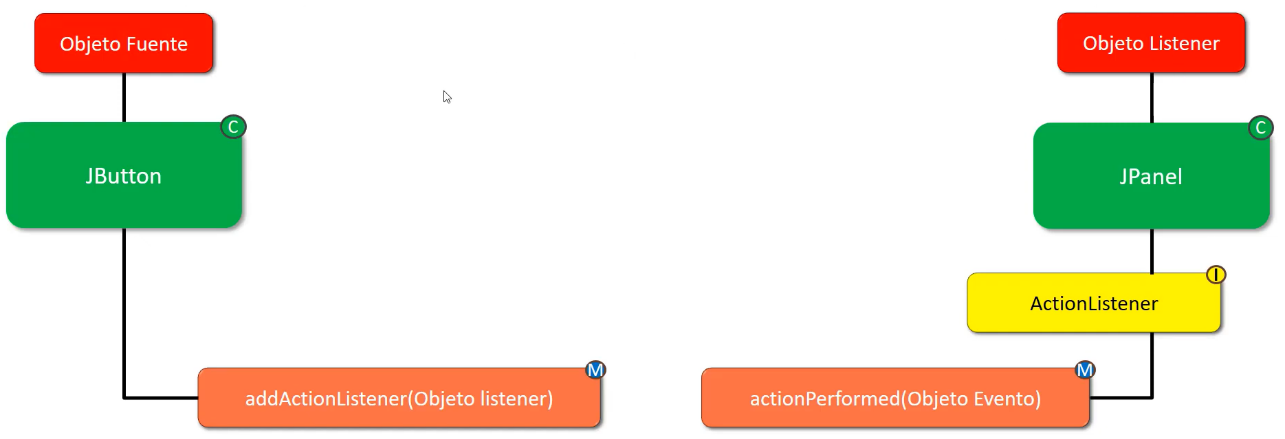
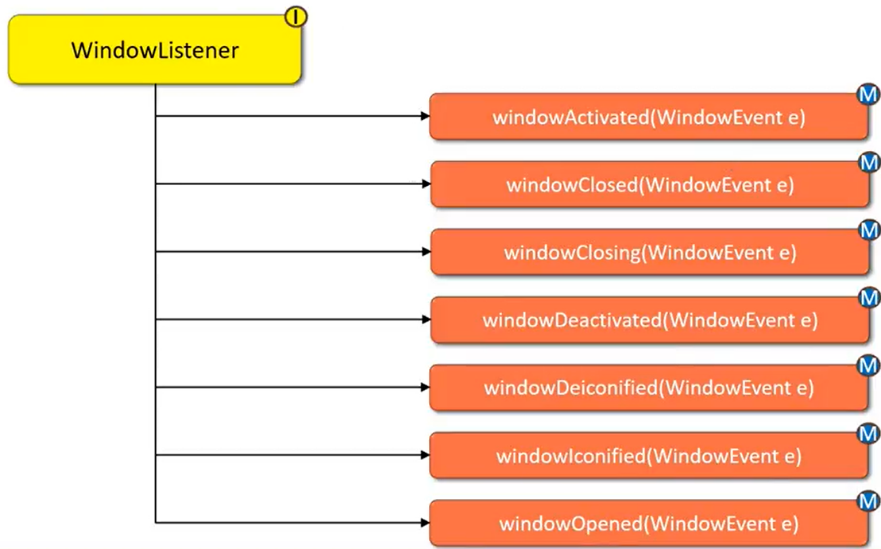
**EVENTOS EN JAVA:**

1. **EVENTOS DE BOTÓN con ActionListener:**



Al accionar el botón que se implemente en el **“JPanel”** se crea un nuevo objeto de tipo evento: **“Objeto Evento”** que lo recibirá “**actionPerformed(Objeto Eevento”** y luego el “**addActionListener(Objeto listener)”** recibirá el objeto de tipo listener que será el **“JPanel”**

1. **EVENTOS DE VENTANA con WindowListener:**

****

El método: **“windowActivated(WindowEvent e)”** cuando una ventana esta activa delante de la pantalla.

El método **“windowClosed(WindowEvent e)”** cuando se cierra una ventana.

El método **“windowClosing(WindowEvent e)”** cuando la ventana está en proceso de cerrarse.

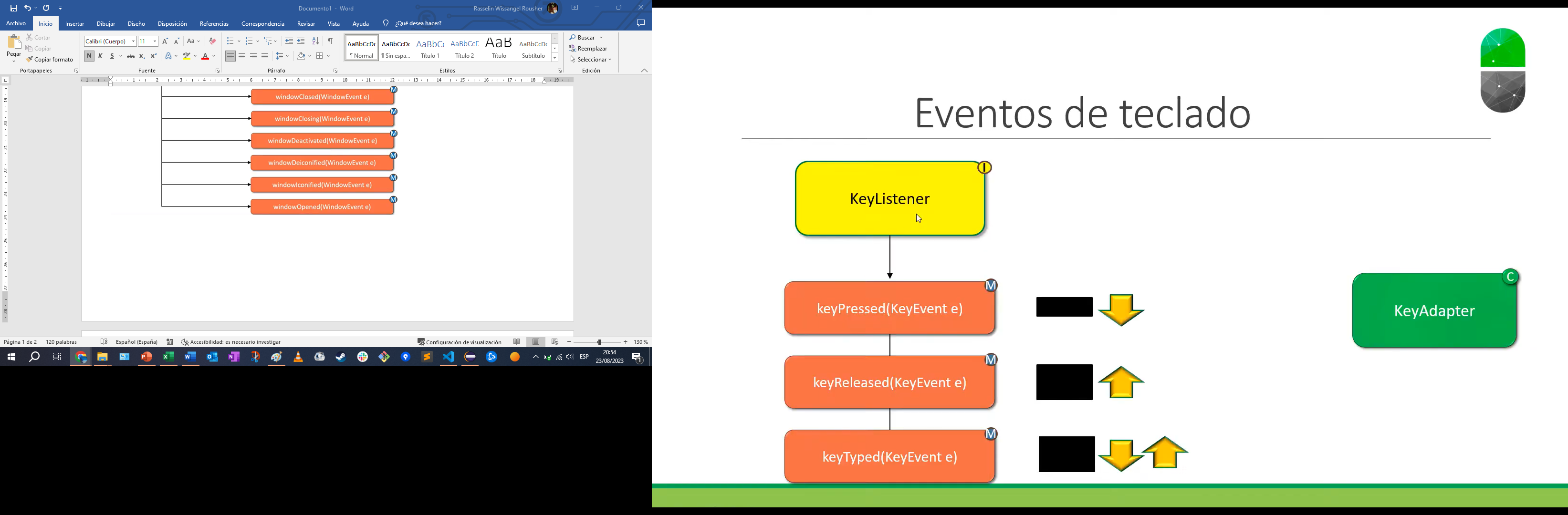
El método **“windowDeactivated(WindowEvent e)”** cuando la ventana deja de estar activada ante la pantalla.

El método **“windowDeiconified(WindowEvent e)”** cuando la ventana se maximiza de haber estado minimizada.

El método **“windowIconified(WindowEvent e)”** cuando la ventana se minimiza.

El método **“windowOpened(WindowEvent e)”** cuando la ventana se abre por primera vez.

1. **EVENTOS DE TECLADO con KeyListener:**

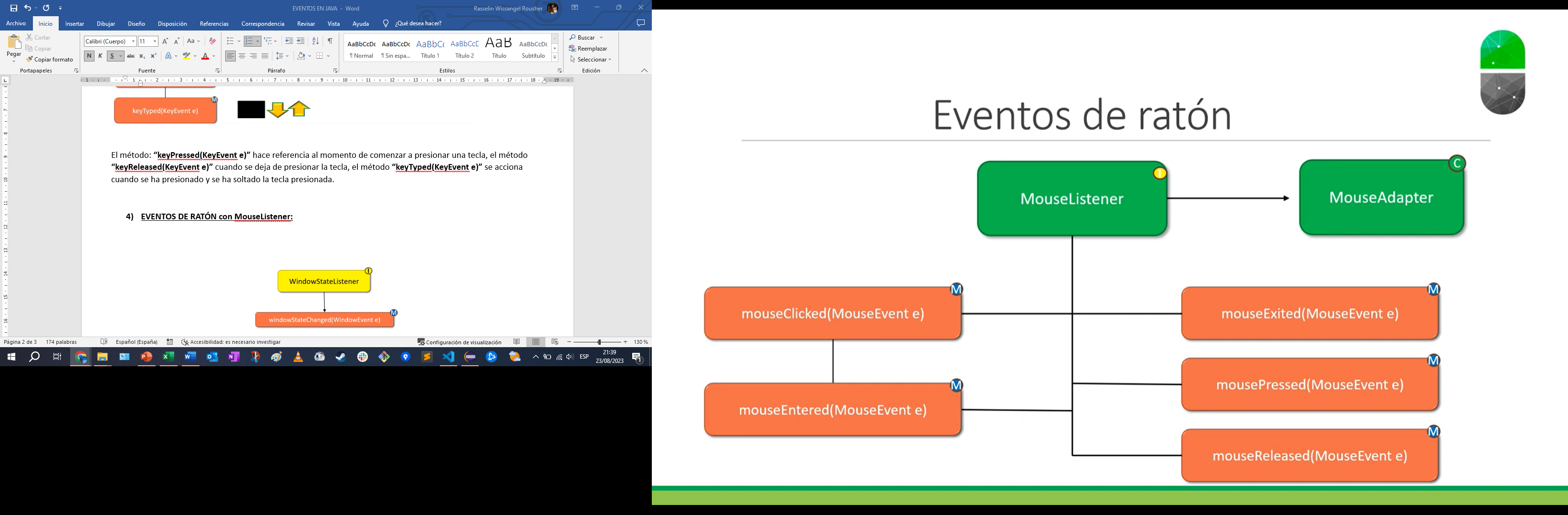


El método: **“keyPressed(KeyEvent e)”** hace referencia al momento de comenzar a presionar una tecla.

El método **“keyReleased(KeyEvent e)”** cuando se deja de presionar la tecla.

El método **“keyTyped(KeyEvent e)”** se acciona cuando se ha presionado y se ha soltado la tecla presionada.

1. **EVENTOS DE RATÓN con MouseListener:**



El método **“mouseClicked(MouseEvent e)”** cuando se presiona el botón izquierdo del ratón sin subir.

El método **“mouseEntered(MouseEvent e)”** cuando entra el ratón dentro de un área que está a la escucha.

El método **“mouseExited(MouseEvent e)”** cuando sale el ratón del interior de un área que está a la escucha.

El método **“mousePressed(MouseEvent e)”** cuando el botón baja, sube y ha sido soltado.

El método **“mouseReleased(MouseEvent e)”** cuando se suelta el botón pulsado.

Hay otra interfaz **“mouseMotionListener”** que contempla otras acciones, como arrastrar un elemento con el ratón y se desplaza.

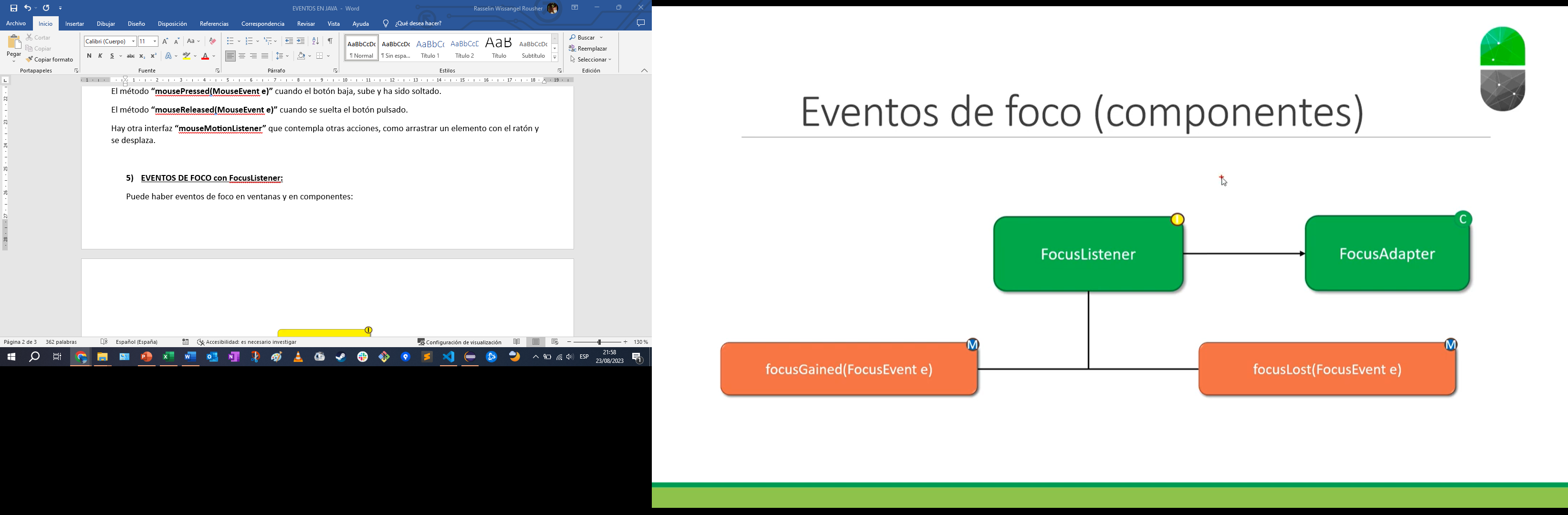
1. **EVENTOS DE FOCO con FocusListener:**

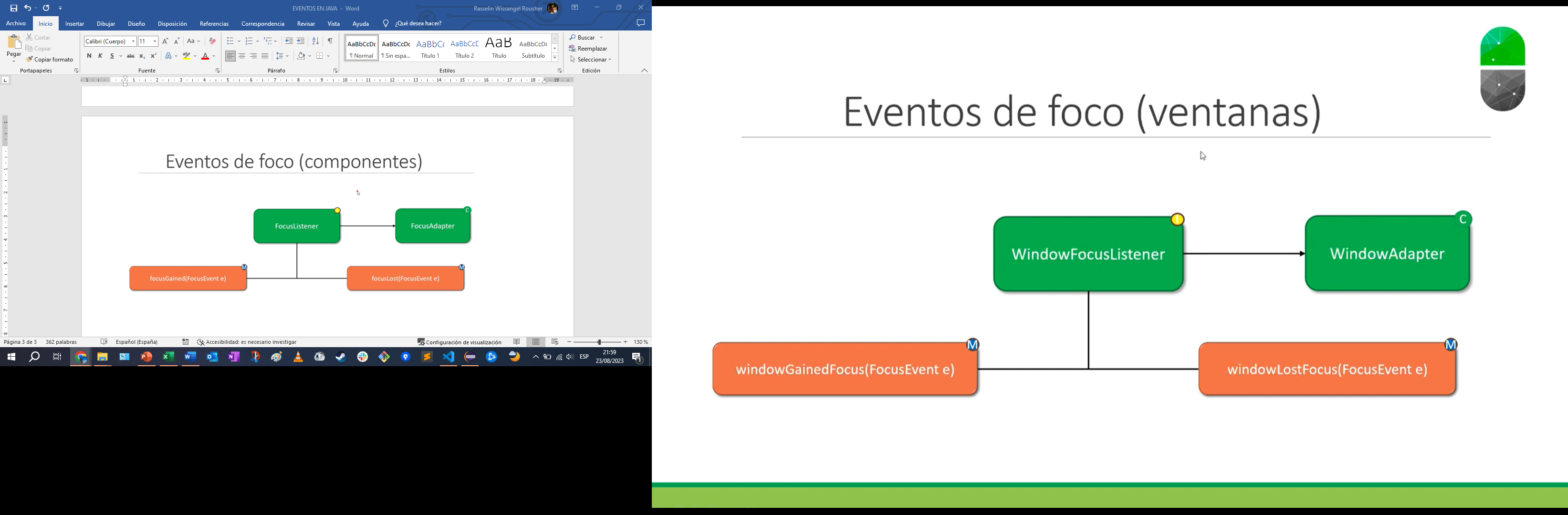
Puede haber eventos de foco en ventanas y en componentes:

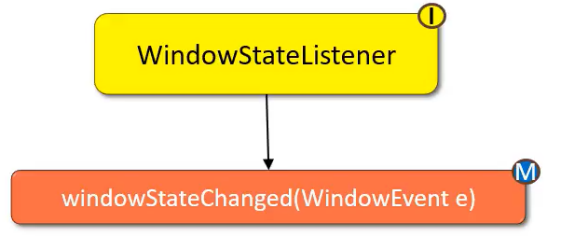
**Componentes**: elementos que hay dentro de una ventana; botones, desplegables…

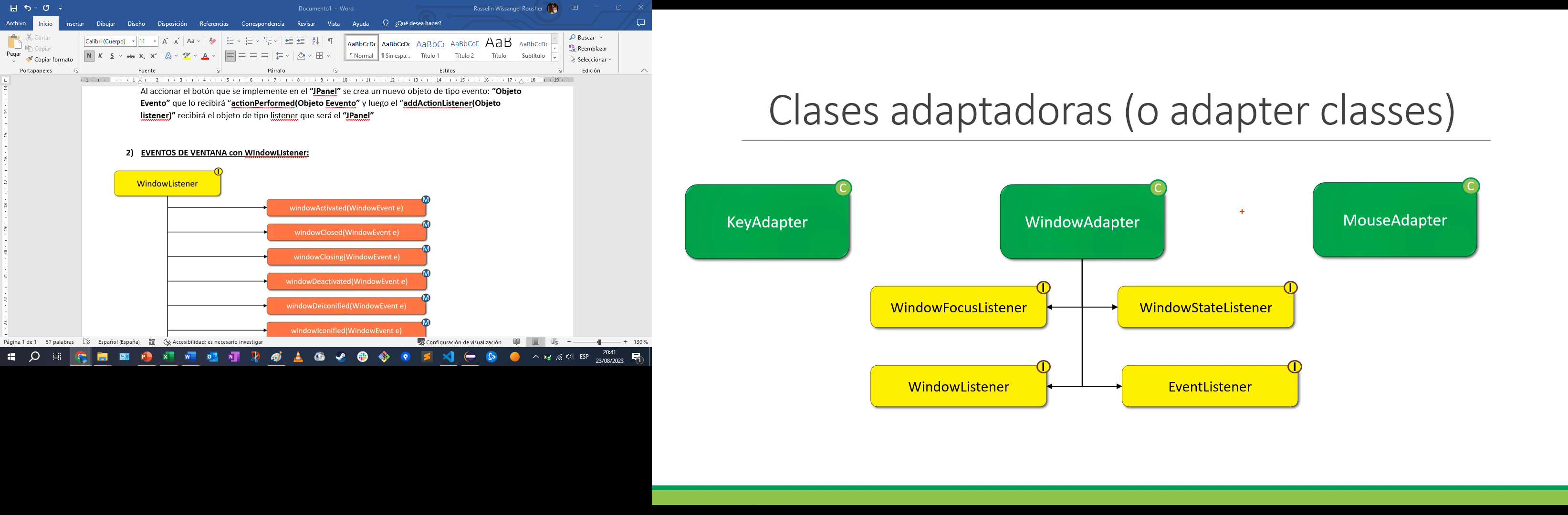
**Ventanas**: es el número de ventanas que se pueden abrir en pantalla.

**Foco**: se define como la ventana activa con la que se esta trabajando en un momento presente tras activarla con el ratón.





****



Las clases adaptadoras, implementan un montón de **inerfaces**, por ejemplo, **WindowAdapter**; que implementa 4 interfaces: **WindowFocusListener**, **WindowStatesListener**, **WindowListener**, **EventListener**. Como son clases y para evitar tener que declarar todos los métodos de una clase con la palabra clase: “**implements**”, se puede heredar de la misma, con la palabra clave: “**extends**”, así sólo necesitaremos declarar los métodos que se necesiten para programar un código.

La clase **“MouseAdapter”** puede implementar las INTERFACES: **MouseListener, MouseMotionListener, MouseWheelListener, EventListener, MouseInputListener**

**CLASE JAVA.TIME**

