**18.- Explique en sus palabras cómo es el mecanismo de escucha utilizado en Swing para permitir**

**los eventos de clic de botón son manejados por el programa**

Los oyentes o escuchadores se encargarán de controlar y esperar eventos para que posteriormente realicen actividades en base al evento deseado. Además, se tienen distintos tipos de “escuchadores” que controlan determinados eventos. También ocurre que se tienen que implementar todos los métodos del “escuchador”, a pesar de que solo usemos uno.

Bueno, para poder colocar un botón dentro de un programa y pueda realizar un evento en nuestros proyectos gráficos primero debemos de colocar nuestros botones en un JFrame, por lo que debemos de extender nuestra clase base de la clase JFrame. Luego añadimos los componentes escogidos, como una etiqueta(JLabel) para poder identificar cuando se utilice el evento y por supuesto un botón(JButton) con el método .add(). Después se añade un evento sobre el botón(por medio del método .addActionlistener()), una clase del tipo ActionListener donde definiremos el método .actionPerformed() para manejar el evento.

Es así como se puede observar el papel que cumplen los “escuchadores” en los eventos, en este caso específico, relacionado a un botón.

**19.- ¿Cuál es la diferencia entre abstracción, encapsulación y modularidad?**

Abstracción se refiere a representar un problema en base a sus características principales, en el caso de un programa, por medio de su interfaz. Encapsulación hace referencia a la capacidad de reunir en una estructura (generalmente una clase), datos y comportamientos de objetos considerados de una misma identidad, según su nivel de abstracción y ocultar la manera en que fue implementada de otras estructuras o del mismo usuario. Por último, modularidad hace mención a una manera de resolver problemas, por medio de la división de un problema grande en problemas pequeños, similares a módulos (parecido a una **caja negra**) que luego de cumplir con su tarea se unen para resolver el problema general.