

## DOCUMENTACION

### Requerimientos funcionales:

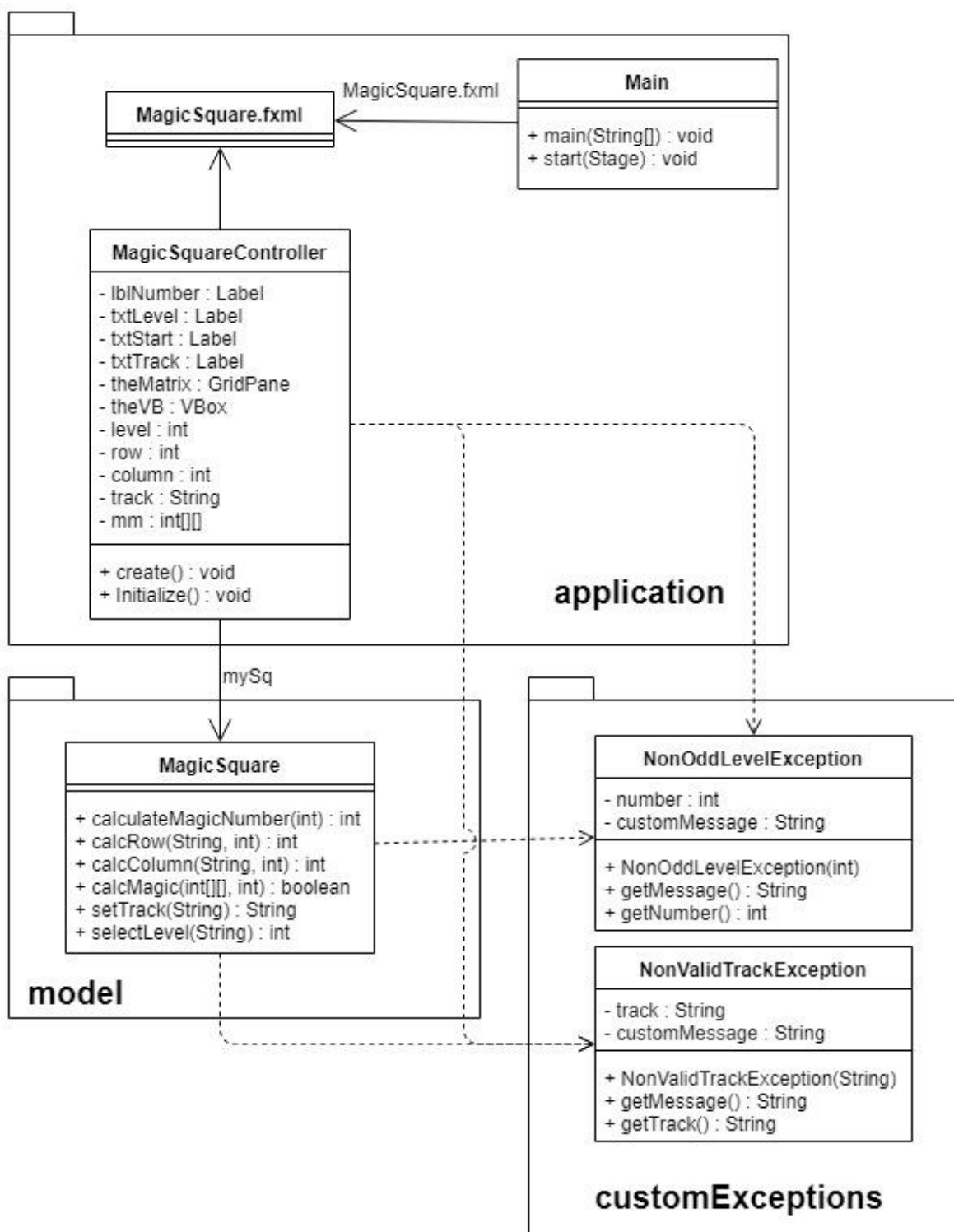
Nombre	R0: Generar la aplicación utilizando javafx
Resumen	Se genera automáticamente un cuadrado mágico según el orden, celda de inicio y dirección dada por el usuario.
Entrada	Int : orden del cuadrado String : celda perimetral de inicio String : dirección de llenado
Salida	Un cuadrado mágico dado a mostrar a través de un GridPane.

Nombre	R1: Generar un cuadrado mágico según especificaciones
Resumen	Se genera automáticamente un cuadrado mágico según el orden, celda de inicio y dirección dada por el usuario.
Entrada	Int : orden del cuadrado String : celda perimetral de inicio String : dirección de llenado
Salida	Un cuadrado mágico dado a mostrar a través de un GridPane.

Nombre	R2: Calcular el número mágico del cuadrado
Resumen	A través del orden se calcula la constante mágica del cuadrado utilizando la formula dada.
Entrada	Int : orden
Salida	Constante mágica del cuadrado.

Nombre	R3: Determinar si el cuadrado es mágico
Resumen	Se cuentan las filas, columnas y diagonal del cuadrado para determinar si la suma de cada una de estas da la constante mágica.
Entrada	Int[][] : matriz del cuadrado mágico generado Int : constante mágica
Salida	Confirmación si es o no mágico.

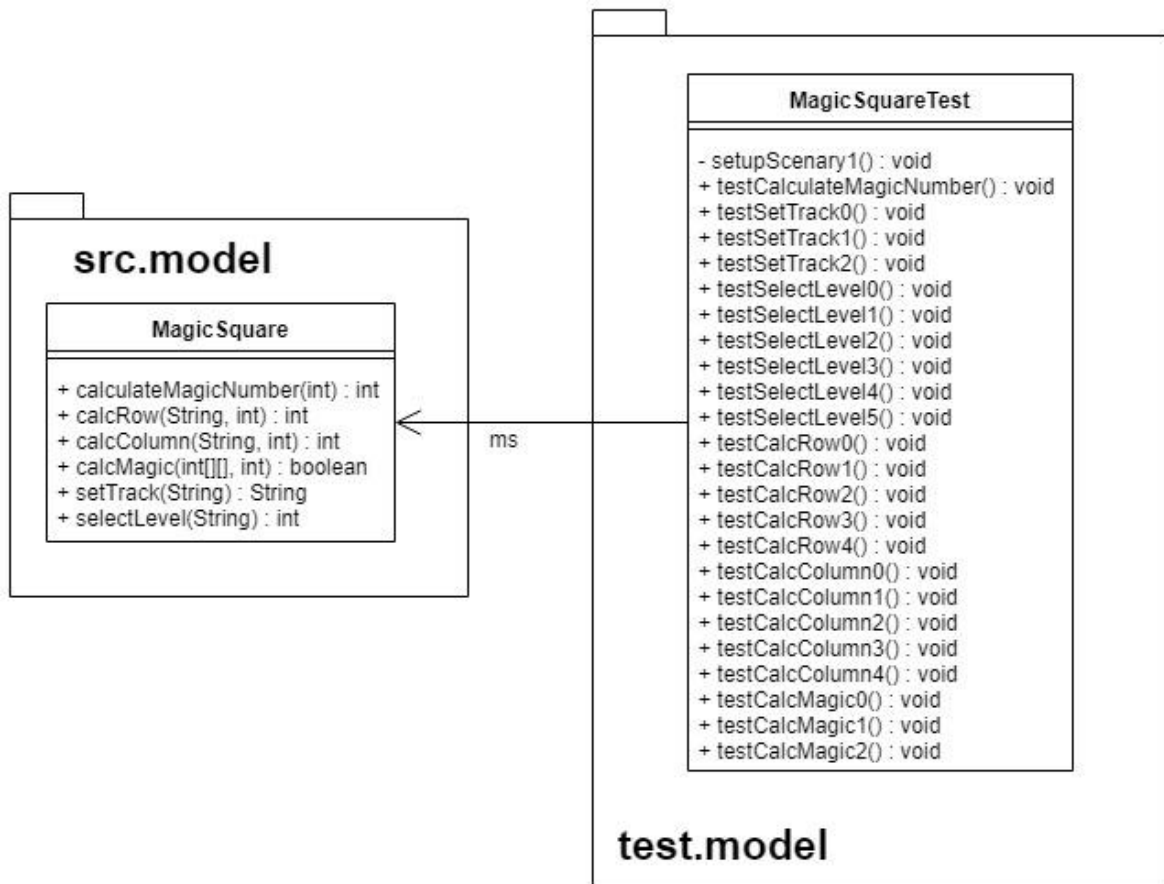
## Diagrama de clases:



**Tabla de trazabilidad del análisis al diseño:**

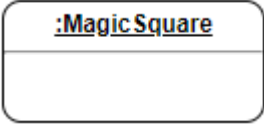
Requerimiento	Clase	Método
<b>Generar la aplicación utilizando javafx</b>	MagicSquare.fxml	
	Main	main(String[]) : void start(Stage) : void
<b>Generar un cuadrado mágico según especificaciones</b>	MagicSquareController	initialize() : void create() : void
	MagicSquare	setTrack(String) : String SelectLevel(String) : int calcRow(String, int) : int calcColumn(String, int) : int
<b>Calcular el número mágico del cuadrado</b>	MagicSquare	calculateMagicNumber(int) : int
<b>Determinar si el cuadrado es mágico</b>	MagicSquare	calcMagic(int[][], int) : boolean

**Diagrama de clases de pruebas unitarias:**



## Casos de pruebas:

Configuración de los escenarios:

Nombre	Clase	Escenario
setupScenary1	MagicSquareTest	

Diseño de casos de prueba:

Objetivo de la prueba: verificar el correcto uso del método				
Clase	Método	Escenario	Valores de entrada	Resultado
MagicSquare	calculateMagicNumber	setupScenary1	n=3	15 El número mágico se calcula correctamente

Objetivo de la prueba: verificar el correcto uso del método				
Clase	Método	Escenario	Valores de entrada	Resultado
MagicSquare	setTrack	setupScenary1	t="nw"	nw se devuelve correctamente la dirección

Objetivo de la prueba: verificar la correcta ejecución de la excepción NonValidTrack				
Clase	Método	Escenario	Valores de entrada	Resultado
MagicSquare	setTrack	setupScenary1	t="epa"	NonValidTrack La entrada no es una dirección correcta

Objetivo de la prueba: verificar la correcta ejecución de la excepción NonValidTrack utilizando un carácter especial				
Clase	Método	Escenario	Valores de entrada	Resultado
MagicSquare	setTrack	setupScenary1	t="&"	NonValidTrack La entrada no es una dirección correcta

<b>Objetivo de la prueba: verificar el correcto uso del método</b>				
Clase	Método	Escenario	Valores de entrada	Resultado
MagicSquare	selectLevel	setupScenary1	l=3	3 Se devuelve el nivel del cuadro mágico como un int

<b>Objetivo de la prueba: verificar la correcta ejecución de la excepción NonOddLevelException</b>				
Clase	Método	Escenario	Valores de entrada	Resultado
MagicSquare	selectLevel	setupScenary1	l=2	NonOddLevelException El valor introducido no es un número impar

<b>Objetivo de la prueba: verificar la correcta ejecución de la excepción IllegalArgumentException con un decimal impar</b>				
Clase	Método	Escenario	Valores de entrada	Resultado
MagicSquare	selectLevel	setupScenary1	l=3.5	IllegalArgumentException El valor introducido no es un entero

<b>Objetivo de la prueba: verificar la correcta ejecución de la excepción IllegalArgumentException con un decimal par</b>				
Clase	Método	Escenario	Valores de entrada	Resultado
MagicSquare	selectLevel	setupScenary1	l=2.5	IllegalArgumentException El valor introducido no es un entero

<b>Objetivo de la prueba: verificar la correcta ejecución de la excepción IllegalArgumentException con un valor negativo</b>				
Clase	Método	Escenario	Valores de entrada	Resultado
MagicSquare	selectLevel	setupScenary1	l=-5	IllegalArgumentException El valor introducido no es un entero

**Objetivo de la prueba: verificar la correcta ejecución de la excepción  
IllegalArgumentException con un carácter especial**

Clase	Método	Escenario	Valores de entrada	Resultado
MagicSquare	selectLevel	setupScenary1	l=&	IllegalArgumentException El valor introducido no es un entero

**Objetivo de la prueba: verificar el correcto uso del método**

Clase	Método	Escenario	Valores de entrada	Resultado
MagicSquare	calcRow	setupScenary1	x="up" y=3	0 Se devuelve correctamente la fila obtenida

**Objetivo de la prueba: verificar el correcto uso del método**

Clase	Método	Escenario	Valores de entrada	Resultado
MagicSquare	calcRow	setupScenary1	x="down" y=3	2 Se devuelve correctamente la fila obtenida

**Objetivo de la prueba: verificar el correcto uso del método**

Clase	Método	Escenario	Valores de entrada	Resultado
MagicSquare	calcRow	setupScenary1	x="left" y=3	1 Se devuelve correctamente la fila obtenida

**Objetivo de la prueba: verificar el correcto uso del método**

Clase	Método	Escenario	Valores de entrada	Resultado
MagicSquare	calcRow	setupScenary1	x="right" y=3	1 Se devuelve correctamente la fila obtenida

**Objetivo de la prueba: verificar la correcta ejecución de la excepción  
IllegalArgumentExcepcion**

Clase	Método	Escenario	Valores de entrada	Resultado
MagicSquare	calcRow	setupScenary1	x="epa" y=3	IllegalArgumentExcepcion La entrada no se encuentra entre las opciones permitidas

**Objetivo de la prueba: verificar el correcto uso del método**

Clase	Método	Escenario	Valores de entrada	Resultado
MagicSquare	calcColumn	setupScenary1	x="up" y=3	1 Se devuelve correctamente la fila obtenida

**Objetivo de la prueba: verificar el correcto uso del método**

Clase	Método	Escenario	Valores de entrada	Resultado
MagicSquare	calcColumn	setupScenary1	x="down" y=3	1 Se devuelve correctamente la fila obtenida

**Objetivo de la prueba: verificar el correcto uso del método**

Clase	Método	Escenario	Valores de entrada	Resultado
MagicSquare	calcColumn	setupScenary1	x="left" y=3	0 Se devuelve correctamente la fila obtenida

**Objetivo de la prueba: verificar el correcto uso del método**

Clase	Método	Escenario	Valores de entrada	Resultado
MagicSquare	calcColumn	setupScenary1	x="right" y=3	2 Se devuelve correctamente la fila obtenida

**Objetivo de la prueba: verificar la correcta ejecución de la excepción  
IllegalArgumentException**

Clase	Método	Escenario	Valores de entrada	Resultado
MagicSquare	calcColumn	setupScenary1	x="epa" y=3	IllegalArgumentException La entrada no se encuentra entre las opciones permitidas

**Objetivo de la prueba: verificar el correcto uso del método**

Clase	Método	Escenario	Valores de entrada	Resultado
MagicSquare	calcMagic	setupScenary1	mm = int[][] //mágico mn = 15	True El cuadrado es mágico

**Objetivo de la prueba: verificar el correcto uso del método con un número mágico incorrecto**

Clase	Método	Escenario	Valores de entrada	Resultado
MagicSquare	calcMagic	setupScenary1	mm = int[][] //mágico mn = 10	False El cuadrado no es mágico porque el número mágico es incorrecto

**Objetivo de la prueba: verificar el correcto uso del método con un cuadrado mágico incorrecto**

Clase	Método	Escenario	Valores de entrada	Resultado
MagicSquare	calcMagic	setupScenary1	mm = int[][] //noMágico mn = 15	False El cuadrado no es mágico porque no está correctamente construido