**RESOLUCION DE ECUACIONES**

**CODIGO UTILIZADO**

function P\_2

clc,clear%limpiar la pantalla

tspan=[0,20];%tiempo de la simulacion

p0=[0;1];%condiciones iniciales

%Posicicon

[t,p]=ode45(@pendulo\_Cartesianas,tspan,p0,[]);

%resolver la ecuacion diferencial con base a una funcion auxiliar

%funcion-Tiempo-valor inicial-parametros para el calculo(se deja vacio si no hay)

plot(t,p(:,1),'r') %dibuja la posición en cartesianas

hold on

[t,p]=ode45(@Pendulo\_Polares,tspan,p0,[]);

plot(t,p(:,1),'g') %dibuja la posición en polares

xlabel('Tiempo');

ylabel('Poscicion');

%Velocidad

[t,p]=ode45(@pendulo\_Cartesianas,tspan,p0,[]);

%resolver la ecuacion diferencial con base a una funcion auxiliar

%funcion-Tiempo-valor inicial-parametros para el calculo(se deja vacio si no hay)

figure;

plot(t,p(:,2),'y') %dibuja la Velocidad en cartesianas

hold on

[t,p]=ode45(@Pendulo\_Polares,tspan,p0,[]);

plot(t,p(:,2),'b') %dibuja la Velocidad en polares

xlabel('Tiempo');

ylabel('Velocidad');

function d = pendulo\_Cartesianas(t,p)

l=5;%longitud de la cuerda

g=9.8;%gravedad

%El vector p guarda las variables Teta=p(1) y Teta.=p(2)

%En el vector v ponemos las derivadas dx/dt=d(1) y dv/dt=d(2)

d=[p(2);

-(g/l)\*(p(1));]

function d = Pendulo\_Polares(t,p)

l=5;%longitud de la cuerda

g=9.8;%gravedad

%El vector p guarda las variables Teta=p(1) y Teta.=p(2)

%En el vector v ponemos las derivadas dx/dt=d(1) y dv/dt=d(2)

d=[p(2);

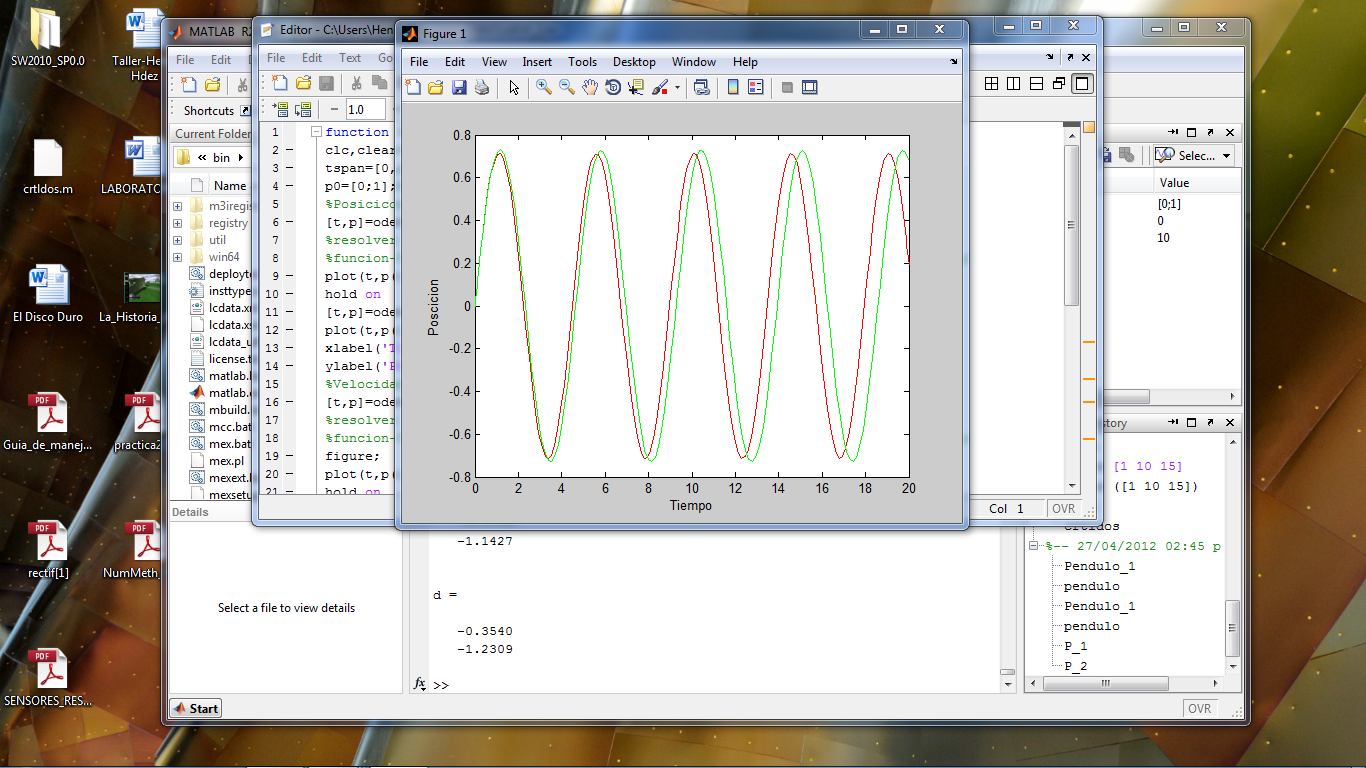
-(g/l)\*sin(p(1));]

**RESULTADOS OBTENIDOS**

**POSICION**

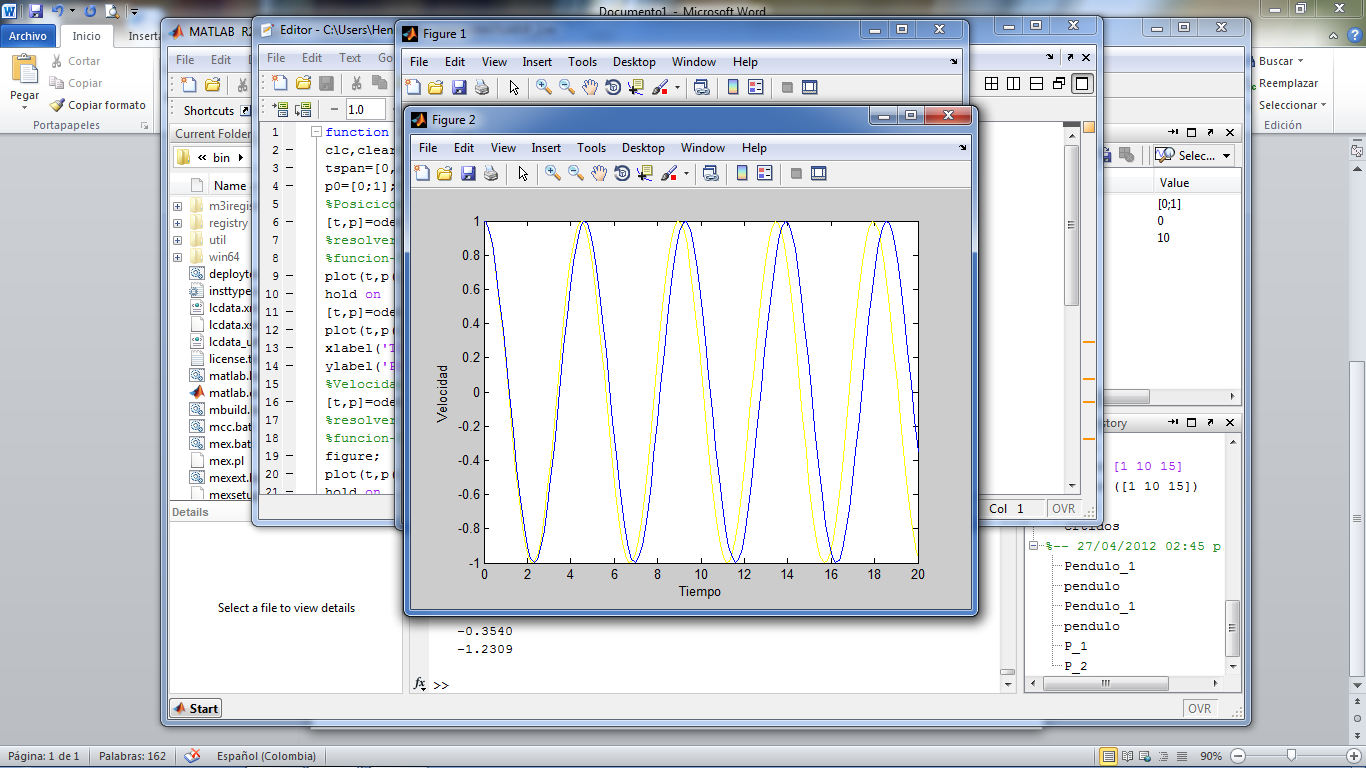
**LINEA ROJA EN COORDENADAS CARTESIANAS**

**LINEA VERDE EN COORDENADASPOLARES**



**LINEA AMARILLA EN COORDENADAS CARTESIANAS**

**LINEA AZUL EN COORDENADAS POLARES**



Donde A=Derivadas

B= equivalencias de las derivadas

C=Condiciones iniciales