

**INSTITUTO POLITÉCNICO NACIONAL**

**ESCUELA SUPERIOR DE CÓMPUTO**

**PROGRAMACIÓN ORIENTADA A OBJETOS**

**2CM3**

**PROFESOR: TECLA PARRA ROBERTO**

**PRÁCTICA 2 MOSAICO DE IMÁGENES**

**VÁZQUEZ MORENO MARCOS OSWALDO 2016601777**



**FECHA DE ENTREGA: 12 DE MARZO DE 2018**

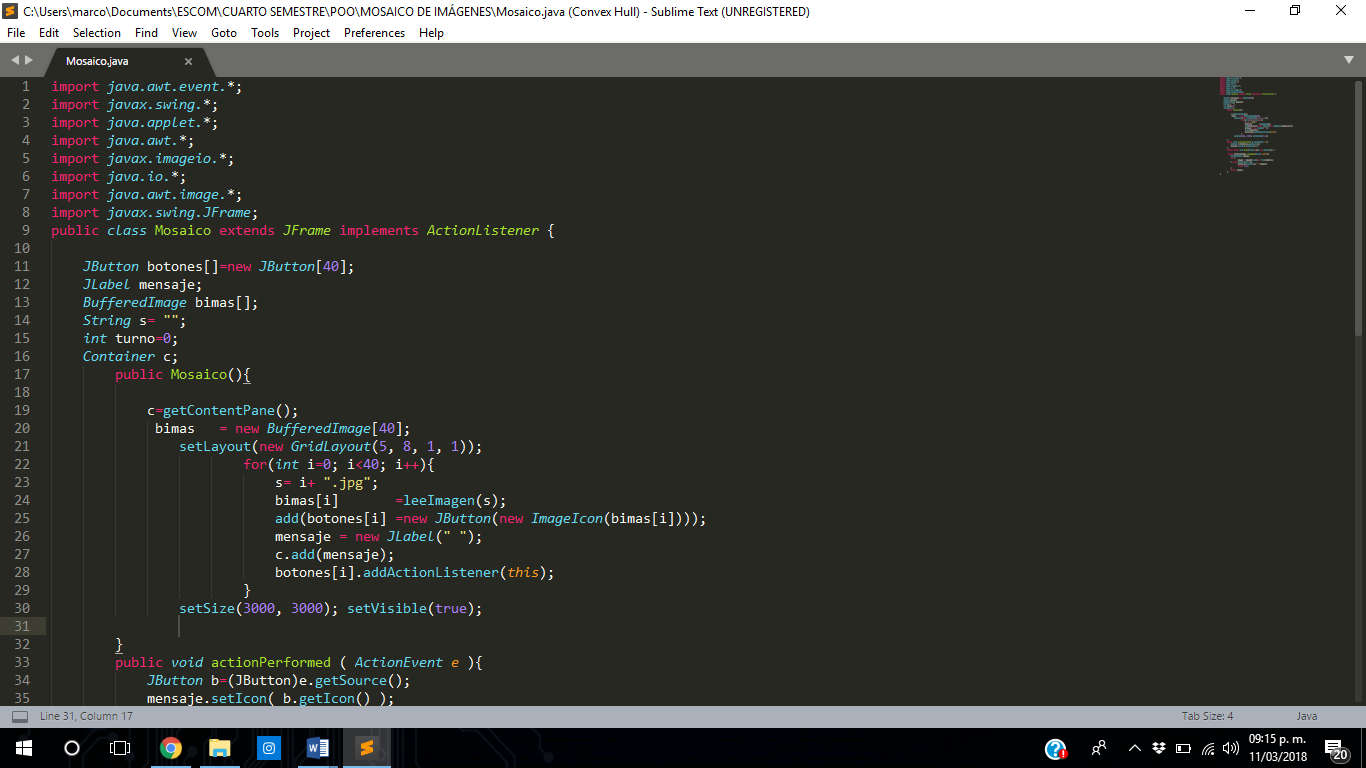
**OBJETIVO**

Realizar un programa en lenguaje de programación Java con arreglo de botones y posicionar dichos botones mediante un GridLayout siendo así un mosaico de 40 imágenes colocadas en botones (JButtons); pero también una etiqueta (JLabel) y que cuando se haga clic en una de las imágenes se muestre la imagen de dicho botón en la etiqueta.

**DESARROLLO**

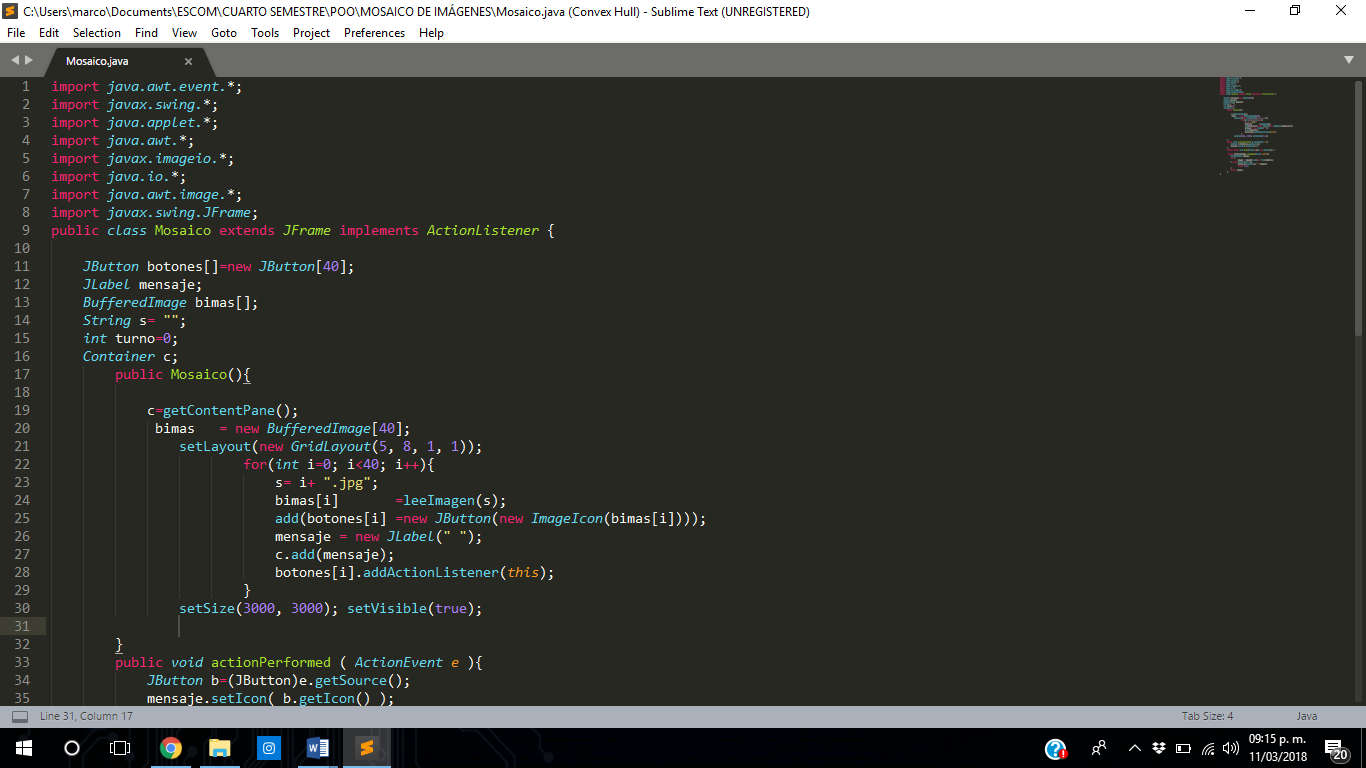
Primeramente, agregamos todas las librerías y extensiones de Java para poder llevar acabo arreglo de imágenes en un entorno gráfico, ya que cada uno de los botones son imágenes y tienen una acción distinta.

En la línea 9 declaramos a nuestra clase de tipo pública de nombre “Mosaico” extendida a un *JFrame*  el cual en un principio era solamente un *Frame* pero al momento de ejecutar el programa, no mostraba nada correctamente, a su vez tiene una acción por cada botón y por eso usaremos el *ActionListener,*  declaramos a las variables de tipo Button*, JLabel, BufferedImage, String, int, Container,* ya que en este programa vamos a utilizar un ciclo for para ir asignando imagen a cada botón del arreglo se creó la variable *int=0.*



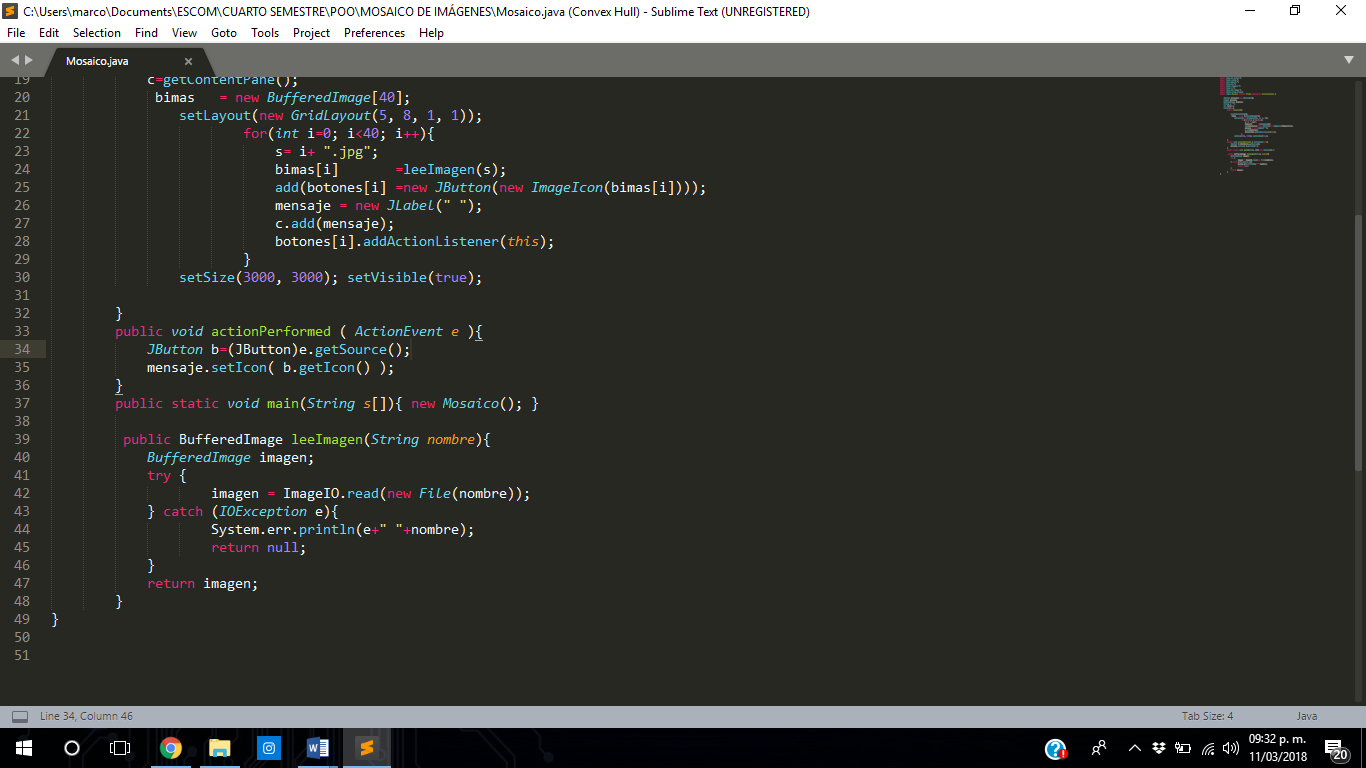
Posteriormente para la línea generamos el contenedor para poder agregar la JLabel al programa, utilizo la variable bimas para crear en ella un arreglo de 40 *BufferedImage,*  en la línea 21 llevamos a cabo la parte fuerte de la práctica pues usamos el método *GridLayout*  haciendo así un arreglo de 5x8, haciendo que el proceso entre a un ciclo con una forma peculiar, ya que las imágenes las enumeré del 0 al 40 y realizo una concatenación de tipo cadena que declaré arriba, después se leen las imágenes, así como se agregan los botones a la pantalla porque claro está decir que no basta con crearlos sino que también debes agregarlos.

En la línea 28 se da la acción a cada uno de los botones, la cual se explicara en unos momentos.



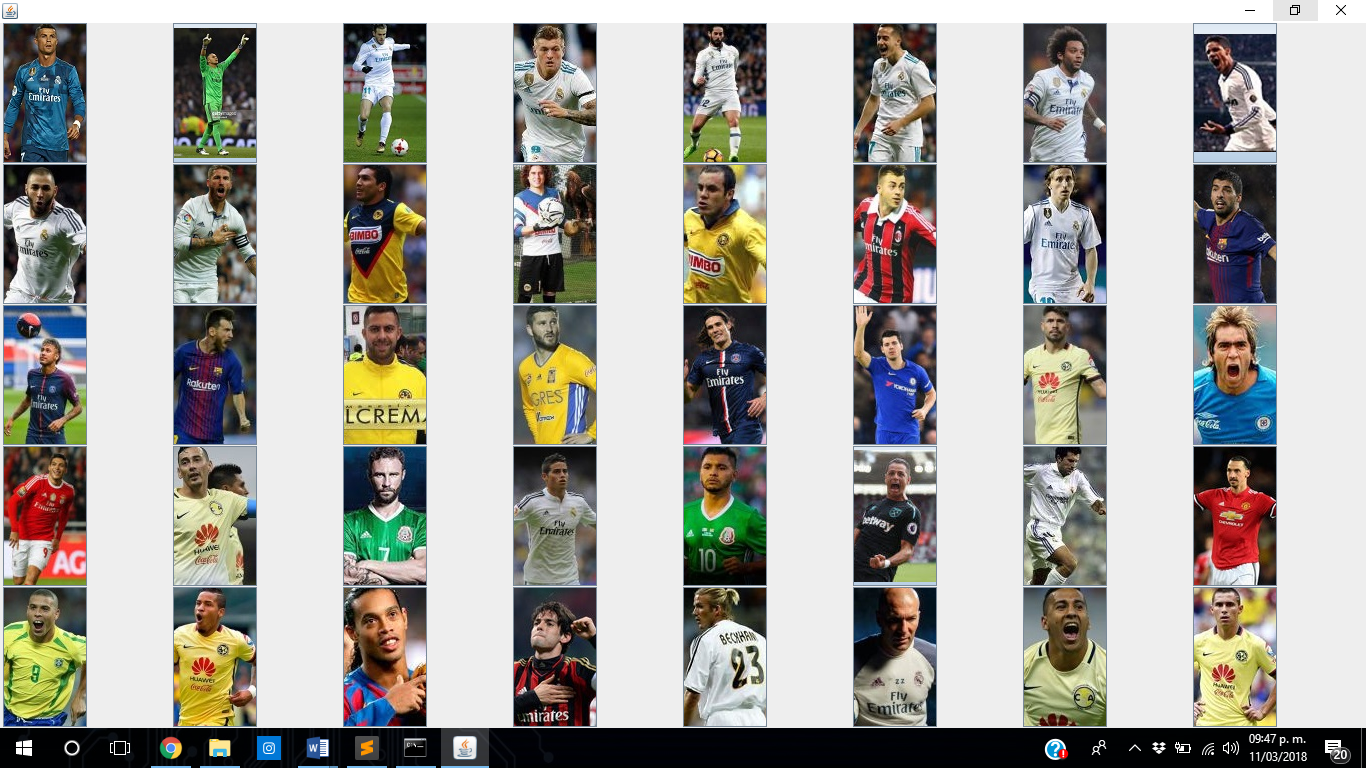
En esta parte se va a dar la acción cuando el evento click sea dado en la imagen o *JButton,*  la cual será igual al mensaje el cual se plantea y se crea.

Por último, en la línea 39 comienza mi constructor del arreglo de imágenes leyéndolas desde la carpeta.



**CONCLUSIÓN**

Concluyendo se puede decir que fue una práctica con un poco más de nivel en cuanto a grado de dificultad, pero con el código ya proporcionado anteriormente por parte del profesor se puede realizar sencillo, aunque me quedo con dudas y preguntas acerca de como eliminar espacios entre las columnas que se crean dentro del arreglo, así como el método para agregar un fondo al JFrame ya que lo busque y no encontré ninguna solución ideal a mi cuestión, por otro lado también quisiera saber por qué la etiqueta se muestra en una esquina de la pantalla y no al centro como yo lo indiqué. Y aunque en cuestión estética el programa no es de lo más bello, es funcional y hace lo que debe hacer por lo tanto el objetivo de la práctica se cumplió. Se anexan imágenes de la resolución del problema como muestra de la funcionabilidad.



En la imagen anterior se puede observar cómo se genera el arreglo de imágenes, se pretendía colocar este fondo para darle más estilo (imagen siguiente).



A continuación, se muestra la correcta funcionalidad del programa mostrando en la esquina inferior izquierda la etiqueta al hacer clic en el *Jbutton.*

