RINCON RUIZ

FIC/UNACH

PROGRAMACION

2ª

Proyecto individual

Analisis de cargas en puentes peatonales colgantes

**Clasificacion de la estructura de un puente peatonal colgante**

Un puente peatonal colgante, es de forma muy similar a los grandes puentes colgantes vehiculares, su estructura se puede clasificar de la siguiente manera: la Subestructura que invluye la cimentación y la estructura de soporte: Superestructura que comprende al piso, baranda, cableado y anclaje.

**Elementos que integran la Subestructura**

La subestructura es la parte del puente colgante que se localiza por debajo del suelo y del nivel de apoyo de la superestructura.

**Cimentacion**

Normalmente esta constituida por una losa de concreto reforzado que sirve de apoyo a las dos columnas o pilares principales de soporte, o bien son cimientos

Individuales. Siempre deben estar asentadas sobre la roca, o tierra firme pero libre del alcance del nivel del agua aun en las máximas crecidas.

La cimentación de los puentes peatonales colgantes al igual que el anclaje de los cables principales, se diseña para lograr una mayor seguridad a los peatones, adecuada contra levantamiento, ladeamiento y delizamiento bajo cualquier posible combinación de fuerzas que entren en acción.

**Estructura de soporte (Torrres)**

Estan constituidas por las columnas o pilares de soporte de la superestructura del puente, conformada a base de concreto reforzado, normalmente unidas entre si por varias vigas de concreto reforzado. Constituyen los elementos encargados de transmitir las cargas de la estructura superios a la cimentación correspondiente. En muchas ocaciones se ha observado que las torres están elaboradas a través de armaduras tridimensionales muy parecidas a las utilizadas por la CFE y a las que se emplean para la radiocomunicación.

Las torres son elementos estructurales que brindan la mayor altura a los cables y a la vez sirven para transmitir la carga. Ha ido siempre los elementos mas difíciles de proyectar para todo tipo de puente colgantes, por que son los que permiten una mayor libertar, por eso en ella se ha dado toda la clase de variantes.

**Elementos que integran la superestructura**

La uperestructura, es la parte superior del puente, se encuentra constituida por el piso, la baranda, el cableado y el anclaje.

**Piso o calzada**

El piso generalmente esta constituido como; tablas de madera y laminas metalicas de determinado grosor, apoyadas sobre los cables, sin embardo es recomendablre que el piso de coloque de forma lingitudinal sobre viguetas de madera o soleras, constituyendo tarimas, que deben apoyarse y asegurarse sobre los cables buscando que tengan una mayor estabilidad longitudinal.

Para la dirección tranversal el sistema rigidisante se consigue a través de los cables tensores laterales adicionales, que partiendo de la cuarta parte de la luz del puente, se anclan a ambas orillas del rio.

Dependiendo de la dimensión y uso de los puentes colgante peatonales ecisten diferentes materiales que se pueden emplear para formar su calzada o tableros.

**Baranda**

Esta constituida básicamente por los cables superiores, también uno o dos cales intermediarios y las pendolas de anclaje de todos los cables, incluyendo a los inferiores que sostienen al piso: resulta de vital importancia proteger el paso de los peatones, agregando maya de alambre galvanizado asegurada a los elementos del barandal.

**Cables principales**

El cable es un elemento flexible, lo que quiere decir que no tiene rigidez y por lo tanto no resiste flexiones. Si se aplica un sistema de fuerzas, tomara la forma necesaria para que en el solo se produzcan esfuerzos axiales. Por lo tanto, la forma del cable coincidirá forzosamente con la línea generada por la trayectoria de una de la posible composición del sistema de fuerzas que actúan sobre el.

Esta línea es funicular de un sistema de cargas, que se define precisamente como la forma que toma un hilo flexible cuando se aplica sobre el un sistema de fuerzas. La curva del cable de un puente colgantes es la combinacion de la catenaria, por que el cable principal persa, y la de la parábola, por que también pesa el tablero; sin embargo la diferencia entre ambas curvas es realmente minima, y por ello en el calculo generalmente se ha utilizado la parábola de segundo grado.

El cable principal es un elemento básico de la estructura resistente del puente colgante. Su montaje debe salvar un claro entre las dos torres y para ello hay que tenderlo en el vacio. Esta fase es la mas complicada de la construcción de los puentes colgantes. El esquema clásico de los puentes colgantes admiten pocas variaciones, los grandes se han hecho siempre con un cable principal en cada borde del tablero.

**Tirantes**

Los tirante se pueden organizar de diferentes formas dentro de cada uno de los haces, por que caben diferentes posibilidades: en primer lugar, es necesario definir el numero de tirantes de cada haz, o lo que es lo mismo, la distancia entre los puntos de anclaje de los tirantes en el tablero. El numero de tirantes es una de las cuestiones que mas ha evolucionado para los puentes colgantes y también los atirantados. Los primeros tenían pocos tirantes, con separación entre el anclaje que llego a pasar de los 50m; Se trata de crear una serie de apoyos intermedio para convertir un puente de claros grandes en uno de los claros medios.

En los puentes actuales el numero de tirantes es mucho mayor que en los iniciales, se utilizan las distancias entre anclajes que varian entre cinco y veinte metros, de forma que la flexion que podemos llamar local, la debida a la distancia entre los apoyos generados por los tirantes, es insignificante debida a la flexion que se produce por la deformación general de la estructura.

Si en un principio la finalidad de los tirantes era la de crear una serie de apoyos adicionales al tablero, para transformas un puente de claros grandes en uno de claros medianos, este planteamiento ha evolucionado hasta considerar a los tirantes como un medio de apoyo cuasi-continuo y elástico del tablero.

**Materiales**

De forma general; en la construcción de puentes, gran parte se resuelve mediante el empleo de concreto reforzado; el principal problema que aquí se presenta, es el análisis y dimensamiento de la estructura a realizar, partiendo de la hipótesis de que el análisis es un termino independiente del tipo de material a emplearse.

Los puentes colgantes peatonales se realizan de concreto reforzado para las estructuras de soporte, como es el caso de los marco y sistema de anclajes o muertos de concreto reforzado.

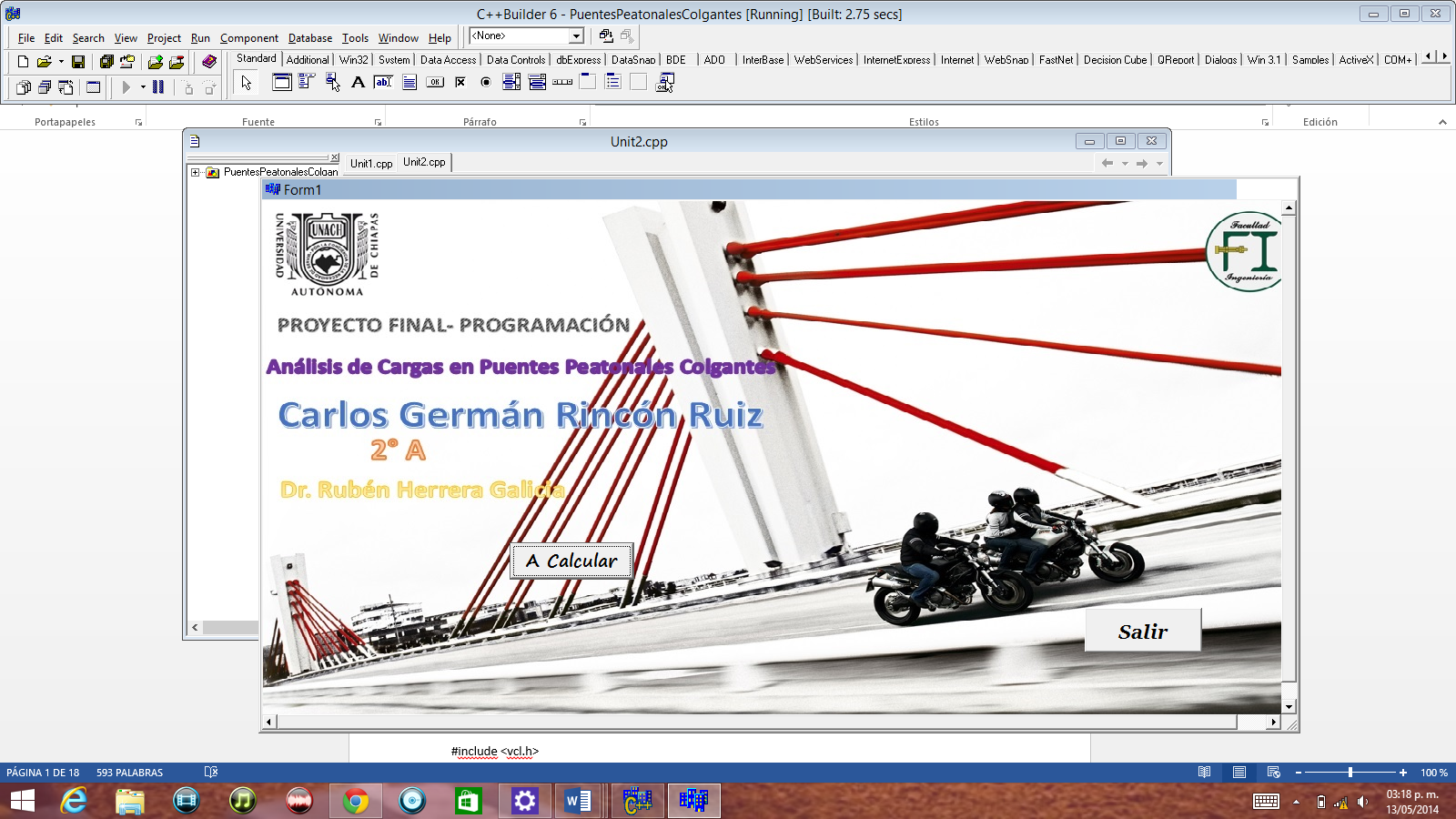
**Diseño del puente peatonal colgante**

Ya determinadas las cargas que actúan sobre el puente peatonal colgante (cargas estaticas y dinamicas), se procede a realizar el calculo de los cables, marcos y del anclaje.

El análisis parte de suposiciones y se demuestra analíticamente a la resistencia de los materiales propuestos, en caso contrario de no ser aptos para el requerimiento del diseño se puede cambiar de acuerdo con las necesidades del calculo o realizar una complementación.Para poder llevar a cabo el desarrollo de los cálculos, se considera una carga uniformemente repartida por unidad de longitud la cual incluye el peso proio de los cables principales, cables secundarios y estructura rigidizantes.

**Uso del programa “Análisis de Cargas en Puentes Peatonales Colgantes”**

Anteriormente se ha descrito el tema Análisis de Cargas en Puentes Peatonales Colgantes, es por eso que se procede a realizar un programa que efectúe las operaciones de este método, para esto creamos un programa en Borland Builder C++. Además que la interfaz gráfica tenga el siguiente aspecto:



1. Cargar Borland Builder C++. Al cargarlo se crea automáticamente un proyecto visual.
2. Guardar el proyecto visual

File/ Save Project As…

1. Entrar a la carpeta designada.
2. Oprimir el botón guardar.
3. Dar nombre al proyecto: “PuentesPeatonalesColgantes”.
4. Oprimir el botón guardar.
5. El siguiente paso será programar el formulario número 1.
6. Declaramos a los dos formularios que usaremos en la parte inicial de los archivos de cabecera:

#include <vcl.h>

#pragma hdrstop

#include "Unit1.h"

#include "Unit2.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm1 \*Form1;

1. Para este formulario se inserta una imagen la cual será el fondo.
   1. Entrar al menú -Aditional-
   2. Se selecciona el complemento imagen.
   3. Insertar la imagen en el formulario.
   4. En la barra de herramientas se selecciona la opción –picture-
   5. Seguido de la opción –load- para seleccionar la imagen deseada.
   6. Para ampliar el tamaño de imagen en la barra de herramientas se selecciona la opción –stretch- y se cambia por la opción true.
2. Poner en el formulario un primer componente tipo button.
   1. Cambiar la leyenda del button1 por texto “A Calcular” para esto ir a la propiedad caption.
   2. Programar al botón1
   3. Seleccionar al botón1
   4. Seleccionar la lista de eventos
   5. En el evento –OnClick hacer doble click en el espacio blanco- Introducir las siguientes líneas de programación:

{

Form2->Show();

}

**Este botón nos llevara al formulario número dos en donde efectuaremos todos los cálculos**

1. Programamos al botón dos “Salir” del mismo modo introduciendo las siguientes líneas de programación:

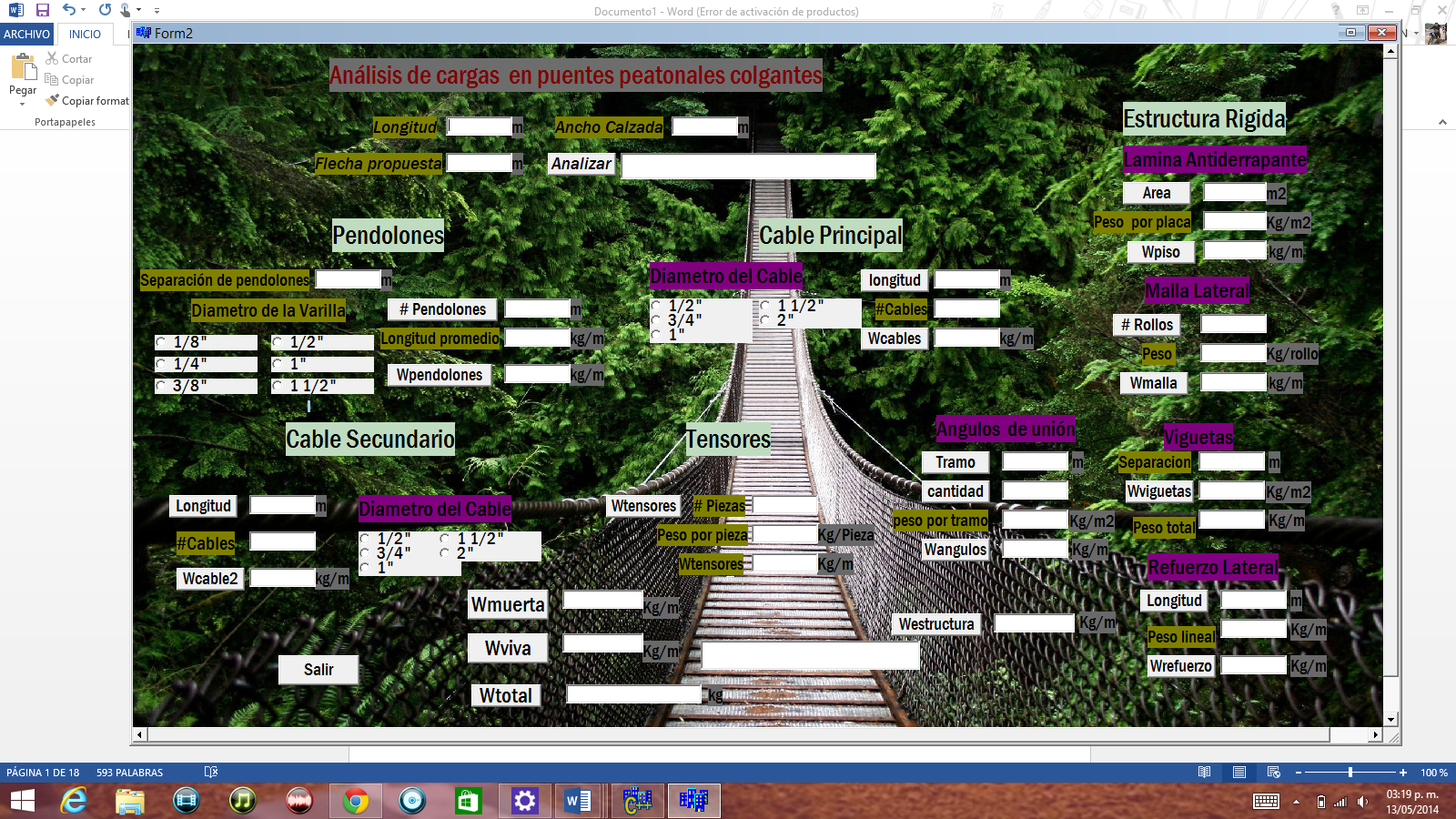
{

Close();

}

**Este botón cerrará el programa.**

1. Insertamos un segundo formulario en donde efectuaremos toda la programación, además que la interfaz gráfica tenga el siguiente aspecto:



1. Poner en el formulario 58 componentes tipo Label
   1. Poner nombre a cada etiqueta:

Label01--------Análisis de cargas en puentes peatonales colgantes

Label02--------Longitud

Label03--------Ancho Calzada

Label04--------Flecha propuesta

Label05--------Pendolones

Label06--------#Cables

Label07--------Cable Principal

Label08--------Separación de Pendolones

Label09--------Diámetro de la Varilla

Label10--------Longitud Promedio

Label11--------m

Label12--------m

Label13--------kg/m

Label14--------kg/m

Label15--------Diámetro del Cable

Label16--------m

Label17--------Kg/m

Label18--------Estructura Rígida

Label19--------Lámina Antiderrapante

Label20--------Peso por placa

Label21--------m2

Label22--------Kg/m2

Label23--------Kg/m

Label24--------Cable Secundario

Label25--------Diámetro del Cable

Label26--------#Cables

Label27--------m

Label28--------Kg/m

Label29--------#Pieza

Label30--------Peso por pieza

Label31--------Wtensores

Label32--------Kg/Pieza

Label33--------Kg/m

Label34--------Ángulos de Unión

Label35--------peso por tramo

Label36--------m

Label37--------Kg/m2

Label38--------Kg/m

Label39--------Malla Lateral

Label40--------Peso

Label41--------Kg/rollo

Label42--------Kg/m

Label43--------Viguetas

Label44--------Separación

Label45--------Peso total

Label46--------m

Label47--------kg/m2

Label48--------kg/m

Label49--------Refuerzo Lateral

Label50--------Peso lineal

Label51--------m

Label52--------kg/m

Label53--------kg/m

Label54--------kg/m

Label55-------- kg/m

Label56-------- kg/m

Label57-------- kg/m

Label58--------Tensores

Para lograr esto seleccionar con el mouse a la etiqueta deseada y en la propiedad caption cambiar el nombre inicial por el nombre deseado.

Cambiar el tipo de fuente en la propiedad –Font-

1. Poner en el formulario 24 componentes tipo button
   1. Nombrar a cada componente tipo button

Button01--------Analizar

Button02--------# Pendolones

Button03--------Wpendolones

Button04--------longitud

Button05--------wcables

Button06--------Longitud

Button07--------Wcable2

Button08--------Wtensores

Button09--------Area

Button10--------Wpiso

Button11--------#Rollo

Button12--------Wmalla

Button13--------Wviguetas

Button14--------Tramo

Button15--------Cantidad

Button16--------Wangulos

Button17--------Longitud

Button18--------Wrefuerzo

Button19--------Westructura

Button20--------Wmuerta

Button21--------Wviva

Button22--------Wtotal

Button23--------Salir

Button24--------Puente

1. Poner en el formulario 11 componentes tipo RadioButton
   1. Poner nombre a cada componente:

RadioButton01-------1/8”

RadioButton02-------1/4”

RadioButton03------- 3/8”

RadioButton04-------½ ”

RadioButton05-------1”

RadioButton06-------1 ½”

RadioButton07------- ½ ”

RadioButton08-------3/4”

RadioButton09-------1”

RadioButton10-------1 1/2”

RadioButton11-------2”

RadioButton12------- 1/2”

RadioButton13-------3/4”

RadioButton14-------1”

RadioButton15-------1 1/2”

RadioButton16-------2”

1. Programamos al botón 1 “Analizar” del mismo modo introduciendo las siguientes líneas de programación:

{

L=Edit1->Text.ToDouble();

AC=Edit2->Text.ToDouble();

Flecha=Edit3->Text.ToDouble();

Nor= (Flecha/L)\*100;

if ( Nor<= 12 && Nor >=8)

{

Edit4->Text="Flecha ideal";

}

if ( Nor <8)

{

Edit4->Text="Longitud de flecha muy pequeña";

}

if (Nor>12)

{

Edit4->Text="Longitud de flecha muy grande";

}

}

1. Programamos al botón 2 “Wpendolones” del mismo modo introduciendo las siguientes líneas de programación:

{

p2= (hp\* PP\*P);

w2=p2/L;

Edit8->Text=AnsiString(w2);

}

1. Programamos al botón 3 “# Pendolones” del mismo modo introduciendo las siguientes líneas de programación:

{

S=Edit5->Text.ToDouble();

P= ((L/2)-s)\*2;

Edit6->Text=AnsiString(P);

h=(4\*Flecha/(L\*L))+1.26 ;

h1=(4\*Flecha/(L\*L))\*(2\*2)+1.26 ;

h2=(4\*Flecha/(L\*L))\*(4\*4)+1.26 ;

h3=(4\*Flecha/(L\*L))\*(6\*6)+1.26 ;

h4=(4\*Flecha/(L\*L))\*(8\*8)+1.26 ;

h5=(4\*Flecha/(L\*L))\*(10\*10)+1.26 ;

h6=(4\*Flecha/(L\*L))\*(12\*12)+1.26 ;

h7=(4\*Flecha/(L\*L))\*(14\*14)+1.26 ;

h8=(4\*Flecha/(L\*L))\*(16\*16)+1.26 ;

h9=(4\*Flecha/(L\*L))\*(18\*18)+1.26 ;

h10=(4\*Flecha/(L\*L))\*(20\*20)+1.26 ;

h11=(4\*Flecha/(L\*L))\*(22\*22)+1.26 ;

h12=(4\*Flecha/(L\*L))\*(24\*24)+1.26 ;

h13=(4\*Flecha/(L\*L))\*(26\*26)+1.26 ;

h14=(4\*Flecha/(L\*L))\*(28\*28)+1.26 ;

h15=(4\*Flecha/(L\*L))\*(30\*30)+1.26 ;

h16=(4\*Flecha/(L\*L))\*(32\*32)+1.26 ;

h17=(4\*Flecha/(L\*L))\*(34\*34)+1.26 ;

h18=(4\*Flecha/(L\*L))\*(36\*36)+1.26 ;

h19=(4\*Flecha/(L\*L))\*(38\*38)+1.26 ;

h20=(4\*Flecha/(L\*L))\*(40\*40)+1.26 ;

h21=(4\*Flecha/(L\*L))\*(42\*42)+1.26 ;

h22=(4\*Flecha/(L\*L))\*(44\*44)+1.26 ;

h23=(4\*Flecha/(L\*L))\*(46\*46)+1.26 ;

h24=(4\*Flecha/(L\*L))\*(48\*48)+1.26;

h0=h1+h2+h3+h4+h5+h6+h7+h8+h9+h10+h11+h12+h13+h14+h15+h16+h17+h18+h19+h20+h21+h22+h23+h24;

hp= h0/24;

Edit7->Text=AnsiString(hp);

}

1. Programamos al botón 4 “Longitud” del mismo modo introduciendo las siguientes líneas de programación:

{

LF= L/Flecha;

LF2=LF\*LF ;

LCP=L\*(1+((3/8)\*LF2));

Edit9->Text=AnsiString(LCP);

}

1. Programamos al botón 5 “Wcables” del mismo modo introduciendo las siguientes líneas de programación:

{

Ncab=Edit10->Text.ToDouble();

Wc= (LCP\*Ncab\*pp1)/L;

Edit11->Text=AnsiString(Wc);

}

1. Insertamos valores determinados al RadioButton1 en la programación:

{

PP= 0.04;

RR= 690;

}

1. Insertamos valores determinados al RadioButton2 en la programación:

{

PP= 0.17;

RR= 2670;

}

1. Insertamos valores determinados al RadioButton3 en la programación:

{

PP= 0.39;

RR= 5950;

}

1. Insertamos valores determinados al RadioButton4 en la programación:

{

PP= 0.68;

RR= 10400;

}

1. Insertamos valores determinados al RadioButton5 en la programación:

{

PP= 2.75;

RR= 40700;

}

1. Insertamos valores determinados al RadioButton6 en la programación:

{

PP= 6.19;

RR= 89700;

}

1. Insertamos valores determinados al RadioButton7 en la programación:

{

pp1= 0.68;

rr1= 10400;

}

1. Insertamos valores determinados al RadioButton8 en la programación:

{

pp1= 1.55;

rr1= 23200;

}

1. Insertamos valores determinados al RadioButton9 en la programación:

{

pp1= 2.75;

rr1= 40700;

}

1. Insertamos valores determinados al RadioButton10 en la programación:

{

pp1= 6.19;

rr1= 89700;

}

1. Insertamos valores determinados al RadioButton11 en la programación:

{

pp1= 11;

rr1= 156000;

}

1. Programamos al botón 6 “Longitud” del mismo modo introduciendo las siguientes líneas de programación:

{

L2=(6\*L)/5;

Edit12->Text=AnsiString(L2);

}

1. Programamos al botón 7 “Wcable2” del mismo modo introduciendo las siguientes líneas de programación:

{

NCable=Edit13->Text.ToDouble();

Wc2=(L2\*pp1\*NCable)/L;

Edit14->Text=AnsiString(Wc2);

}

1. Programamos al botón 8 “Wtensores” del mismo modo introduciendo las siguientes líneas de programación:

{

Npieza= Edit16->Text.ToDouble();

Wpieza= 8.4;

Edit15->Text=AnsiString(Wpieza);

Wtenso= (Wpieza\*Npieza)/L;

Edit17->Text=AnsiString(Wtenso);

}

1. Programamos al botón 9 “Area” del mismo modo introduciendo las siguientes líneas de programación:

{

Ar= L\*AC;

Edit18->Text=AnsiString(Ar);

Wplaca= 42.3;

Edit19->Text=AnsiString(Wplaca);

}

1. Programamos al botón 10 “Wpiso” del mismo modo introduciendo las siguientes líneas de programación:

{

Wpiso= (Wplaca\*Ar)/L;

Edit20->Text=AnsiString(Wpiso);

}

1. Programamos al botón 11 “# Rollos” del mismo modo introduciendo las siguientes líneas de programación:

{

rollo=(L/20)\*2;

Edit21->Text=AnsiString(rollo);

Wrollo= 45;

Edit22->Text=AnsiString(Wrollo);

}

1. Programamos al botón 12 “Wmalla” del mismo modo introduciendo las siguientes líneas de programación:

{

Wmalla=(rollo\*Wrollo)/L;

Edit23->Text=AnsiString(Wmalla);

}

1. Programamos al botón 13 “Wviguetas” del mismo modo introduciendo las siguientes líneas de programación:

{

Sp=Edit24->Text.ToDouble();

Wperfil= 4.54;

Edit25->Text=AnsiString(Wperfil);

Wvigueta= ((L/Sp)\* Wperfil)/L;

Edit26->Text=AnsiString(Wvigueta);

}

1. Programamos al botón 14 “Tramo” del mismo modo introduciendo las siguientes líneas de programación:

{

Tr=(Sp-0.05);

Edit27->Text=AnsiString(Tr);

}

1. Programamos al botón 15 “Calidad” del mismo modo introduciendo las siguientes líneas de programación:

{

Cnt= (L/Sp);

Edit28->Text=AnsiString(Cnt);

Wtramo= (2.46\*Tr);

Edit29->Text=AnsiString(Wtramo);

}

1. Programamos al botón 16 “Wangulos” del mismo modo introduciendo las siguientes líneas de programación:

{

Wangulos= (Wtramo\*Cnt\*Tr)/L;

Edit30->Text=AnsiString(Wangulos);

}

1. Programamos al botón 17 “Longitud” del mismo modo introduciendo las siguientes líneas de programación:

{

Rf= L\*2;

Edit31->Text=AnsiString(Rf);

Edit32->Text=AnsiString(Wperfil);

}

1. Programamos al botón 18 “Wrefuerzo” del mismo modo introduciendo las siguientes líneas de programación:

{

Wrefuerzo= (Wperfil\*2);

Edit33->Text=AnsiString(Wrefuerzo);

}

1. Programamos al botón 19 “Westructura” del mismo modo introduciendo las siguientes líneas de programación:

{

Westructura= Wpiso+ Wmalla+ Wvigueta+ Wangulos+ Wrefuerzo;

Edit34->Text=AnsiString(Westructura);

}

1. Programamos al botón 20 “Wmuerta” del mismo modo introduciendo las siguientes líneas de programación:

{

WMuerta= w2+Wc+Wc2+Westructura;

WMuerta= WMuerta\*1.05;

Edit35->Text=AnsiString(WMuerta);

}

1. Programamos al botón 21 “Wviva” del mismo modo introduciendo las siguientes líneas de programación:

{

lk= AC/2;

Pv= (146+(4464/L))\*((16.67-lk)/15.24) ;

if (Pv<=293)

{

Edit37->Text="Dentro de la Norma";

}

if (Pv>293)

{

Edit37->Text="Excede lo permitido por la Norma";

}

Wviva=Pv\*lk;

Edit36->Text=AnsiString(Wviva);

}

1. Programamos al botón 22 “Wtotal” del mismo modo introduciendo las siguientes líneas de programación:

{

WTotal= (Wviva+WMuerta);

Edit38->Text=AnsiString(WTotal);

}

1. Programamos al botón 23 “Salir” del mismo modo introduciendo las siguientes líneas de programación:

{

Form1->Show();

}

1. Programamos al botón 24 “Puente” del mismo modo introduciendo las siguientes líneas de programación:

{

WP=WTotal\*L;

Edit39->Text=AnsiString(WP) ;

}