**PROGRAMACIÓN ORIENTADA A OBJETOS**

**Herencia e interfaces**

**ADEMÁS, Java desde consola**

**2020-1**

**Laboratorio 3/6**

**CONOCIENDO**

**Número uno.**

1. **¿Cuántos paquetes tiene?**

Tiene 3 paquetes. (1. Autómata, 2. Aplicación, 3. Presentación)

1. **¿Cuántas clases tiene en total? ¿Cuántas tienen fuentes?**

Tiene cinco clases en total,

1. AutomataGUI, sí tiene fuente.
2. AutomataCelular, no tiene fuente.
3. Celula, sí tiene fuente.
4. Elemento, no tiene fuete.
5. FotoAutomata, sí tiene fuente.
6. **¿Cuál es la clase ejecutiva? ¿Por qué?**

La clase ejecutiva es “AutomataGUI” porque tiene el método main que conecta un paquete al otro.

**Número dos.**

**Ejecuten el programa. ¿Qué funcionalidades ofrece? ¿Qué hace actualmente? ¿Por qué?**

Por ahora crea la cuadricula inicial porque las clases sin fuentes no están implementadas aún.

**ARQUITECTURA GENERAL.**

1. **Packaje:** Se utiliza para crear los paquetes en el código en curso.

**Import:** Se utiliza para poder utilizar los elementos de un paquete, importando el paquete en el módulo de código en curso.

Los paquetes son el mecanismo que usa Java para facilitar el modularidad del código. Un paquete puede contener una o más definiciones de interfaces y clases, distribuyéndose habitualmente como un archivo.

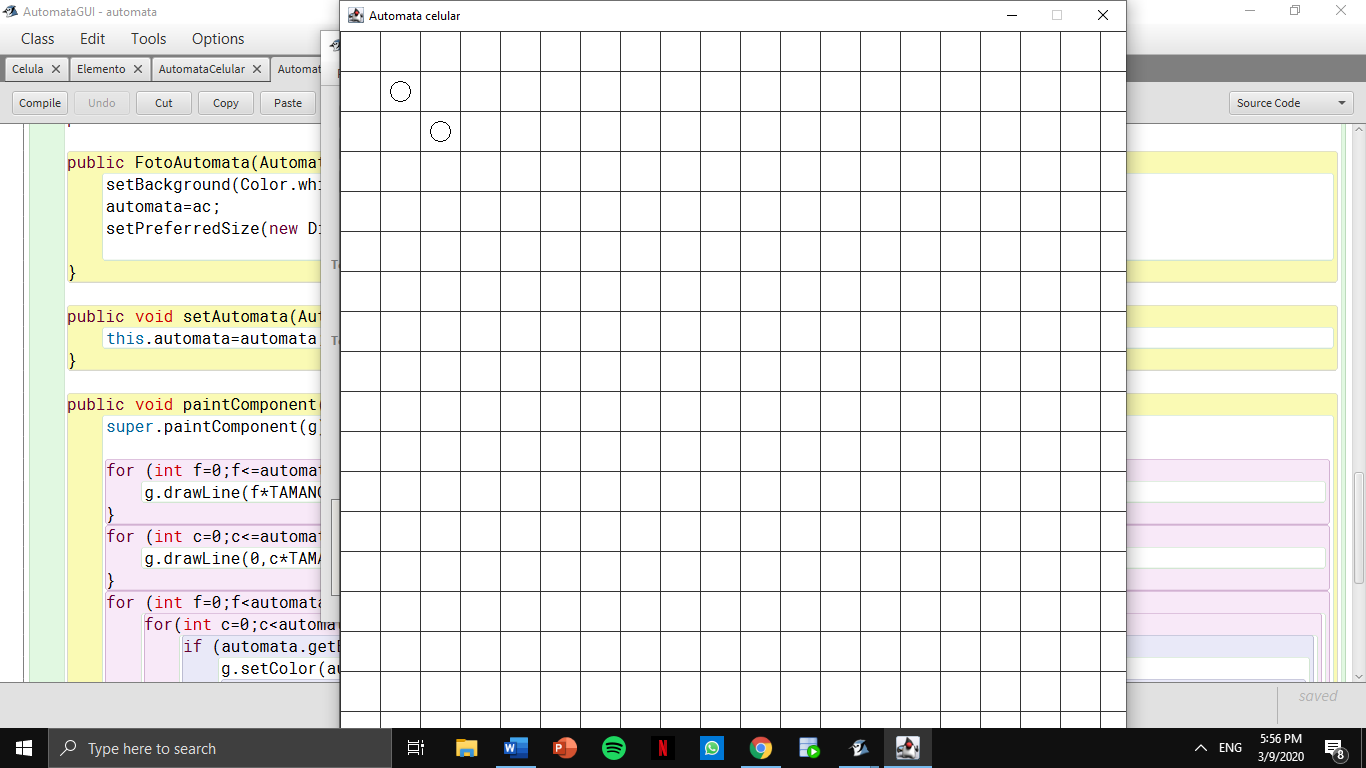
1. El directorio de trabajo que tenemos es donde están contenidos los paquetes, la coincidencia que se obtiene es que ambos contienen el código en curso para trabajar.

**ARQUITECTURA DETALLADA.**

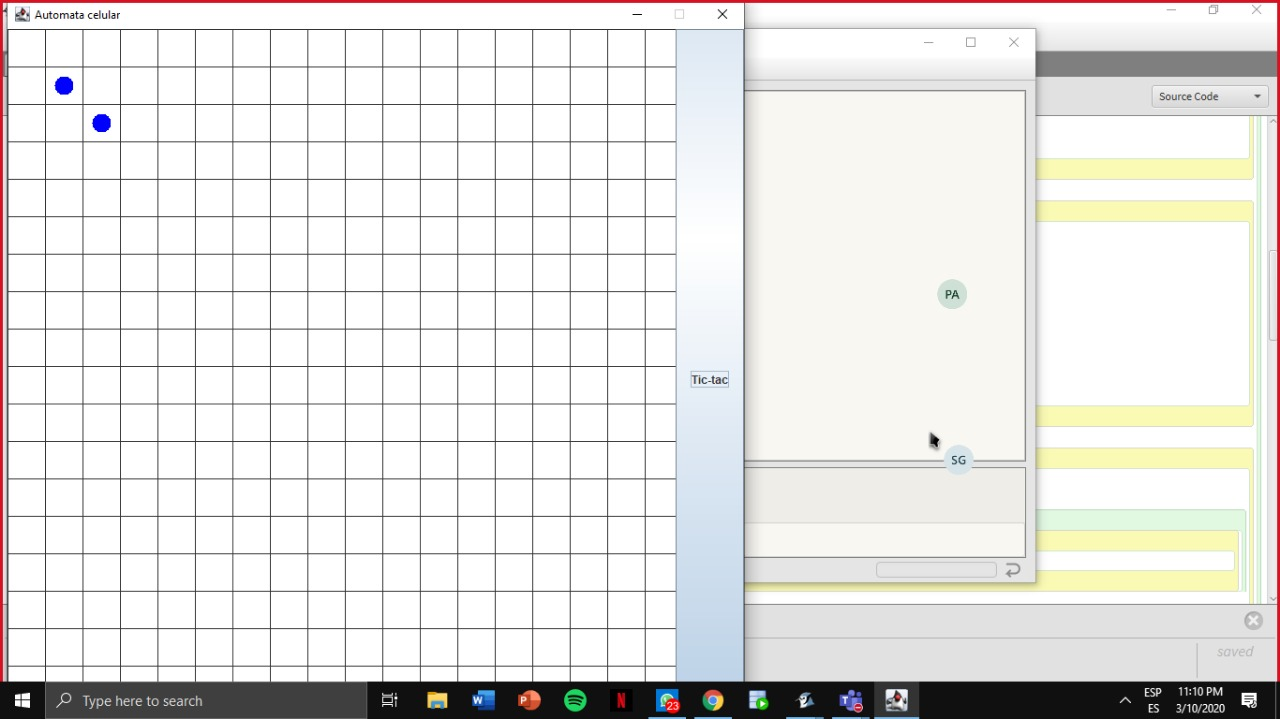
**2.** Consideramos que las pruebas deben estar en el paquete de presentación ya que en la clase AutomataGUI está el método “main” que llamará los demás métodos y así se probaran todos.

**CICLO 1. INICIANDO CON LAS CÉLULAS NORMALES**

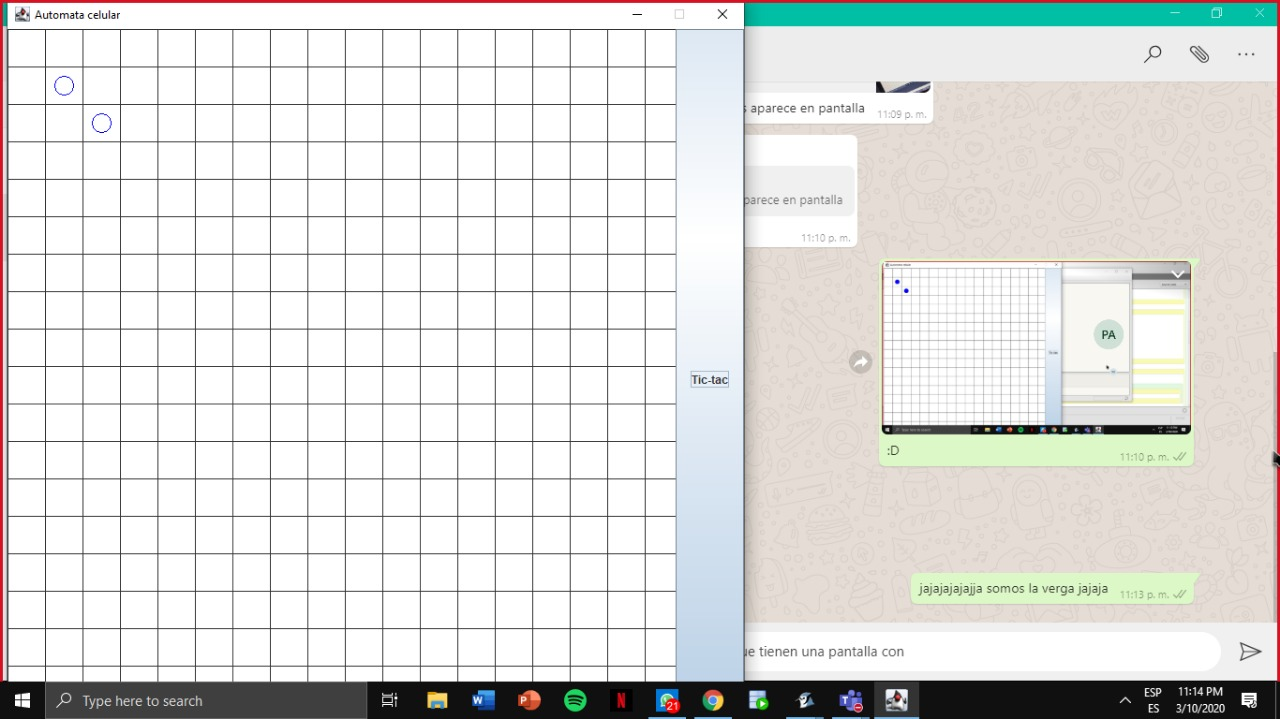
1. El tipo de colección que usa es una matriz de tipo Elemento, si puede recibir células ya que al estar llamando a la matriz de elementos esta es una interfaz y tiene una conexión con célula
2. Las otras clases que definen a Célula son Autómata celular y la interface Elemento la interface tiene los métodos que esta compartiendo con Célula sin especificarlos.
3. Las células conocen su color, el autómata en la que viven, la posición en la que están en ese autómata, su estado actual y el estado que van a tomar en el siguiente instante, puede hacer lo mismo que la interface menos el getForma debe aprender a obtener la forma en la que actualmente esta.
4. El elemento tiene los métodos decida (), cambie (), getForma (), getColor (), IsVivo(), puede hacer todas las anteriores menos getForma ya que es una privada y esta inicializada. Debe aprender a….
5. Según la documentación una célula únicamente podría ser de color azul, es una variable protegida la cual decide su color en el constructor de célula, si se quiere cambiar de color solo tiene que cambiarse su inicialización en el constructor y poner el color deseado



1. BlueJ
2. Bluej
3. En el primer y segundo tic tac,

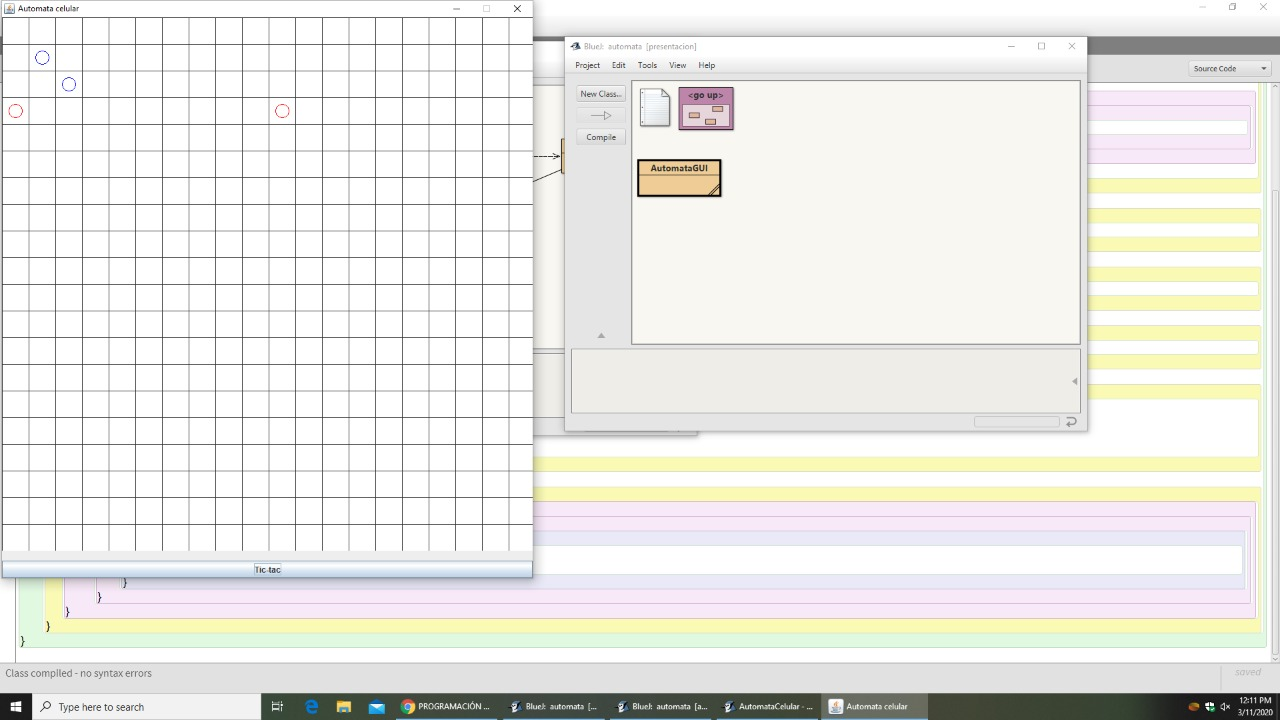


En el tercer tic tac,

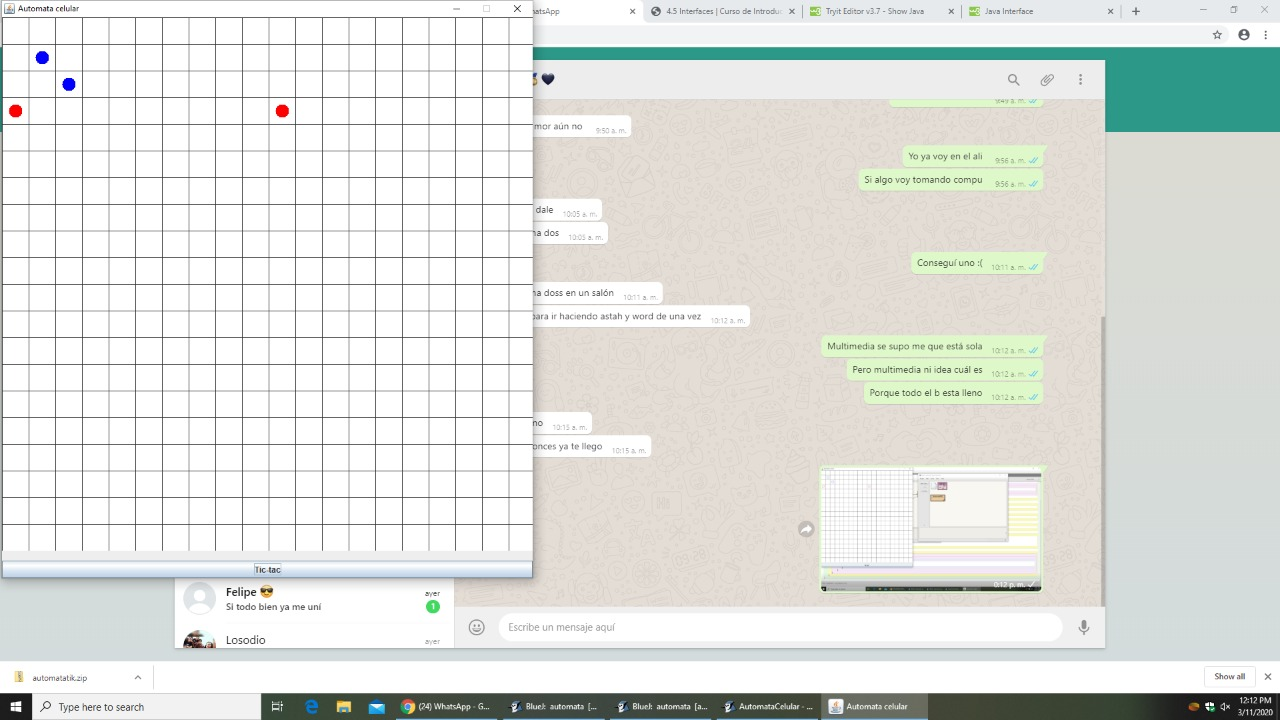


**CICLO 2. INCLUYENDO A LAS CÉLULAS IZQUIERDOSAS**

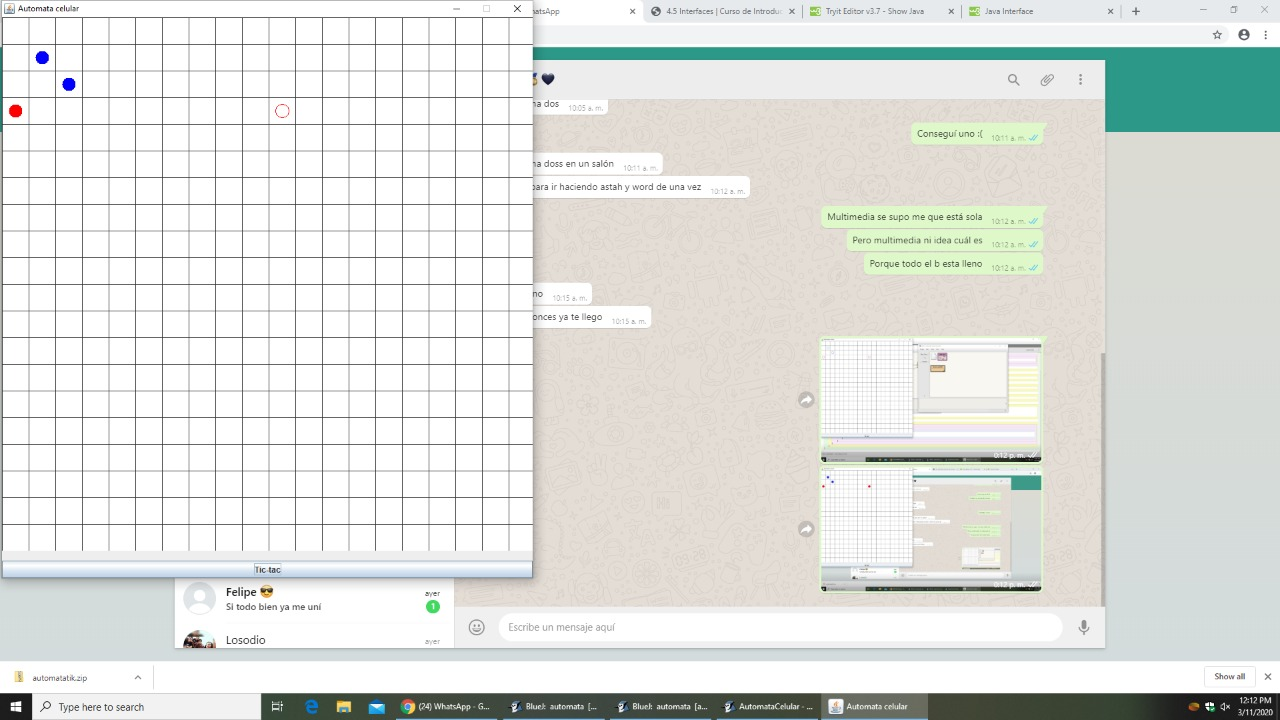
1. Si se tienen dos células izquierdosas en la misma fila, en el primer tic ambas estarían vivas, en el segundo la célula de la izquierda moriría y la derecha seguiría viva, en el último tic tac estarían ambas células muertas.
2. Astah.
3. Para poder identificar la célula izquierdosa dentro del autómata, solo se realizó el cambio de color que depende de un atributo booleano de la célula.
4. Los tic tac funcionarían con en el inciso uno.
5. Blue J
6. Creando,



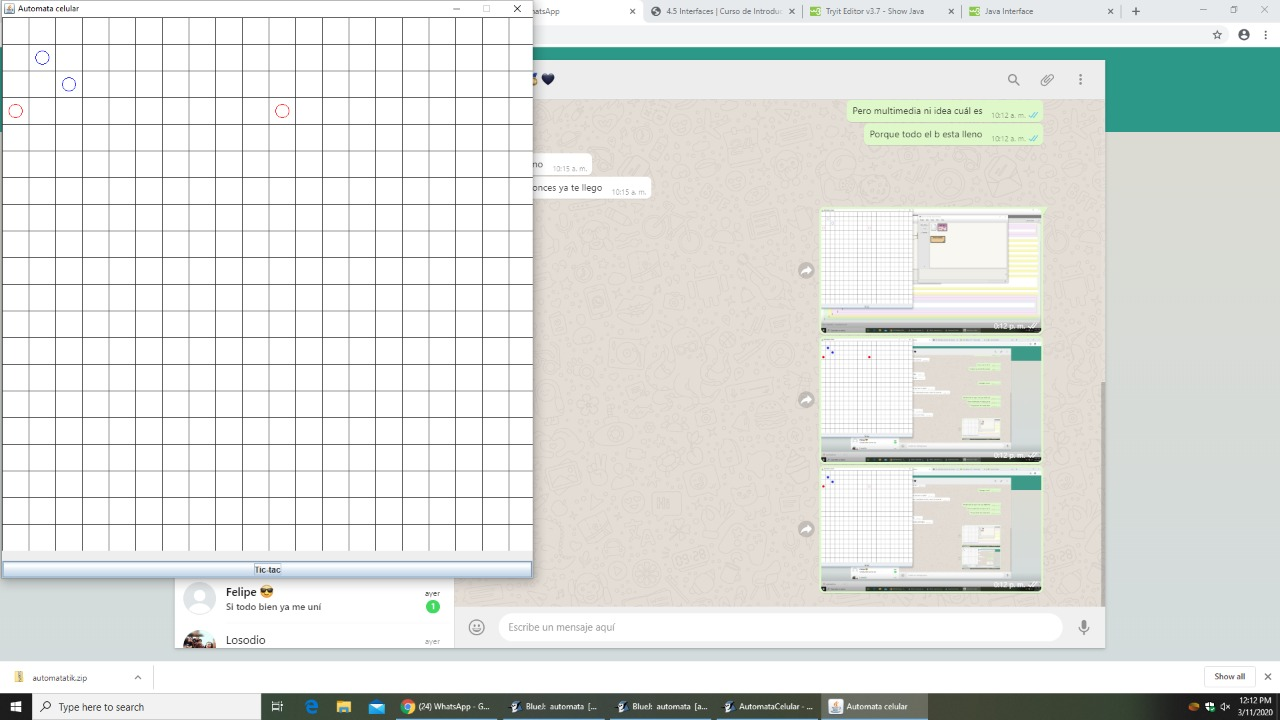
Primer clic,



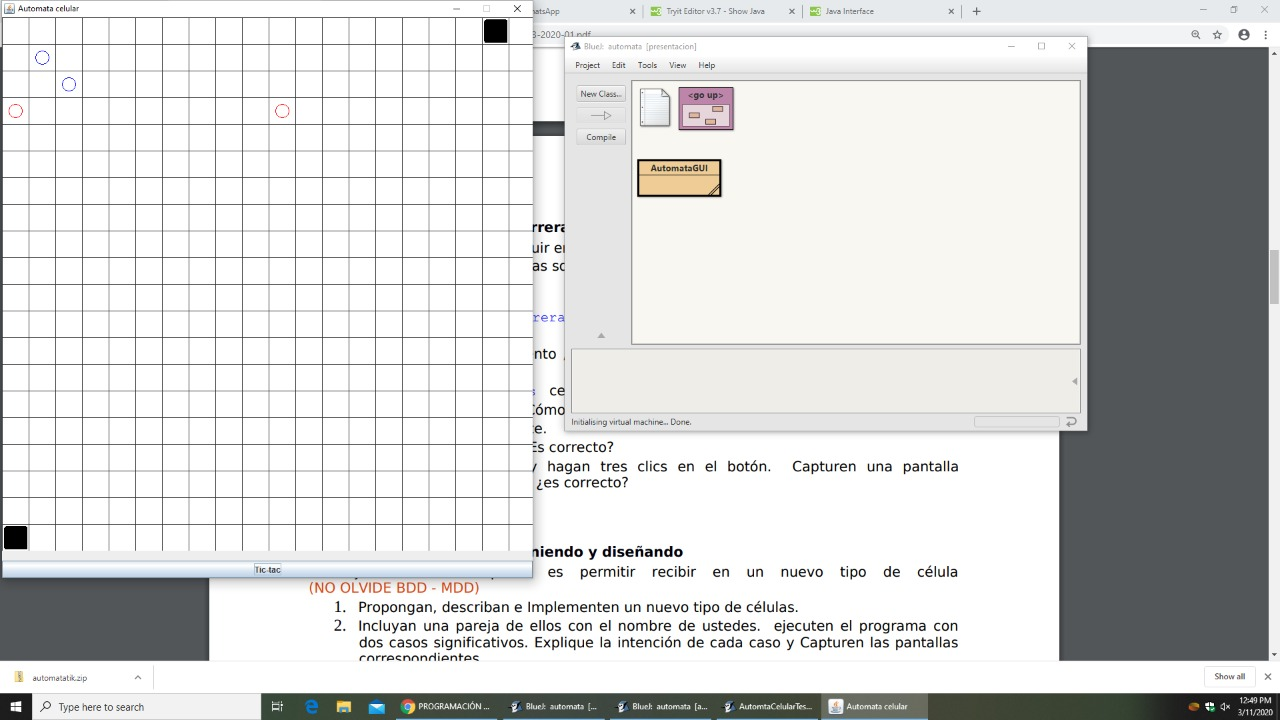
Segundo clic,

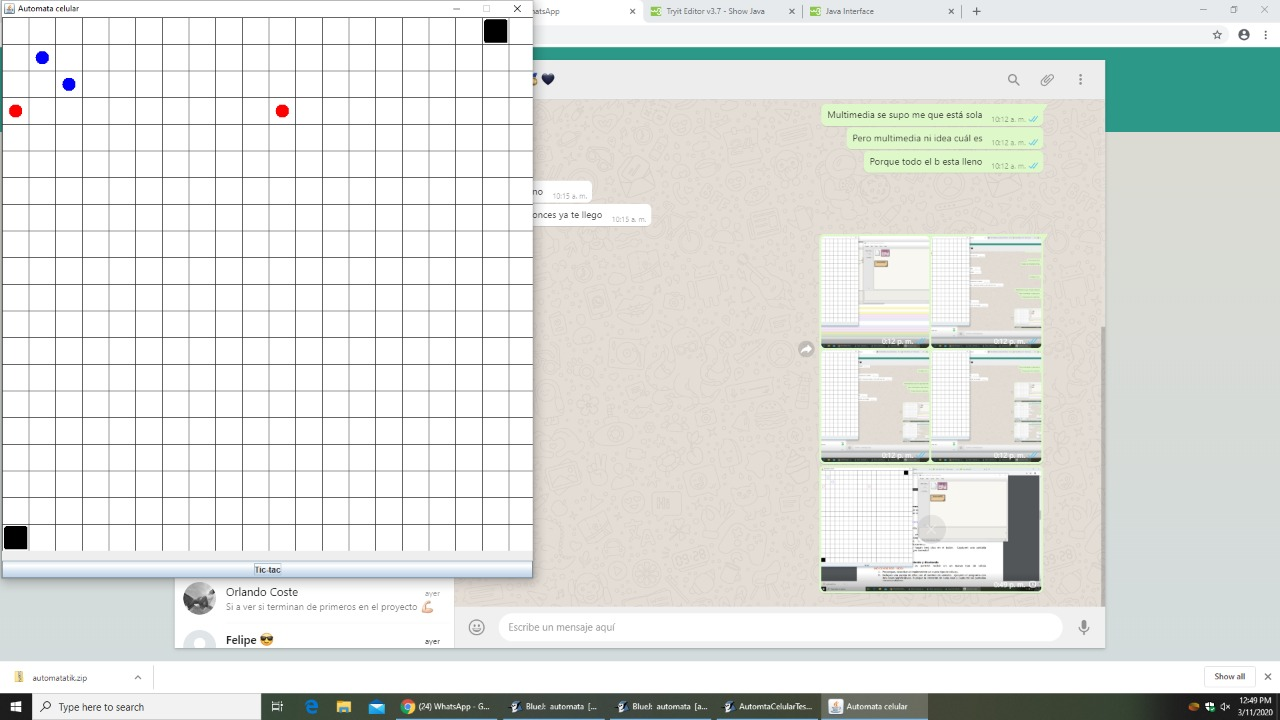


Tercer clic,

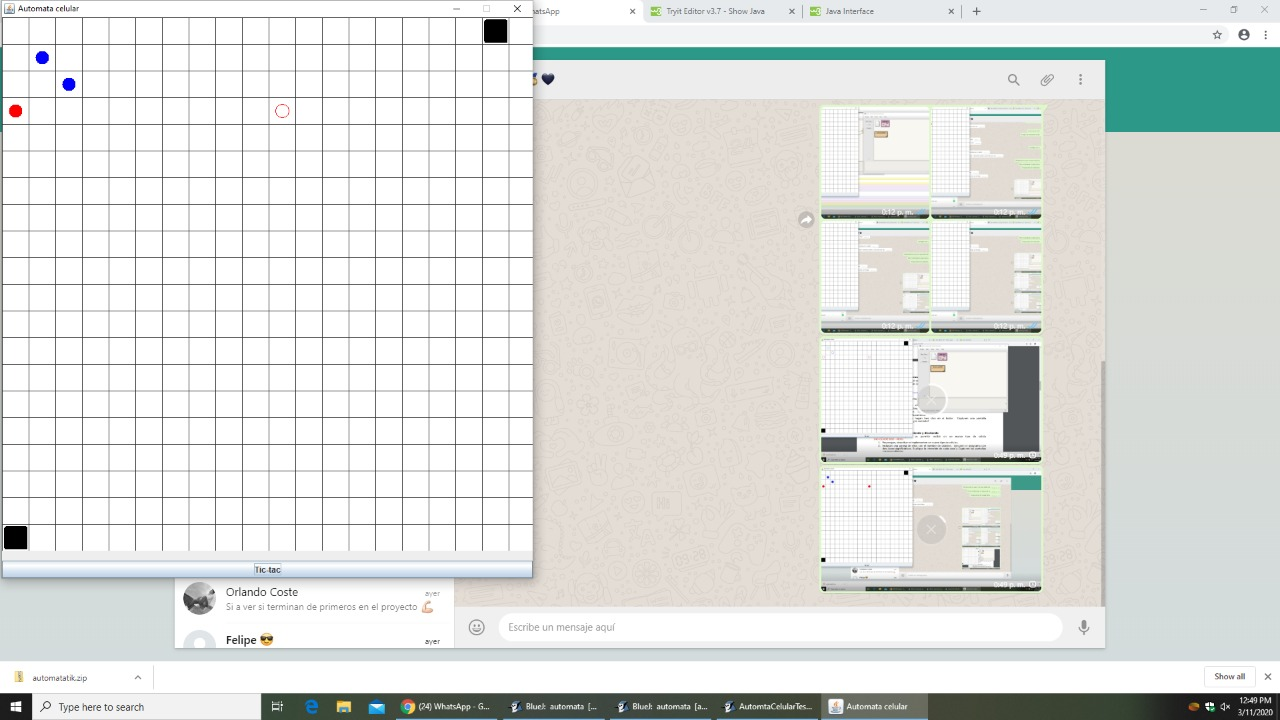


**CICLO 3. ADICIONANDO UNA BARRERA**

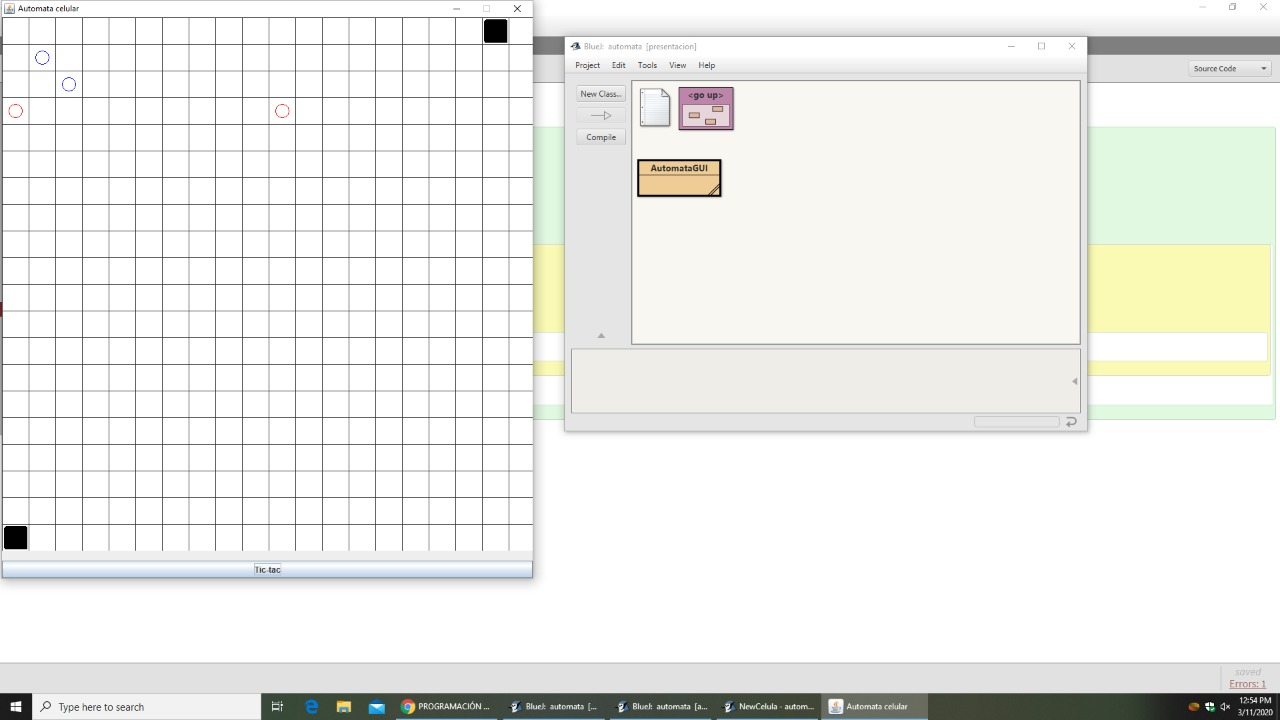
1. Para adicionar la clase barrera y se pueda utilizar dentro del autómata celular, la clase hace una implementación de la interfaz de elemento.
2. No hubo cambios.
3. No sucedería nada ya que las barreras son elementos muertos.
4. BlueJ
5. Creando,



Primer clic,



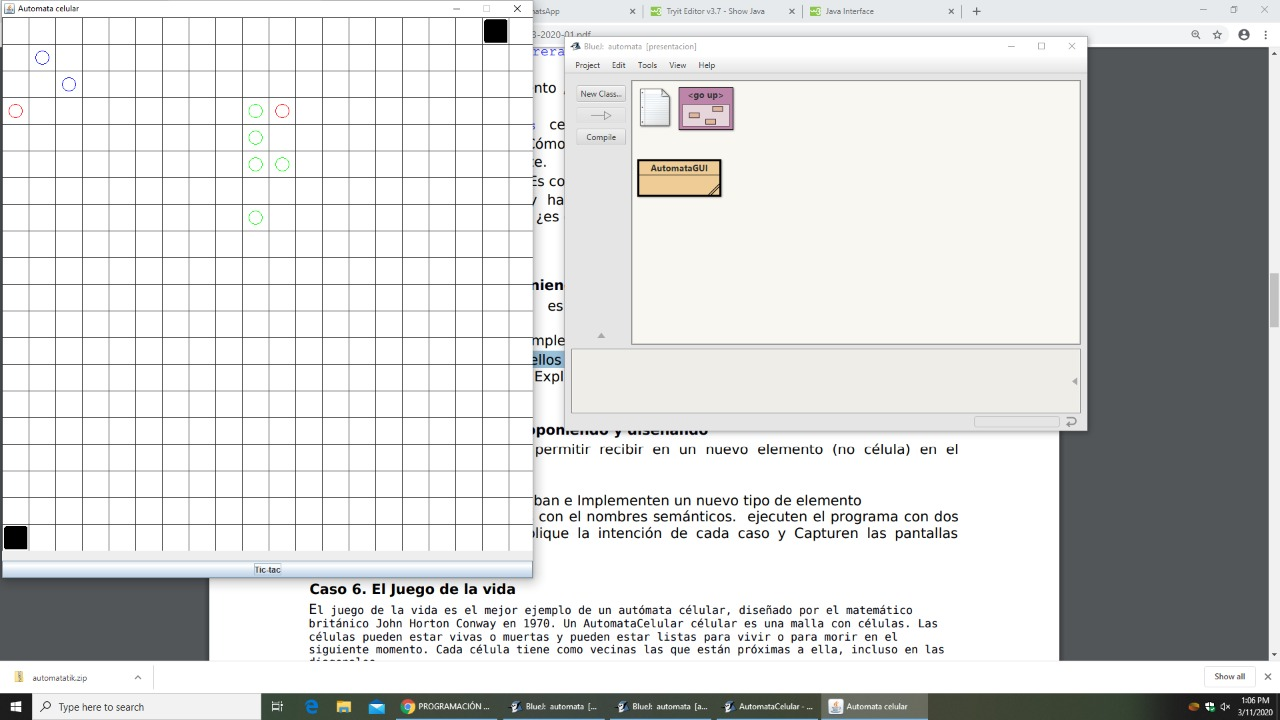
Segundo clic,



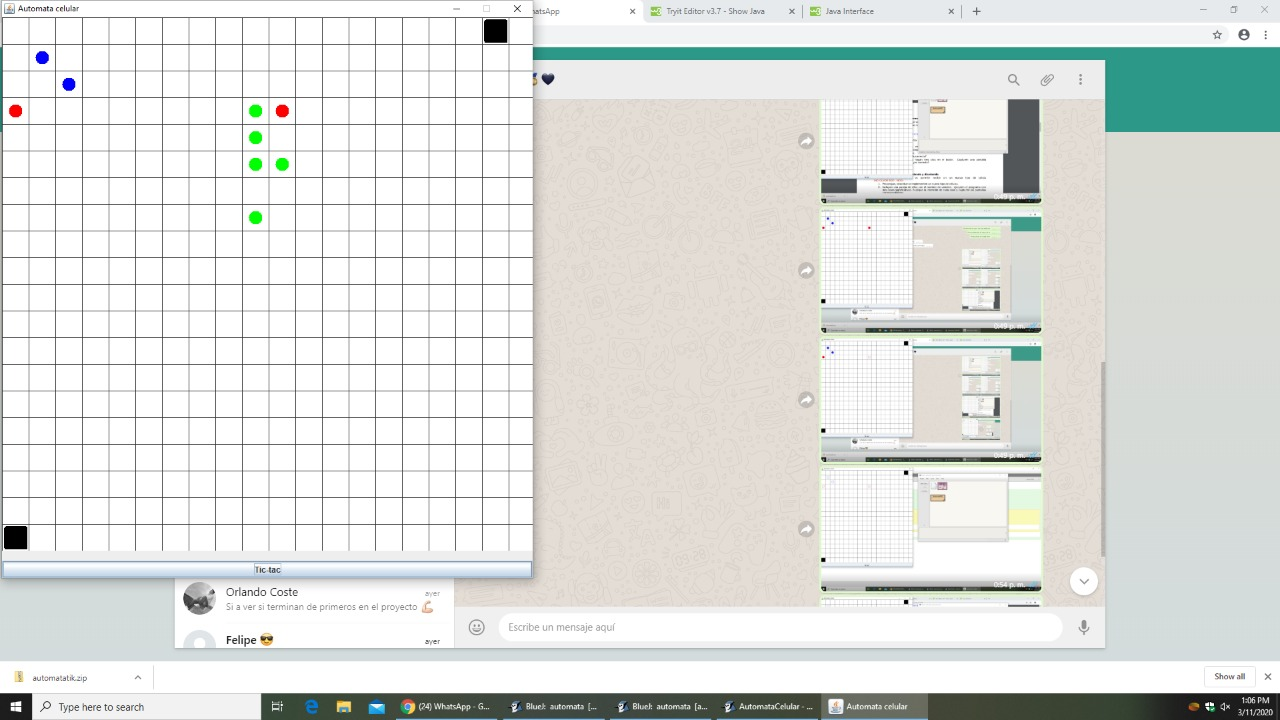
Tercer clic,

**CICLO 4. NUEVA CELULA: PROPONIENDO Y DISEÑANDO**

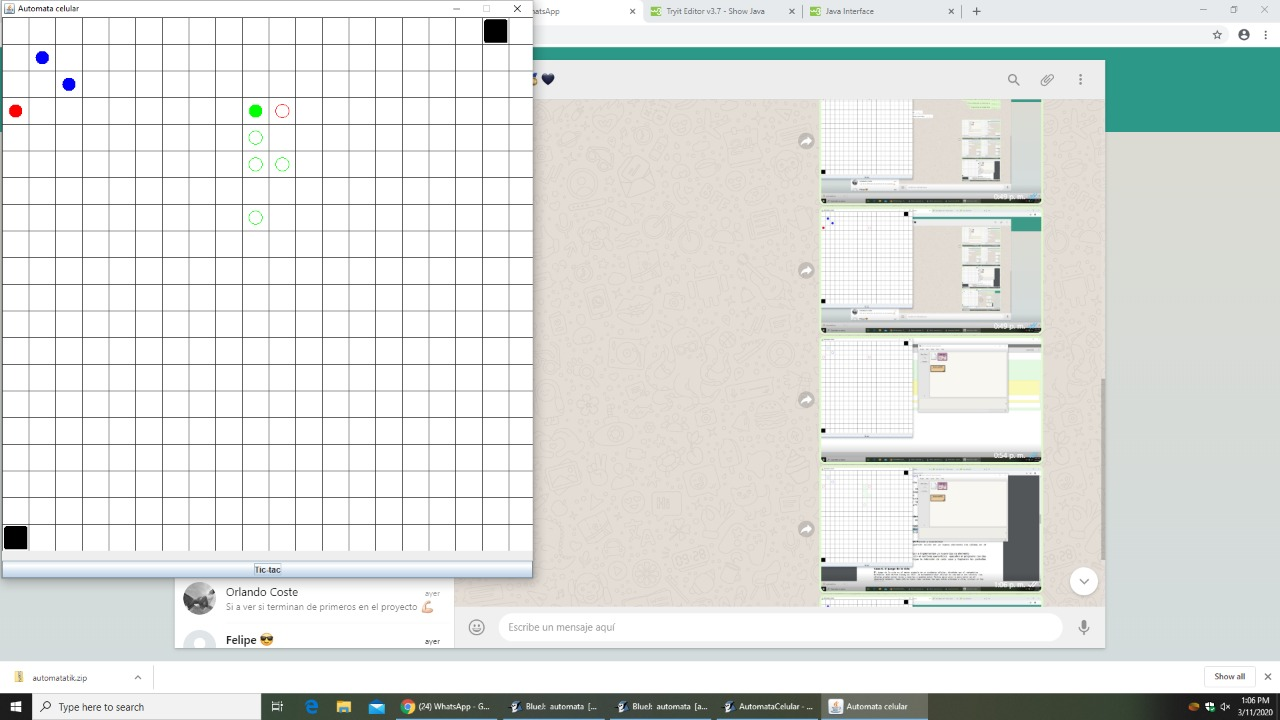
1. Nuestro nuevo tipo de célula será de color verde, la función que realizará será que muera si tiene vecinos en las filas mayores a ella.



1. Creando,



Primer clic,



Segundo clic,