UAEM | Universidad Autónoma del Estado de México

CENTRO UNIVERSITARIO UAEM ZUMPANGO

Ingeniería en computación

Programación Avanzada

LIRA ALANIS ESTEBAN

#Cuenta : 1624256

Profesor: Asdrúbal López Chau

3 / Noviembre / 2018

UAEM | Universidad Autónoma del Estado de México

Índice

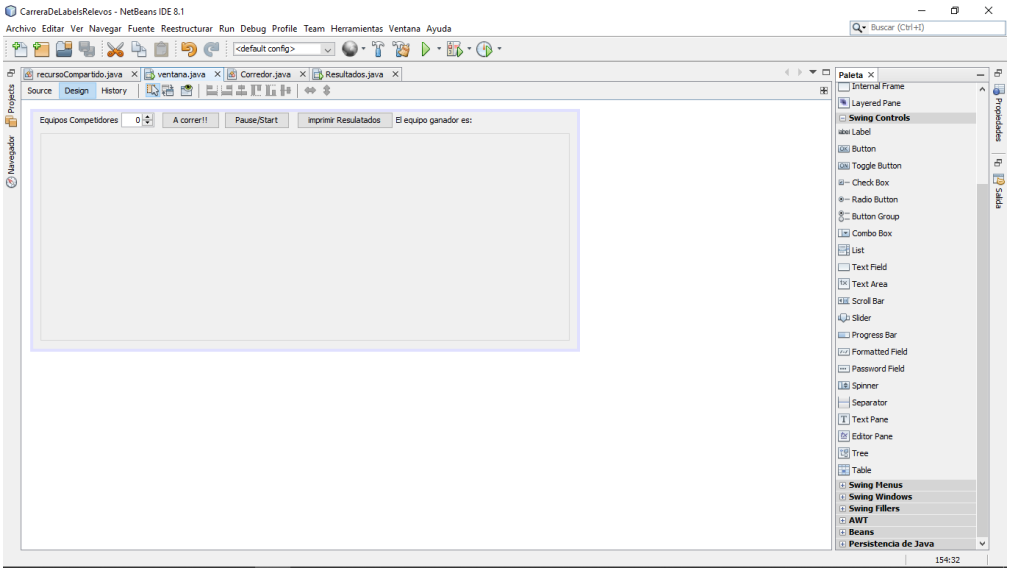
Introducción

La carrera de labels es un proyecto que da lugar a un trabajo de lleno de componentes, métodos y trucos, que llevan un cierto orden con variables que tienen un motivo en específico con respecto a los threads como lo son las funciones sincronizadas entre si con una cierta cantidad de variables y haciendo uso de las llamadas barreras cíclicas de java para Threads clásicos de java swing.

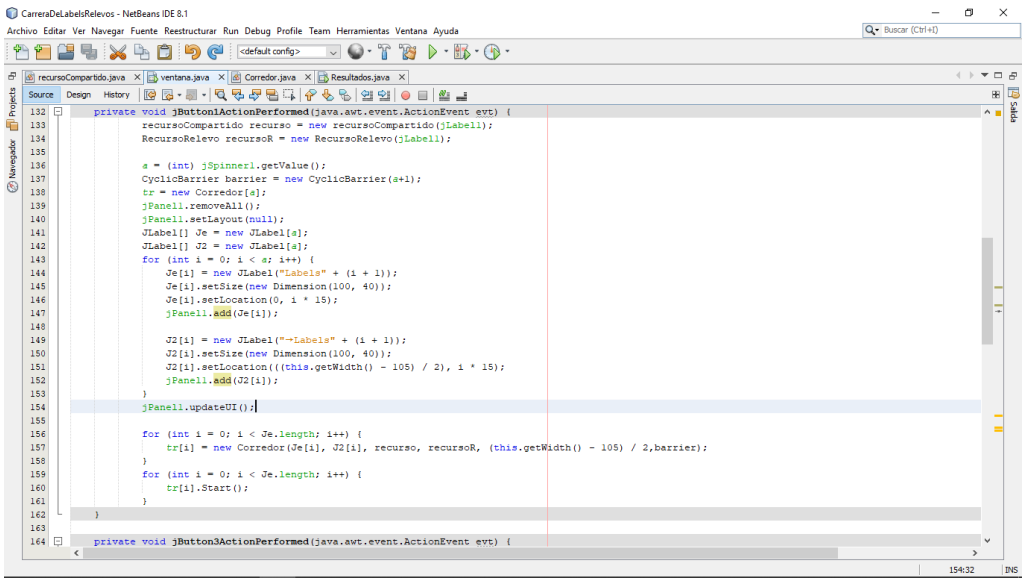
Java pone lo mejor de lo que tiene a sus proyectos y lo interesantes que son los hilos el caso de poder tener varios por decirlo así main´s para que hagan cada uno una función específica en palabras comunes “que cada quien haga su chamba“ y para el uso de la barrera cíclica “el que termine primero espera que los demás terminen”.

Desarrollo

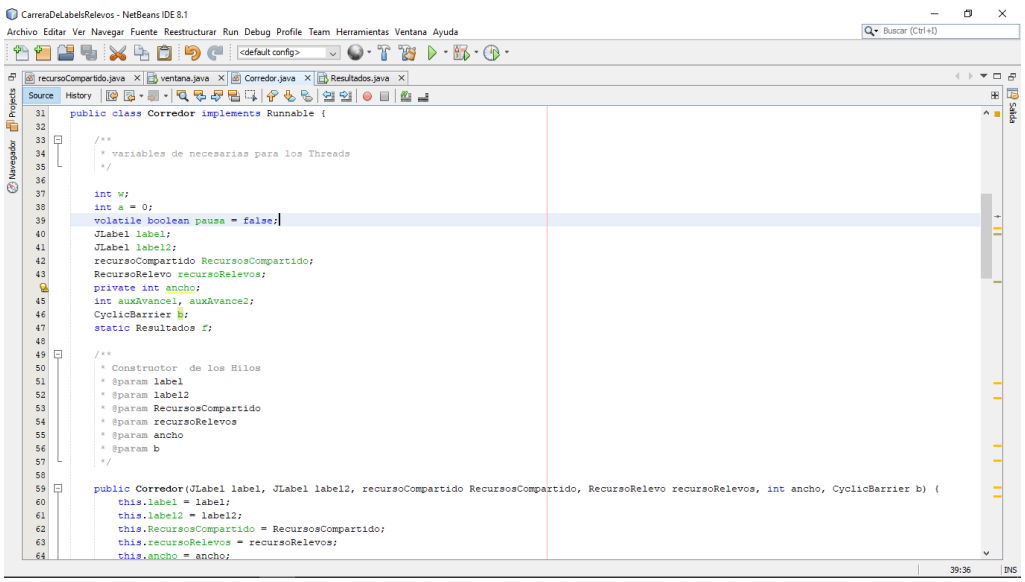
Primero fue necesario crear el proyecto de netbeans y con una Clase JFrame para la GUI y que esta carrera sea emocionante viendo cómo se rebasan una y otra pero para ello es necesario de vez en cuando usar y bueno es más cómodo usar la paleta de componentes pero en el ámbito grafico solo colocaremos los controles



Estos constan de un par de labels para decirle al usuario de que está manipulando y tres botones que están destinados para lanzar los hilos que controlan las labels y otro para detener la carrera y poder reanudarla y otro para mostrar una tabla con las posiciones una vez terminada la carrera



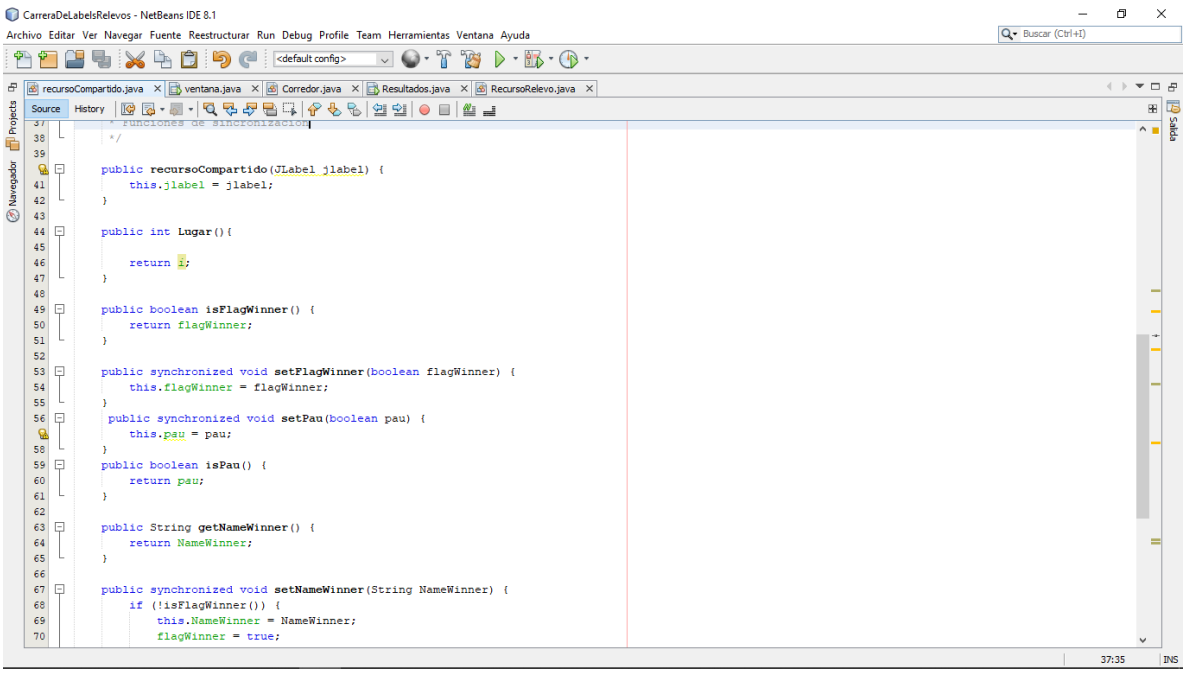
El botón que genera la llamada de los Threads es “A correr” que es este Tramo de código crea los hilos y los echa a andar claro antes creamos las labels que controlaran y le asignamos el equipo de dos Threads al arreglo de objetos de la clase entonces ahora será un arreglo de equipos de Threads Corredor como ya se dijo asignando sus recursos como el que ocupa para el relevo y para que obtenga la bandera de ganador y que sepa cuan larga es la carrera y por supuesto la barrera cíclica.



Una vez creado la GUI se continúa con la parte más importante de estos proyectos como lo es programar el Thread.

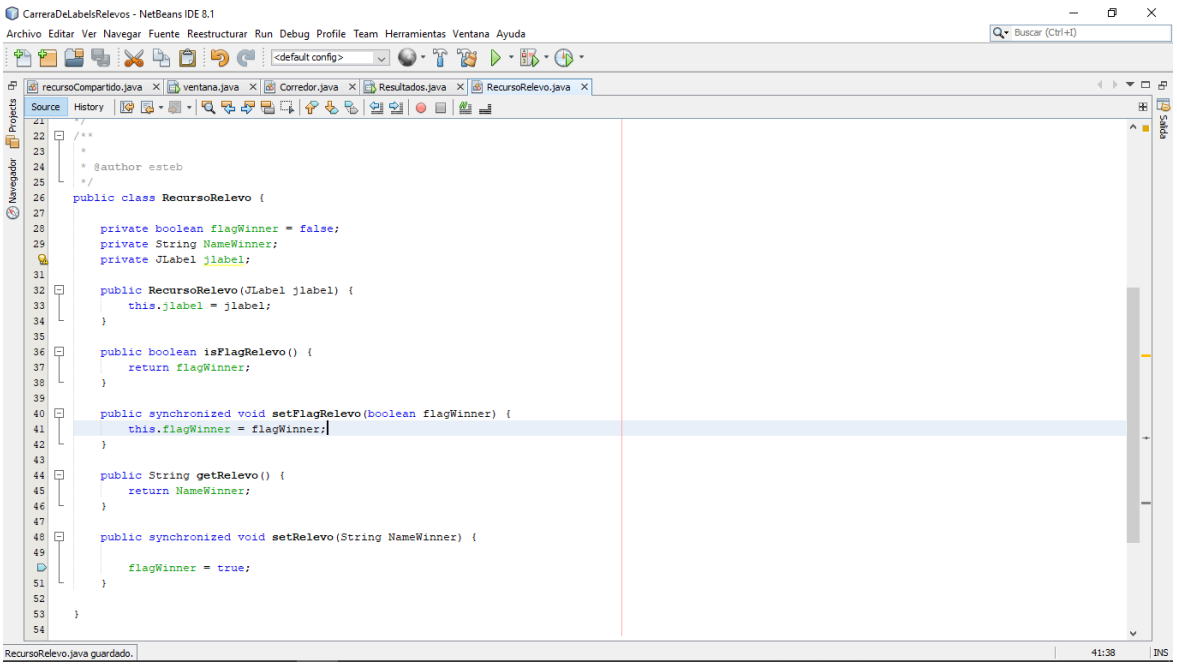
Primero Creamos el Contructor con todos sus variables que necesitamos de la GUI además se incorporan unas variables como una JFrame de Resultado donde se alojan los resultados y un par de contadores que se ocupan para definir unos bucles for asi como un variable boolean que es la que se encarga de detener o reanudar un Thread.

La parte que más causa conflictos como es la función run la cual reescribimos con el proceso que el Thread va a ejecutar y pues para no hacerla tan robusta se creó una función llamada proceso que contiene todo lo que el Thread necesita hacer como lo es crear una variable random para que de manera aleatoria se mueva la label una cierta distancia asi mismo que el hilo que se esté ejecutando duerma un momento para que los demás puedan avanzar y cuando termine de llegar a la meta definida por ancho esté llegue y tome la FlagWinner de la variable sincronizada y en la barrea ciclica ahí espere a los demás hilos .

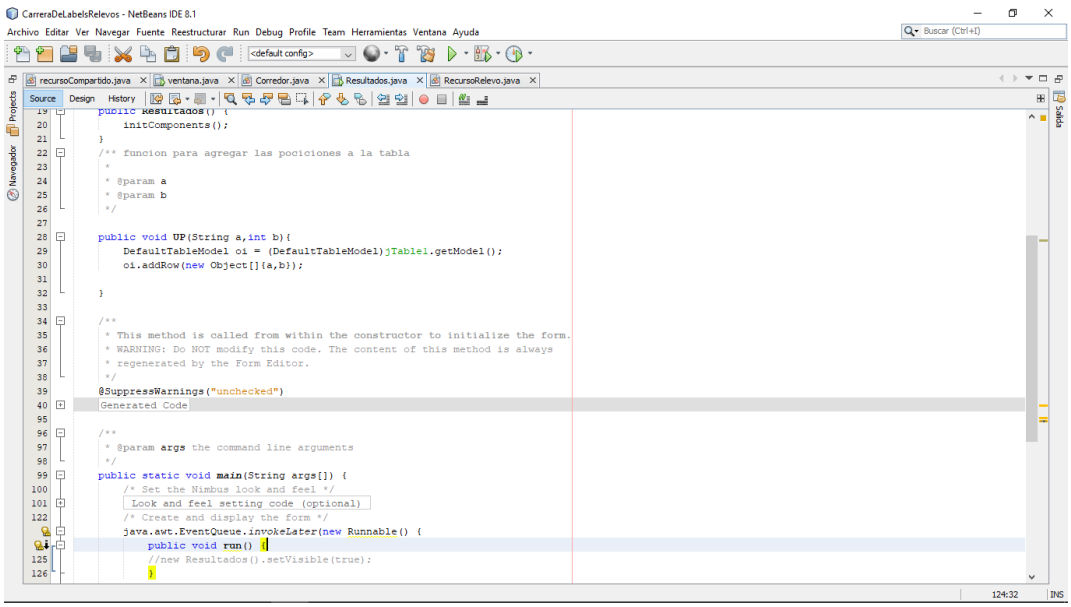


En la Parte de la Clase de RecursoCompartido es la clase que aloja a la función y variable que están sincronizadas con los hilos en ejecución incluso con el hilo principal utilizadas para hacer exclusión mutua y que solo haya un ganador.

De la misma clase se obtiene en orden el lugar y nombre en el que las labels llegan a la meta y estos datos son ocupados para enviárselos a la clase Resultados para mostrarlos en la tabla de posiciones.



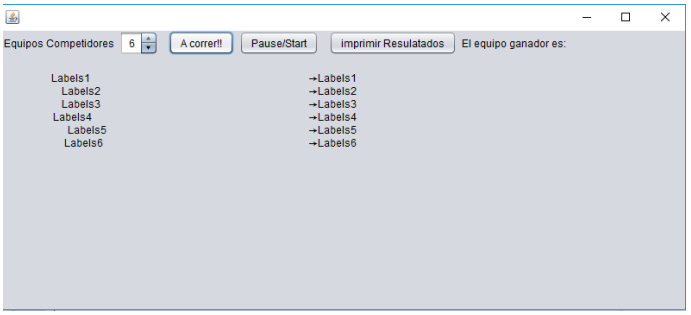
La clase recursoRelevo hace algo parecido a RecursoCompartido ya que esta solo les dice a los hilos que su pareja ya llego y que ya pueden avanzar para esta Clase se ocupó la misma estructura de la clase RecursoCompartido ya que hacen cosas parecidas.



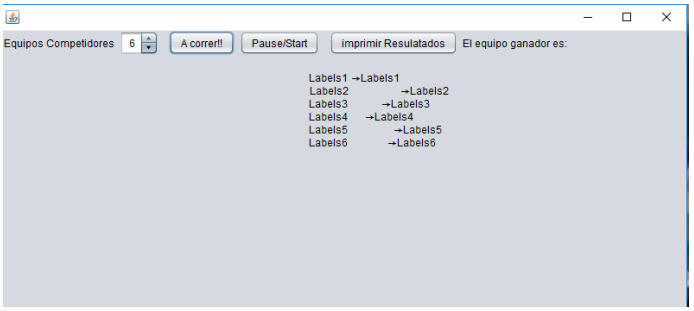
En la clase Resultados se reciben de la clase ventana desde el botón de Imprimir resultado el nombre del equipo y su puesto y son agregados a la tabla con la función UP con un string y un int de parámetros.

Funciona asi:

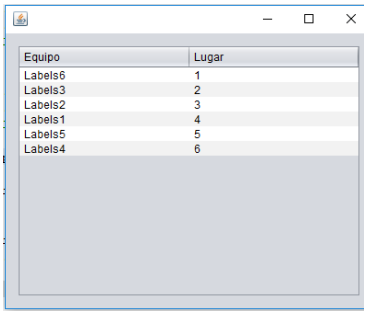
Colocamos 6 Labels en el Jspinner y a continuación damos clic en “A correr”.



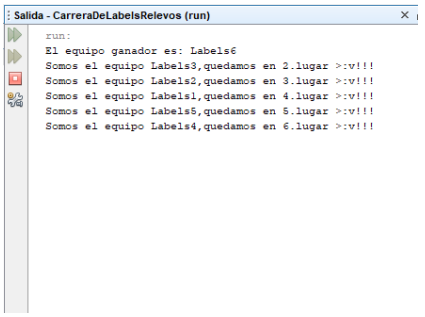
Asi comienza el relevo sale el segundo grupo.



Y gano solo una y fue el equipo de labels6 y asi son las estadisticas en la tabla de posiciones y en la consola de Netbeans.



Equipo	Lugar
Labels6	1
Labels3	2
Labels2	3
Labels1	4
Labels5	5
Labels4	6



Conclusiones.

Es interesante la forma de trabajar con Threads y con CyclicBarriers así como con exclusión mutua que en hilos de programación avanzada bueno son unos trucos que facilitan en cierto modo la programación paralela de subprocesos que son bastante eficientes si son usados de la manera más correcta y bueno el uso de esto es como ya lo vimos un juego y como no puede dar lugar a mas juegos con threads aleatorios.

Incluso mientras uno esta ejecutando una tarea otros ya hicieron dos procesos cabe decir que es muy curiosos que la clase main es un Thread y en las barreras cíclicas debe tomarse en cuenta. La exclusión mutua también tiene múltiples usos como en bancos, en cuentas de clubes de ocio, juegos y aplicaciones móviles.

Referencias

https://jarroba.com/multitarea-e-hilos-en-java-con-ejemplos-thread-runnable/

https://docs.oracle.com/javase/7/docs/api/java/lang/Thread.html