

Unidad 1: Construcción de la interfaz gráfica.

OE1.1. Utilizar una arquitectura de tres capas para el desarrollo de un programa de computador, repartiendo de manera adecuada las responsabilidades entre la interfaz de usuario, el control de la interfaz y el modelo. El estudiante deberá poder explicar la importancia de mantener separadas las clases de estos tres dominios.

OE1.2. Construir las clases que implementan una interfaz de usuario.

OE1.3. Aplicar la técnica de descomposición de requerimientos para cumplir con la funcionalidad de un programa de computador.

Entregables. Unidad 2.

1. Requerimientos (completarlos).
2. Diagrama de clases de modelo e interfaz (no generado automáticamente). El modelo debe incluir las clases de las Excepciones a que haya lugar y su relación con las demás clases.
3. Diseño de Casos de Pruebas Unitarias y Diagrama de Clases de las Pruebas Unitarias.
4. Implementación completa de todos los requerimientos en Java incluyendo la generación y el manejo de Excepciones.
5. Implementación de todas las pruebas unitarias automáticas corriendo perfectamente.

➤ Solución:

1. Especificación de los requerimientos funcionales:

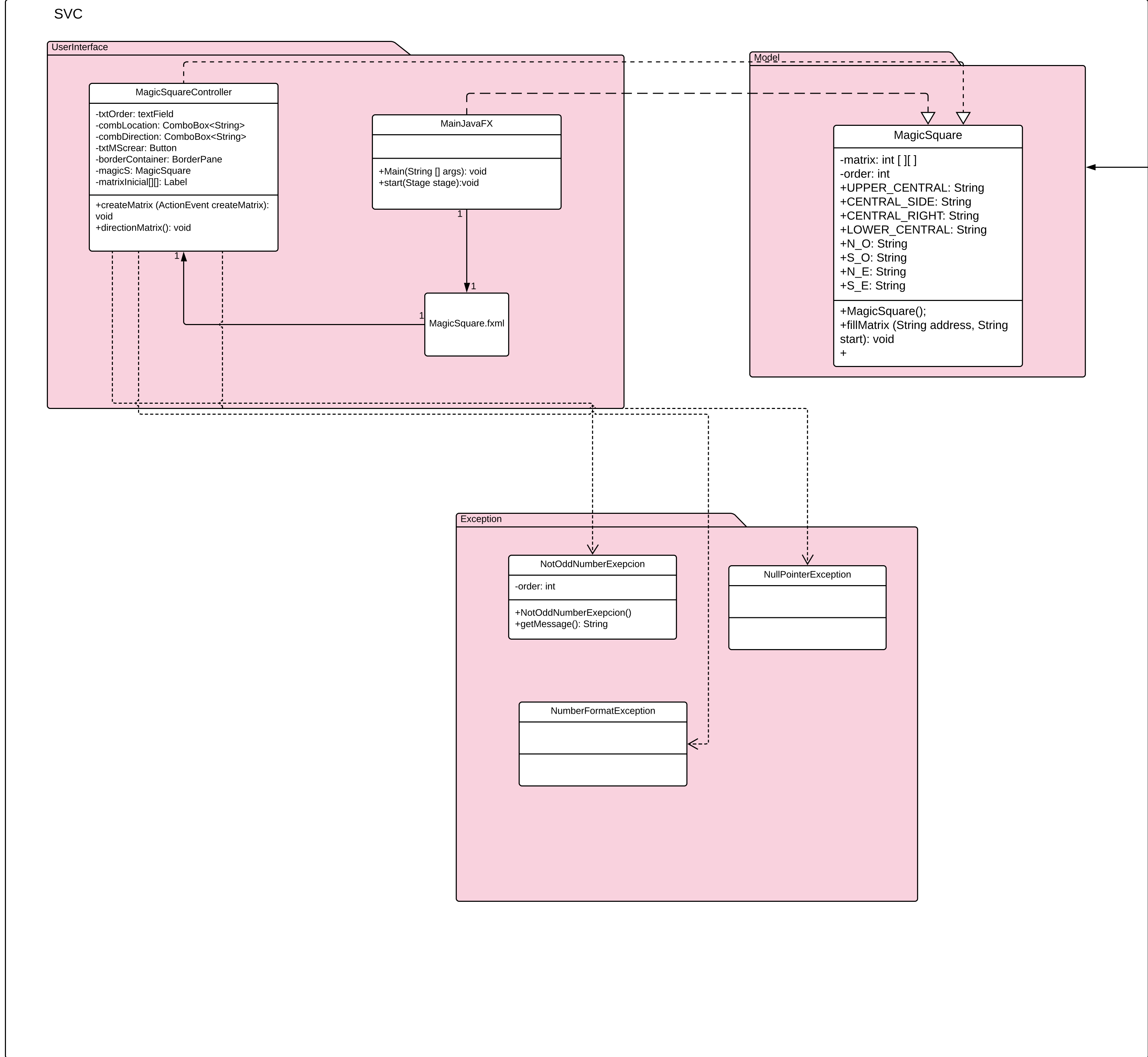
<b>Nombre</b>	R1: generar un cuadrado mágico automáticamente después de dar el orden (número impar).
<b>Resumen</b>	En la ventana de inicio, el usuario debe ingresar un número que necesariamente debe ser impar, y automáticamente se debe generar una matriz de la cual cada uno de sus lados es del tamaño del número ingresado.
<b>Entradas</b>	Un numero entero impar
<b>Resultado</b>	Se espera que al ingresar el orden se genere una matriz de la cual cada lado es del tamaño del orden ingresado.

<b>Nombre</b>	R2: preguntar al usuario en qué ubicación y hacia qué dirección desea iniciar.
<b>Resumen</b>	Para llenar la matriz de forma correcta se debe tener la ubicación (el centro de cualquiera de los lados del cuadro) y la dirección (que es hacia los lados, después arriba abajo y viceversa). Estos datos deben ser ingresados por el usuario.
<b>Entradas</b>	Una de las opciones del menú ubicación. Una de las opciones del menú dirección.
<b>Resultado</b>	Al tener la ubicación y la dirección con que se llena la matriz, se procede a llenarla.

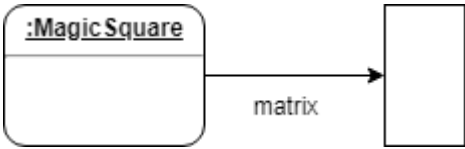
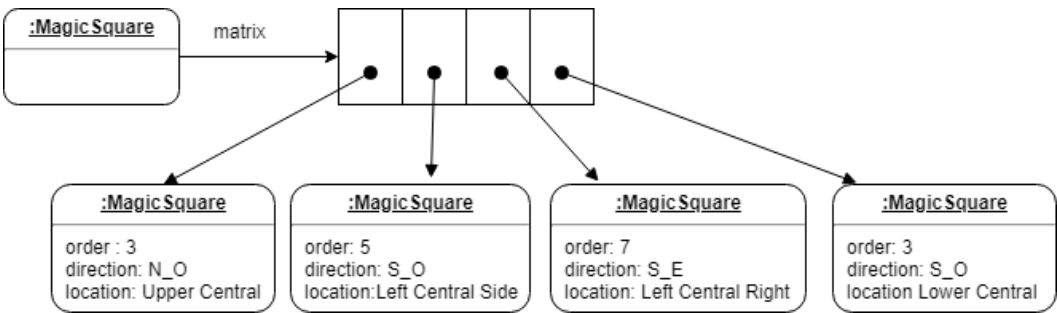
<b>Nombre</b>	R3: generar pruebas eficaces para la comprobación del funcionamiento eficaz del programa.
<b>Resumen</b>	Para saber si el programa funciona correctamente, se deben realizar las pruebas necesarias para que este funcione, por consiguiente se deben realizar las pruebas que se crean convenientes para corroborar el perfecto funcionamiento del programa.
<b>Entradas</b>	Order. Direction. Location.
<b>Resultado</b>	Se comprueba si el programa funciona correctamente.

<b>Nombre</b>	R4: descubrir las posibles excepciones que pueda lanzar el programa, para así tener un funcionamiento más eficaz al momento de la ejecución del programa.
<b>Resumen</b>	El programa debe estar en la capacidad de mostrar y atrapar las excepciones que este genera.
<b>Entradas</b>	
<b>Resultado</b>	El programa maneja las excepciones propias y automáticas correctamente.

## 2. Diagrama de clases



### Configuración de los Escenarios

Nombre	Clase	Escenario
setupScenary1	MagicSquare Test	
setupScenary2	MagicSquare Test	

### Diseño de Casos de Prueba:

<b>Objetivo de la Prueba:</b> Al ingresar un número, se genere una matriz de ese tamaño.				
Clase	Método	Escenario	Valores de Entrada	Resultado
MagicSquare	MagicSquare()	setupScenary1	Order = 3	Se genera exitosamente una matriz del tamaño esperado.

<b>Objetivo de la Prueba:</b> Se genera un cuadrado mágico según las especificaciones brindadas por el usuario.				
Clase	Método	Escenario	Valores de Entrada	Resultado
MagicSquare	fillMatrix()	setupScenary2	Order = 3 Direction: N_O Location: Upper Central	Se genera exitosamente un cuadrado mágico, según las especificaciones ingresadas por el usuario

<b>Objetivo de la Prueba:</b> Verificar que no se crea una matriz con un numero par.				
Clase	Método	Escenario	Valores de Entrada	Resultado
MagicSquare	fillMatrix()	setupScenary2	Order: 4 Location: Lower Central Direction: S_O	No se crea un cuadro mágico con el orden ingresado.

**Objetivo de la Prueba:**

Clase	Método	Escenario	Valores de Entrada	Resultado
MagicSquare	fillMatrix()	SetupScenario2	Order: 5 Location: "Lower Central" Direction: S_O	Se genera exitosamente un cuadrado mágico, según las especificaciones ingresadas por el usuario

**Objetivo de la Prueba:**

Clase	Método	Escenario	Valores de Entrada	Resultado
MagicSquare	fillMatrix()	SetupScenario2	Order: 7 Location: "Left Central Right" Direction: S_E	Se genera exitosamente un cuadrado mágico, según las especificaciones ingresadas por el usuario

**Objetivo de la Prueba:**

Clase	Método	Escenario	Valores de Entrada	Resultado
MagicSquare	fillMatrix()	SetupScenario2	Order: 3 Location: "Lower Central" Direction: S_O	Se genera exitosamente un cuadrado mágico, según las especificaciones ingresadas por el usuario