

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERIA  
ESCUELA DE CIENCIAS Y SISTEMAS  
MODELACION Y SIMULACION 1

# MANUAL TECNICO

## GRUPO 10

<b>Rony Alberto De León González</b>	<b>200915038</b>
<b>Robson Giovanni Cruz Guerrero</b>	<b>200915453</b>
<b>Erick Geovani Dávila Reyes</b>	<b>200915455</b>

## **Introducción**

La aplicación consiste en el desarrollo de un pequeño programa que consiste en un simulador de movimiento rectilíneo uniformemente acelerado el cual a partir de los datos iniciales del tiempo que se tarda un auto en ir de 0 a 200 km/h realiza los cálculos de distancia, tiempo, aceleración, etc.

## CONCEPTOS PRINCIPALES

### MRUA

El movimiento rectilíneo uniformemente acelerado (MRUA), también conocido como movimiento rectilíneo uniformemente variado (MRUV), es aquel en el que un móvil se desplaza sobre una trayectoria recta estando sometido a una aceleración constante.

### Fórmulas utilizadas

#### Nomenclatura:

$d_f$  = distancia final

$d_o$  = distancia inicial

$v_o$  = velocidad inicial

$v_f$  = velocidad final

$t$  = tiempo

$a$  = aceleracion

### DISTIANCIA FINAL

$$d_f = d_o + \left( \frac{v_f + v_o}{2} \right) t$$

$$d_f = d_o + v_o t + \frac{at^2}{2}$$

### ACELERACION

$$a = \frac{v_f - v_o}{t}$$

### VELOCIDAD FINAL

$$V_f^2 = V_o^2 + 2a (df - do )$$

$$v_f = v_o + at$$

## **JAVA**

Lenguaje de programación de propósito general, concurrente, orientado a objetos y basado en clases que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como WORA, o "write once, run anywhere"), lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra.

## **Netbeans**

NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos.

### **DETALLES TECNICOS DE DESARROLLO DE LA APLICACION**

<b>Lenguaje de programación utilizado:</b>	<b>Java</b>
<b>Versión de java</b>	<b>Jdk 7u55</b>
<b>IDE</b>	<b>Netbeans 7.3</b>
<b>Repositorio</b>	<b>GITHUB</b>
<b>Dirección del repositorio</b>	<b><a href="https://github.com/cruzriverita/MYS1_PROYECTO">https://github.com/cruzriverita/MYS1_PROYECTO</a></b>

### Equipos de desarrollo (características básicas)

CARACTERISTICA	EQUIPO 1	EQUIPO 2	EQUIPO 3
Modelo	Samsung 355v4c	Elit	HP Pavilion 17
Procesador	AMD A6-4400 APU	Intel Pentium dual-core 1.4 hz	AMD A8-5550M APU
Memoria Ram	4 GB	2 GB	8 GB
Disco Duro	500 GB	250 GB	750 GB
Sistema Operativo	Windows 8 Enterprise 64 bits.	Windows 7 Enterprise 64 bits	Windows 8.1 Pro 64 bits

### DESCRIPCION DEL CODIGO DE LA APLICACIÓN

#### LIBRERIAS

Nombre	Descripción
JDK 1.7	Las librerías que nos brinda SUN para con los distintos objetos de java
JfreeChart 1.0.4	librería/extensión Gratuita para el lenguaje de Programación Java la cual te permite utilizar Gráficos de todo tipo y de gran calidad en tus Proyectos.
j-Commom 1.0.8	Complemento para jfreechart, necesario para poder visualizar las imágenes.

## PAQUETES

Nombre	Descripción
Principal	Contiene las clases de la todos los GUI
lógica	Contiene las clases en al cual está basado todo el funcionamiento de la aplicación
Imágenes	Contiene todas las imágenes utilizadas en las GUI's

## CLASES (paquete)

Ventana Principal (principal)	Creada en netbeans 7.3, esta clase con tiene la interfaz gráfica principal de la aplicación, es un formulario que interactúa con el usuario.
Graficar (principal)	Clase encargada de llamar a la librería JfreeChart y se encarga de graficar los puntos en la gráfica lineal para mostrar la velocidad vs tiempo y desplazamiento vs tiempo de los automóviles disponibles en la simulación de la aplicación.
Datos_estaticos (lógica)	Clase que contiene las variables con los valores predeterminados de los carros como nombre, velocidad maxi ...
Nodo_simulacion (logica)	Clase que contiene la información que los carros que se están manejando durante la simulación. Datos como distancia a recorrer, el panel que se está utilizando para la pista, la imagen del carro...
Hilo_para_simulacion (logica)	Esta clase hereda thread, ya que en esta clase se ejecuta el hilo para poder manejar la simulación de un carro. Esta clase está contenida en el nodo_simulacion.

Coloca_imagen	Esta clase hereda Jpanel, ya que es utilizada para dibujar la pista y el carro. Que se utiliza en la simulación, contiene la función Graphics la cual se utiliza para pintar los gráficos en el panel.
---------------	--

## METODOS

### Clase Ventana Principal

Elimina carro(): método que es llamado cuando se elimina un carro que haya sido agregado a la simulación. El cual elimina todos los elementos del panel y luego solo agrega los que quedan.

Pone visible eliminar(int): este método recibe un numero entre 0-3 el cual le pone la propiedad al botón de eliminar según sea la posición colocada a través de la entrada.

Pone invisible eliminar(): este método le coloca la propiedad de visibilidad en false a todos los botones de eliminar.

Poner datos(int): este método recibe un numero como entrada, y según sea el número setea los valores predeterminados al panel de características que se encuentra en el Jdialog de agregar carro , 0 para Lamborghini, 1 McLaren, 2 Volkswagen y 3 Ford.

### Clase Nodo simulación

Nodo simulación (ImageIcon,coloca\_image,Jlabel,int,int,int,int):

constructor de la clase, el cual recibe todos los valores que se va a utilizar para la simulación, como lo son la distancia, el panel que se va a utilizar, la velocidad y la aceleración.

Pone \*\* (int):(pone\_tiempo,pone\_posx,pone\_velocidad,...) este método setea valores nuevos a las variables que se utilizan para la simulación;

Obtén \*\*(): \_\_\_\_\_(obten\_tiempo,obten\_posx,obten\_velocidad,...)  
retorna el valor de la variable solicitada.

Iniciar simulacion(): crea el objeto hilo\_para\_simulacion  
y lo pone en estado start. Para que inicie la simulación.

### Clase hilo para simulacion

Hilo para simulación \_\_\_\_\_(coloca\_image,int,int):  
este método es el constructor de la clase, recibe los valores  
para la simulación, como el panel donde se mostrara el carro,  
la distancia que recorrerá y la aceleración.

Detener hilo(): método que pone en estado stop, destruye el  
hilo que está corriendo actualmente.

Parar hilo(): pone al hilo en estado suspend, el cual  
suspende el hilo, hasta recibir la señal de reanudación.

Reanudar hilo(): reanuda el hilo, para poder seguir con la  
simulación.

### Clase Graficar

public static void graph (String,String ,double,double  
,double ,double ,double ,double ,double , double , double  
,double , double ,double , double ,double ,double , double  
,String): Método que se manda a llamar desde la  
ventana\_principal.java para graficar la velocidad y  
desplazamiento de los vehículos. Recibe como paramentros el  
nombre del grafico, las etiquetas de sus ejes y los puntos  
en el eje "x" y en el eje "y" a graficar.

### Clase Ventana Principal

public void calculos (): Método que realiza todos los  
cálculos del MRUA de la aplicación, dentro de el se llevan  
a cabo los cálculos para cada uno de los automóviles  
disponibles en la aplicación, se obtienen los resultados y  
las gráficas de los mismos.

### **VARIABLES**



### Clase Datos estaticos

Variables usadas para los datos predeterminados de los carros seleccionados, con los prefijos L para Lamborghini, M para McLaren, Vw para Volkswagen y F para Ford.

\*\_peso: guarda el peso del auto.

\*\_Velocidad\_max: guarda la velocidad maxima que puede alcanzar el auto.

\*\_aceleracion: contiene el tiempo en el cual alcanza la velocidad de 100 km/h .

\*\_aceleracion2: contiene el tiempo en el cual alcanza la velocidad de 200 km/h.

\*\_imagen: contiene la ruta de la imagen del carro .

\*\_nombre: contiene el nombre completo del carro.

### Variables para la elección de carro

Lista\_simulacion: es una lista que va contener los carros que se deseen simular.

Distancia: contiene el valor de la distancia seleccionada cuando se agrega un carro.

tipo\_carro: contiene el tipo de carro que fue seleccionado, 0 para Lamborghini, 1 McLaren, 2 Volkswagen y 3 Ford.