

SOFTWARE EDUCATIVO PARA LA AUTOMATIZACIÓN DEL MÉTODO PUSHOVER EN MATLAB.

EDUCATIONAL SOFTWARE FOR THE AUTOMATION OF THE PUSHOVER METHOD IN MATLAB.

Miguel Ángel Escobar García

miguelescobar@usantotomas.edu.co

Resumen: Este proyecto de investigación nace de la idea de ser un apoyo para los espacios académicos que corresponden al área de estructuras en la facultad de Ingeniería Civil, fortaleciendo las competencias que van adquiriendo los estudiantes y favoreciendo las labores del docente a través de la representación didáctica de los componentes prácticos. Los cálculos estructurales que son necesarios para el análisis del comportamiento de un pórtico estructural siguen una metodología continua, lo que supone una ventaja teórica que este software aprovecha de manera puntual suministrando numérica y gráficamente los resultados obtenidos. El software educativo habilita la posibilidad de estructurar gráficamente el comportamiento de capacidad sísmica de un pórtico simple en 2d de hasta tres niveles describiendo sus desplazamientos máximos en los tres momentos importantes de análisis como lo son el previo a la fisura, justo después de la fisura y en el momento de la rotura.

Palabras clave: Pórtico simple, análisis, pushover, MatLAB, cortante basal, desplazamientos, ductilidad y rotura.

Abstract: This research project was born from the idea of being a support for the academic spaces that correspond to the area of structures in the Faculty of Civil Engineering, strengthening the skills that students are acquiring and favoring the work of the teacher through the didactic representation of the practical components. The structural calculations that are necessary for the analysis of the behavior of a structural frame follow a continuous methodology, which represents a theoretical advantage that this software takes advantage of in a specific way, supplying numerically and graphically the results obtained. The educational software enables the possibility of graphically structuring the seismic capacity behavior of a simple 2d frame of up to three levels, describing its maximum displacements in the three important moments of analysis such as the one before the crack, just after the crack and at the time of breakage.

Keywords: simple frame, analysis, pushover, MatLAB, basal shear, displacements, ductility and rupture.

1. INTRODUCCIÓN

El análisis sismorresistente es de indispensable conocimiento por parte de cualquier ingeniero, así mismo de estar al tanto de las metodologías que estudia y publica la academia con el objetivo de ampliar la gama o crear nuevas soluciones posibles para el estudio del comportamiento de las estructuras en presencia de fuerzas externas, como lo es el caso de un sismo. El diseño de una estructura se resume en la obtención de las secciones y cantidad de refuerzos que van a ser utilizados, pero es el paso que inicia la verdadera ciencia detrás del análisis estructural, disciplina que estudia y determina las capacidades de una estructura dependiendo su diseño para con su entorno y estructura interna. El análisis estructural permite la apreciación de la infraestructura en términos lineales y no lineales. La no linealidad aprovecha el comportamiento de los materiales a través de percepciones distintas a su linealidad geométrica, de esta forma se puede acercar a la realidad omitiendo el principio de superposición [1]. El análisis no lineal pushover es la metodología por la cual se analizan las estructuras en el caso de aparición de una fuerza horizontal equivalente y determinar en tres estados (ductilidad, límite de servicio y colapso) la capacidad propia de la misma, previo al momento de alcanzar el colapso.

2. DESARROLLO DEL ARTÍCULO

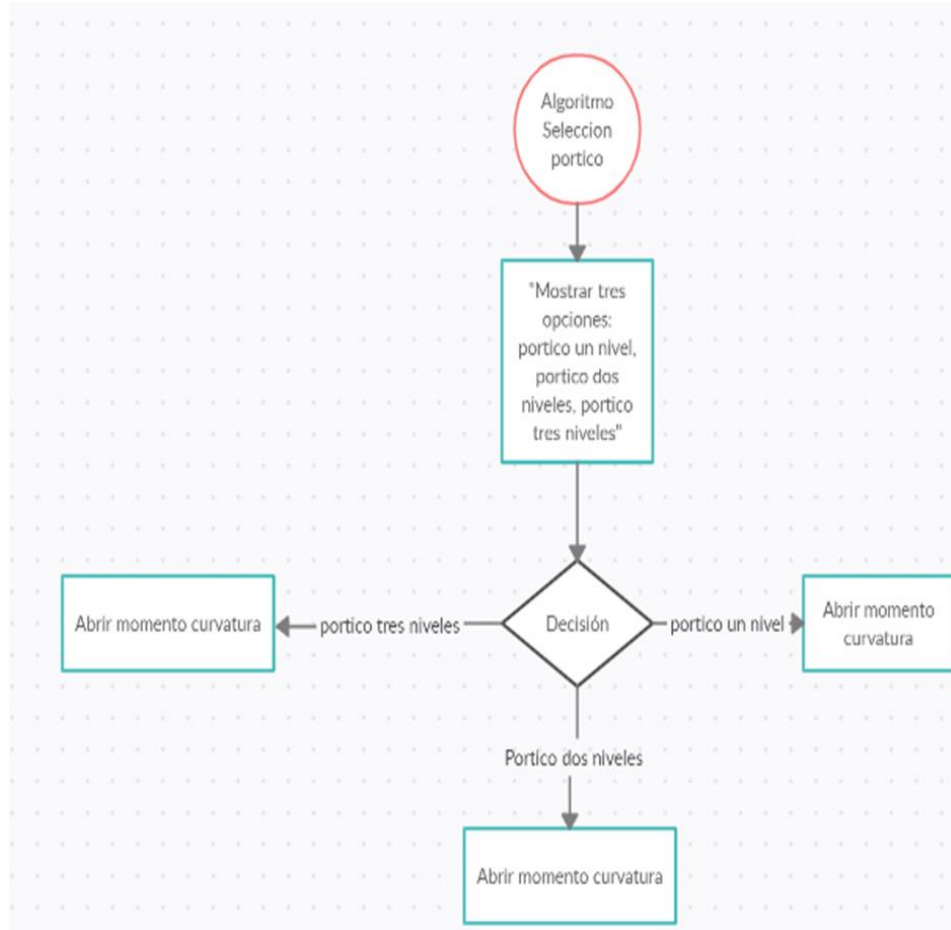
Para la construcción de un software de cualquier índole, desde el lenguaje más básico hasta el más profesional presente en la actualidad deben estructurarse los conocidos algoritmos, que permitan la correcta adaptabilidad de los procesos a través de un lineamiento delineado en un diagrama de flujo en el que se evidencie que la información “viaje” correctamente a través del código interno del software y arroje el resultado que las matemáticas permitan correcto.

En el desarrollo de este software educativo se construyeron algoritmos para cada ventana que se planteó diseñar de manera que la dirección de los datos siguiera un orden jerárquico correcto en el control de la información almacenada y posteriormente computada para la obtención de los resultados gráficos esperados.

2.1. Selección de pórtico

El software educativo se planteó en el desarrollo del anteproyecto como una posibilidad de análisis de pórticos simples en dos dimensiones de hasta tres niveles, por lo que fue necesaria la planeación de un algoritmo que permitiera la lectura por parte del código del pórtico y la cantidad de niveles a calcular.

Fig.1.ALGORITMO SELECCIÓN DE PÓRTICO



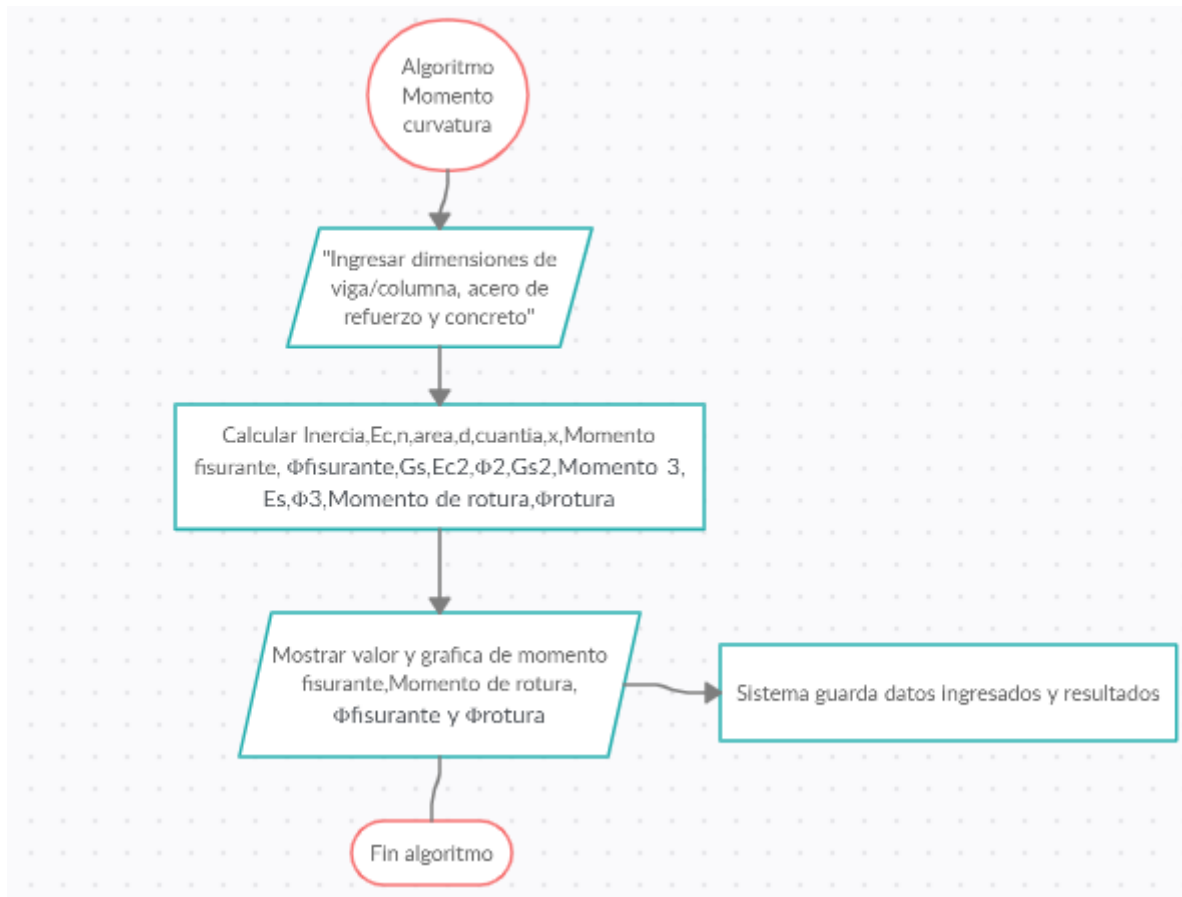
Fuente: Creately

Este diagrama de flujo lo que indica es que el software debe mostrar tres (3) alternativas al usuario con posibilidad de selección de cualquier opción, el código lee la elección a través del Click del usuario y le abre la ventana "Momento curvatura" para continuar con el manejo de la información.

2.2. Calculo momento curvatura

Después de la lectura del pórtico que desea analizar el usuario, el código tiene como orden la apertura de una ventana independiente correspondiente al cálculo del momento curvatura que es la fase uno de dos correspondiente al análisis pushover para la obtención de la curva capacidad del pórtico.

Fig.2. ALGORITMO MOMENTO-CURVATURA



Fuente: Creately

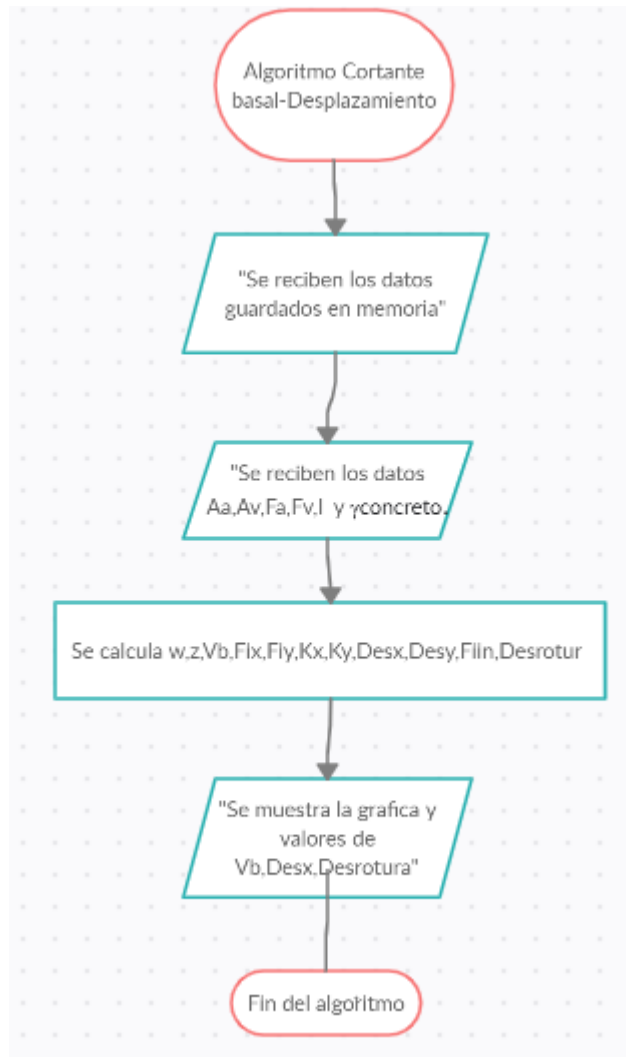
A través de este algoritmo lo que se le ordena al código la lectura de los datos pertenecientes a las dimensiones de las vigas y columnas, que como se expresó con anterioridad así sea el pórtico de tres niveles las dimensiones de las vigas y columnas serán las mismas, de esta manera acompañando a las dimensiones se leerán las características del concreto y acero de refuerzo que requiera el usuario.

Con estos datos el código deberá calcular las cifras correspondientes a el momento de fisura, Φ_{fisura} , momento de rotura y Φ_{rotura} . Para con estos datos realizar la representación gráfica del comportamiento de momento-curvatura describiendo así la fase uno de dos del análisis no lineal pushover que el usuario requiere.

2.3. Calculo Curva Capacidad

Para la obtención de la curva capacidad es necesaria, según la teoría presentada en el marco teórico, la obtención de las cifras correspondientes a los cortante basal y desplazamientos máximos de cada nivel de la estructura. Es por esto que se habilita una nueva ventana al usuario en el que se desglosa el siguiente algoritmo:

Fig.3. ALGORITMO CURVA CAPACIDAD



Fuente: Creately

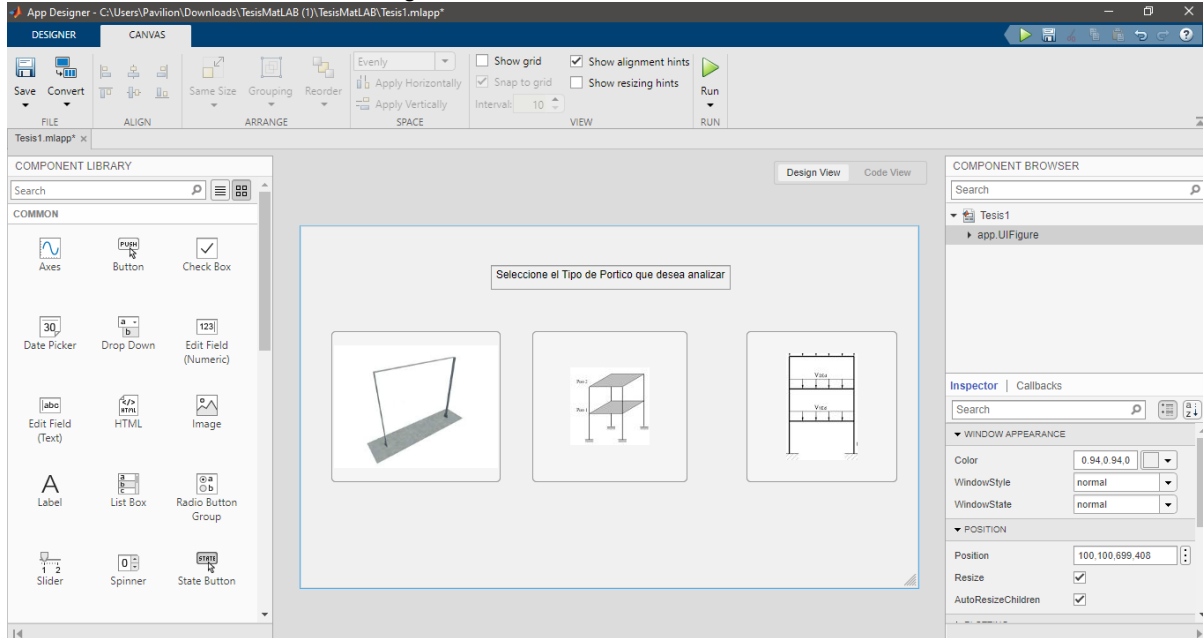
Con la delineación de este algoritmo se describe que el código almacena de la ventana anterior los datos necesarios para el cálculo de las variables de cortante basal y desplazamientos máximos como lo son las dimensiones de las vigas y momento de fisura, Φ_{fisura} , momento de rotura y Φ_{rotura} . Con esto se leen las inserciones del usuario, datos pertenecientes a la sismicidad del elemento, para realizar los cálculos correspondientes y exponer los valores de cortante basal, desplazamiento permisible y desplazamiento de ruptura. Estos valores describen su comportamiento a través de una gráfica en la que el usuario recibe la curva capacidad de su estructura, completando así el análisis no lineal pushover.

3 RESULTADOS

3.1 Selección de pórtico

Para la selección del pórtico se creó una ventana independiente para la selección del pórtico a gusto del usuario, con tres botones a los que se les programó una intercomunicación de ventanas para acceder al siguiente paso, cálculo de momento-curvatura, la interfaz se diseñó para ser amigable y clara para el usuario.

Fig. 4. INTERFAZ SELECCIÓN DE PÓRTICO.



Fuente: autores

1. Si se selecciona el pórtico de un nivel:

```
% Button pushed function: Button
function ButtonPushed2(app, event)
    Tesis3
end
```
2. Si se selecciona el pórtico de dos niveles:

```
% Button pushed function: Button_2
function Button_2Pushed (app, event)
    Tesis2niveles
end
```
3. Si se selecciona el pórtico de tres niveles:

```
% Button pushed function: Button_3
function Button_3Pushed (app, event)
    Tesis3niveles
end
```

3.2 Calculo momento-curvatura

Para el cálculo del momento-curvatura se creó una interfaz gráfica de una ventana con la opción scrollable del menú interactivo activada para que al momento de correr la aplicación debido a su magnitud pueda ser desplazado por el usuario para la obtención de gráficas y resultados.

FIG. 1 INTERFAZ CALCULO MOMENTO CURVATURA

Calculo del momento curvatura de elementos estructurales

Inserte los siguientes datos de su viga

Base Viga (m) Numero de varillas Viga esu viga
 Altura Viga (m) Numero de varilla viga Ec (KN/m²) Viga
 Longitud Viga (m) fc concreto viga (KN/m²)
 Fct (KN/m²) Viga Fy (KN/m²) Viga
 Recubrimiento (m) viga Es (KN/m²) Viga

Inserte los siguientes datos de su columna

Base Columna (m) Numero de varillas columna Recubrimiento (m) Columna
 Altura Columna (m) Numero de varilla columna esu columna
 Longitud Columna (m) fc concreto columna (KN/m²)
 Ec (KN/m²) Columna Fy (KN/m²) Columna
 Es (KN/m²) Columna Fct (KN/m²) Columna

A continuacion se calcularan las graficas de momento curvatura de sus vigas y columnas que cumplan los datos previamente consignados:

Momento fisorante
☐ fisorante
☐ rotura
 Momento Rotura

Momento fisorante
☐ fisorante
☐ rotura
 Momento Rotura

Calcular

Gráfico de Momento Curvatura

Volver a fase 2

Fuente: autores

%1. Asignar Variables

```
Bviga = app.BaseVigam.Value;
Hviga = app.AlturaViga.Value;
Lviga = app.LongitudViga.Value;
NoVarillas = app.NumerodevarillasViga.Value;
fcviga = app.fcconcretovigaKNm2.Value;
Ecvigam = app.EcKNm2Viga.Value;
fyvigam = app.FyKNm2Viga.Value;
fctvigam = app.FctKNm2Viga.Value;
Esviga = app.EsKNm2Viga.Value;
recubrimientoviga = app.Recubrimientomviga.Value;
numeroavarilla = app.Numerodevarillaviga.Value;
Bcolumna = app.BaseColumna.Value;
Hcolumna = app.AlturaColumna.Value;
Lcolumna = app.LongitudColumna.Value;
NoVarillas = app.Numerodevarillascolumna.Value;
fccolumna = app.fcconcretocolumnaKNm2.Value;
Eccolumna = app.EcKNm2Columna.Value;
fycolumna = app.FyKNm2Columna.Value;
fctcolumna = app.FctKNm2Columna.Value;
Escolumna = app.EsKNm2Columna.Value;
recubrimientocolumna = app.RecubrimientomColumna.Value;
numeroavarillac = app.Numerodevarillacolumna.Value;
esuviga = app.esuviga.Value;
esucolumna = app.esucolumna.Value;
```

%2 Definir valores para Iviga y Ec1

```
Iviga = ((Bviga*(Hviga^3))/12);
Ec1 = fctvigam/Ecvigam;
```

%3 mostrar cálculo de Momento curvatura viga fisura

```
app.Momentofisurante.Value = (fctvigam * Iviga)/(Hviga/2);
app.Ofisurante.Value = Ec1/(Hviga/2);
```

%4 Definir valores de , Ec2, x, M1, M2, M3, M31,M32, M4, Fi1, Fi2, Fi31, Fi32, Fi4, pviga, areaviga, dviga, Gsviga, nviga,Gs2viga,x2,Esviga1

```

nviga = Esviga / Ec2viga;
areaviga = NoVarillas * pi * (numerovarilla * 0.0254/16)^2;
dviga = Hviga - recubrimientoviga;
pviga = areaviga/(Bviga * dviga);
x = dviga * nviga * pviga * (-1+ sqrt((1+(2/(nviga*pviga)))));
M1 = app.Momentofisurante.Value;
Fi1 = app.Ofisurante.Value;
Gsviga = (2*M1)/(Bviga*x*(dviga-(x/3)));
Ec2 = Gsviga/Ecviga;
Fi2 = Ec2/x;
Gs2viga = (2*areaviga*fyviga)/(Bviga*x);
M31 = (1/2)*Gs2viga*Bviga*x*(dviga-(x/3));
Esviga1 = fyviga/Esviga;
Fi31 = Esviga1/(dviga-x);
x2 = (areaviga*fyviga)/(0.8*fcviga*Bviga);
M32 = x2*0.8*fcviga*Bviga*(dviga-(0.8*x2/2));
Fi32 = Esviga1/(dviga-x2);
M3 = max(M31,M32);
M4 = M3;
Fi4 = esuviga/(dviga-x2);
app.MomentoRotura.Value = M4;
app.Orotura.Value = Fi4;

```

%5 Definir valores para Icolumna y Ec1columna

```

Icolumna = ((Bcolumna*(Hcolumna^3))/12);
Ec1columna = fctcolumna/Eccolumna;

```

%6 mostrar cálculo de Momento curvatura columna fisura

```

app.Momentofisurantecolumna.Value = (fctcolumna * Icolumna)/(Hcolumna/2);
app.Ofisurantecolumna.Value = Ec1columna/(Hcolumna/2);

```

%7 Definir valores de , Ec2, xcolumna, M1columna, M2columna, M3columna, M31columna, M32columna, M4columna, Fi1columna, Fi2columna, Fi31columna, Fi32columna, Fi4columna, pcolumna, areacolumna, dcolumna, Gscolumna, ncolumna,Gs2columna,x2,Escolumna1,z,I

```

ncolumna = Escolumna / Eccolumna;
areacolumna = NoVarillasc * pi * (numerovarillac * 0.0254/16)^2;
dcolumna = Hcolumna - recubrimientocolumna;
pcolumna = areacolumna/(Bcolumna * dcolumna);
xcolumna = dcolumna * ncolumna * pcolumna * (-1+ sqrt((1+(2/(ncolumna*pcolumna)))));
M1columna = app.Momentofisurantecolumna.Value;
Fi1columna = app.Ofisurantecolumna.Value;
Gscolumna = (2*M1columna)/(Bcolumna*xcolumna*(dcolumna-(xcolumna/3)));
Ec2columna = Gscolumna/Eccolumna;
Fi2columna = Ec2columna/xcolumna;
Gs2columna = (2*areacolumna*fycolumna)/(Bcolumna*xcolumna);
M31columna = (1/2)*Gs2columna*Bcolumna*xcolumna*(dcolumna-(x/3));
Escolumna1 = fycolumna/Escolumna;
Fi31columna = Escolumna1/(dcolumna-xcolumna);
x2columna = (areacolumna*fycolumna)/(0.8*fccolumna*Bcolumna);
M32columna = x2columna*0.8*fccolumna*Bcolumna*(dcolumna-(0.8*x2columna/2));
Fi32columna = Escolumna1/(dcolumna-x2columna);
M3columna = max(M31columna,M32columna);
M4columna = M3columna;
Fi4columna = esucolumna/(dcolumna-x2columna);
z = Fi4 + 0.0002;
I = M4+10
app.MomentoRoturacolumna.Value = M4columna;

```

```

    app.Oroturacolumna.Value = Fi4columna;
%8 Calcular Elviga, Elcolumna
cla(app.eje,'reset');
datosx = [0,Fi1,Fi32,Fi4];
datosy = [0,M1,M3,M4];
plot(app.eje,datosx,datosy,z,l,'-m');
end
% Button pushed function: Irafase2Button
function Irafase2ButtonPushed(app, event)
    Bviga = app.BaseVigam.Value;
    Hviga = app.AlturaViga.Value;
    Lviga = app.LongitudViga.Value;
    Bcolumna = app.BaseColumna.Value;
    Hcolumna = app.AlturaColumna.Value;
    Lcolumna = app.LongitudColumna.Value;
    Fi1 = app.Ofisurante.Value;
    Fi2 = app.Orotura.Value;
    Fc = app.fcconcretocolumnaKNm2.Value;
    Ec = app.EcKNm2Columna.Value;
    Tesis2(Bviga,Hviga,Lviga,Bcolumna,Hcolumna,Lcolumna,Fi1,Fi2,Fc,Ec);
End

```

3.3 Cálculo cortante basal-desplazamiento

Para el cálculo de la gráfica del cortante basal-desplazamiento nuevamente se construyó una interfaz con la opción scrollable activada en los casos de pórticos de dos niveles y tres niveles debido a su extensión. Para todos los casos se muestran los datos que vienen de la fase de momento-curvatura en la parte superior de la pantalla de manera que se agilice el proceso de cálculo y se evite que por causa de las limitaciones del software el usuario no tenga que replicar datos.

FIG. 2 INTERFAZ CORTANTE BASAL-DESPLAZAMIENTO PÓRTICO UN NIVEL.

Fuente: autores

FIG. 3 INTERFAZ CALCULO CORTANTE BASAL-DESPLAZAMIENTO DOS NIVELES.

Datos iniciales

Base Columna Altura Columna Longitud Columna $\phi 1$ F_c
 Base Viga Altura viga Longitud Viga $\phi 2$ E_c

Inserte los siguientes datos (procedentes de su espectro sísmico de diseño)

A_s A_v F_a F_v I
 $\gamma_{overstress}$ (K(N/m³))

Calculos primer nivel

Peso del primer nivel (Kn)
 Cortante basal (Kn)
 F_x (Kn)
 Desplazamiento en X
 Desplazamiento Reducido
 Desplazamiento Brutosa

Calculos segundo nivel

Peso del segundo nivel (Kn)
 Cortante basal (Kn)
 F_x (Kn)
 Desplazamiento en X
 Desplazamiento Reducido
 Desplazamiento Brutosa

Calculos estructurales

S_a Periodo aproximado K Cortante basal (Kn)
 Peso total de la estructura (Kn)

Plot

Calcular

Fuente: autores

FIG. 4 INTERFAZ CALCULO CORTANTE BASAL-DESPLAZAMIENTO TRES NIVELES.

Datos iniciales

Base Columna Altura Columna Longitud Columna $\phi 1$ F_c
 Base Viga Altura viga Longitud Viga $\phi 2$ E_c

Inserte los siguientes datos (procedentes de su espectro sísmico de diseño)

A_s A_v F_a F_v I
 $\gamma_{overstress}$ (K(N/m³))

Calculos primer nivel

Peso del primer nivel (Kn)
 Cortante basal (Kn)
 F_x (Kn)
 Desplazamiento en X
 Desplazamiento Reducido
 Desplazamiento Brutosa

Calculos segundo nivel

Peso del segundo nivel (Kn)
 Cortante basal (Kn)
 F_x (Kn)
 Desplazamiento en X
 Desplazamiento Reducido
 Desplazamiento Brutosa

Calculos tercer nivel

Peso del tercer nivel (Kn)
 Cortante basal (Kn)
 F_x (Kn)
 Desplazamiento en X
 Desplazamiento Reducido
 Desplazamiento Brutosa

Calculos estructurales

S_a Periodo aproximado K Cortante basal (Kn)
 Peso total de la estructura (Kn)

Plot

Calcular

Fuente: autores

% Code that executes after component creation

```
function startupFcn(app, Bviga, Hviga, Lviga, Bcolumna, Hcolumna, Lcolumna, Fi1, Fi2, Fc, Ec)
    app.BaseColumna.Value = Bcolumna;
    app.BaseViga.Value = Bviga;
    app.Alturaviga.Value = Hviga;
    app.LongitudViga.Value = Lviga;
    app.AlturaColumna.Value = Hcolumna;
    app.LongitudColumna.Value = Lcolumna;
    app.EditField.Value = Fi1;
    app.EditField_2.Value = Fi2;
    app.Fc.Value = Fc;
```

```

app.Ec.Value = Ec;
end
% Button pushed function: CalcularButton
function CalcularButtonPushed(app, event)
    Bcolumna = app.BaseColumna.Value;
    Bviga = app.BaseViga.Value;
    Hviga = app.Alturaviga.Value;
    Lviga = app.LongitudViga.Value;
    Hcolumna = app.AlturaColumna.Value;
    Lcolumna = app.LongitudColumna.Value;
    Fi1 = app.EditField.Value;
    Fi2 = app.EditField_2.Value;
    Fc = app.Fc.Value;
    Ec = app.Ec.Value;
    Aa = app.Aa.Value;
    Av = app.Av.Value;
    Fa = app.Fa.Value;
    Fv = app.Fv.Value;
    I = app.I.Value;
    GamaConcreto = app.concretoKNm3.Value;
    Desplazamientoductil = app.Desplazamientoductil.Value;
    Desplazamientorotura = app.Desplazamientorotura.Value;
%1 Nombrar Variables lxx, lyy, Sa, Ta, K, w, z,vb,Fix,Fiy,Kx,Ky,Desx,Desy,lxx2, lyy2, Sa2, Ta2, K2, w2,
z2,vb2,Fix2,Fiy2,Kx2,Ky2,Desx2,Desy2
    lxx = Bcolumna * ((Hcolumna^3)/12;
    lyy = (Bcolumna^3 * Hcolumna)/12;
    w = 2*(GamaConcreto * Bcolumna * Hcolumna * (Lcolumna)) + (GamaConcreto * Bviga * Hviga * Lviga);
    Kx = 2 * (12 * Ec * lyy)/(Lcolumna^3);
    Ky = 2 * (12 * Ec * lxx)/(Lcolumna^3);
    Fiin = Fi2 - Fi1;
    Desrotura = Fiin * Lcolumna;
    lxx2 = Bcolumna * ((Hcolumna^3)/12;
    lyy2 = (Bcolumna^3 * Hcolumna)/12;
    Sa2 = Aa * Fa * I * 2.5;
    Ta2 = ((2*Lcolumna)^0.9) * 0.047;
    if Ta2 <= 0.5
        K2 = 1;
    elseif Ta > 2.5
        K2 = 2
    else
        K2 = 0.75 + 0.5*Ta
    end
    w2 = 2*(GamaConcreto * Bcolumna * Hcolumna * (Lcolumna/2)) + (GamaConcreto * Bviga * Hviga *
    Lviga);
    WT = w + w2;
    WHK = w * (Lcolumna^K2);
    WHK2 = w2 * (Lcolumna^K2);
    WHKT = WHK + WHK2;
    CVI1 = WHK/WHKT;
    CVI2 = WHK2/WHKT;
    VBT = WT * Sa2;
    vb1 = VBT * CVI1;
    Vb2 = VBT * CVI2;
    Fix = vb1;
    Fix2 = Vb2;
    fix3 = Fix + Fix2

```

```

Fiy = vb1;
Fiy2 = Vb2;
F = [Fix;Fix2];
Kx2 = 2 * (12 * Ec * Iyy2)/(Lcolumna^3);
Ky2 = 2 * (12 * Ec * Ixx2)/(Lcolumna^3);
A = [(Kx2+Kx) (-Kx2); (-Kx2) (Kx2)];
B = inv(A);
D = B * F;
Desx = D(1,1);
Desx2 = D(2,1);
Fiin2 = Fi2 - Fi1;
Desrotura2 = Fiin * (2*Lcolumna);
%4 Mostrar Calculos
app.PesoprimernivelKn.Value = w;
app.Cortantebasal1Kn.Value = vb1;
app.FxKn.Value = Fix;
app.DesplazamientoenX.Value = Desx;
app.Desplazamientorotura.Value = Desrotura;
app.PesodelaestructuraKn_2.Value = w2;
app.CortantebasalKn_2.Value = Vb2;
app.FxKn_2.Value = Fix2;
app.DesplazamientoenX_2.Value = Desx2;
app.Desplazamientorotura_2.Value = Desrotura2;
app.Sa_2.Value = Sa2;
app.Períodoaproximado_2.Value = Ta2;
app.K_2.Value = Kx2;
app.Cortantebasal.Value = VBT;
app.Pesodelaestructuratotal.Value = WT;
app.Desplazamientoductil_2.Value = Desx2;
app.Desplazamientoductil.Value = Desx;
wl1 = Desrotura2 +0.001;
xl1 = fix3 + 10;
%5 Graficar
cl1 = [0,Desx,Desx2,Desrotura2];
dl1 = [0,Fix,fix3,fix3];
plot(app.UIAxes,cl1,dl1,wl1,xl1,'r')
end
Código De Programación Tres Niveles
% Code that executes after component creation
function startupFcn(app, Bviga, Hviga, Lviga, Bcolumna, Hcolumna, Lcolumna, Fi1, Fi2, Fc, Ec)
    app.BaseColumna.Value = Bcolumna;
    app.BaseViga.Value = Bviga;
    app.Alturaviga.Value = Hviga;
    app.LongitudViga.Value = Lviga;
    app.AlturaColumna.Value = Hcolumna;
    app.LongitudColumna.Value = Lcolumna;
    app.EditField.Value = Fi1;
    app.EditField_2.Value = Fi2;
    app.Fc.Value = Fc;
    app.Ec.Value = Ec;
end
% Button pushed function: CalcularButton
function CalcularButtonPushed(app, event)
    Bcolumna = app.BaseColumna.Value;
    Bviga = app.BaseViga.Value;
    Hviga = app.Alturaviga.Value;

```

```

Lviga = app.LongitudViga.Value;
Hcolumna = app.AlturaColumna.Value;
Lcolumna = app.LongitudColumna.Value;
Fi1 = app.EditField.Value;
Fi2 = app.EditField_2.Value;
Fc = app.Fc.Value;
Ec = app.Ec.Value;
Aa = app.Aa.Value;
Av = app.Av.Value;
Fa = app.Fa.Value;
Fv = app.Fv.Value;
I = app.I.Value;
GamaConcreto = app.concretoKNm3.Value;
Desplazamientoductil = app.Desplazamientoductil.Value;
Desplazamientorotura = app.Desplazamientorotura.Value;
%1 Nombrar Variables lxx, lyy, Sa, Ta, K, w,z,vb,Fix,Fiy,Kx,Ky,Desx,Desy,lxx2, lyy2, Sa2, Ta2, K2, w2,
z2,vb2,Fix2,Fiy2,Kx2,Ky2,Desx2,Desy2
lxx = Bcolumna * ((Hcolumna^3)/12;
lyy = (Bcolumna^3 * Hcolumna)/12;
w = 2*(GamaConcreto * Bcolumna * Hcolumna * (Lcolumna)) + (GamaConcreto * Bviga * Hviga * Lviga);
Kx = 2 * (12 * Ec * lyy)/(Lcolumna^3);
Ky = 2 * (12 * Ec * lxx)/(Lcolumna^3);
Fiin = Fi2 - Fi1;
Desrotura = Fiin * Lcolumna
lxx2 = Bcolumna * ((Hcolumna^3)/12;
lyy2 = (Bcolumna^3 * Hcolumna)/12;
Sa2 = Aa * Fa * I * 2.5;
Ta2 = ((2*Lcolumna)^0.9) * 0.047;
if Ta2 <= 0.5
K2 = 1;
elseif Ta > 2.5
K2 = 2
else
K2 = 0.75 + 0.5*Ta
end
w2 = 2*(GamaConcreto * Bcolumna * Hcolumna * (Lcolumna)) + (GamaConcreto * Bviga * Hviga *
Lviga);
w3 = 2*(GamaConcreto * Bcolumna * Hcolumna * (Lcolumna/2)) + (GamaConcreto * Bviga * Hviga *
Lviga)
WT = w + w2 + w3;
WHK = w * (Lcolumna^K2);
WHK2 = w2 * (Lcolumna^K2);
WHK3 = w3 * (Lcolumna^K2);
WHKT = WHK + WHK2 + WHK3;
CVI1 = WHK/WHKT;
CVI2 = WHK2/WHKT;
CVI3 = WHK3/WHKT;
VBT = WT * Sa2;
vb1 = VBT * CVI1;
Vb2 = VBT * CVI2;
Vb3 = VBT * CVI3;
Fix = vb1;
Fix2 = Vb2;
Fix3 = Vb3;
fix4 = Fix + Fix2 + Fix3;
Fiy = vb1;

```

```

Fiy2 = Vb2;
F = [Fix;Fix2;Fix3];
Kx2 = 2 * (12 * Ec * Iyy2)/(Lcolumna^3);
Kx3 = 2 * (12 * Ec * Iyy2)/(Lcolumna^3);
Ky2 = 2 * (12 * Ec * Ixx2)/(Lcolumna^3);
A = [(Kx2+Kx) (-Kx2) (0); (-Kx2) (Kx2+Kx3) (-Kx3); (0) (-Kx3) (Kx3)];
B = inv(A);
D = B * F;
Desx = D(1,1);
Desx2 = D(2,1);
Desx3 = D(3,1);
Fiin2 = Fi2 - Fi1;
Desrotura2 = Fiin * (2*Lcolumna);
Desrotura3 = Fiin * (3*Lcolumna);
%4 Mostrar Calculos
app.PesoprimernivelKn.Value = w;
app.Cortantebasal1Kn.Value = vb1;
app.FxKn.Value = Fix;
app.DesplazamientoenX.Value = Desx;
app.Desplazamientorotura.Value = Desrotura;
app.PesodelaestructuraKn_2.Value = w2;
app.CortantebasalKn_2.Value = Vb2;
app.FxKn_2.Value = Fix2;
app.DesplazamientoenX_2.Value = Desx2;
app.Desplazamientorotura_2.Value = Desrotura2;
app.Sa_2.Value = Sa2;
app.Periodoaproximado_2.Value = Ta2;
app.K_2.Value = Kx2;
app.Cortantebasal.Value = VBT;
app.Pesodelaestructuratotal.Value = WT;
app.Desplazamientoductil_2.Value = Desx2;
app.Desplazamientoductil.Value = Desx3;
app.PesodelaestructuraKn_3.Value = w3;
app.DesplazamientoenX_3.Value = Desx3;
app.Desplazamientoductil_3.Value = Desx3;
app.CortantebasalKn_3.Value = Vb3;
app.DesplazamientoenX_3.Value = Desx3;
app.FxKn_3.Value = Fix3;
wl1 = Desrotura2 +0.001;
yl1 = fix4 + 10;
%3 Graficar
DI1 = [0,Desx,Desx2,Desx3,Desrotura3];
ql1 = [0,Fix,(Fix+Fix2),fix4,fix4];
plot(app.UIAxes,dl1,ql1, wl1,yl1,'r')
end

```

4. CONCLUSIONES

Este trabajo de grado, proyecto de investigación, funciona como una base teórica suficiente para comprender conceptos reales sobre el comportamiento y análisis de estructuras en concreto armado debido a que en su estructura interna funcionan elementos y parámetros pertenecientes al análisis sísmico, comportamientos a tensión, deformaciones, desplazamientos y rigidez.

El software didáctico contribuye en el ámbito educativo a que el estudiante goce de herramientas que acerquen sus conocimientos a nuevos conceptos, en este caso el análisis no lineal, y motivar al continuo aprendizaje practico en el que los conceptos teóricos se describan en una situación real y expongan las causales por las que funcionan y las que lo afectan en un entorno natural, como la sismicidad.

De igual manera la creación y desarrollo de un proyecto de investigación de este calibre sirve como ejemplo

de que el aprendizaje es continuo y desde las etapas finales de la academia se pueden pensar en suplir necesidades que se presentan en los inicios de la misma que siguen ahí por la falta de interés de muchos próximos a egresados de realizar el esfuerzo de proveer un producto educativo importante y complejo y fuera de lo común, como se observa anualmente con el continuo interés de los mismos en probar aditivos en el concreto que a día de hoy se han probado cientos de veces.

Las posibilidades que ofrece al estudiante este aplicativo, población objetivo, son infinitas combinaciones de estructuras bidimensionales en el que puede dar rienda suelta a su imaginación cuya guía sea la curiosidad de reconocer el impacto de la sismicidad en una estructura de concreto armado con variables reales que nos ofrecen los espectros sísmicos cuya base conceptual y practica son establecidas por la Norma Colombiana Sismo resistente.

AGRADECIMIENTOS

Agradecer primero que todo a Dios por permitirme lograr estos acontecimientos que son resultado de mucho esfuerzo y trabajo durante mucho tiempo. Agradecer a mi familia por ser siempre el apoyo en los momentos fáciles y difíciles de todo este proceso, a mi director de grado por su conocimiento, paciencia y cooperación en la elaboración de este documento y a todos los docentes involucrados directa e indirectamente en el transcurso de mi carrera.

REFERENCIAS

- [1] R. A. M. Tapia, Sensibilidad y seguridad de estructuras de hormigon en regimen no lineal, Barcelona: Escola Tecnica Superior d'Enginyers de Camins, 2012.
- [2] Comunidad para la Ingeniería Civil, «Pushover-Analisis no lineal estatico en edificios de concreto armado,» CIC, Culiacan, 2015.
- [3] Y. H. S. Barco, Desempeño sísmico por el metodo Pushover de un edificio construido en la ciudad de Barranquilla bajo la norma NSR-10, Barranquilla: Universidad de la Costa, 2017.
- [4] C. Duarte, M. Marlon y J. Santamaria, Analisis estatico no lineal (Pushover) del cuerpo central del edificio de la facultad de medicina de la Universidad de El Slavador, Ciudad Universitaria: Universidad de El Salvador, 2017.
- [5] A. Z. Jorquera, Modelado y analisis Pushover de la respuesta sísmica de estructuras de acero, Cartagena: Universidad Politecnica de Cartagena, 2017.
- [6] J. D. V. Vasconez, Analisis del metodo modal pushover y su incidencia en el calculo de porticos de hormigon armado en el Canton de Ambato, Ambato: Universidad Tecnica de Ambato, 2012.