[[1]](#footnote-2)

Reporte IEEE Proyecto Final(junio 2021)

Carlos, Alvarado (20200118) Mario Pisquiy (20200399) Javier Mazariegos (20200223)

Computer Science, UFM

*Abstract*— A program was developed for the measurement and control of a diabetic person. This requires a database. The main focus of the program is for doctors but it can also be used by patients.

# INTRODUCCIÓN

Al partir de una necesidad como lo es el control de una enfermedad como lo es la diabetes, se desarrolló un programa para ayudar a médicos y pacientes a llevar un registro y en cierto modo, una ayuda para el análisis de su desarrollo.

Para los análisis, se realizó un histograma de la glucosa medida correspondiente al tiempo, también una interpolación usando el método de Newton para la relación glucosa-tiempo. Luego, usando los métodos de diferencias finitas se encontraron las razones de cambio de la glucosa con respecto al tiempo (velocidad) y luego la segunda derivada de los mismos datos (aceleración). Luego, a través de encontrar el valor promedio de una función y utilizando una derivada a través del método del trapecio compuesto con intervalos indefinidos. Para poder encontrar la meta de la glucosa para el paciente, se utilizó una regresión lineal simple y un método de prueba y error, para hallar la tendencia también se utilizó regresión lineal simple. Finalmente, para el resumen estadístico se utilizaron herramientas de estadística descriptiva.

Todas estas herramientas están puestas para la disposición del análisis de datos ingresados y poder ser de ayuda para médicos y pacientes de diabetes.

# Scripts

//CODIGO PRINCIPAL

pkg load io

%leer fechas

excelDates = xlsread('datos.xlsx','B1:B138');

nanElements = isnan(excelDates); %indices donde estan los valores NaN

excelDates(nanElements) = []; %eliminar las casillas

matlabDates = 693960 + excelDates;

excelDates = datestr(matlabDates,2);

%leer horas

excelhoras = xlsread('datos.xlsx','D1:D138');

nanElements = isnan(excelhoras);

excelhoras(nanElements) = [];

exhoras = excelhoras;

excelhoras = datestr(excelhoras,'HH:MM');

%leer mg/dlmread

excelglucosa = xlsread('datos.xlsx','C1:C138');

nanElements = isnan(excelglucosa);

excelglucosa(nanElements) = [];

%leer condicion

[n, condiciones] = xlsread('datos.xlsx','E3:E138');

printf('\n\n\n\n\n');

%p = condiciones(136,1);

%a=condiciones{136,1};

%\_\_\_\_\_\_ Pregunta: Primera dosis de mediacemento \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

hora\_mediacamento = inputdlg ("Ingrese la hora en la que tom su primera dosis de medicamento (HH:MM)");

hora\_mediacamento = hora\_mediacamento{1,1};

tiempo = strsplit(hora\_mediacamento,":");

horas = str2double(tiempo{1,1});

hora\_entera\_medicamento = horas;

minutos = str2double(tiempo{1,2})/60;

hora\_decimal\_medicamento = minutos;

hora\_mediacamento = (horas+minutos)/24;

hora\_total\_mediacmento = hora\_mediacamento\*24;

%\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Fin Pregunta \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

%\_\_\_\_\_\_\_ Clculo numero de horas (hora glocosa menos toma de medicamento) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

tiempo =[];

temporal = 0.00;

for i=1: length(excelhoras);

temporal = (exhoras(i) - hora\_mediacamento);

if temporal <0;

temporal2 = datestr(exhoras(i),'HH:MM');

temporal2 = strsplit(temporal2,":");

h2 = str2num(temporal2{1,1});

m2 = str2num(temporal2{1,2});

temporal = datestr(hora\_mediacamento,'HH:MM');

temporal = strsplit(temporal,":");

h = str2num(temporal{1,1});

m = str2num(temporal{1,2});

h = h2-h;

m = (m2-m)/60;

temporal = h+m;

tiempo(i) = temporal;

else

temporal = datestr(temporal,'HH:MM');

temporal = strsplit(temporal,":");

h = str2num(temporal{1,1});

m = str2num(temporal{1,2})/60;

temporal = h + m;

tiempo(i) = temporal;

endif

endfor

%\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Fin Clculo \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

%\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Pregunta: Rango de analisis \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

fecha\_inicio = inputdlg ("Ingrese fecha de inicio (mm/dd/yy): ");

fecha\_inicio = datenum(fecha\_inicio{1,1}, "mm/dd/yy");

while(fecha\_inicio < matlabDates(1));

fecha\_inicio = inputdlg ("Ingrese otra vez la fecha de inicio (mm/dd/yy): ");

fecha\_inicio = datenum(fecha\_inicio{1,1}, "mm/dd/yy");

endwhile

fecha\_final = inputdlg ("Ingrese fecha final (mm/dd/yy): ");

fecha\_final = datenum(fecha\_final{1,1}, "mm/dd/yy");

while(fecha\_final>matlabDates(length(matlabDates)));

fecha\_final = inputdlg ("Ingrese otra vez la fecha final (mm/dd/yy): ");

fecha\_final = datenum(fecha\_final{1,1}, "mm/dd/yy");

endwhile

x = 1;

indices=[];

for i=1:length(excelDates);

if matlabDates(i) >= fecha\_inicio && matlabDates(i) <= fecha\_final;

indices(x) = i;

x = x+1;

endif

endfor

random = indices;

if length(indices) > 10;

random = [indices(1): 1: indices(x-1)];

random = random(randperm (10,10));

endif

for i=1: length(indices)

datestr(matlabDates(indices(i)))

endfor

%\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Fin Pregunta \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

%\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Menu de opciones \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

randomSort=sort(random);

tiempoSort=[];

glucosaSort=[];

tiempoSortGrafica = [];

glucosaSortGrafica= [];

for i=1:length(randomSort);

tiempoSort(i)=tiempo(randomSort(i));

tiempoSortGrafica(i)=tiempo(randomSort(i));

glucosaSort(i)=excelglucosa(randomSort(i),1);

glucosaSortGrafica(i) = excelglucosa(randomSort(i),1);

endfor

for i=1:length(tiempoSort);

Nindices = find(tiempoSortGrafica == tiempoSort(i));

if length(Nindices>1)

for k=2:length(Nindices);

tiempoSortGrafica(Nindices(k)) = [];

glucosaSortGrafica(Nindices(k)) = [];

endfor

endif

endfor

%tiempoSort

tiempoSortGrafica;

%glucosaSort

glucosaSortGrafica;

while 1;

choice = menu('Seleccione una opcion: ','Grficas','Tabla de metabolizacin de glucosa','Aceleracin metablica de glucosa','Glucosa Promedio','Glucosa-Meta','Tendencia','Resumen Estadistico','Salir');

clf

clc

switch choice

case 1

%Funcion graficas

choice2 = menu('Eleccin de graficas','Puntos','Polinomio')

switch choice2

case 1

%Grafica por puntos

graficas(tiempoSortGrafica,glucosaSortGrafica,1);

case 2

%Grafica por polinomio

Yinter = graficas(tiempoSortGrafica,glucosaSortGrafica,2);

grid on

case 3

otherwise

endswitch

case 2

%Tabla de metabolizacion de glucos

derivada = primeraDerivada(tiempoSortGrafica,glucosaSortGrafica);

fprintf(' Fecha Razon de Cambio Condicion\n');

for i=1:length(derivada)

if(derivada(i) < 0)

if(derivada(i)<=-10)

fprintf('%s %4.5f %4s\n',datestr(matlabDates(randomSort(i))),derivada(i),condiciones{randomSort(i),1});

else

fprintf('%s %4.5f %4s\n',datestr(matlabDates(randomSort(i))),derivada(i),condiciones{randomSort(i),1});

endif

else

if(derivada(i)>=10)

fprintf('%s +%4.5f %4s\n',datestr(matlabDates(randomSort(i))),derivada(i),condiciones{randomSort(i),1});

else

fprintf('%s +%4.5f %4s\n',datestr(matlabDates(randomSort(i))),derivada(i),condiciones{randomSort(i),1});

endif

endif

endfor

case 3

%Aceleracion metabolica

acelMeta=segundaDerivada(tiempoSortGrafica,glucosaSortGrafica);

maxAcelMeta=max(acelMeta);

minAcelMeta=min(acelMeta);

fprintf('La aceleraci?n maxima es: %4.5f\nLa aceleraci?n minima es: %4.5f\n\n',maxAcelMeta,minAcelMeta);

case 4

promGluco=(1/(max(tiempoSortGrafica)-min(tiempoSortGrafica)))\*integral(tiempoSortGrafica,glucosaSortGrafica);

printf('Glucosa promedio: %4.5f\n',promGluco);

%Glucosa Promedio

case 5

%Glucosa-Meta

nuevas\_horas =[];

glucosa\_meta = inputdlg("Ingrese el nievl del glucosa: ");

glucosa\_meta = str2num(glucosa\_meta{1,1});

horas\_aproximadas = encontrarGlucosa(tiempoSortGrafica,glucosaSortGrafica,glucosa\_meta);

for i=1:length(horas\_aproximadas);

if horas\_aproximadas(i) > 0

entera\_aprox = floor(horas\_aproximadas(i));

entera\_aprox = floor(horas\_aproximadas(i)) + floor(hora\_entera\_medicamento);

decimal\_aprox = horas\_aproximadas(i) - floor(horas\_aproximadas(i));

decimal\_aprox = (decimal\_aprox + hora\_decimal\_medicamento)\*60;

fprintf('%0.0f:%0.0f\n',entera\_aprox,decimal\_aprox);

else

hora\_aprox = horas\_aproximadas(i) + hora\_total\_mediacmento;

entera\_aprox = floor(hora\_aprox);

decimal\_aprox = (hora\_aprox - entera\_aprox)\*60;

fprintf('%0.0f:%0.0f\n',entera\_aprox,decimal\_aprox);

endif

endfor

case 6

clf

puntos=regLineal(tiempoSort,glucosaSort);

fprintf('La ecuacin es: y=%4.5f\*x+%4.5f\nEl r2 es: %4.5f\n',puntos(1),puntos(2),puntos(3));

%Tendencia

case 7

%Resumen Estadistico

# MEDIA DEL RANGO DE FECHAS

MEDIA = mean(excelglucosa(indices,1))

MEDIANA = median(excelglucosa(indices,1))

MODA = mode(excelglucosa(indices,1))

Glucosa\_Maxima = max(excelglucosa(indices,1))

Glucosa\_Minima = min(excelglucosa(indices,1))

Desviasion\_estandar = std(excelglucosa(indices,1))

case 8

break

otherwise

endswitch

endwhile

%\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Fin menu de opciones

//CODIGO DE GRAFICAS.

function Yinter = graficas (x,y,tipo);

if tipo ==1;

plot(x,y,'\*')

xlabel('Tiempo (Primera dosis - hora de medicion)');

ylabel('Nivel de glucosa (mg,dL)');

title('Glucosa - Tiempo');

grid on;

else

n = length(x); % La lingitud del vector x nos da el numero de coeficientes y terminos para el polinomio interpolante de newton.

a(1) = y(1); %Aqui esta encontrando el primer coeficiente del polinomio.

for i=1:n-1;

divDIF(i,1) = (y(i+1)-y(i))/(x(i+1)-x(i));

endfor

%de la linea 7 a la 9 se trabaja la primera diferencia divididas

for j=2:n-1;

for i=1: n-j;

divDIF(i,j) = (divDIF(i+1,j-1)-divDIF(i,j-1))/(x(j+i)-x(i));

endfor

endfor

%De la linea 12 a la 16 se calculan las diferencias dividdidas de orden superior

for j=2:n;

a(j) = divDIF(1,j-1); % Aqui se asignan las diferencias divididas a los coeficientes restantes

endfor

Xint = x;

len = length(Xint); %Se obtiene el tama�o del vector para saber donde finalizar el ciclo

for i=1:len;

Yinter(i) = a(1);

xn =1;

for k=2:n;

xn = xn\*(Xint(i)-x(k-1));

Yinter(i) = Yinter(i) + a(k)\*xn; %Se utiliza un vector para alamecnar todas las imagenes

endfor

endfor

plot(Xint, Yinter, 'g')

xlabel('Tiempo (Primera dosis - hora de medicion)');

ylabel('Nivel de glucosa (mg,dL)');

title('Curva aproximada (Polinomio Interpolante de Newton)');

endif

endfunction

——————————————————————————————————

//CODIGO DE GLUCOSA-META

function aprox = encontrarGlucosa (x,y,glucosa\_meta);

n = length(x); % La lingitud del vector x nos da el numero de coeficientes y terminos para el polinomio interpolante de newton.

a(1) = y(1); %Aqui esta encontrando el primer coeficiente del polinomio.

for i=1:n-1;

divDIF(i,1) = (y(i+1)-y(i))/(x(i+1)-x(i));

endfor

%de la linea 7 a la 9 se trabaja la primera diferencia divididas

for j=2:n-1;

for i=1: n-j;

divDIF(i,j) = (divDIF(i+1,j-1)-divDIF(i,j-1))/(x(j+i)-x(i));

endfor

endfor

%De la linea 12 a la 16 se calculan las diferencias dividdidas de orden superior

for j=2:n;

a(j) = divDIF(1,j-1); % Aqui se asignan las diferencias divididas a los coeficientes restantes

endfor

intersectos = [];

aprox = [];

repetidos = [];

for m=1: length(x)-1;

%si el siguiente numero es mayor al actual. el minimo va a ser el actual y el maximo el siguiente

if(x(m+1)> x(m))

Xint = x(m):0.05:x(m+1);

else

%si el siguiente numero es menor al actual, el minimo va a ser el siguiente y el maximo va a ser el actual

%se hace un flip

Xint = flip(x(m+1):0.05:x(m));

endif

Yinter = [];

len = length(Xint);

for i=1:len;

Yinter(i) = a(1);

xn =1;

for k=2:n;

xn = xn\*(Xint(i)-x(k-1));

Yinter(i) = Yinter(i) + a(k)\*xn;

endfor

endfor

menores = find(Yinter <= (glucosa\_meta+2));

mayores = find(Yinter >= (glucosa\_meta-2));

intersectos = intersect(menores,mayores);

if length(intersectos)>0;

for t=1:length(intersectos);

aprox = [aprox,Xint(intersectos(t))];

endfor

endif

%fprintf("Iteracion %0.0f\n",m)

%Xint

%Yinter

endfor

imagen\_aprox = [];

for i=1: length(aprox)

aprox(i) = str2num(sprintf('%0.4f',aprox(i)));

endfor

for i=1: length(aprox)

imagen\_aprox(i) = str2num(sprintf('%0.4f',aprox(i)));

endfor

indice\_N = [];

for i=1: length(imagen\_aprox)

indice\_N = find(aprox == imagen\_aprox(i));

if length(indice\_N) >1

if length(indice\_N) == 2

aprox(indice\_N(2)) = [];

else

for k = 2: length(indice\_N)-1

aprox(indice\_N(k)) = [];

endfor

endif

endif

endfor

aprox;

endfunction

——————————————————————————————————————

//INTEGRACIÓN POR MEDIO DE TRAPECIO

%Se utilizar� el m�todo del trapecio con intervalos irregulares

function I=integral(x,y);

n=length(x);

suma=0;

for i=1:n-1;

suma=suma+(y(i)+y(i+1))\*(x(i+1)-x(i));

endfor

I=0.5\*suma;

endfunction

——————————————————————————————————————

//DIFERENCIACIÓN NUMERICA A TRAVEZ DE DIFERENCIAS FINITAS

function dx = primeraDerivada(x,y)

%calcula la derivada de una funcion dada por un conjunto de puntos.

%Se utilizara la derivacion numerica por diferencias finitas.

% x es un vector que almacena las coordenadas x de los puntos

% y es un vector que almacena las coordenadas y de los puntos

% La variable de salida es un vector dx

n = length(x);

dx(1) = (y(2) - y(1))/(x(2)-x(1)); %diferencia finita hacia adelante

for i=2: n-1;

dx(i) = (y(i+1)-y(i-1))/(x(i+1)-x(i-1)); %diferencia finita centrada

endfor

dx(n) = (y(n)-y(n-1))/(x(n)-x(n-1)); %Diferencia finita hacia atras

endfunction

——————————————————————————————————————

// REGRESION LINEAL

function puntos=regLineal(x,y);

n=length(x);

x=x(:); y=y(:);

sx=sum(x); sy=sum(y); %suma de las columnas de "x" y de "y"

sx2=sum(x.\*x); sxy=sum(x.\*y); sy2=sum(y.\*y);

puntos(1)=(n\*sxy-sx\*sy)/(n\*sx2-sx^2);

puntos(2)=sy/n-puntos(1)\*sx/n;

puntos(3)=((n\*sxy-sx\*sy)/sqrt(n\*sx2-sx^2)/sqrt(n\*sy2-sy^2))^2;

xp=linspace(min(x),max(x),2);

yp=puntos(1)\*xp+puntos(2);

plot(x,y,'\*',xp,yp);

grid on;

endfunction

——————————————————————————————————————

// SEGUNDA DERIVADA POR DIFERENCIACIÓN NUMERICA POR DIFERENCIAS FINITAS

unction d2x=segundaDerivada(x,y);

n=length(x);

d2x(1)=(y(1)-2\*y(2)+y(3))/((x(2)-x(1))^2); %Segunda derivada, dif. finitas hacia adelante

for i=2:n-1;

d2x(i)=(y(i-1)-2\*y(i)+y(i+1))/((x(i)-x(i-1))^2);

endfor

d2x(n)=(y(n-2)-2\*y(n-1)+y(n))/((x(n)-x(n-1))^2);

endfunction

# Resultados

# Rango de fechas utilizado:

# 

Chart, scatter chart

Description automatically generated

Diagrama de dispersión de nivel de glucosa con respecto al tiempo.

La dispersión de los niveles de glucosa obtenida en el rango de fechas seleccionada.

Chart, line chart

Description automatically generated

Polinomio interpolante de Newton.

Polinomio dado por el método de interpolación de Newton obtenido gracias a los niveles de glucosa con respeto a las horas que fueron medidas y la hora en la que se toma el medicamento el paciente. Tomando esta última como punto 0 y los demás son las observaciones obtenidas en el rango de fechas seleccionado.

Tabla de metabolización de glucosa

Text

Description automatically generated

Registro de la razón de cambio instantánea de la glucosa con respecto al tiempo, según el rango de fechas seleccionado.

Aceleración metabólica de la glucosa.



La aceleración máxima y mínima de la metabolización de la glucosa con respecto al tiempo en el rango de fechas observado.

Nivel de glucosa promedio.



Siendo este el promedio de la función de la glucosa con respecto al tiempo.

Horas relacionadas al nivel de glucosa observado.

Text

Description automatically generated with medium confidence

Las horas mostradas son aquellas que cumplen con una aproximación de cierto nivel de glucosa que el usuario desee encontrar a lo largo del rango de fechas en el que se obtuvieron las observaciones. En este caso el rango fue de 6/1/20 (formato mm/dd/yy) hasta 6/26/20, con un nivel de glucosa de 185 mg/dL. Se debe tomar en cuenta que estas horas aproximadas son dadas por un margen de error de ±2.

Regresión lineal.



Se muestra la ecuación de la regresión lineal obtenida y el coeficiente de determinación el cual podemos ver es muy bajo.

Chart, scatter chart

Description automatically generated

Diagrama de dispersión y recta de regresión lineal.

Se muestra la dispersión de los valores obtenidos junto a la recta de regresión lineal para la muestra de que el coeficiente de determinación es bajo.

Resumen estadístico

Text, letter

Description automatically generated

Estos datos son por el rango sin ser limitado a 10 fechas.

Histograma de glucosa en el rango de fechas.

Chart, histogram

Description automatically generated

Frecuencia de los niveles de glucosa.

# Discusión de Resultados

El diagrama de dispersión, el polinomio interpolante y la regresión lineal son únicos para cada rango de fechas que se haya seleccionado, por lo que dar una aproximación del comportamiento de los datos será correcta si se habla solo en el rango especificado.

Cuando se desea hacer un análisis del comportamiento de los datos, la regresión lineal muestra una mejor representación de la dispersión de los datos, cuando el coeficiente de determinación es muy alto, debido a que el polinomio interpolante de newton utiliza puntos más alejados de los reales con el objetivo de aproximarse a los puntos dados. Sin embargo, si el coeficiente de determinación es muy bajo el polinomio muestra una mejor representación de la dispersión de los datos.

La metabolización de la glucosa puede ser utilizada por el médico para observar el cambio de los niveles de glucosa. La aceleración le permite ver al doctor la evolución de la metabolización del paciente, y de esta forma ver los cambios a través de diferentes intervalos de tiempo y poder concluir si el medicamento es efectivo o se necesita una mayor intervención.

El apartado de Glucosa-meta ayudará a que el médico pueda estimar una posible nueva hora de toma de medicamente o de una segunda dosis para controlar algunos niveles específicos de glucosa, siempre y cuando tome en cuenta que son aproximaciones dadas por el intervalo de fechas.

# Conclusiones

Una conclusión que general de los resultados obtenidos con respecto a las expectativas es que todo dependerá de los parámetros de medición, porque estos harán que la exactitud o el margen de error de los datos varié. Por ejemplo, las razones de cambio podrían ser más exactas si se amplia los datos observados porque se podría utilizar diferenciación por medio de más puntos que converge con O(h) mayor a la que utiliza menos puntos.

Al segmentar la información de datos en rangos de fechas de las cuales solo se aceptan 10 datos como máximo hacen que los resultados obtenidos estén muy distorsionados y no puedan dar una aproximación más amplia con respeto al paciente.

Esto puede dar pie a la diferenciación entre qué métodos utilizar para dar resultados más cercanos a la realidad, un ejemplo claro es que, si la dispersión de los datos es baja y obtenemos un coeficiente de determinación alto, la regresión lineal es la mejor opción.

El diagrama de dispersión depende mucho del contexto en el que se hayan tomado las lecturas, puesto que en una misma diferencia de horas puede dar diferentes medidas de glucosa debido a que fue tomada en fechas distintas después de diferentes situaciones, lo que hace que varíe la lectura, por lo que esta gráfica nos puede ayudar únicamente a ver la dispersión entre las medidas, mas que para ver la evolución del paciente.

# RECOMENDACIONES

Si se busca la utilización del mismo programa se aconseja que se acepten más puntos de observación para una mayor exactitud del comportamiento del paciente.

Para encontrar el valor promedio de una función, podemos utilizar el método de Simpson 1/3, que es más exacto que el de Trapecio Compuesto con Intervalos Indefinidos.

Para realizar la curva aproximada de Glucosa – Tiempo se recomienda utilizar el Polinomio Interpolante de Lagrange para obtener una mejor aproximación.

Bibliografía

1. Steven C. Chapra “Métodos Numéricos para ingenieros” Séptima edición.

1. [↑](#footnote-ref-2)