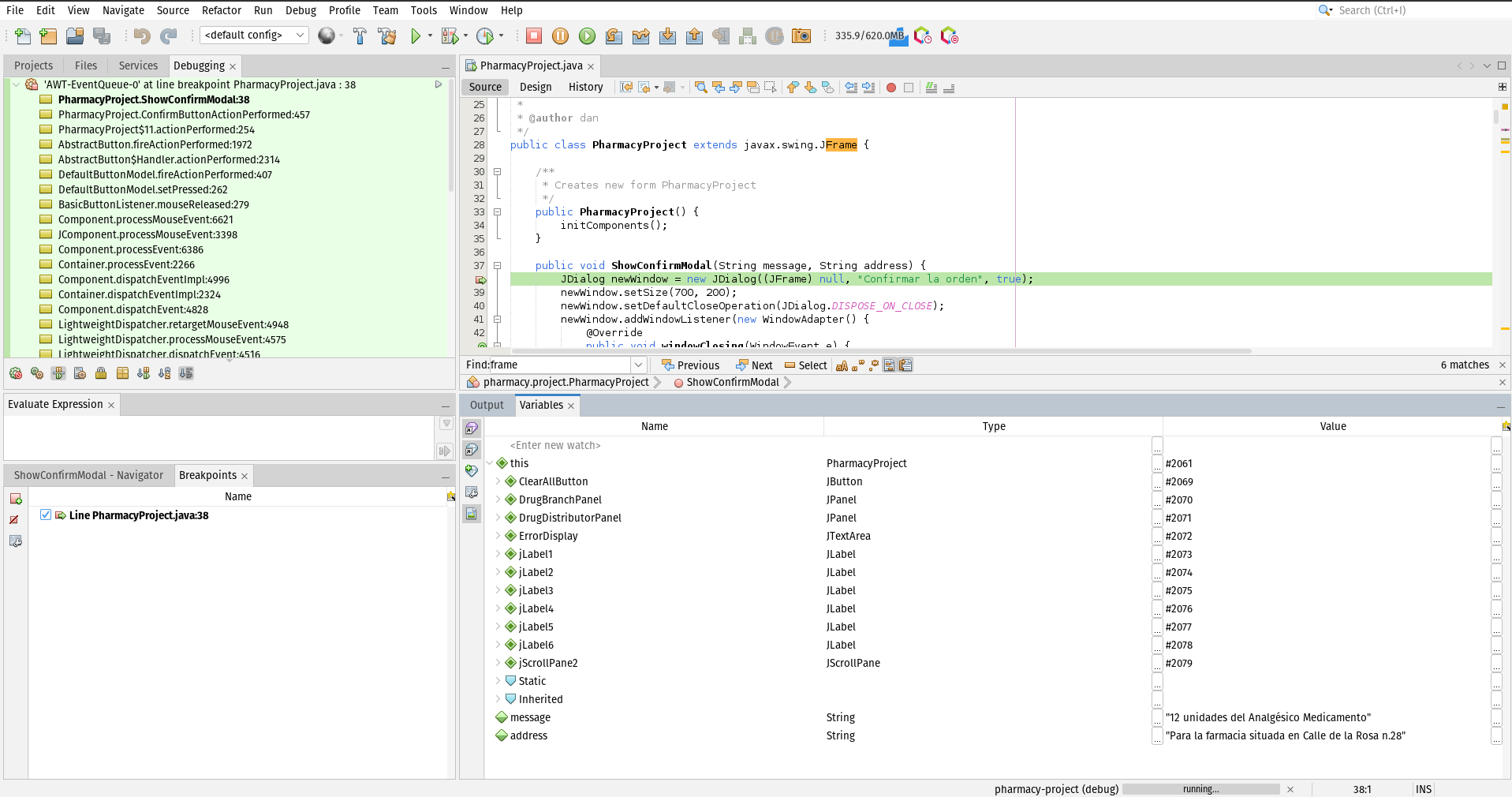


*Figura 5: campos sin seleccionar.*



*Figura 6: datos de tipo incorrecto.*

En el proceso de prueba del software, se agregan *breakpoints* para depurar la aplicación y comprobar los datos ingresados hasta ese punto en la ejecución (*Figura 7)*



*Figura 7: depuración con breakpoints.*

**Conclusiones**

1. Se puede evidenciar que con Java y Swing se pueden crear aplicativos funcionales en espacios de tiempo limitados, que pueden realizar procesos exitosos para el usuario y pueden automatizar procesos grandes en pocos pasos.
2. También se evidencia que la librería Swing contiene todos los elementos necesarios para crear los aplicativos, sin tener que recurrir a librerías externas adicionales.
3. Se evidencia que la depuración a través de *breakpoints* de código es efectiva y ayuda a la correcta escritura de código.

**Referencias**

* Package javax.swing. javax.swing (Java Platform SE 7 ). (2020, June 24). https://docs.oracle.com/javase/7/docs/api/javax/swing/package-summary.html
* JDK 21 documentation - home. Oracle Help Center. (2023, September 19). https://docs.oracle.com/en/java/javase/21/index.html