**PROGRAMACIÓN ORIENTADA A OBJETOS**

**Introducción. Clases y objetos.**

**2020-1**

**Laboratorio 1/6**

**Integrantes:**

* Juan Andrés Pico Machuca
* Ana Gabriela Silva Briceño

**SHAPES**

**Conociendo el proyecto shapes.**

1. **Descarga.**
2. Considerando el **diagrama de clases** de “shapes”, las clases que ofrece son: Canvas, Circle, Rectangle, Triangle. Las últimas tres clases se relacionan con la clase Canvas ya que este será el lienzo donde se podrán graficar.
3. Como se mencionó anteriormente las clases que se encuentran en el paquete “Shapes” son Canvas, Rectangle, Circle, Triangle.

Los atributos que tiene la clase Circle son:

* Diametro
* Posición en x
* Posición en y
* Color
* Visible o invisible

Los métodos que tiene la clase Circle son:

* makeVisible
* makeInvisible
* moveRight
* moveHorizontal
* moveLeft
* moveUp
* moveVertical
* moveDown
* slowMoveHorizontal
* slowMoveVertical
* changeSize
* changeColor
* draw
* erase

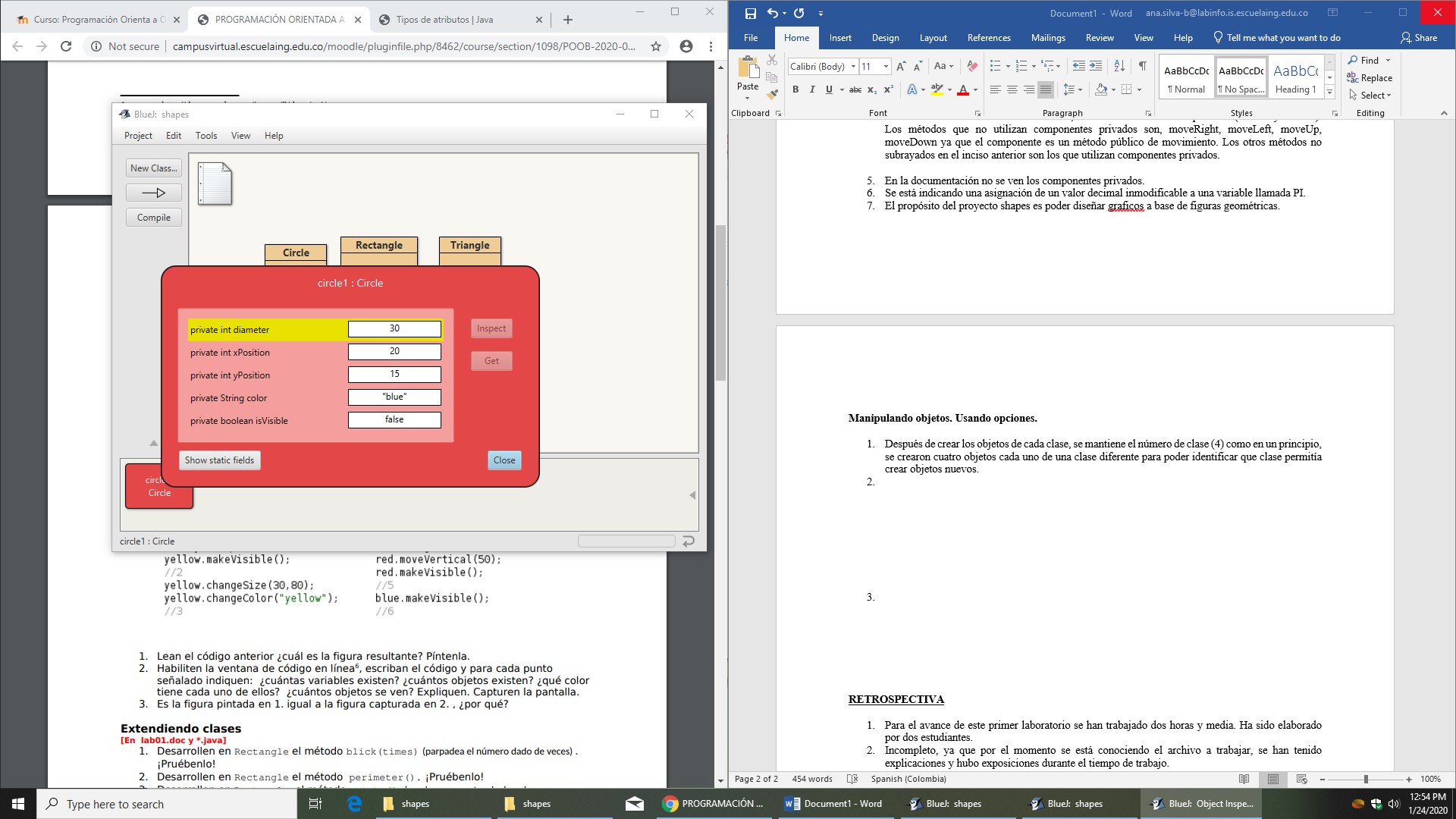
1. Los atributos importantes de la clase Circle, son los que modifican su forma y posición. (Diametro, Posición en x, Posición en y) De igual forma se tiene que ver la figura para poder conocerla gráfica y visualmente. (isVisible). Todos sus atributos son privados.

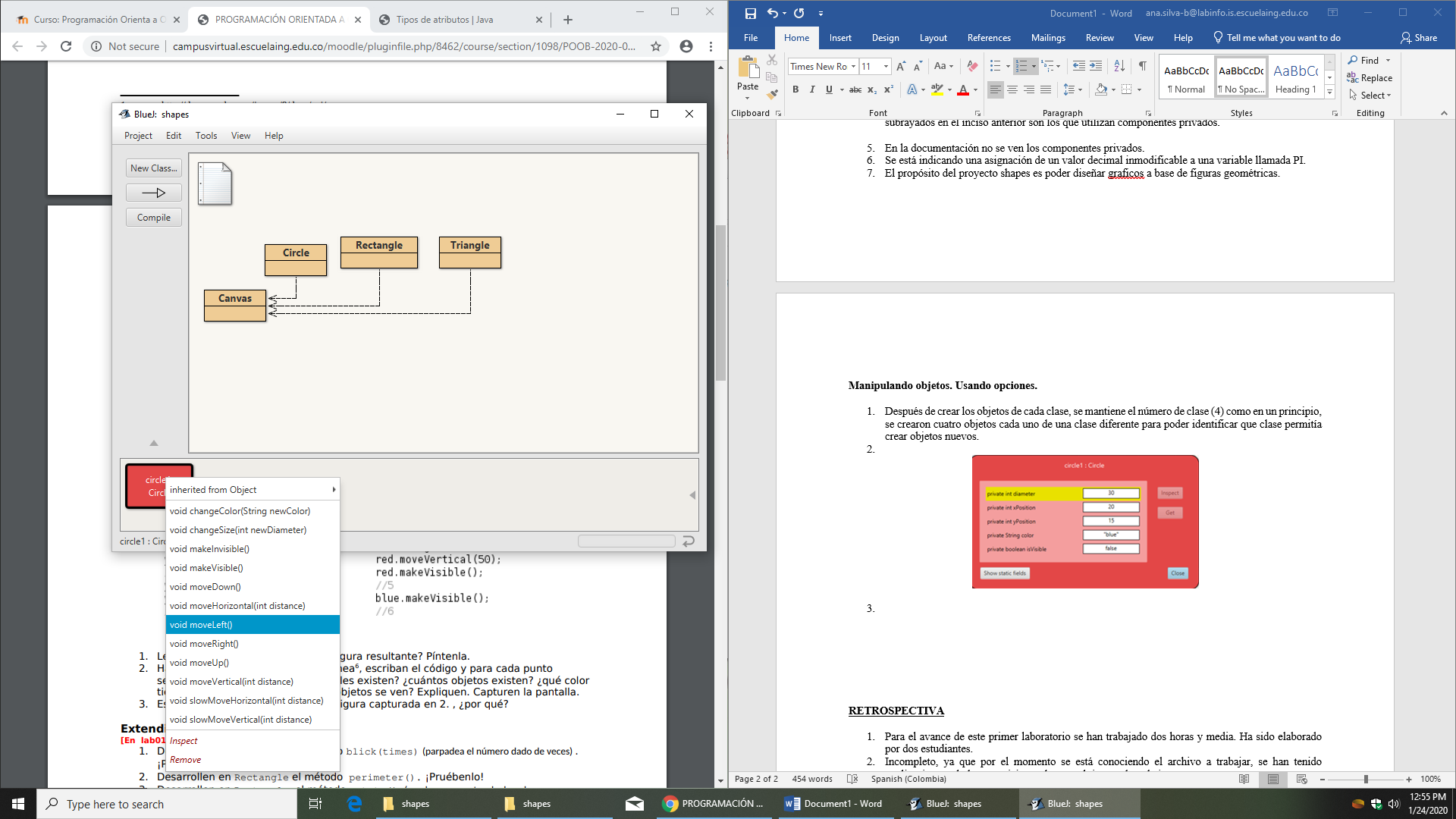
La clase Circle tiene en total 14 métodos, de estos dos de ellos son privados (“draw” y “erase”). Los métodos que no utilizan componentes privados son, moveRight, moveLeft, moveUp, moveDown ya que el componente es un método público de movimiento. Los otros métodos no subrayados en el inciso anterior son los que utilizan componentes privados.

1. En la documentación no se ven los componentes privados.
2. Se está indicando una asignación de un valor decimal inmodificable a una variable llamada PI.
3. El propósito del proyecto shapes es poder diseñar graficos a base de figuras geométricas.

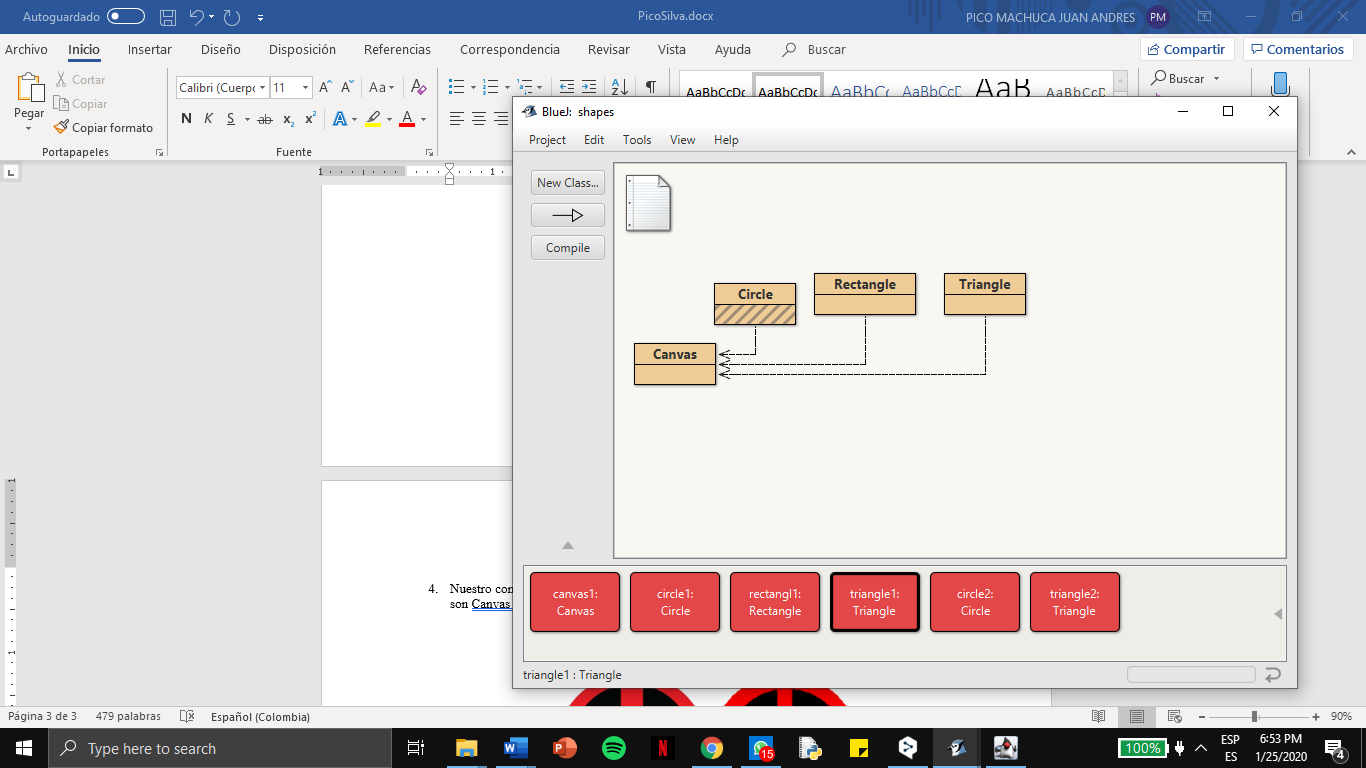
**Manipulando objetos. Usando opciones.**

1. Después de crear los objetos de cada clase, se mantiene el número de clase (4) como en un principio, se crearon cuatro objetos cada uno de una clase diferente para poder identificar que clase permitía crear objetos nuevos.





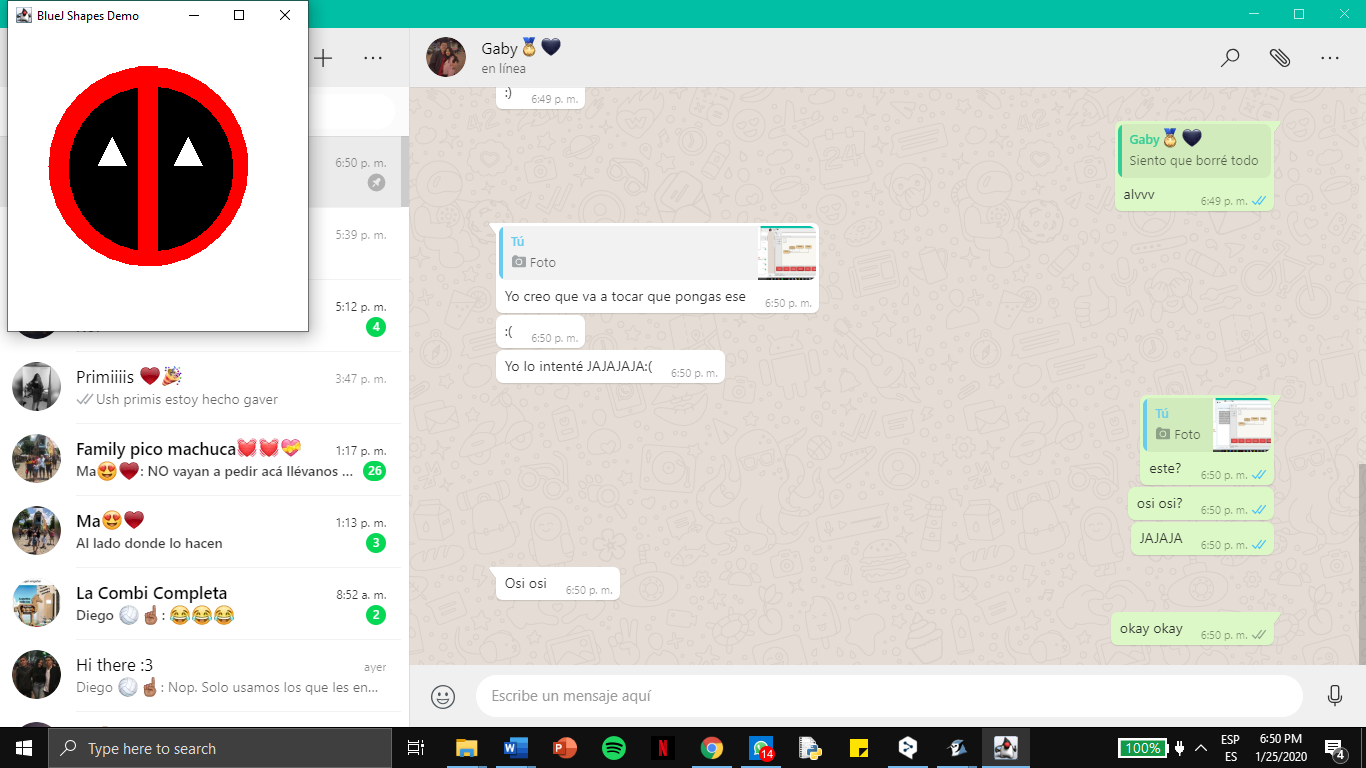
1. Nuestro comic favorito es el de Deadpool donde para crear su logo se necesitan cuatro clases que son Canvas, Circle ,Rectangle y Triangle.



Logo real:

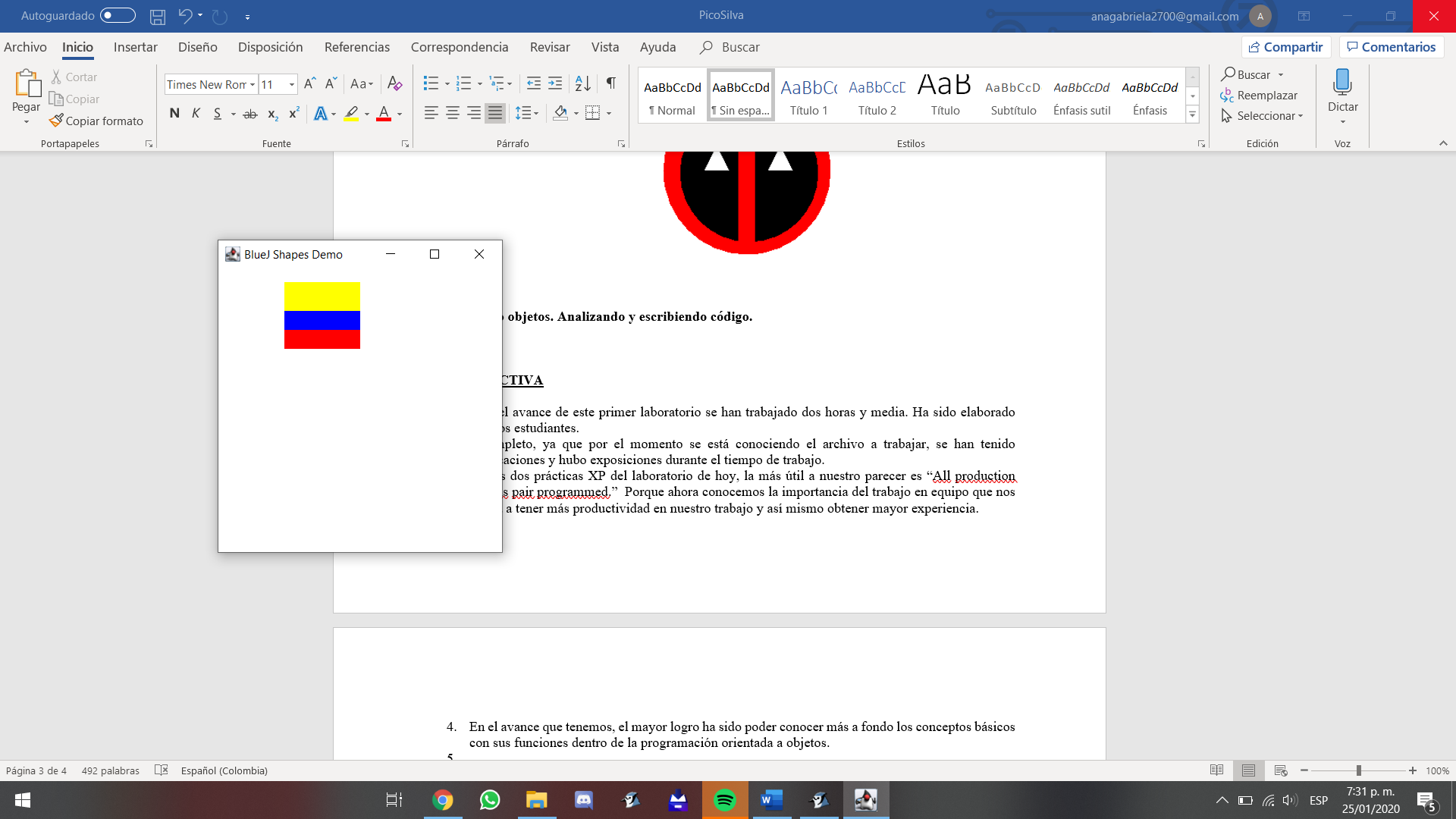


Logo realizado con los objetos dados:



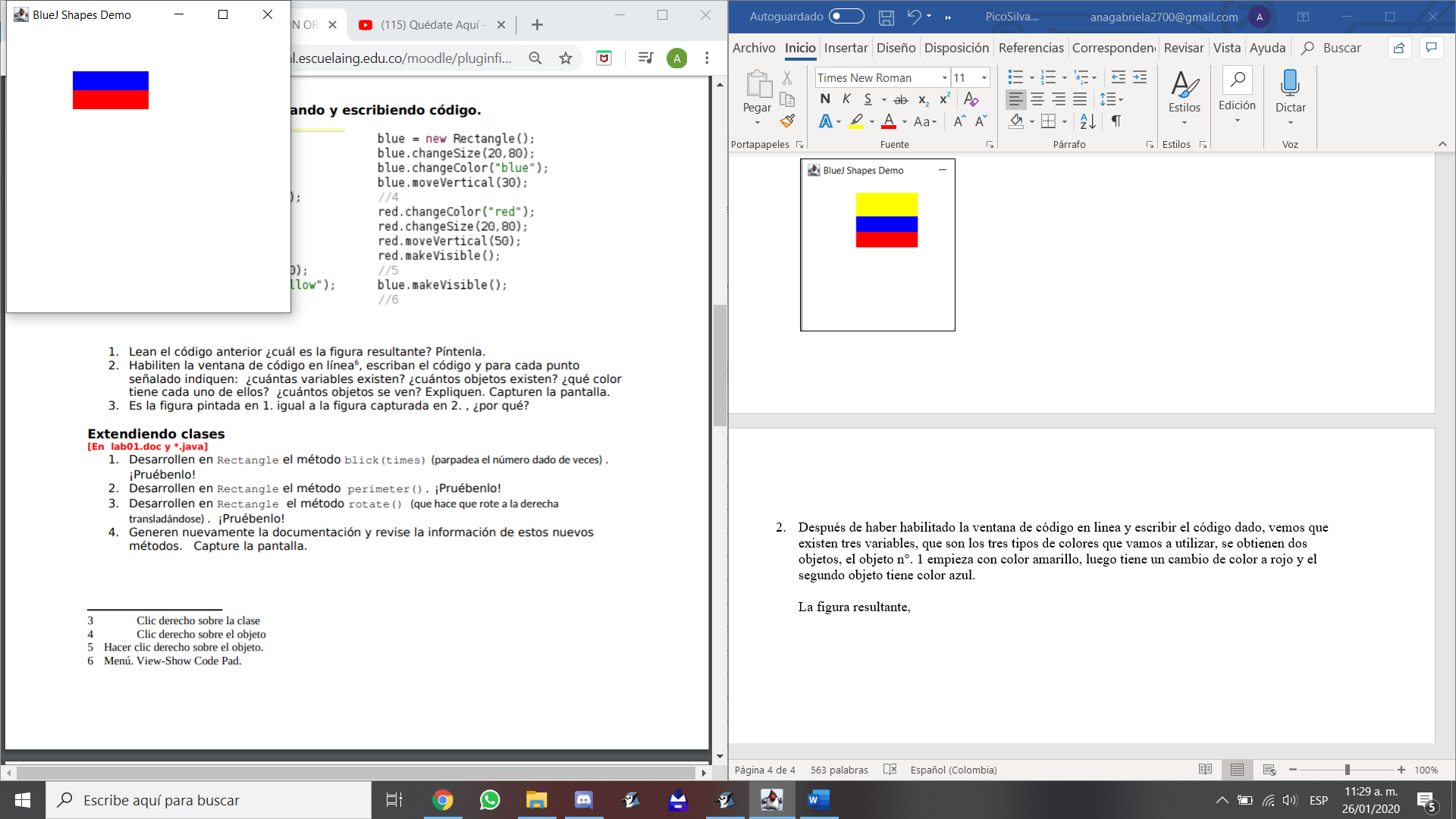
**Manipulando objetos. Analizando y escribiendo código.**

1. La figura resultante, modificando simplemente los objetos.



1. Después de haber habilitado la ventana de código en linea y escribir el código dado, vemos que existen tres variables, que son los tres tipos de colores que vamos a utilizar, se obtienen dos objetos, el objeto n°. 1 empieza con color amarillo, luego tiene un cambio de color a rojo y el segundo objeto tiene color azul.

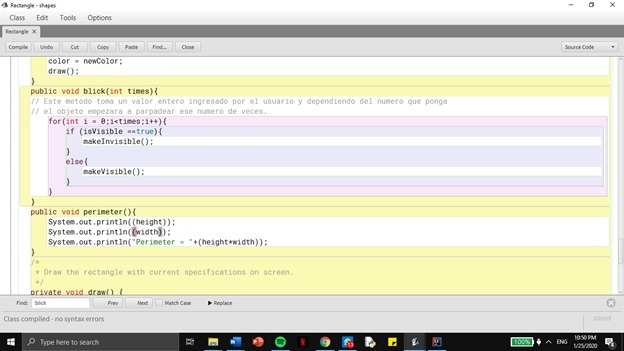
La figura resultante,



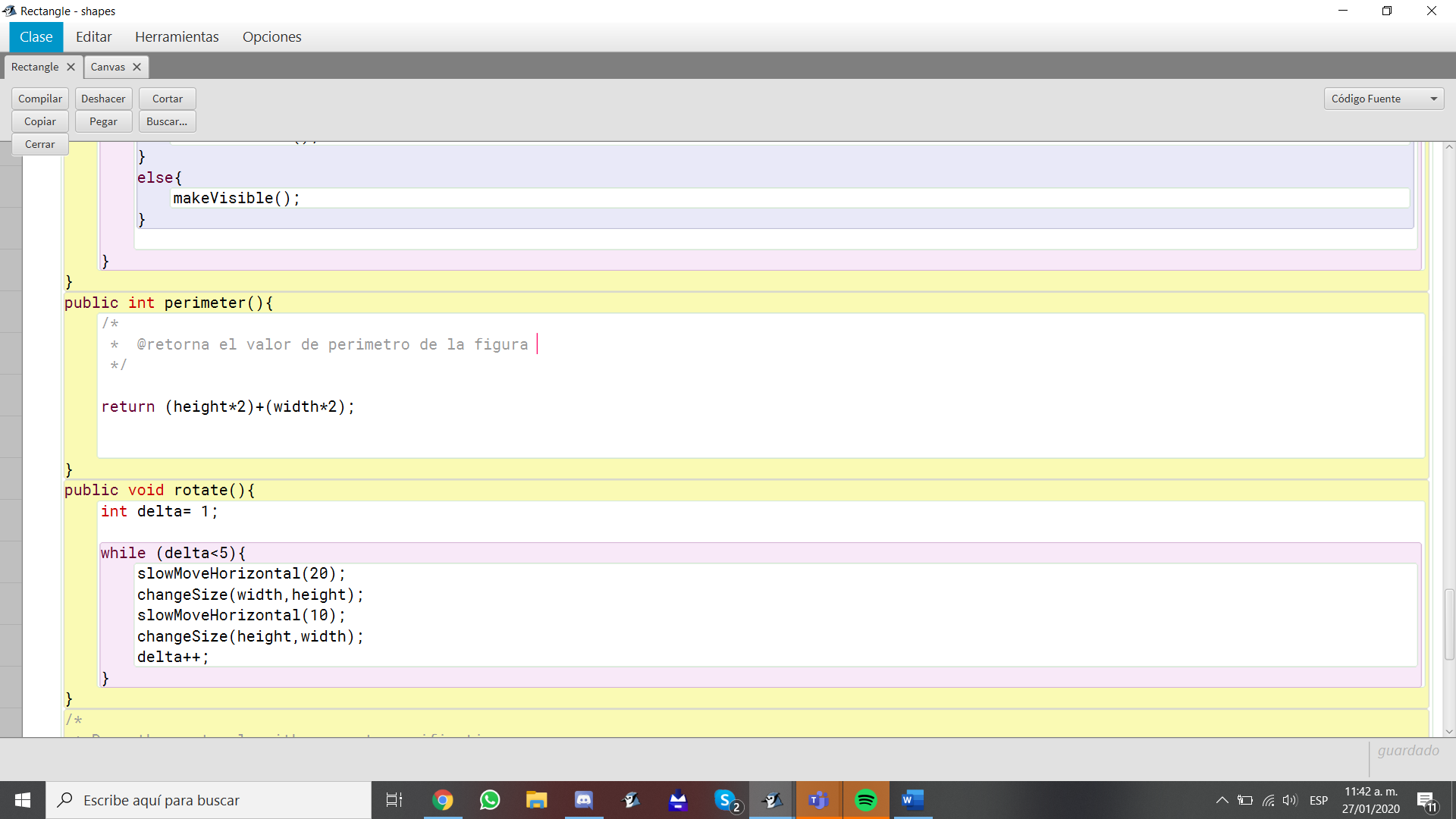
1. No obtenemos las mismas figuras ya que en el primer inciso lo único que se realizo fue crear tres objetos diferentes y cada uno modificarlo según el codigo. En cambio, en el segundo inciso se ejecuta el código paso a paso y allí se ve que una variable se le asigna a otra por lo tanto se obtendrán solo dos objetos que son los que se ven en el inciso 2.

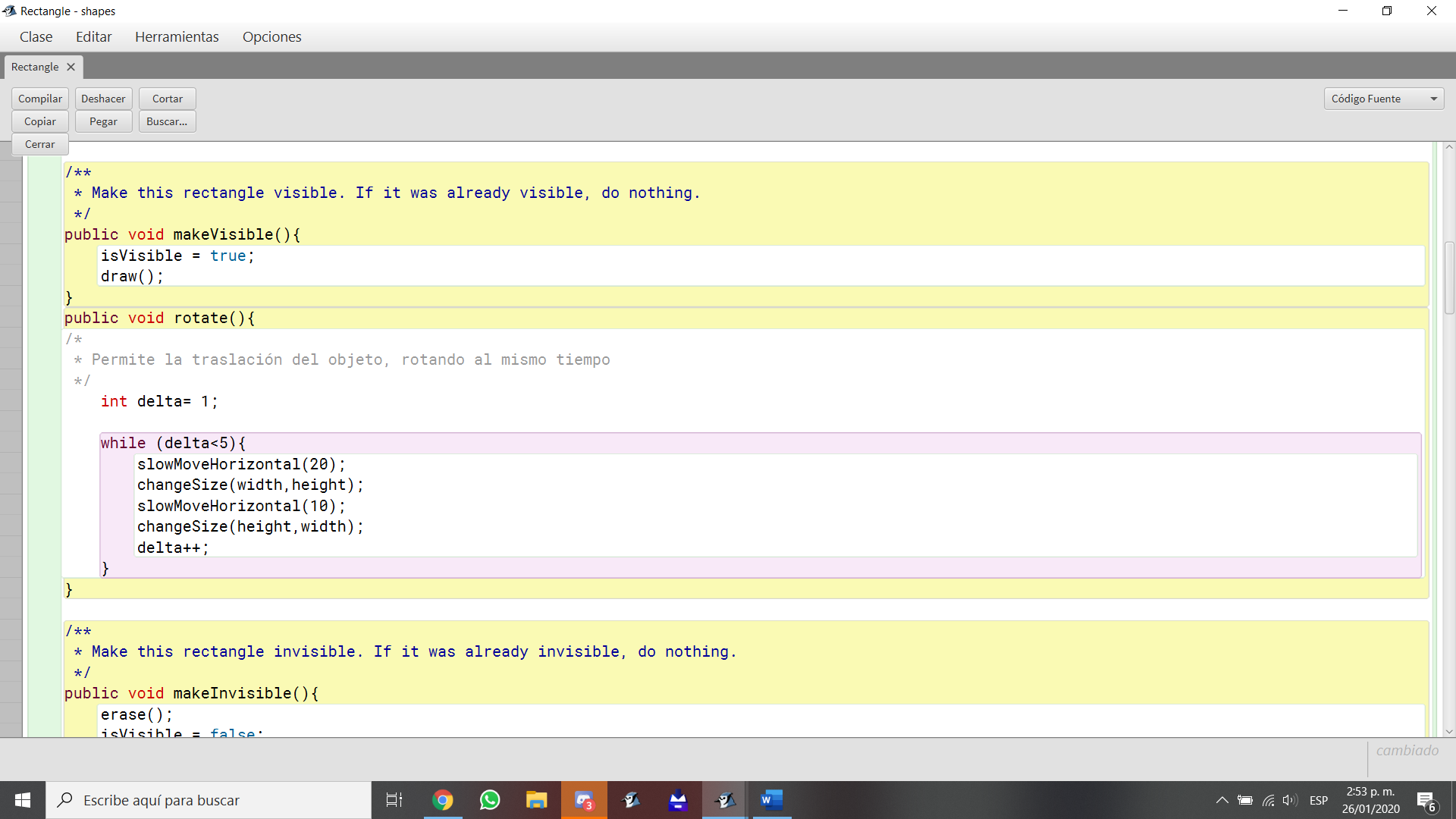
**Extendiendo clases**

1. Método “blick”



1. Método “perimeter”



1. Método “rotate”
2. La documentación está denotada con los símbolos \*/, \* ,\*/ en cada captura de pantalla de los métodos extendidos en el código.

**NÚMEROS MAYAS**

**Implementando una nueva clase. Xook.**

1. El número de objetos que conforman un “Xook” son dos, un círculo y un rectángulo.

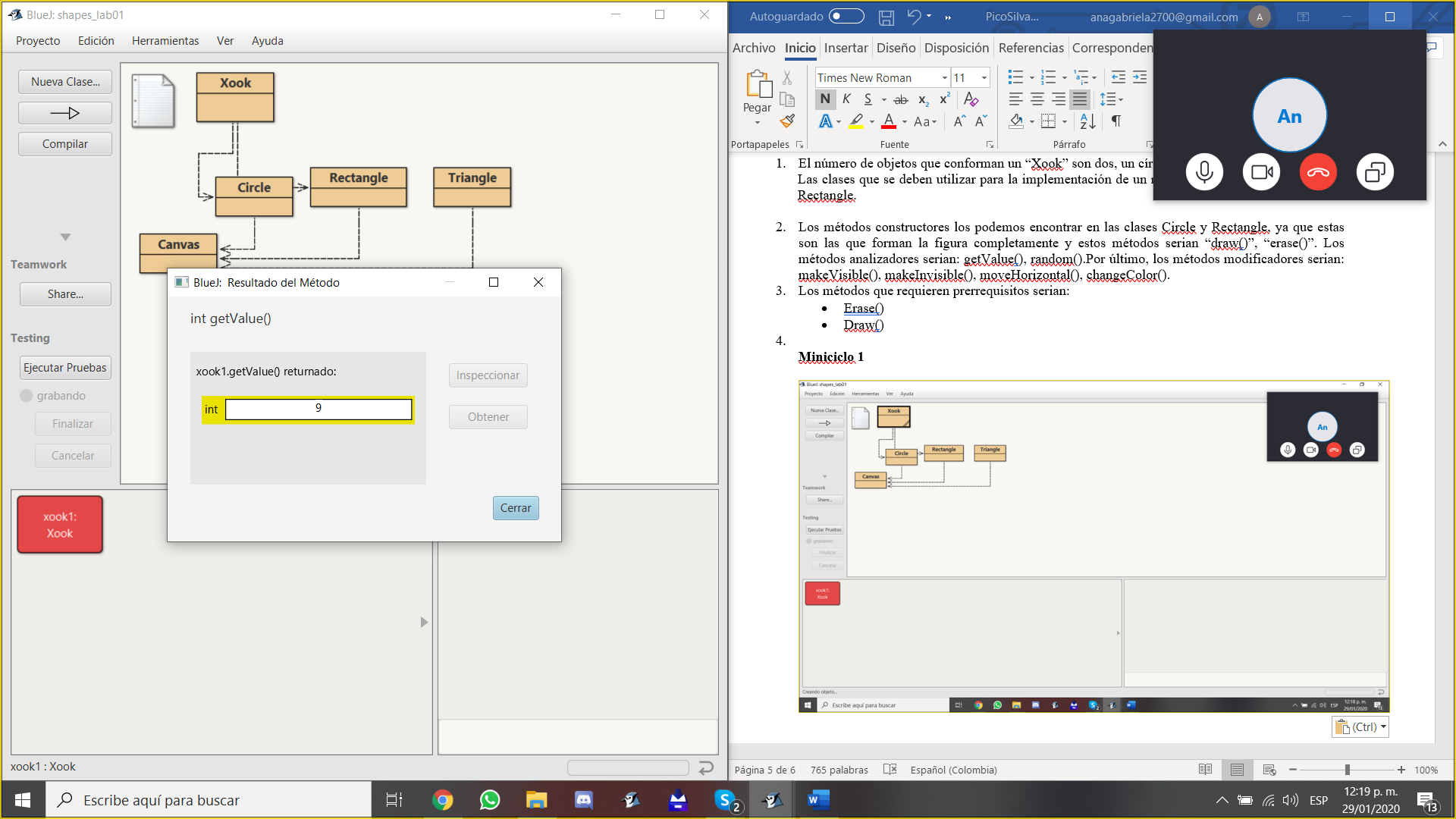
Las clases que se deben utilizar para la implementación de un nuevo Xook son, Canvas, Circle y Rectangle.

1. Los métodos constructores los podemos encontrar en las clases Circle y Rectangle, ya que estas son las que forman la figura completamente y estos métodos serian “draw()”, “erase()”. Los métodos analizadores serian: getValue(), random().Por último, los métodos modificadores serian: makeVisible(), makeInvisible(), moveHorizontal(), changeColor().
2. Los métodos que requieren prerrequisitos serian:

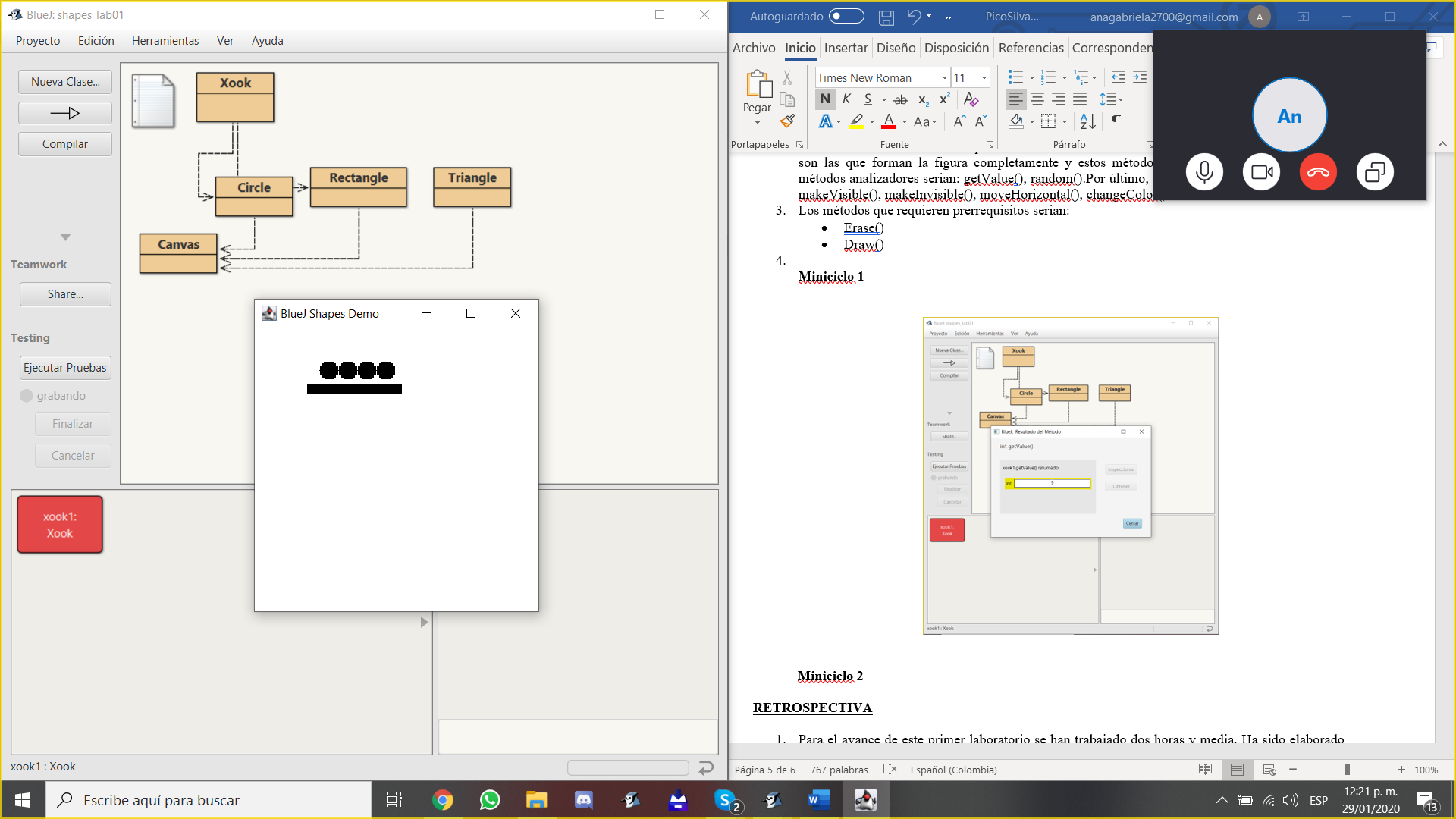
* Erase()
* Draw()



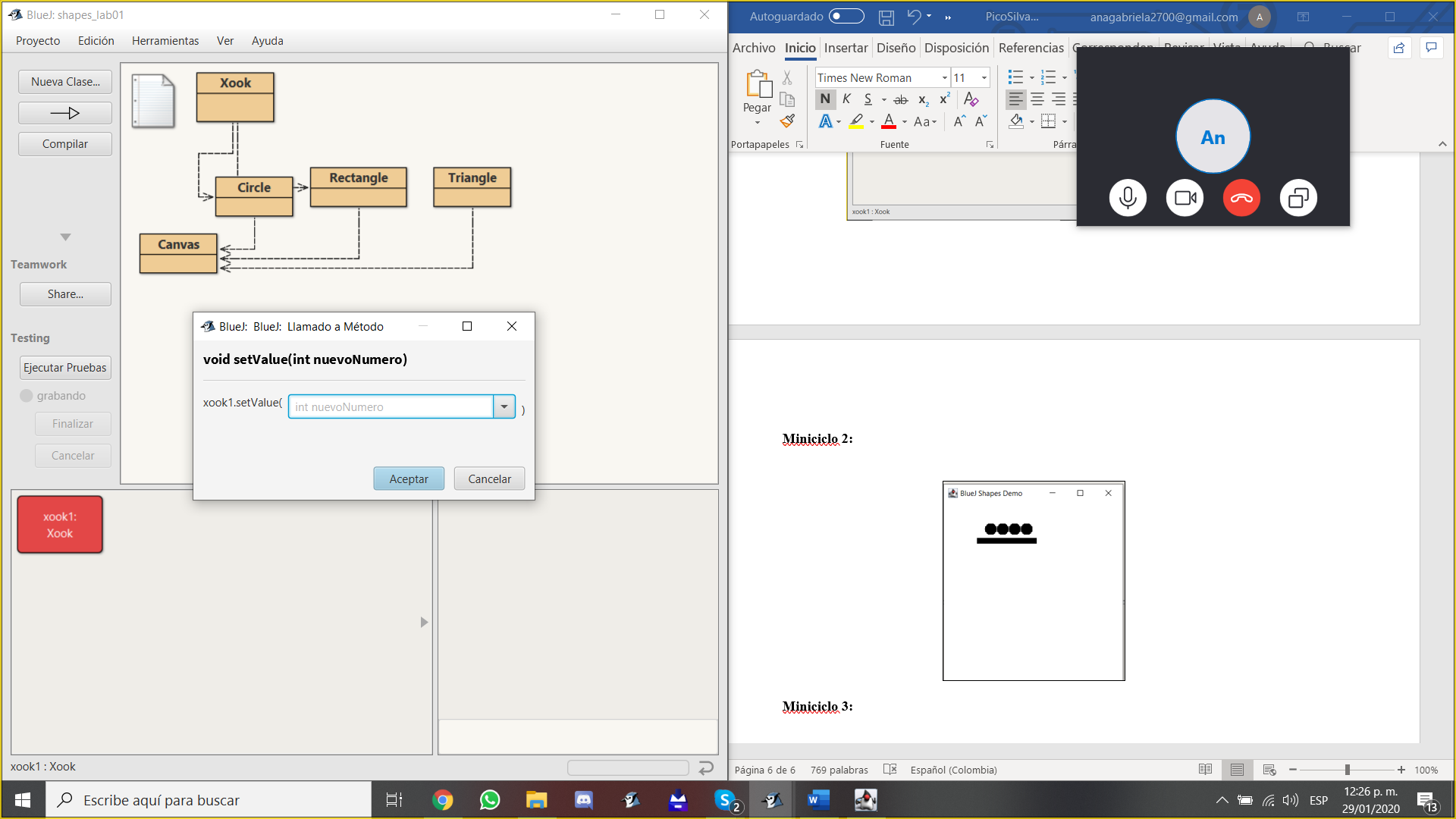
**Miniciclo 1:**



**Miniciclo 2:**

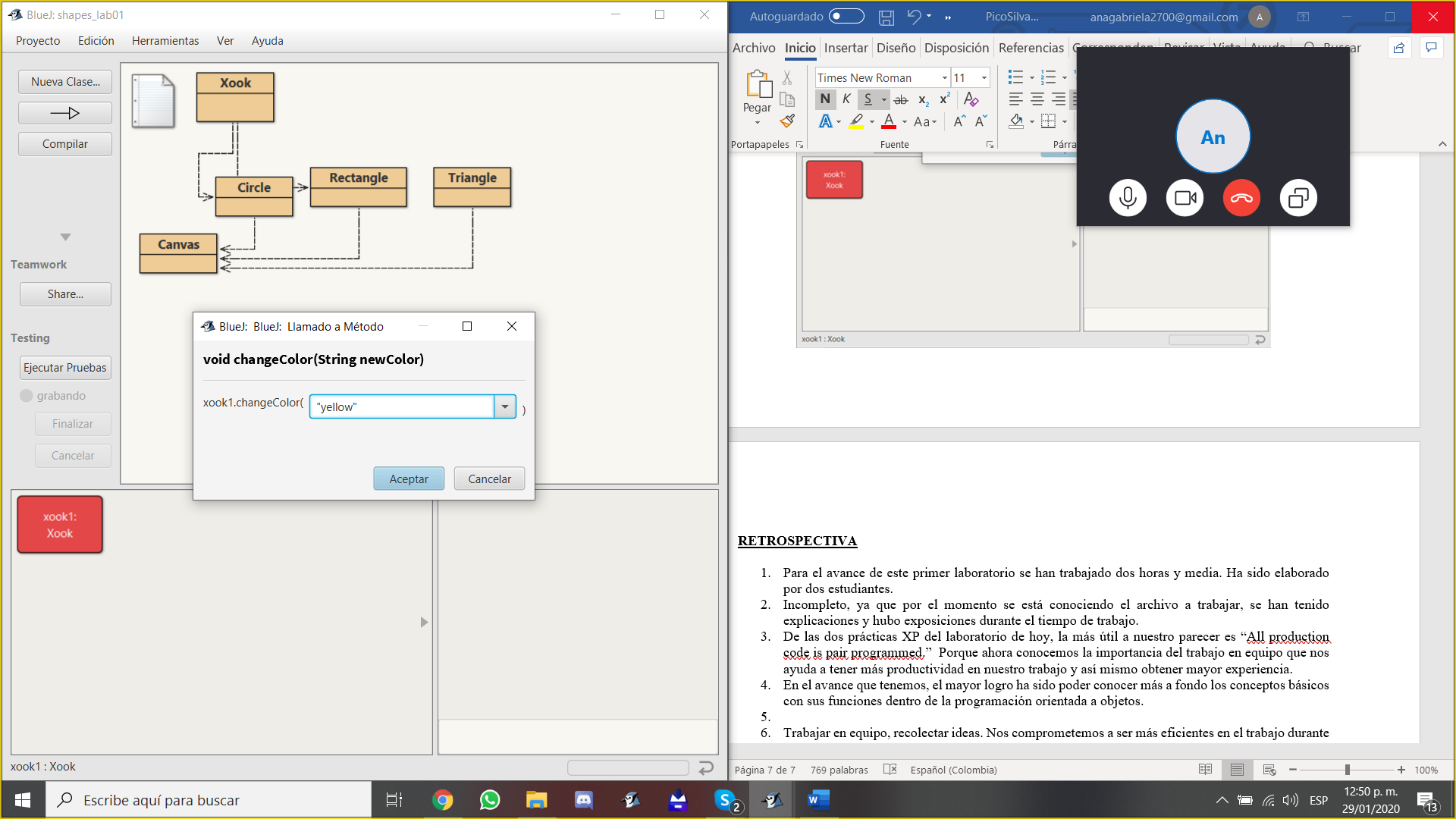
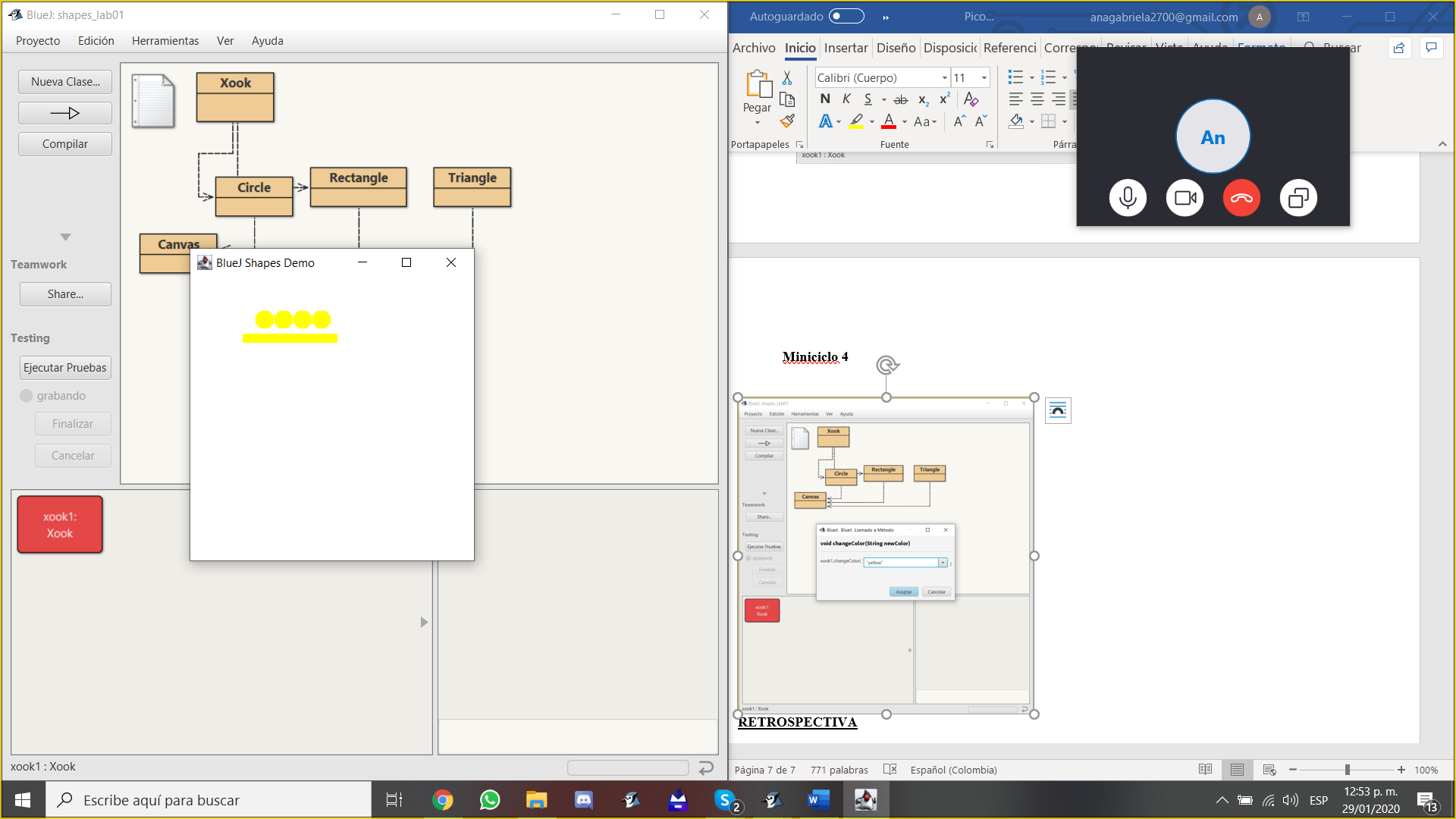


**Miniciclo 3:**

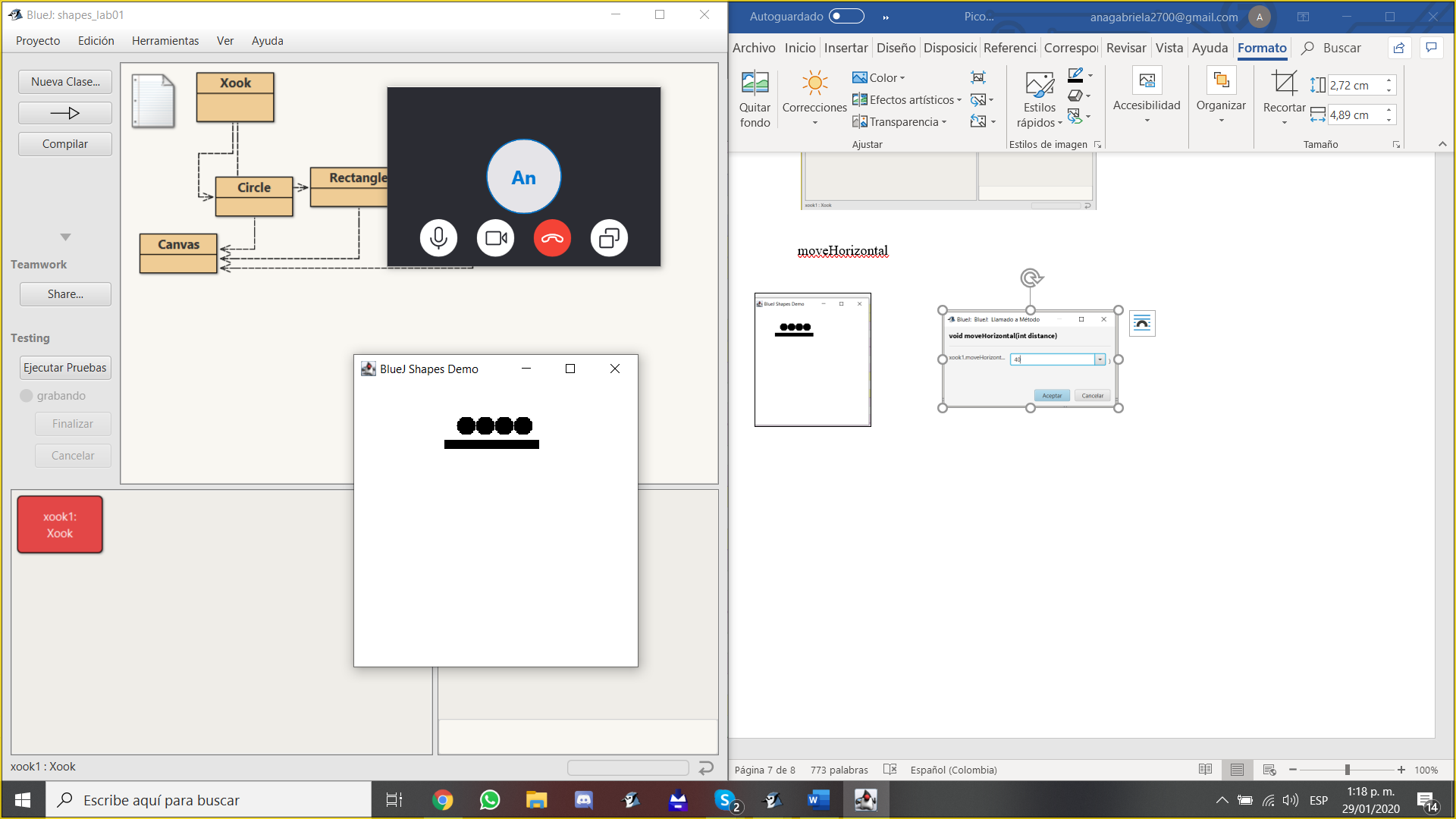
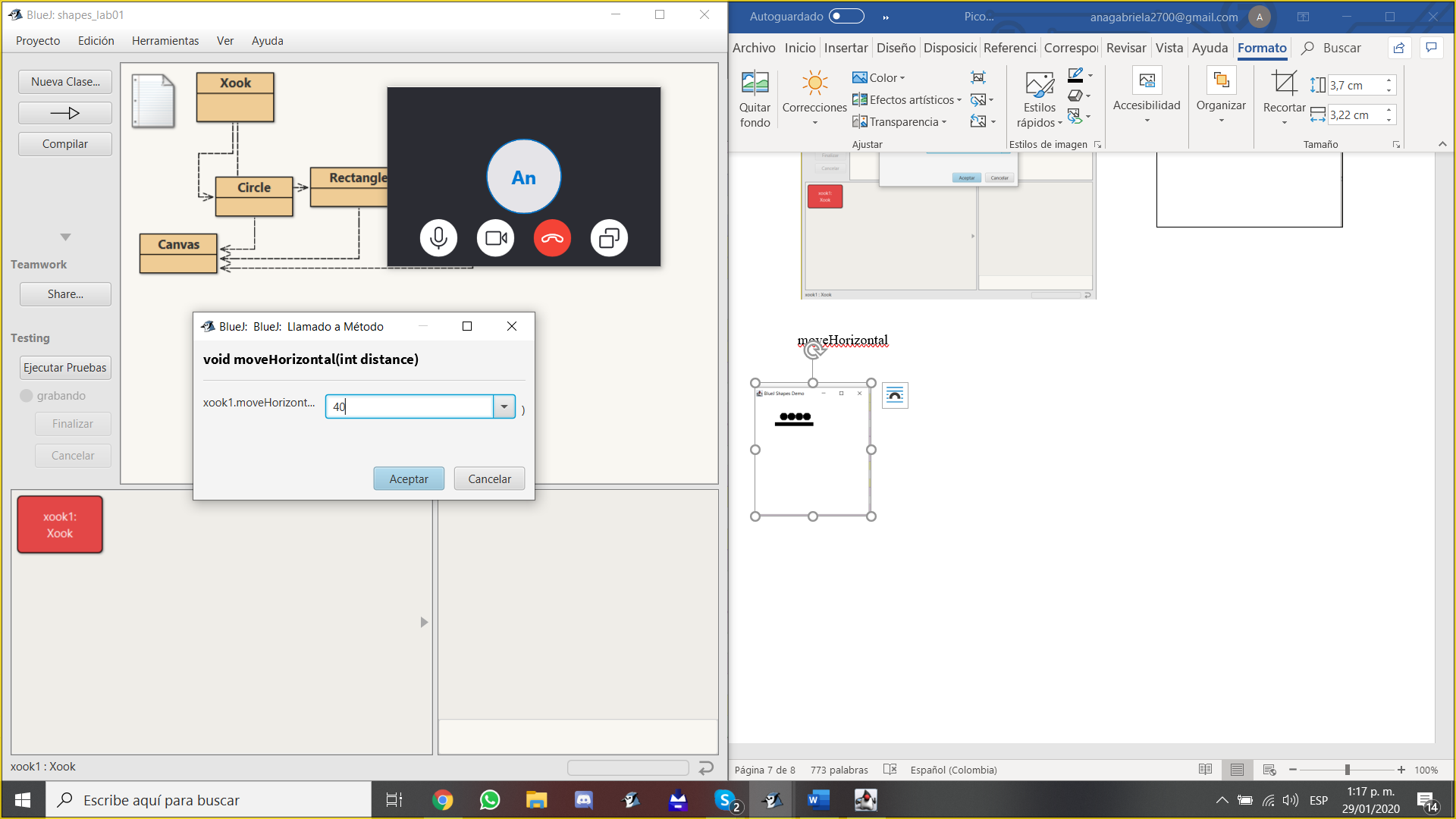
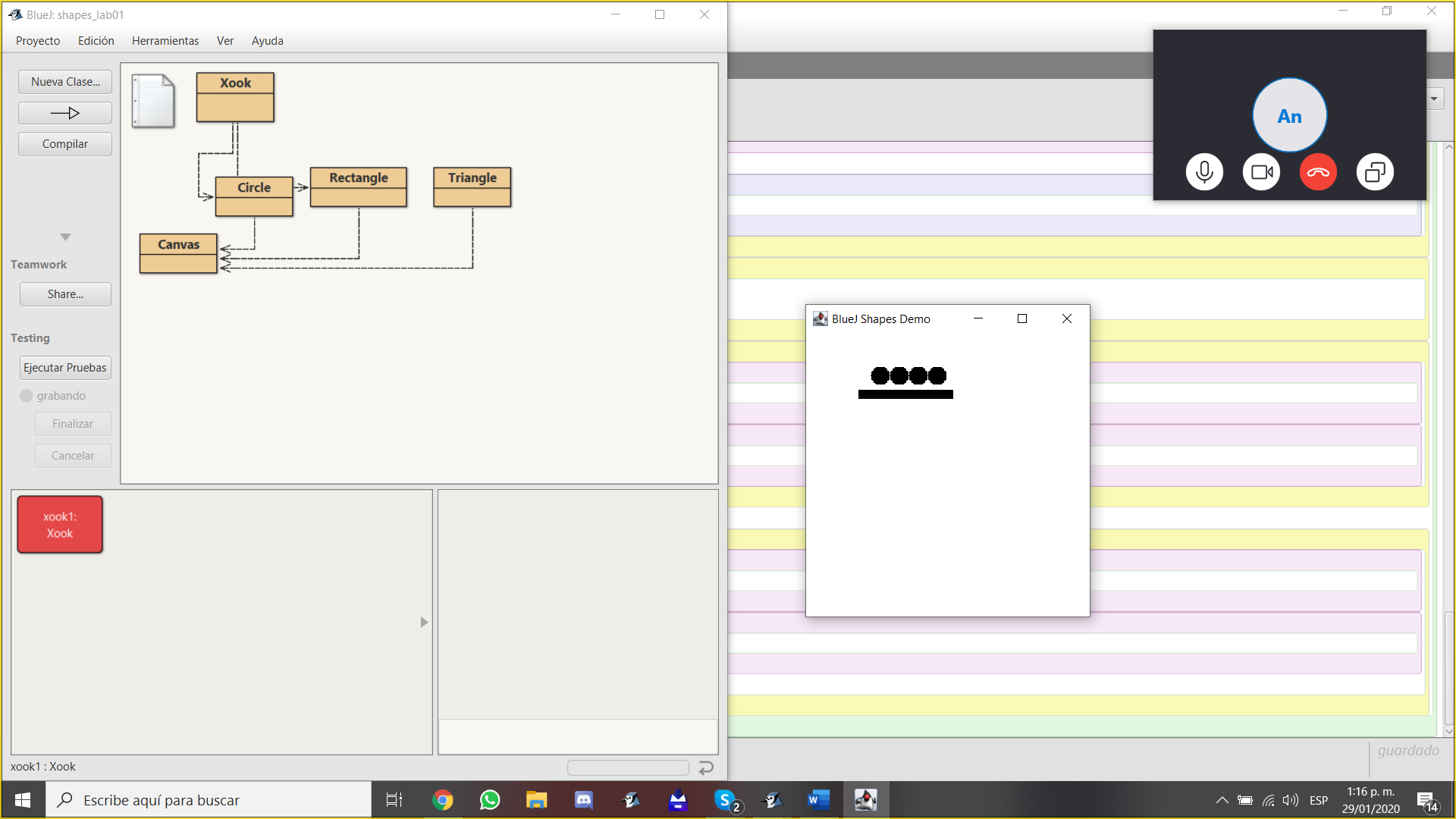


**Miniciclo 4**

changeColor



moveHorizontal



**Definiendo y creando una nueva clase. / Extendiendo una clase. – CalcXook**

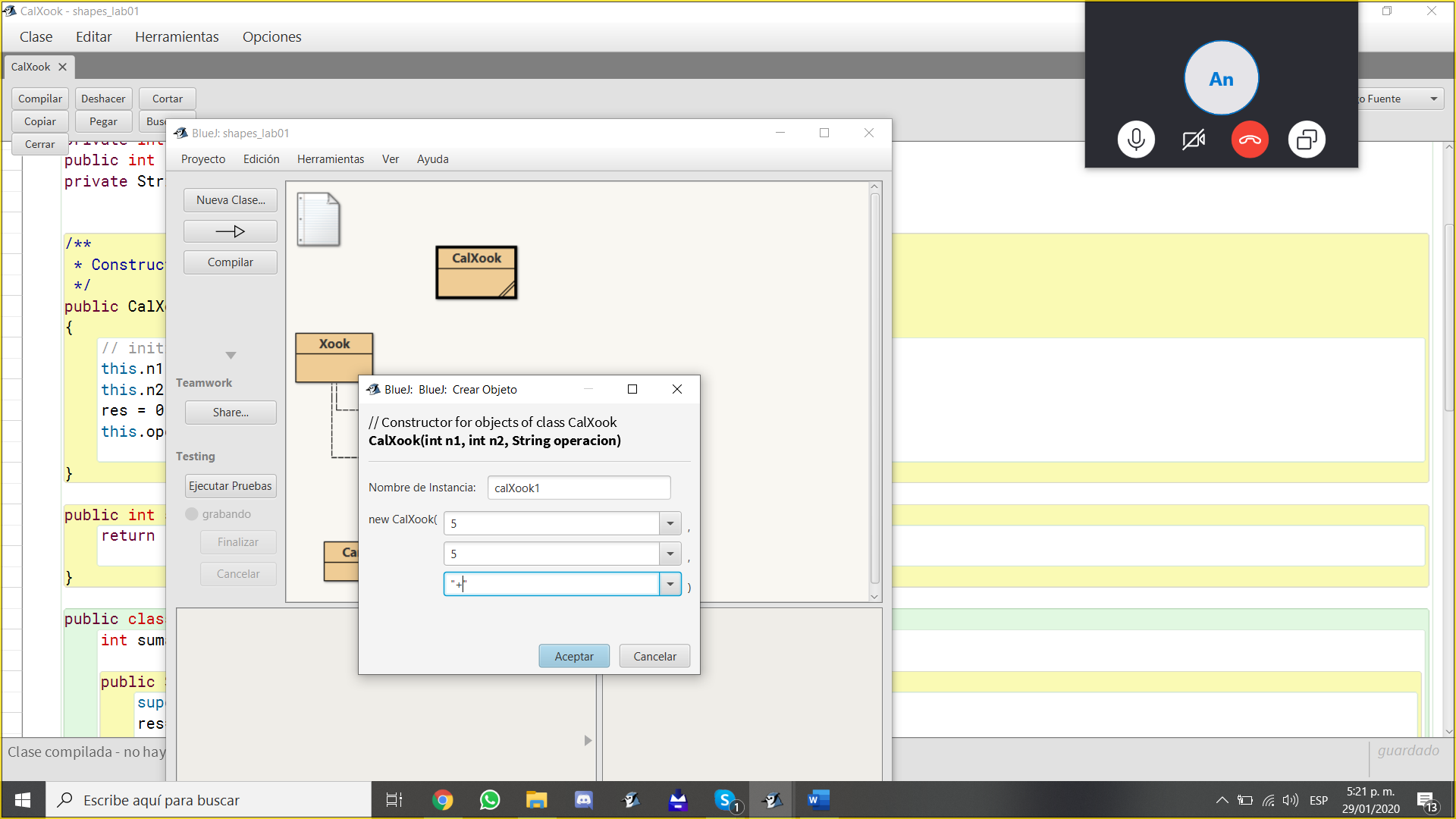
1. Para crear la clase CalcXook se necesitan los atributos privados enteros de los numero a operar y la respuesta, de igual forma se necesitará un string que indica la operación que se va a realizar con los números dados.

El método que va a obtener será el de mostrar el resultado ya que las operaciones se harán en clases extendidas a la principal.

1. La construcción se planificó en:

* Tener los números a operar
* Mostrar resultado en número maya





1. Como lo indicamos anteriormente, se realizó extensión para cada clase que fuera a realizar una operación.

* “public class Suma extends CalcXook”
* “public class Resta extends CalcXook”
* “public class Multiplicación extends CalcXook”
* “public class División extends CalcXook”

**RETROSPECTIVA**

1. Para el este primer laboratorio lo trabajamos 48 horas cada uno. Ha sido elaborado por dos estudiantes.
2. Medianamente terminado, ya que se nos dificultó algunos temas recientemente aprendidos en un nuevo lenguaje. (Herencia)
3. De las dos prácticas XP del laboratorio de hoy, la más útil a nuestro parecer es “All production code is pair programmed.” Porque ahora conocemos la importancia del trabajo en equipo que nos ayuda a tener más productividad en nuestro trabajo y así mismo obtener mayor experiencia.
4. Consideramos que nuestro mayor logro fue haber podido realiza la mayor parte la practica de los números mayas.
5. Consideramos que el mayor problema técnico que tuvimos fue realizar el método draw en una implementación nueva de clases, lo solucionamos buscando información acerca de los atributos que contiene.
6. Trabajar en equipo, recolectar ideas. Nos comprometemos a ser más eficientes en el trabajo durante un tiempo determinado.