

Trabajo Fin de Máster

Diseño de una aplicación informática para la generación de gráficos de interacción y la resolución de problemas de flexión compuesta y compresión compuesta, en elementos de hormigón armado

Javier Ortiz Avilés

Curso | 2012-2013
Convocatoria | Septiembre

Director/es:

Antonio Jesús Álvarez Martínez

Resumen

Las estructuras de hormigón armado son elementos que están sujetos a complejos cálculos. La vía más utilizada para la simplificación de estos cálculos es la de los llamados diagramas de interacción. Sin embargo, éstos, aunque útiles, presentan ciertas limitaciones de tipo práctico. El objetivo de este trabajo fue solventar dichos problemas desarrollando para ello una aplicación informática apoyada en Excel mediante el lenguaje de programación Visual Basic for Applications y para la que seguimos, como base de cálculo, la normativa actual vigente en España, la Instrucción Española del Hormigón Estructural del año 2008. La aplicación se halla estructurada en formularios (Userforms) que comunican al usuario con el programa de forma que introduciendo los parámetros pertinentes para que la aplicación realice el cálculo requerido se obtengan los resultados que la aplicación le devuelve. La aplicación es capaz de realizar los cálculos para un caso práctico determinado, así como generar un diagrama de interacción general. Se ha obtenido así una aplicación cuyo formato atiende a las necesidades actuales facilitando su uso, a la vez que aumenta en versatilidad, rapidez y eficiencia, con respecto al empleo de diagramas con los formatos actuales.

Palabras clave: *Visual Basic, Excel, Diagramas de interacción.*



Abstract

Reinforced concrete structures are components which are subject to complex calculations. The use of the so-called interaction diagrams is the most common way to simplify these calculations. However, these calculations, despite being useful, come up with certain practical constraints. The aim of this project was to solve such problems by developing an Excel-based application by means of Visual Basic for Applications computing language, together with, as a basis of calculation, current normative in Spain, the Code on Structural Concrete (EHE-08). The application is structured by Userforms, which connect the user to the programme, so that after introducing required values for the application to calculate, the application will generate the corresponding results. The application is able to make calculations for a given practical case, as well as to generate a general interaction diagram. Thus, we have created an application whose format meets current needs, making it easier to use, increasing, at the same time, its speed and efficiency, in relation to the use of diagrams in current formats.

Key words: Visual Basic, Excel, Interaction Diagrams



Índice

1. Introducción.....	pag.4
2. Materiales y métodos.....	pag.8
2.1. Materiales	
2.1.1. Microsoft Visual Basic for Applications.....	pag.9
2.1.2. Excel 2007	
2.2. Metodología	
2.2.1. Planteamiento del problema	
2.2.2. Bases de cálculo.....	pag.10
2.2.3. Secuencia de cálculo.....	pag.12
2.2.4. Resolución manual.....	pag.14
2.2.5. Construcción de una hoja Excel 2007.....	pag.17
2.2.6. Programación en VBA	
2.2.6.1. Código: UserForm3_Usuario_Contraseña.....	pag.18
2.2.6.2. Código: Userform1_Parámetros.....	pag.26
2.2.6.3. Código: Userform2.....	pag.36
2.2.6.4. Código: Módulo Funciones_No_Predefinidas	
2.2.7. Fase de prueba de fallos.....	pag.45
2.2.8. Depuración del código y mejoras	
3. Resultados.....	pag.46
4. Discusión.....	pag.49
5. Conclusiones.....	pag.50
6. Referencias.....	pag.51



1. Introducción

Hoy en día son de gran importancia todos los elementos estructurales de hormigón armado, tanto en el campo de la ingeniería como en el de la arquitectura, sosteniendo al resto de los elementos que son necesarios, para cubrir los espacios, mantener en altura a otros y abarcar instalaciones.

Los elementos estructurales generalmente tienen acción combinada de momento y carga axial, estado que se denomina flexo-compresión, este estado puede definirse como la acción de la carga a compresión con una excentricidad dada desde los ejes centroidales de la sección.

A la hora de analizar el esfuerzo de flexo-compresión resulta muy complicado debido a las diversas combinaciones que pueden producir el fallo, al tratarse de una sollicitación que origina un estado de deformación último que puede hallarse en cualquiera de los dominios de fallo, dependiendo de los esfuerzos que soliciten a la sección, exigiendo entonces el análisis de las ecuaciones de campo para todo el rango de valores de la profundidad relativa de la línea neutra, situados en el intervalo $0 < X < +\infty$; estas ecuaciones varían en dependencia del *dominio* en que se encuentre.

Debido a la complejidad del diseño y revisión de las estructuras de hormigón armado, tenemos la necesidad de disponer de instrumentos de cálculo que faciliten la labor de los facultativos competentes en el ámbito. Estos instrumentos proporcionan soluciones completas para las estructuras tipo, pero siempre resulta conveniente poder efectuar comprobaciones específicas sobre algunos elementos y secciones.

Por ello es conveniente tener a mano tablas, ábacos y diagramas que permiten resolver los problemas más habituales que plantea el cálculo de secciones de hormigón armado de una manera rápida, y precisa Montoya y col., (2001).

En la práctica está extendido el uso de los diagramas de interacción, ya que resuelve algunos inconvenientes que pueden presentar alguno de los otros métodos que mencionados.

Por esta razón, y para vencer a la complejidad vamos a utilizar los *diagramas de interacción* que no son más que “el lugar geométrico de todas las combinaciones de las cargas actuantes que llevan a un elemento a alcanzar su estado límite de resistencia”, Valdez (2001). A lo largo de la historia diferentes autores han tratado el tema, y cada época ha venido condicionada por características propias, ya sea por los materiales y técnicas



constructivas o por lo métodos de diseño, sin mencionar las condiciones geográficas de la región a las que se adaptaban.

Así en la primera edición de su libro, Montoya (1971) nos brinda diferentes ábacos, dedicados principalmente al cálculo de soportes sometidos a flexión o compresión compuesta, para los casos más usados.

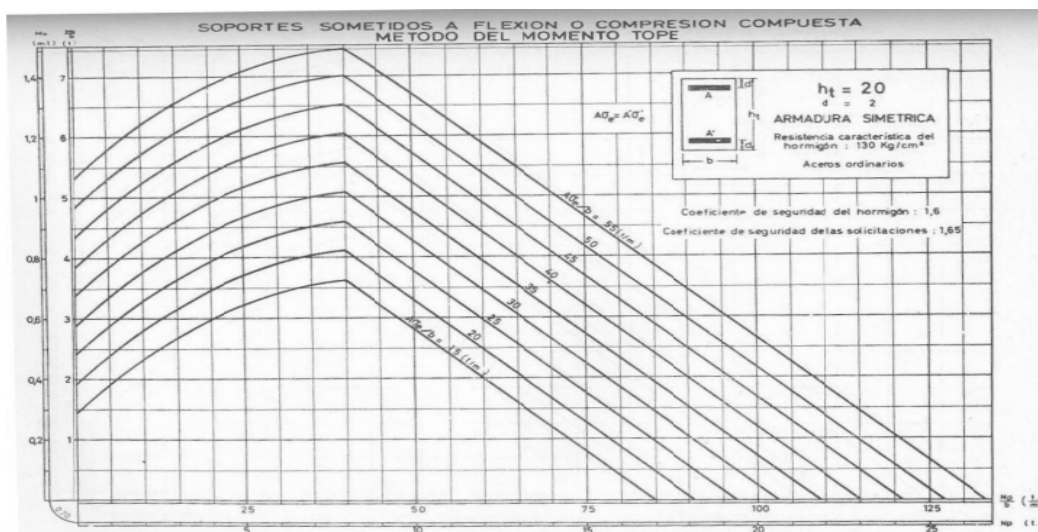


Fig.1 Diagrama de interacción, Jiménez Montoya, 1971.

Fuente: Hormigón Armado 1ra Edición. Jiménez Montoya, Pedro. 1971

Como se puede apreciar en la figura 1, los diagramas de interacción son obtenidos por el método de momento máximo. Estos diagramas constan de unas series, cada una correspondiente a las diferentes resistencias características del hormigón. Cada ábaco corresponde a un cálculo determinado, y consta de dos escalas principales cartesianas.

En el año 1997 el Instituto Americano Del Concreto (ACI) publica "ACI Design Handbook Sp-17-1997 "manual del ACI donde se presenta gráficas de interacción para el diseño de columnas y que se muestra en la figura 2.



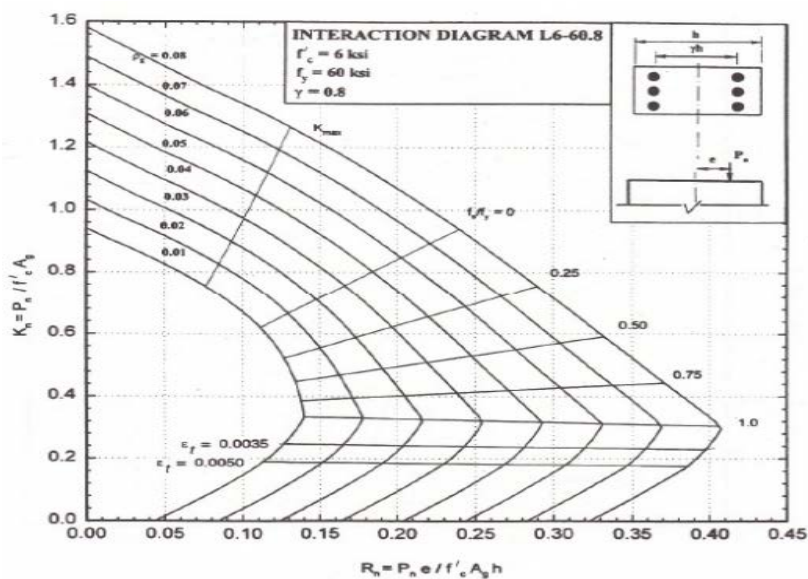


Fig. 2 Diagrama de interacción, ACI. 1997.

Fuente: ACI Design Handbook, ACI. Sp-17-1997

Otro de los diagramas históricamente importante son los brindados por Valdez (2000), que son una relación de cinco diagramas, según el autor los más comunes. En la figura 3 se muestra un ejemplo de los cinco.

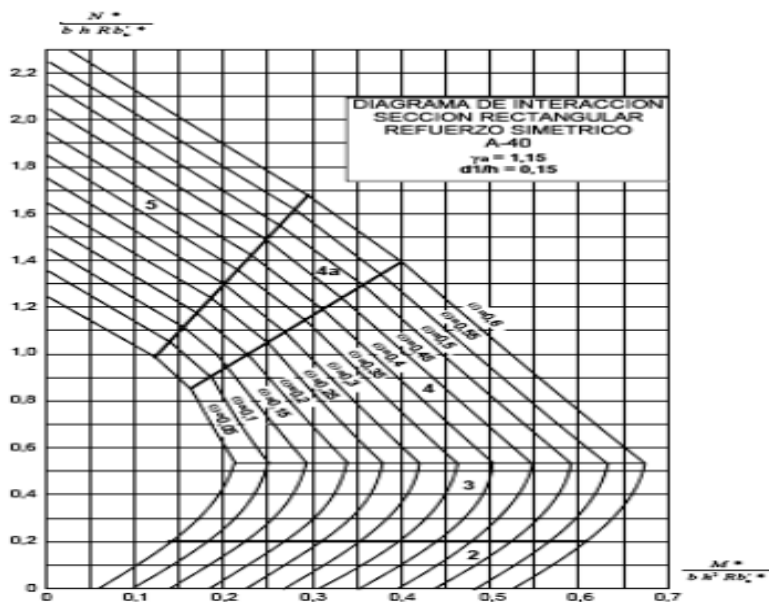


Fig.3 Diagrama de interacción, Valdez Avellaneda, 2000.

Fuente: Flexión compuesta en Hormigón Armado. Valdéz 2001

Estos son del tipo dimensional, por tanto son independientes de la calidad del hormigón y de las dimensiones de la sección. En cada ábaco se considera el tipo de acero, el recubrimiento mecánico, traducido en recubrimiento relativo, y el coeficiente reductor del acero.

Montoya, y col., (2001).en su texto Hormigón Armado, prepararon una colección de diagramas dimensionales de interacción que consta de 30 diagramas correspondientes a secciones rectangulares, en cajón, circulares y anulares, con distintas distribuciones de armaduras y recubrimientos.

En esta edición se observa como los autores completan la idea que comenzó el autor principal en la primera edición del texto, ya obteniendo los diagramas totalmente de forma adimensional; concepto también usado por el profesor Valdez Avellaneda en su texto. La figura 4 muestra uno de los 30 ejemplos.

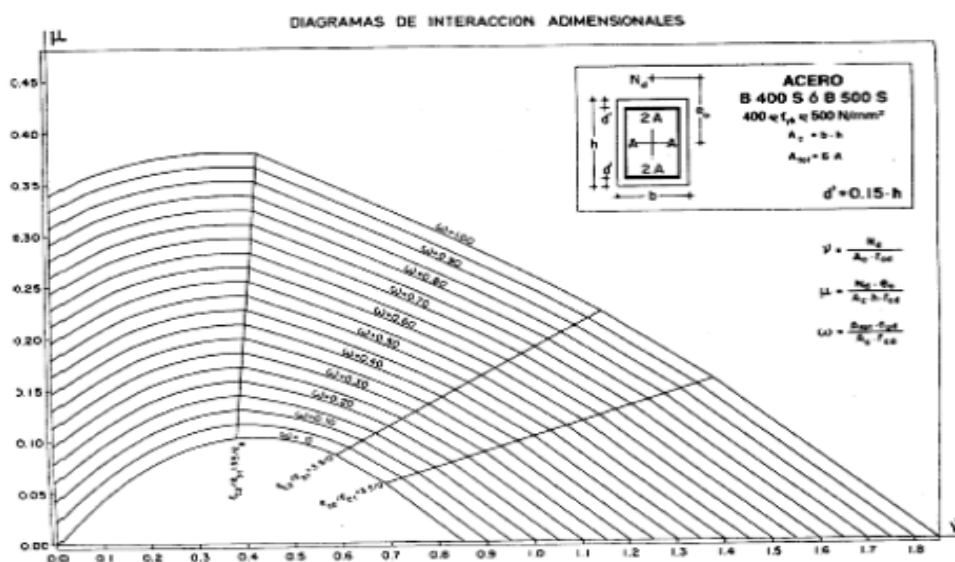


Fig.4 Diagrama de Interacción. Montoya et al., 2001.

Fuente: Montoya, col., 2001.

En dependencia de las condiciones de cada región, las regulaciones vigentes y las características propias de la construcción será entonces necesario crear estos ábacos para el diseño. En el análisis realizado se observa como los excelentes diagramas brindados por el ingeniero Jiménez Montoya son los que más se adaptan a las condiciones de España, por lo tanto son los que tomé como referencia para este trabajo.



Sin embargo los diagramas de interacción presentan limitaciones prácticas que muchas veces dificultan o imposibilitan su uso:

- Estos sólo valen para las características de sección para los que fueron generados; esto implica una gran limitación práctica inmediata, ya que necesitaríamos disponer de un diagrama generado concretamente para las características de la sección que deseamos calcular y esto muchas veces no ocurre así. De este modo entendemos, que una colección de tal naturaleza no puede contemplar todos los casos posibles, por lo que para registrarlos en bibliografía es necesario seleccionar aquellos diagramas que ayudan a resolver los problemas más frecuentes.
- Por otra parte generar un diagrama de interacción dimensional no es tarea que se pueda realizar de forma rápida por lo cual si en un momento dado no se dispone del diagrama de interacción previamente generado para las características de la sección, sería imposible resolver los problemas de manera rápida como pudiera ser necesario.
- El material disponible en bibliografía, en papel implica un inconveniente adicional de manejo a pie de Obra; ya que éste puede ensuciarse, romperse, perderse, etc. Lo que lo hace poco manejable en ese formato.

El objetivo de nuestro trabajo se centra en dar solución a las limitaciones prácticas que tiene el sistema clásico de diagramas de interacción en formato papel, aún utilizado, así como en proporcionar una nueva vía para la evolución que aún se apoya en el formato clásico y consideramos arcaico para el año en el que nos encontramos. Para ello desarrollamos una aplicación informática apoyada en Excel, mediante el lenguaje de programación Visual Basic for Applications.

* No se ha encontrado bibliografía referente al uso de VBA en Excel para el diseño de una aplicación que genere diagramas de interacción.

2. Materiales y métodos

2.1. Materiales

Los materiales utilizados para el desarrollo de este trabajo fueron los que siguen:



2.1.1. Microsoft Visual Basic for Applications

Microsoft Visual Basic for Applications (VBA) es el lenguaje de macros de Microsoft Visual Basic (VB), que se utiliza para programar aplicaciones Windows y que se incluye en varias aplicaciones Microsoft. En sí VBA es un subconjunto casi completo de Visual Basic 5.0 y 6.0.

Aunque Microsoft VBA viene integrado en las aplicaciones de Microsoft Office, como Word, Excel, Access y Powerpoint, en el presente trabajo hemos utilizado la aplicación Excel en su versión 2007.

2.1.2. Excel 2007

Excel2007 es una hoja de cálculo integrada en Microsoft Office. Una hoja de cálculo es un programa que es capaz de trabajar con números de forma sencilla e intuitiva. Para ello se utiliza una cuadrícula donde en cada celda de la cuadrícula se pueden introducir números, letras y gráficos.

2.2. Metodología

El proceso de generación y diseño de la aplicación siguió las siguientes fases:

2.2.1. Planteamiento del problema

* Tomamos como referencia el diagrama de interacción a construir (ver apartado 2.2.2 Bases de cálculo-Diagramas de interacción dimensionales).

El objetivo de este paso fue conocer, al detalle, tanto las operaciones que tendría que ser capaz de realizar la aplicación por sí misma para poder llegar a generar el diagrama de interacción deseado, como los resultados numéricos que se le exigirían que mostrara al final del proceso, para cumplir con los objetivos marcados.

Para ello lo primero que hicimos fue plantear un problema tipo para un caso concreto (con unas características geométricas y de materiales, de la sección, determinadas), y limitamos su estudio a los casos de interés para el presente trabajo, flexión compuesta y compresión compuesta.

Enunciado:

Construir el diagrama de interacción dimensional para una sección simétrica de HA-30 de 452,4 mm², con armaduras de acero B400S (4 \varnothing 12), $r= 40$ mm; sometida a flexión compuesta y compresión compuesta.



2.2.2. Bases de cálculo

Una vez fue conocido el enunciado del problema, se estudiaron las bases de cálculo a considerar para la correcta resolución manual, según la normativa EHE-08. El problema se resolvió siguiendo rigurosamente la normativa sin hacer ninguna variación, por lo que no la desarrollaremos a continuación. Pueden consultarse los documentos en internet. Por el contrario si describimos el diagrama de interacción que hay que construir pues hay muy diversos.

a) Diagramas de interacción:

Los diagramas de interacción que hemos tomado como referencia para el presente trabajo, son diagramas que ligan el momento de cálculo, M_d , con el esfuerzo normal de cálculo, N_d . Los diagramas de *interacción dimensional* son propios de una sección determinada (h =canto, b =ancho, r =recubrimiento mecánico), la armadura colocada (A_1 =armadura que trabaja a flexión y A_2 =armadura que trabaja a compresión) y las curvas de comportamiento de los materiales de ese caso. A partir de ahí, cualquier combinación (M_d =Momento, N_d =Axil) que caiga dentro de la curva o en la curva misma, identificará una situación de resistencia posible para la sección. Dicho en otras palabras, la sección es capaz de resistir las sollicitaciones propuestas. Un punto que esté fuera de la zona cubierta por la curva indica una combinación de esfuerzos no válida para la sección en cuestión. El diagrama de interacción es, entonces, un diagrama de resistencia de una sección, que se indica por una curva límite de resistencia y por toda el área delimitada bajo la propia curva para distintas combinaciones de M_d y N_d posibles. Tiene un uso muy sencillo que permite saber (con pocos parámetros adoptados) si la armadura hallada es idónea o lógica, desde el punto de vista resistente, o no. Por ejemplo, caer en la zona baja del diagrama (donde no existen valores) significará que la sección pre-dimensionada está sobredimensionada y no se necesita armadura porque el hormigón es suficiente (lo que implicaría seguramente, tener que dimensionara la armadura por cuantía mínima, que es antieconómico) y, por el contrario, caer fuera del diagrama (por arriba o por la derecha) significará que la sección está sub-dimensionada y debe ser redimensionada.



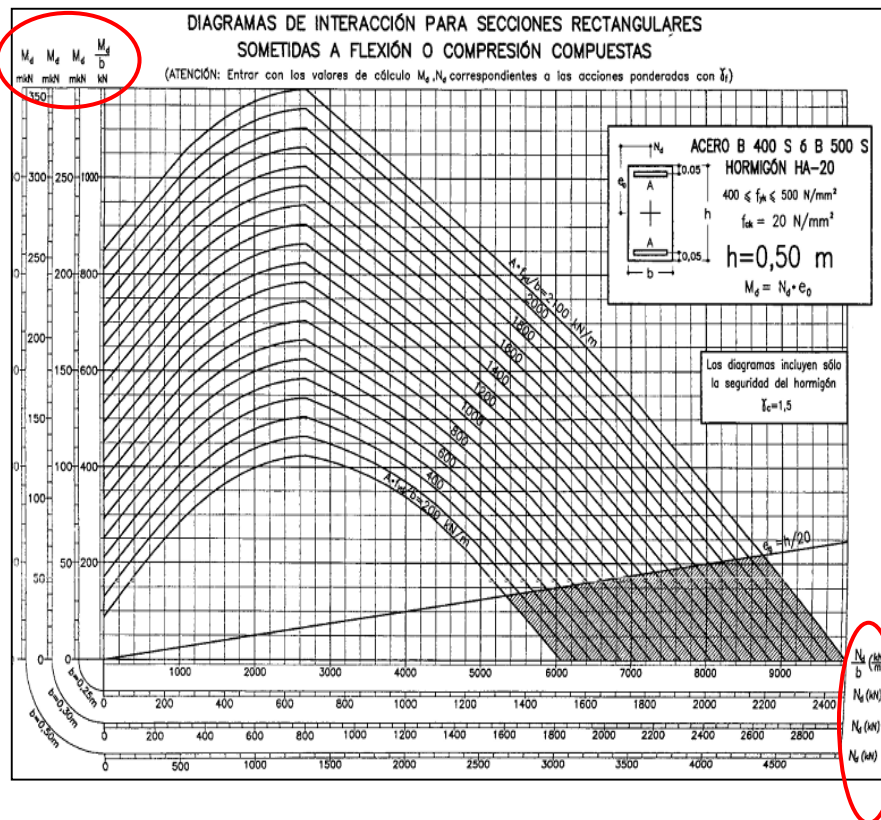


Fig.5. Diagrama de interacción, Montoya col., 2001

Fuente: Hormigón Armado, Montoya col., 2001.

- b) EHE-08 Art. 42.1.2 **Hipótesis básicas:**
- c) EHE-08 Art. 42.1.3 **Dominios de deformación**
- d) EHE-08 Art. 39.5.a **Diagrama tensión-deformación de cálculo del hormigón**
Diagrama parábola-rectángulo.
- e) EHE-08 Art. 38.4 **Diagrama tensión- deformación de cálculo del acero en las armaduras pasivas**
- f) EHE-08 Art. 15.3. **Coefficientes parciales de seguridad de los materiales**
- g) Condiciones de equilibrio



2.2.3. Secuencia de cálculo

Teniendo en cuenta las *bases de cálculo* y el problema planteado, una vez fueron definidas las dimensiones de una sección (h , b , r), la armadura colocada (A_1 y A_2), el tipo de hormigón, el acero tipificado y las curvas de comportamiento de los materiales, fue posible construir un diagrama o curva de interacción propia para la misma, teniendo en cuenta los diagramas de deformación (*dominios*).

Para la elaboración de la curva de interacción, para una sección dada, se utilizó el siguiente procedimiento:

1. Se definen diferentes posiciones del eje neutro
2. Para cada posición del eje neutro se calculan las deformaciones unitarias en cada fibra de la pieza, tomando como base una deformación máxima en el hormigón de $\varepsilon_c = 0.0035$ (en flexión) y la del acero $\varepsilon_s = 0.0010$. Para cada armadura.
3. En función de las deformaciones en el acero y en el hormigón se determinan desde los diagramas de comportamiento, los esfuerzos en el hormigón y en el acero. En cada armadura.
4. Se aplican las condiciones de equilibrio para despejar N_d
5. Por medio de la excentricidad de primer orden (e_e) consigo determinar M_d



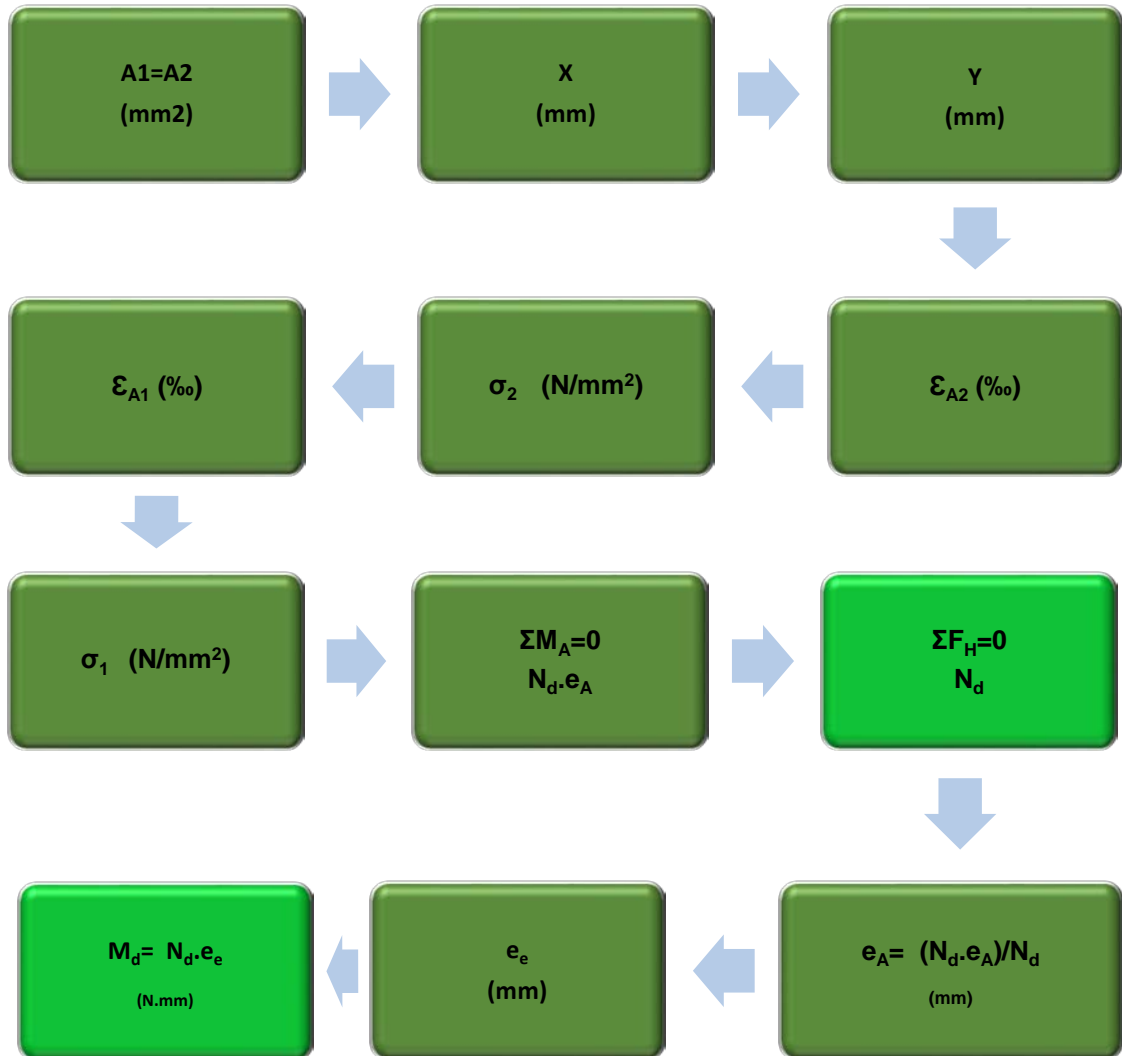


Fig.6 Secuencia de cálculo que se sigue para obtener las incógnitas M_d y N_d

Fuente: elaboración propia

2.2.4. Resolución manual

Una vez conocida la secuencia de cálculo para generar el diagrama de interacción que vimos en el apartado 2.2.2. Diagramas de interacción, se pudo calcular manualmente las incógnitas N_d y M_d que son necesarias (estos son el valor en el eje de abscisas y en el eje de ordenadas), para conocer al detalle el proceso. Así tomamos como incógnitas M_d y N_d respectivamente y aplicando las bases de cálculo, se resolvió como sigue para ambos casos planteados, en el dominio 2:

a) Caso flexión compuesta.

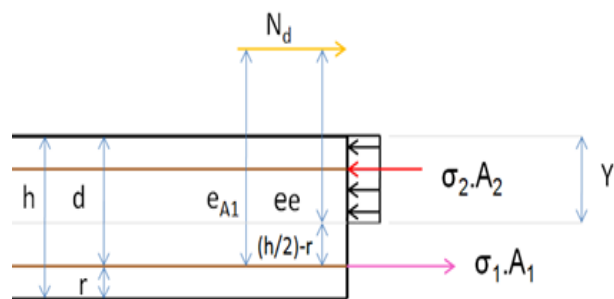


Fig.7 Esquema de la sección sometida a flexión compuesta.

Fuente: elaboración propia

b) Caso compresión compuesta

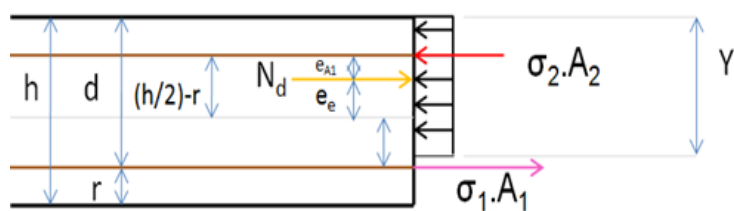


Fig.8 Esquema de la sección sometida a flexión compuesta

Fuente: elaboración propia



Dominio 2: $0 < X < 0.259 \cdot d$	
X (mm) = Profundidad de la línea neutra	Y(mm)= $0.8 \cdot X$ Desplazamiento en Y
ϵ_{A1} = 10% límite de deformación a tracción del acero	$\epsilon_{A2} = ((10 \cdot (X-r)) (d-X)) \cdot 0.001$ ·Límite de deformación a compresión del acero
A (mm ²)= área de la armadura	d (canto útil en mm) = h - r
h (mm)= canto de la estructura	b (mm) = ancho de la estructura
r (mm) = recubrimiento mecánico	E _s (Módulo de Young) = 200000 (N/mm ²)
σ_2 (N/mm ²) = $\epsilon_2 \cdot E_s$ Tensión en la armadura que trabaja a compresión compuesta	σ_1 (N/mm ²) = f_{yd} Tensión en la armadura que trabaja a flexión compuesta
Y _s = Coeficiente de minoración del acero =1.5	Y _c = Coeficiente de minoración del hormigón = 1.15
$f_{yd} = f_{yk} / Y_s$ Resistencia de cálculo del acero	f_{yk} =Límite elástico característico del acero (N/mm ²)
$f_{cd} = f_{ck} / Y_c$ Resistencia de cálculo del hormigón	f_{ck} = límite elástico característico del hormigón (N/mm ²)
ee= eA1- ((0.5*h)-r) excentricidad de primer orden (mm)	eA= (Nd.eA)/Nd en (mm)
$N_d = (N_d + \sigma_1 \cdot A_1) - (\sigma_2 \cdot A_2) - (f_{cd} \cdot b \cdot y)$	Axil (N)
$N_d \cdot e_{A1} = (\sigma_2 \cdot A_2 \cdot (d-r)) + (f_{cd} \cdot b \cdot y \cdot (d-y/2)) - (0.5 \cdot f_{cd} \cdot b \cdot y)$	
$M_d = N_d \cdot ee$ Momento (KN*m)	

Tabla 2. Incógnitas y fórmulas para el dominio 2. Fuente: elaboración propia

- En el Dominio 2 se cumple que: $0 < X < 0.259 \cdot d$, que puede trabajar a F_s o F_c , $\epsilon_s = 10\%$
- Calculo las incógnitas en el orden marcado en la secuencia de cálculo ya que son dependientes unas de las otras (ver apartado 2.2.3. Secuencia de cálculo):

Primero: calculo para las características de mi sección, cuales son los intervalos de cada dominio (en este caso solo calculo los del dominio 2).



→ Como conozco: A (área de la armadura), h (ancho) y r (recubrimiento mecánico)

$$A = 452.4 \text{ mm}^2$$

$$h = 400 \text{ mm}$$

$$r = 40 \text{ mm}$$

→ Puedo calcular: d (canto útil) y X crítica

$$d = h - r = 360 \text{ mm}$$

$$X_{\text{crítica}} = 0.259 \cdot d = 93.2 \text{ mm}$$

Con lo cual se puede afirmar que para estas características de sección, la profundidad del eje neutro en el dominio 2 varía de:

$$0 < x < 93.2$$

↓

Segundo: Conocido el intervalo del dominio 2 se puede hacer los cálculos para una X situada en dicho dominio, siguiendo la secuencia y aplicando las fórmulas pertenecientes al dominio 2.

→ Si X = 60 mm, entonces como nos encontramos en el dominio 2:

$$Y = 0.8 X = 48 \text{ mm}$$

→ Para el dominio 2 por semejanza de triángulos:

$$\epsilon_{A2} = ((10 \cdot (X-r)) / (d-X)) \cdot 0.001 = 6.7 \cdot 10^{-4}$$

→ Como el dominio 2 la armadura de tracción A₁ trabaja a su máximo

$$\text{Potencial entonces } \epsilon_{A1} = 0.010$$

$$\text{Como además } \epsilon_{A2} < \epsilon_{yd} \text{ entonces } \sigma_2 = \epsilon_2 \cdot E_s = 133.32 \text{ N/mm}^2$$

$$\text{Como además } \epsilon_{A1} \geq \epsilon_{yd} \text{ entonces } \sigma_1 = f_{yd} = f_{yk} / \gamma_s = 347.8 \text{ N/mm}^2$$

→ Si aplicamos condiciones de equilibrio:



$$\Sigma M_{A1} = 0; (N_d \cdot e_{A1}) - (\sigma_2 \cdot A_2 \cdot (d-r)) - (f_{cd} \cdot b \cdot y \cdot (d-(y/2))) = 0 \rightarrow N_d \cdot e_{A1} = (\sigma_2 \cdot A_2 \cdot (d-r)) + (f_{cd} \cdot b \cdot y \cdot (d-(y/2))) - (0.5 \cdot f_{cd} \cdot b \cdot y) = 116068.5 \text{ kN}\cdot\text{mm}$$

$$\Sigma F_H = 0; (N_d + \sigma_1 \cdot A_1) - (\sigma_2 \cdot A_2) - (f_{cd} \cdot b \cdot y) = 0 \rightarrow N_d = (N_d + \sigma_1 \cdot A_1) - (\sigma_2 \cdot A_2) - (f_{cd} \cdot b \cdot y) = 191.0 \text{ kN}$$

(X del diagrama de interacción)

Para el dominio 2 como es flexión compuesta:

$$e_{A1} = (N_d \cdot e_{A1}) / N_d = 607.7 \text{ mm}$$

$$e_e = e_{A1} - ((h/2) - r) = 447.8 \text{ mm}$$

Ya puedo calcular el valor de la coordenada Y del diagrama de interacción:

$$M_d = N_d \cdot e_e = 85529.8 \text{ mm}\cdot\text{kN}$$

2.2.5. Construcción de una hoja Excel 2007

Con este paso se conoció la manera de implantar dichos cálculos en Excel, así como las limitaciones de Excel, o dificultades que supone para el usuario.

Se construye una hoja de cálculo en Excel que calcule el proceso del problema (sin programación en VBA). Se comienza el proceso informático.

2.2.6. Programación VBA

Ya se conocía lo que debía hacer la aplicación, los cálculos, la forma en la que el usuario debía relacionarse con la aplicación, la forma en la que la aplicación debía de obtener los parámetros del usuario, etc. En definitiva, ya se tenía toda la información necesaria para poder empezar a programar según el lenguaje de programación VBA.

2.2.6.1. Código: UserForm3_Usuario_Contraseña:



El código que se escribió para obtener el UserForm3_Usuario_Contraseña (con las prestaciones deseadas) es decir el formulario para meter el usuario y la contraseña, fue el siguiente:

```
'En este formulario escribo el código referente al formulario 3 y su trabajo a realizar.
Option Explicit
    Dim contador As Byte
    Dim usuario, Contraseña As String
'Declaro las variables como String ya que las voy a tomar como texto (no diferencio
entre números y letras para evitarme problemas)

'CUANDO SE INICIA EL FORMULARIO 3 DEBE DE:
Private Sub UserForm_Initialize() 'iniciarse el formulario la variable contador toma el
valor 3
    contador = 3 'me creo un contador para ayudar a llevar las cuentas. Se ejecutará
el proceso de nuevo cada vez que tome un valor distinto
    Botón_Aceptar_UserForm3_Usuario_Contraseña.Enabled = False
'el botón aceptar esté inactivo
'decimos que la propiedad enabled no se cumple
' Modifico desde código algunas de las propiedades del formulario que no quiero que
aparezcan como las que muestra por defecto:
'Evidentemente esto se puede hacer también desde la ventana de propiedades pero
es interesante ver como se puede modificar desde el código (no voy a mostrar todas
las propiedades modificadas ya que la ventana propiedades permite hacerlo más
rápidamente).
'Le dice a la Label de name Usuario_Label que muestre Usuario
    Usuario_Label.Caption = "Usuario" 'La propiedad Caption hace referencia a lo
que el usuario verá escrito
'Le dice a la Label de name Contraseña_Label que muestre Contraseña
    Contraseña_Label.Caption = "Contraseña"
'Le dice al botón Botón_Aceptar_UserForm3_Usuario_Contraseña que muestre el
texto Aceptar en él.
    Botón_Aceptar_UserForm3_Usuario_Contraseña.Caption = "Aceptar"
'Le indico cuanto tiene que medir de alto el formulario
    UserForm3_Usuario_Contraseña.Height = 220 'La propiedad alto del formulario
es Height
'Le indico cuanto tiene que medir de ancho el fomulario
    UserForm3_Usuario_Contraseña.Width = 333
```



```
'La propiedad Width del formulario es la del ancho del mismo
'Le indico cuanto tiene que medir de alto la textbox de nombre Usuario_TextBox:
    Usuario_TextBox.Height = 24
'Le indico cuanto tiene que medir de ancho la textbox de nombre Usuario_TextBox:
    Usuario_TextBox.Width = 120
'Le puedo indicar el color que quiero que tenga cualquier parte del formulario, aquí lo
haré de la TextBox del Usuario.
    Usuario_TextBox.BackColor = &H80000005 'Le indico que el color de fondo sea
blanco (código del blanco)
'Puedo cambiar muchas más propiedades del formulario desde el código
    End Sub

'FUNCIONAMIENTO DEL FORMULARIO USERFORM3 DE USUARIO Y
CONTRASEÑA:

Private Sub Botón_Aceptar_UserForm3_Usuario_Contraseña_Click() 'Lo que dice este
procedimiento va a ocurrir después de hacer click en el botón aceptar del mismo
formulario
    usuario = Usuario_TextBox.Text 'Inicio la variable Usuario
    Contraseña = Contraseña_TextBox.Text 'Inicio la variable contraseña
    If contador > 0 Then 'Abro un condicional que va a leerse siempre que sea positivo
(osea 2 veces porque el contador es 3)
        If Usuario_TextBox.Text = "TFM" And Contraseña_TextBox.Text = "2013" Then
'Si el usuario es TFM y la contraseña 2013 entonces
            UserForm3_Usuario_Contraseña.Hide 'Que se quite el formulario 3
            UserForm1_Parámetros.Show False
'que se muestre el formulario parámetros de forma no modal (que deje que se pueda
acceder a otras ventanas)
            contador = 3
        Else 'en caso de que no se cumpla lo anterior sucederá lo siguiente
            contador = contador - 1 'El contador valdrá uno menos
            If contador = 0 Then 'Si el contador es 0, es porque ya ha entrado 3 veces y
es el máximo de oportunidades
                Application.ActiveWorkbook.Close 'en ese caso se cierra el libro
                Else 'Si no es cero el contador
                    MsgBox "Compruebe que estén correctamente rellenos los campos
Usuario y Contraseña" & vbCrLf & "¡Tiene solo (" & contador & ") intentos más antes de
```



```
que se cierre la aplicación!", vbExclamation 'se muestra dicho mensaje. vbCrLf  
(Proporciona salto de línea), & (concatena) y vbExclamation (muestra el icono de  
Mensaje de Advertencia)
```

```
    Usuario_TextBox.Value = ""  
    Contraseña_TextBox.Value = ""  
    Usuario_TextBox.SetFocus  
End If
```

```
End If
```

```
Else
```

```
    Application.ActiveWorkbook.Close (False)
```

```
    Unload Me 'Descarga el formulario
```

```
End If
```

```
End Sub
```

' CUANDO DEBE HABILITARSE EL BOTÓN "ACEPTAR" DEL FORMULARIO 3:

'que se active el botón "aceptar" del Userform3 solo cuando hayan sido rellenado el dato de contraseña, que como tengo indicado por set focus que aparezca en Usuario el puntero, será por lógica el último en rellenarse.(En el evento "iniciar el formulario 3" ya le indiqué que al iniciarse el formulario este botón esté inhabilitado. Creo un procedimiento para que compruebe si están completos los dos campos necesarios antes de activar el botón aceptar

```
Private Sub comprobacióndecasillasrellenadas()
```

```
If Contraseña_TextBox <> "" And Usuario_TextBox.Value <> "" Then 'Si la TextBox de  
la contraseña están completadas entonces
```

```
    Botón_Aceptar_UserForm3_Usuario_Contraseña.Enabled = True 'El botón  
aceptar del Userform3 está habilitado
```

```
    Else: Botón_Aceptar_UserForm3_Usuario_Contraseña.Enabled = False 'Pero si no  
entonces está inhabilitado
```

```
End If
```

```
End Sub
```

'Le indico que cuando haya un cambio en la casilla Contraseña_TextBox compruebe si están completos los dos cambios necesarios para activar o no el botón aceptar.

```
Private Sub Contraseña_TextBox_Change()
```



```
Call comprobacióndecasillasrellenadas 'Con la instrucción call puedo llamar a un
procedimiento
End Sub
'Le indico que cuando haya un cambio en la casilla Usuario_TextBox compruebe si están
completos los dos cambios necesarios para activar o no el botón aceptar.
Private Sub Usuario_TextBox_Change()
Call comprobacióndecasillasrellenadas
End Sub
```

2.2.6.2. Código: Userform1_parámetros:

El código que se "picó" para obtener el formulario para introducir los parámetros con las prestaciones deseadas fue el siguiente:

```
'Aquí escribo el código referente al formulario 1 y el trabajo que vaya a realizar:
Option Explicit 'Fuerza la declaración explícita de todas las variables de un archivo mediante
instrucciones Dim o ReDim. Recomendable para evitar confusiones al escribir variables en el
código.
Public AumentoDeX As Integer
Dim Gráfico As Chart
Dim NombreGIF$
Private Sub Diagrama_CommandButton_Click()
NombreGIF = ActiveWorkbook.Path & "\temporal.gif"
'indicamos qué gráfico vamos a insertar como imagen en el Formulario
Set Gráfico = Worksheets("Hoja6").ChartObjects(1).Chart
Dim graficoactivo As Object
Gráfico.Export Filename:=NombreGIF, FilterName:="GIF"
VAYA_Userform.Show
End Sub
Private Sub Sección_Marco_Click()
End Sub
'CUANDO SE INICIE EL FORMULARIO 1 QUE OCURRA:
Private Sub UserForm_Initialize()
'Coloco el cursor en la casilla Ancho_Textbox:
Ancho_TextBox.SetFocus
'Que el botón calcular esté inhabilitado
AceptarCalcular_CommandButton2.Enabled = False
'Que el botón mostrar diagrama esté inhabilitado
Diagrama_CommandButton.Enabled = False
```



```
'Le indico el ancho que quiero que tenga al iniciarse el formulario
Me.Width = 610 'Me hace referencia al formulario ya que estoy dentro de él
'Le indico el alto que quiero que tenga al iniciarse el formulario
Me.Height = 395
End Sub
'CUANDO SE ACTIVE EL FORMULARIO 1 QUE OCURRA:
Private Sub UserForm_Activate()
'Se añadan los siguientes (más probables, aunque podrán ser modificables) valores a las
TextBox siguientes:
    Yc_TextBox.Value = 1.5
    Ys_TextBox.Value = 1.15
    Young_TextBox = 200000
'Se añadan los siguientes valores a los combobox correspondientes:
    With Fck_ComboBox 'Ejecuta una serie de instrucciones que hacen referencia repetidas
veces a un único objeto, simplificando código.
        .AddItem 25 'Gracias al método AddItem de los comboBox
        .AddItem 30
        .AddItem 35
        .AddItem 40
        .AddItem 45
        .AddItem 50
        .AddItem 55
        .AddItem 60
        .AddItem 70
        .AddItem 80
        .AddItem 90
        .AddItem 100
    End With
    Fyk_ComboBox.AddItem 400
    Fyk_ComboBox.AddItem 500

    AceptarCalcular_CommandButton2.Enabled = False

End Sub
'Procedimiento para comprobar que no falta ningún campo por rellenar y se puede activar el
botón calcular
```



```
Private Sub comprobarparaactivarcalcular()
    If Ancho_TextBox <> "" And Canto_TextBox <> "" And Recubrimiento_TextBox <> ""
        And Armadura_TextBox <> "" And Fck_ComboBox <> "" And Yc_TextBox <> "" And
        Fyk_ComboBox <> "" And Ys_TextBox <> "" And Young_TextBox <> "" And X_TextBox
        <> "" Then

        If Ancho_TextBox = "." Or Canto_TextBox = "." Or Recubrimiento_TextBox = "." Or
        Armadura_TextBox = "." Or Fck_ComboBox = "." Or Yc_TextBox = "." Or
        Fyk_ComboBox = "." Or Ys_TextBox = "." Or Young_TextBox = "." Or X_TextBox = "."
    Then
        AceptarCalcular_CommandButton2.Enabled = False
    Else
        AceptarCalcular_CommandButton2.Enabled = True
    End If
Else
    AceptarCalcular_CommandButton2.Enabled = False
End If
End Sub

Private Sub Ancho_TextBox_Change()
    'La variable b toma el valor que se escriba dentro de la casilla Ancho_TextBox del formulario 1
    (hay que especificar qué valor toma de entre todos los que puede tomar la variable)
    b = Ancho_TextBox.Value
        Call comprobarparaactivarcalcular 'llamo al procedimiento
End Sub

Private Sub Canto_TextBox_Change()
    'La variable h toma el valor que se escriba dentro de la casilla Canto_TextBox
    h = Canto_TextBox.Value
        Call comprobarparaactivarcalcular
End Sub

Private Sub Recubrimiento_TextBox_Change()
    'La variable r toma el valor que se anote en la casilla Recubrimiento_TextBox
    r = Recubrimiento_TextBox.Value
        Call comprobarparaactivarcalcular
End Sub

Private Sub Armadura_TextBox_Change()
    'La variable A toma el valor que se anote en la casilla Armadura_TextBox
```




```
A = Armadura_TextBox.Value
    If Armadura_TextBox.Value = 0 Then Armadura_TextBox.Text = ""
    Call comprobarparaactivarcalcular
End Sub
Private Sub Fck_ComboBox_Change()
'La variable Fck tome el valor del Fck_ComboBox cuando este sea rellenado (cuando cambie)
    Fck = Fck_ComboBox.Value
    Call comprobarparaactivarcalcular
End Sub

Private Sub Yc_TextBox_Change()
    If Yc_TextBox.Value = 0 Then Yc_TextBox.Text = ""
    Call comprobarparaactivarcalcular
End Sub
Private Sub Fyk_ComboBox_Change()
'Que la variable Fyk tome el valor del Fyk_ComboBox cuando este sea rellenado (cuando cambie)
    Fyk = Fyk_ComboBox.Value
    Call comprobarparaactivarcalcular
End Sub
Private Sub Ys_TextBox_Change()
'La Yc toma el valor
    Ys = Ys_TextBox.Value
    If Ys_TextBox.Value = 0 Then Ys_TextBox.Text = ""
    Call comprobarparaactivarcalcular
End Sub
Private Sub Young_TextBox_Change()
'La Es toma el valor introducido en Young_TextBox
    Es = Young_TextBox.Value
    Call comprobarparaactivarcalcular
End Sub
Private Sub X_TextBox_Change()
'La X toma el valor introducido en X_TextBox
    X = X_TextBox.Value
    If X_TextBox.Value = 0 Then X_TextBox.Text = ""
Call comprobarparaactivarcalcular
End Sub
```



'CUANDO SE HAGA CLICK AL BOTÓN CALCULAR DEL USERFORM1 OCURRIRÁ: Cuando haga click se realizarán los cálculos y se generará el diagrama de interacción (para ello debemos llamar a las funciones).

'A) Cálculos para una X determinada:

```
Private Sub AceptarCalcular_CommandButton2_Click()  
    MsgBox "¡Atención!: Esta operación puede tardar unos minutos"  
    Ycc = CalcularYcc(h, X)  
    Yfc = CalcularYfc(X)  
    Ys = Ys_TextBox  
    Yc = Yc_TextBox  
    d = Calculard(h, r)  
    xc = XCrítica(d)  
    xl = XLímite(d)  
    Fck = Fck_ComboBox  
    Fyk = Fyk_ComboBox  
    Fcd = CalcularFcd(Fck, Yc)  
    Fyd = CalcularFyd(Fyk, Ys)  
    Es = Young_TextBox.Value  
    Label7.Caption = "X (mm): " & X_TextBox.Value 'El nombre de la etiqueta será el texto  
entre comillas seguido del valor que tenga la textbox de la x  
    If X > 0 And X < h Then 'Las X que corresponden a los dominios 2,3,4, y 4a  
        Yfc = CalcularYfc(X)  
        Y_Label.Caption = "Y (mm): " & CalcularYfc(X)  
    ElseIf X >= h Then  
        Ycc = CalcularYcc(h, X) 'Las X que corresponden al dominio 5. Las X<0 no es  
necesario ponerlas ya que no se permite introducir en la ficha el signo "-" con lo cual es  
imposible.  
        Y_Label.Caption = "Y (mm): " & CalcularYfc(X)  
    End If  
    Fyd = CalcularFyd(Fyk, Ys)  
    If X > 0 And X < xc Then 'Usar esta en el intervalo del dominio 2  
        EA22 = CalcularEA22(X, r, d)  
        DefA2_Label.Caption = "Deformación en A2 (%): " & CalcularEA22(X, r, d) * 1000  
        Dominio_Label.Caption = "Dominio de deformación:" & " 2"  
    ElseIf X >= xc And X < xl Then 'Usar esta en el intervalo del dominio 3  
        EA23 = CalcularEA23(X, r)
```



```
DefA2_Label.Caption = "Deformación en A2 (%): " & CalcularEA23(X, r) * 1000
Dominio_Label.Caption = "Dominio de deformación:" & " 3"
Elseif X >= xl And X < d Then 'Usar esta en el intervalo del dominio 4
EA24 = CalcularEA24(X, r)
DefA2_Label.Caption = "Deformación en A2 (%): " & CalcularEA24(X, r) * 1000
Dominio_Label.Caption = "Dominio de deformación:" & " 4"
Elseif X >= d And X < h Then 'Usar esta en el intervalo del dominio 4a
EA24a = CalcularEA24a(X, r)
DefA2_Label.Caption = "Deformación en A2 (%): " & CalcularEA24a(X, r) * 1000
Dominio_Label.Caption = "Dominio de deformación:" & " 4a"
Elseif X >= h Then 'Usar esta en el intervalo del dominio 5
EA25 = CalcularEA25(X, r, h)
DefA2_Label.Caption = "Deformación en A2 (%): " & CalcularEA25(X, r, h) * 1000
Dominio_Label.Caption = "Dominio de deformación:" & " 5"
End If
If X > 0 And X < xc Then 'Usar esta en el intervalo del dominio 2
EA12 = CalcularEA12()
DefA1_Label.Caption = "Deformación en A1 (%): " & CalcularEA12() * 1000
Elseif X >= xc And X < xl Then 'Usar esta en el intervalo del dominio 3
EA13 = CalcularEA13(d, X)
DefA1_Label.Caption = "Deformación en A1 (%): " & CalcularEA13(d, X) * 1000
Elseif X >= xl And X < d Then 'Usar esta en el intervalo del dominio 4
EA14 = CalcularEA14(d, X)
DefA1_Label.Caption = "Deformación en A1 (%): " & CalcularEA14(d, X) * 1000
Elseif X >= d And X < h Then 'Usar esta en el intervalo del dominio 4a
EA14a = CalcularEA14a(X, d)
DefA1_Label.Caption = "Deformación en A1 (%): " & CalcularEA14a(X, d) * 1000
Elseif X >= h Then 'Usar esta en el intervalo del dominio 5
EA15 = CalcularEA15(X, d, h)
DefA1_Label.Caption = "Deformación en A1 (%): " & CalcularEA15(X, d, h) * 1000
End If
If X > 0 And X < xc Then 'Usar esta en el intervalo del dominio 2
TA22 = CalcularTA22(EA22, Es)
TensiónA2_Label.Caption = "Tensión de la armadura en A2 (N/mm2): " &
CalcularTA22(EA22, Es)
Elseif X >= xc And X < xl Then 'Usar esta en el intervalo del dominio 3
TA23 = CalcularTA23(EA13, Es)
```



```
TensiónA2_Label.Caption = "Tensión de la armadura en A2 (N/mm2): " &
CalcularTA23(EA13, Es)
  Elself X >= xl And X < d Then 'Usar esta en el intervalo del dominio 4
    TA24 = CalcularTA24(Fyd)
    TensiónA2_Label.Caption = "Tensión de la armadura en A2 (N/mm2): " &
CalcularTA24(Fyd)
  Elself X >= d And X < h Then 'Usar esta en el intervalo del dominio 4a
    TA24a = CalcularTA24a(Fyd)
    TensiónA2_Label.Caption = "Tensión de la armadura en A2 (N/mm2): " &
CalcularTA24a(Fyd)
  Elself X >= h Then 'Usar esta en el intervalo del dominio 5
    TA25 = CalcularTA25(Fyd)
    TensiónA2_Label.Caption = "Tensión de la armadura en A2 (N/mm2): " &
CalcularTA25(Fyd)
End If
  If X > 0 And X < xc Then 'Usar esta en el intervalo del dominio 2
    TA12 = CalcularTA12(Fyd)
    TensiónA1_Label.Caption = "Tensión de la armadura en A1 (N/mm2): " &
CalcularTA12(Fyd)
  Elself X >= xc And X < xl Then 'Usar esta en el intervalo del dominio 3
    TA13 = CalcularTA13(Fyd)
    TensiónA1_Label.Caption = "Tensión de la armadura en A1 (N/mm2): " &
CalcularTA13(Fyd)
  Elself X >= xl And X < d Then 'Usar esta en el intervalo del dominio 4
    TA14 = CalcularTA14(EA14, Es)
    TensiónA1_Label.Caption = "Tensión de la armadura en A1 (N/mm2): " &
CalcularTA14(EA14, Es)
  Elself X >= d And X < h Then 'Usar esta en el intervalo del dominio 4a
    TA14a = CalcularTA14a(EA14a, Es)
    TensiónA1_Label.Caption = "Tensión de la armadura en A1 (N/mm2): " &
CalcularTA14a(EA14a, Es)
  Elself X >= h Then 'Usar esta en el intervalo del dominio 5
    TA15 = CalcularTA15(EA15, Es)
    TensiónA1_Label.Caption = "Tensión de la armadura en A1 (N/mm2): " &
CalcularTA15(EA15, Es)
End If
  If X > 0 And X < xc Then 'Usar esta en el intervalo del dominio 2
```



```
NdporeA12 = CalcularNdporeA12(TA22, A, d, r, Fcd, b, Yfc)
Elseif X >= xc And X < xl Then 'Usar esta en el intervalo del dominio 3
    NdporeA13 = CalcularNdporeA13(TA23, A, d, r, Fcd, b, Yfc)
Elseif X >= xl And X < d Then 'Usar esta en el intervalo del dominio 4
    NdporeA14 = CalcularNdporeA14(TA24, A, d, r, Fcd, b, Yfc)
Elseif X >= d And X < h Then 'Usar esta en el intervalo del dominio 4a
    NdporeA24a = CalcularNdporeA24a(TA14a, A, d, r, Fcd, b, Yfc)
Elseif X >= h Then 'Usar esta en el intervalo del dominio 5
    NdporeA25 = CalcularNdporeA25(TA15, A, d, r, Fcd, b, Ycc)
End If
If X > 0 And X < xc Then 'Usar esta en el intervalo del dominio 2
    Nd2 = CalcularNd2(TA12, A, TA22, Fcd, b, Yfc)
    Nd_Label.Caption = "Axil máximo Nd(kN): " & CalcularNd2(TA12, A, TA22, Fcd, b, Yfc)
Elseif X >= xc And X < xl Then 'Usar esta en el intervalo del dominio 3
    Nd3 = CalcularNd3(TA13, A, TA23, Fcd, b, Yfc)
    Nd_Label.Caption = "Axil máximo Nd(kN): " & CalcularNd3(TA13, A, TA23, Fcd, b, Yfc)
Elseif X >= xl And X < d Then 'Usar esta en el intervalo del dominio 4
    Nd4 = CalcularNd4(TA14, A, TA24, Fcd, b, Yfc)
    Nd_Label.Caption = "Axil máximo Nd(kN): " & CalcularNd4(TA14, A, TA24, Fcd, b, Yfc)
Elseif X >= d And X < h Then 'Usar esta en el intervalo del dominio 4a
    Nd4a = CalcularNd4a(TA14a, A, TA24a, Fcd, b, Yfc)
    Nd_Label.Caption = "Axil máximo Nd(kN): " & CalcularNd4a(TA14a, A, TA24a, Fcd, b, Yfc)
Elseif X >= h Then 'Usar esta en el intervalo del dominio 5
    Nd5 = CalcularNd5(TA15, A, TA25, Fcd, b, Ycc)
    Nd_Label.Caption = "Axil máximo Nd(kN): " & CalcularNd5(TA15, A, TA25, Fcd, b,
Ycc)
End If
If X > 0 And X < xc Then 'Usar esta en el intervalo del dominio 2
    eA2 = CalcularA2(NdporeA12, Nd2)
Elseif X >= xc And X < xl Then 'Usar esta en el intervalo del dominio 3
    eA3 = CalcularA3(NdporeA13, Nd3)
Elseif X >= xl And X < d Then 'Usar esta en el intervalo del dominio 4
    eA4 = CalcularA4(NdporeA14, Nd4)
Elseif X >= d And X < h Then 'Usar esta en el intervalo del dominio 4a
    eA4a = CalcularA4a(NdporeA24a, Nd4a)
Elseif X >= h Then 'Usar esta en el intervalo del dominio 5
    eA5 = CalcularA5(NdporeA25, Nd5)
```



```
End If
  If X > 0 And X < xc Then 'Usar esta en el intervalo del dominio 2
    ee2 = Calcular ee2(eA2, h, r)
  ElseIf X >= xc And X < xl Then 'Usar esta en el intervalo del dominio 3
    ee3 = Calcular ee3(eA3, h, r)
  ElseIf X >= xl And X < d Then 'Usar esta en el intervalo del dominio 4
    ee4 = Calcular ee4(eA4, h, r)
  ElseIf X >= d And X < h Then 'Usar esta en el intervalo del dominio 4a
    ee4a = Calcular ee4a(eA4a, h, r)
  ElseIf X >= h Then 'Usar esta en el intervalo del dominio 5
    ee5 = Calcular ee5 (eA5, h, r)
End If
If X > 0 And X < xc Then 'Usar esta en el intervalo del dominio 2
  Md2 = Calcular Md2(Nd2, ee2)
  Md_Label.Caption = "Momento máximo Md(kN.M): " & Calcular Md2(Nd2, ee2)
Elseif X >= xc And X < xl Then 'Usar esta en el intervalo del dominio 3
  Md3 = Calcular Md3(Nd3, ee3)
  Md_Label.Caption = "Momento máximo Md(kN.M): " & Calcular Md3(Nd3, ee3)
Elseif X >= xl And X < d Then 'Usar esta en el intervalo del dominio 4
  Md4 = Calcular Md4(Nd4, ee4)
  Md_Label.Caption = "Momento máximo Md(kN.M): " & Calcular Md4(Nd4, ee4)
Elseif X >= d And X < h Then 'Usar esta en el intervalo del dominio 4a
  Md4a = Calcular Md4a(Nd4a, ee4a)
  Md_Label.Caption = "Momento máximo Md(kN.M): " & Calcular Md4a(Nd4a, ee4a)
Elseif X >= h Then 'Usar esta en el intervalo del dominio 5
  Md5 = Calcular Md5(Nd5, ee5)
  Md_Label.Caption = "Momento máximo Md(kN.M): " & Calcular Md2(Nd5, ee5)
End If
  AceptarCalcular_CommandButton2.Enabled = False

'B)Generación del diagrama de interacción para cualquier X y localización de nuestra X
determinada:
  Worksheets("Hoja6").Activate
  Worksheets("Hoja6").Range("a2:b504").Select
Selection.Clear
For X = 1 To 504
  d = Calcular d(h, r)
```



```
xc = XCrítica(d)
xl = XLímite(d)
Ys = Ys_TextBox
Yc = Yc_TextBox
Fck = Fck_Combobox
Fyk = Fyk_Combobox
Es = Young_TextBox.Value
If X > 0 And X < h Then 'Las X que corresponden a los dominios 2,3,4, y 4a
    Yfc = CalcularYfc(X)
Elseif X >= h Then
    Ycc = CalcularYcc(h, X) 'Las X que corresponden al dominio 5. Las X<0 no es necesario
    ponerlas ya que no se permite introducir en la ficha el signo "-" con lo cual es imposible.
End If
    Fyd = CalcularFyd(Fyk, Ys)
If X > 0 And X < xc Then 'Usar esta en el intervalo del dominio 2
    EA22 = CalcularEA22(X, r, d)
Elseif X >= xc And X < xl Then 'Usar esta en el intervalo del dominio 3
    EA23 = CalcularEA23(X, r)
Elseif X >= xl And X < d Then 'Usar esta en el intervalo del dominio 4
    EA24 = CalcularEA24(X, r)
Elseif X >= d And X < h Then 'Usar esta en el intervalo del dominio 4a
    EA24a = CalcularEA24a(X, r)
Elseif X >= h Then 'Usar esta en el intervalo del dominio 5
    EA25 = CalcularEA25(X, r, h)
End If
If X > 0 And X < xc Then 'Usar esta en el intervalo del dominio 2
    EA12 = CalcularEA12()
Elseif X >= xc And X < xl Then 'Usar esta en el intervalo del dominio 3
    EA13 = CalcularEA13(d, d)
Elseif X >= xl And X < d Then 'Usar esta en el intervalo del dominio 4
    EA14 = CalcularEA14(d, X)
Elseif X >= d And X < h Then 'Usar esta en el intervalo del dominio 4a
    EA14a = CalcularEA14a(X, d)
Elseif X >= h Then 'Usar esta en el intervalo del dominio 5
    EA15 = CalcularEA15(X, d, h)
End If
If X > 0 And X < xc Then 'Usar esta en el intervalo del dominio 2
```



```
TA22 = CalcularTA22(EA22, Es)
Elseif X >= xc And X < xl Then 'Usar esta en el intervalo del dominio 3
    TA23 = CalcularTA23(EA13, Es)
Elseif X >= xl And X < d Then 'Usar esta en el intervalo del dominio 4
    TA24 = CalcularTA24(Fyd)
Elseif X >= d And X < h Then 'Usar esta en el intervalo del dominio 4a
    TA24a = CalcularTA24a(Fyd)
Elseif X >= h Then 'Usar esta en el intervalo del dominio 5
    TA25 = CalcularTA25(Fyd)
End If
If X > 0 And X < xc Then 'Usar esta en el intervalo del dominio 2
    TA12 = CalcularTA12(Fyd)
Elseif X >= xc And X < xl Then 'Usar esta en el intervalo del dominio 3
    TA13 = CalcularTA13(Fyd)
Elseif X >= xl And X < d Then 'Usar esta en el intervalo del dominio 4
    TA14 = CalcularTA14(EA14, Es)
Elseif X >= d And X < h Then 'Usar esta en el intervalo del dominio 4a
    TA14a = CalcularTA14a(EA14a, Es)
Elseif X >= h Then 'Usar esta en el intervalo del dominio 5
    TA15 = CalcularTA15(EA15, Es)
End If
If X > 0 And X < xc Then 'Usar esta en el intervalo del dominio 2
    NdporeA12 = CalcularNdporeA12(TA22, A, d, r, Fcd, b, Yfc)
Elseif X >= xc And X < xl Then 'Usar esta en el intervalo del dominio 3
    NdporeA13 = CalcularNdporeA13(TA23, A, d, r, Fcd, b, Yfc)
Elseif X >= xl And X < d Then 'Usar esta en el intervalo del dominio 4
    NdporeA14 = CalcularNdporeA14(TA24, A, d, r, Fcd, b, Yfc)
Elseif X >= d And X < h Then 'Usar esta en el intervalo del dominio 4a
    NdporeA24a = CalcularNdporeA24a(TA14a, A, d, r, Fcd, b, Yfc)
Elseif X >= h Then 'Usar esta en el intervalo del dominio 5
    NdporeA25 = CalcularNdporeA25(TA15, A, d, r, Fcd, b, Ycc)
End If
If X > 0 And X < xc Then 'Usar esta en el intervalo del dominio 2
    Nd2 = CalcularNd2(TA12, A, TA22, Fcd, b, Yfc)
    Worksheets("Hoja6").Range("a504").End(xlUp).Offset(1, 0).Value = Nd2
Elseif X >= xc And X < xl Then 'Usar esta en el intervalo del dominio 3
    Nd3 = CalcularNd3(TA13, A, TA23, Fcd, b, Yfc)
```




```
Worksheets("Hoja6").Range("a504").End(xlUp).Offset(1, 0).Value = Nd3
Elseif X >= xl And X < d Then 'Usar esta en el intervalo del dominio 4
    Nd4 = CalcularNd4(TA14, A, TA24, Fcd, b, Yfc)
    Worksheets("Hoja6").Range("a504").End(xlUp).Offset(1, 0).Value = Nd4
Elseif X >= d And X < h Then 'Usar esta en el intervalo del dominio 4a
    Nd4a = CalcularNd4a(TA14a, A, TA24a, Fcd, b, Yfc)
    Worksheets("Hoja6").Range("a504").End(xlUp).Offset(1, 0).Value = Nd4a
Elseif X >= h Then 'Usar esta en el intervalo del dominio 5
    Nd5 = CalcularNd5(TA15, A, TA25, Fcd, b, Ycc)
    Worksheets("Hoja6").Range("a504").End(xlUp).Offset(1, 0).Value = Nd5
End If
If X > 0 And X < xc Then 'Usar esta en el intervalo del dominio 2
    eA2 = CalcularA2(NdporeA12, Nd2)
    Elseif X >= xc And X < xl Then 'Usar esta en el intervalo del dominio 3
        eA3 = CalcularA3(NdporeA13, Nd3)
    Elseif X >= xl And X < d Then 'Usar esta en el intervalo del dominio 4
        eA4 = CalcularA4(NdporeA14, Nd4)
    Elseif X >= d And X < h Then 'Usar esta en el intervalo del dominio 4a
        eA4a = CalcularA4a(NdporeA24a, Nd4a)
    Elseif X >= h Then 'Usar esta en el intervalo del dominio 5
        eA5 = CalcularA5(NdporeA25, Nd5)
End If
If X > 0 And X < xc Then 'Usar esta en el intervalo del dominio 2
    ee2 = CalcularA2(eA2, h, r)
    Elseif X >= xc And X < xl Then 'Usar esta en el intervalo del dominio 3
        ee3 = CalcularA3(eA3, h, r)
    Elseif X >= xl And X < d Then 'Usar esta en el intervalo del dominio 4
        ee4 = CalcularA4(eA4, h, r)
    Elseif X >= d And X < h Then 'Usar esta en el intervalo del dominio 4a
        ee4a = CalcularA4a(eA4a, h, r)

    Elseif X >= h Then 'Usar esta en el intervalo del dominio 5
        ee5 = CalcularA5(eA5, h, r)
End If
If X > 0 And X < xc Then 'Usar esta en el intervalo del dominio 2
    Md2 = CalcularMd2(Nd2, ee2)
    Md2 = Round(Md2, 2)
```



```
Worksheets("Hoja6").Range("b504").End(xlUp).Offset(1, 0).Value = Md2
Elseif X >= xc And X < xl Then 'Usar esta en el intervalo del dominio 3
    Md3 = CalcularMd3(Nd3, ee3)
    Md3 = Round(Md3, 2)
    Worksheets("Hoja6").Range("b504").End(xlUp).Offset(1, 0).Value = Md3
Elseif X >= xl And X < d Then 'Usar esta en el intervalo del dominio 4
    Md4 = CalcularMd4(Nd4, ee4)
    Md4 = Round(Md4, 2)
    Worksheets("Hoja6").Range("b504").End(xlUp).Offset(1, 0).Value = Md4
Elseif X >= d And X < h Then 'Usar esta en el intervalo del dominio 4a
    Md4a = CalcularMd4a(Nd4a, ee4a)
    Md4 = Round(Md4a, 2)
    Worksheets("Hoja6").Range("b504").End(xlUp).Offset(1, 0).Value = Md4a
Elseif X >= h Then 'Usar esta en el intervalo del dominio 5
    Md5 = CalcularMd5(Nd5, ee5)
    Md5 = Round(Md5, 2)
    Worksheets("Hoja6").Range("b504").End(xlUp).Offset(1, 0).Value = Md5
End If

Next X
MsgBox "¡El proceso ha sido completado con éxito!"
Diagrama_CommandButton.Enabled = True
End Sub

'INDICAR QUE SOLO SE ACEPTEN VALORES NUMÉRICOS (PARA EVITAR ERRORES DE
EJECUCIÓN):
    'Los ascii que corresponden a los números son los siguientes
    '0 le corresponde el 48, 1 = 49, 2 = 50, 3 = 51, 4 = 52, 5 = 53,
    '6 = 54, 7= 55, 8 = 56, 9= 57. Se crea un código que condicione dichas teclas
    'en cada entrada que se quiera hacer.
Private Sub Ancho_TextBox_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
    If KeyAscii = 46 Or KeyAscii >= 48 And KeyAscii <= 57 Then
        KeyAscii = KeyAscii
    Else
        KeyAscii = 0
    End If
End Sub
Private Sub Canto_TextBox_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
```



```
        If KeyAscii = 46 Or KeyAscii >= 48 And KeyAscii <= 57 Then
            KeyAscii = KeyAscii
        Else
            KeyAscii = 0
        EndIf
    End Sub
Private Sub Recubrimiento_TextBox_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
        If KeyAscii = 46 Or KeyAscii >= 48 And KeyAscii <= 57 Then
            KeyAscii = KeyAscii
        Else
            KeyAscii = 0
        End If
    End Sub
Private Sub Armadura_TextBox_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
    If KeyAscii = 46 Or KeyAscii >= 48 And KeyAscii <= 57 Then
        KeyAscii = KeyAscii
    Else
        KeyAscii = 0
    End If
End Sub
Private Sub Fck_ComboBox_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
    If KeyAscii = 46 Or KeyAscii >= 48 And KeyAscii <= 57 Then
        KeyAscii = KeyAscii
    Else
        KeyAscii = 0
    End If
End Sub
Private Sub Yc_TextBox_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
    If KeyAscii = 46 Or KeyAscii >= 48 And KeyAscii <= 57 Then
        KeyAscii = KeyAscii
    Else
        KeyAscii = 0
    End If
End Sub
Private Sub Fyk_ComboBox_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
    If KeyAscii = 46 Or KeyAscii >= 48 And KeyAscii <= 57 Then
```



```
    KeyAscii = KeyAscii
Else
    KeyAscii = 0
End If
End Sub

Private Sub Ys_TextBox_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
If KeyAscii = 46 Or KeyAscii >= 48 And KeyAscii <= 57 Then
    KeyAscii = KeyAscii
Else
    KeyAscii = 0
End If
End Sub

Private Sub Young_TextBox_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
If KeyAscii = 46 Or KeyAscii >= 48 And KeyAscii <= 57 Then
    KeyAscii = KeyAscii
Else
    KeyAscii = 0
End If
End Sub

Private Sub X_TextBox_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
If KeyAscii = 46 Or KeyAscii >= 48 And KeyAscii <= 57 Then
    KeyAscii = KeyAscii
Else
    KeyAscii = 0
End If
End Sub
```



2.2.6.3. Código: Userform:

Este formulario tan solo se utilizó para cargar una foto gif

```
Private Sub UserForm_Initialize()  
  
IMAGEVAYA.Picture = LoadPicture(ActiveWorkbook.Path & "temporal.gif")  
  
End Sub
```

2.2.6.4. Módulo: Funciones_No_Predefinidas:

En este módulo se picó el código necesario para crear las funciones que se necesitaban y que no venían por defecto en Excel.

"CONTENIDO DEL MÓDULO: En este módulo voy a organizar y almacenar, por un lado las "declaraciones de las variables" que voy a usar en las funciones no predefinidas que debo construir, para poder realizar los cálculos que necesito, y por otro lado los procedimientos para dichas "funciones no predefinidas"(Procedimientos Function).

"A) DECLARACIÓN DE VARIABLES: Al declarar variables lo que estoy haciendo es guardar un espacio en la memoria Ram del ordenador un dato que puede variar su valor a lo largo del proyecto y que será necesario que el ordenador conozca su valor a la hora de utilizar la variable en los procedimientos. En VBA para declarar una variable se utiliza la sintaxis: Dim NombreDeLaVariable As TipodeDatoPrimitivoQueAlmacena si quiero que trabaje en el ámbito local (a nivel de módulo o de procedimiento) y sustituyo Dim por Public si la quiero disponible para todo el proyecto.

Option Explicit 'Obliga a declarar todas las variables que aparecen en el módulo (muy útil para no dejarme ninguna sin declarar, ya que voy a manejar bastantes y podría errarse.

Public d As Single

Public Dom As Byte 'Dominio de deformación

Public A As Variant ' Armadura simétrica en (mm²)

Public b As Variant 'Ancho de la viga de hormigón (mm)

Public h As Variant 'Canto de la viga de hormigón (mm)

Public r As Variant 'Recubrimiento mecánico de la viga de hormigón (mm)

Public X As Variant 'Profundidad de la línea neutra (mm)

Public Fck As Variant 'Resistencia característica del hormigón (N/mm²).

Public Fyd As Variant

Public Fcd As Variant



Public Fyk As Variant 'Límite elástico característico del acero (N/mm2)

Public Yc As Variant 'Coeficiente de seguridad del hormigón (escojo el tipo de dato byte pues es de los posibles el que menos memoria reserva. Como no va a cambiar su valor lo de claro como constante

Public Ys As Variant 'Coeficiente de seguridad del acero de la armadura (escojo el tipo de dato byte pues es de los posibles el que menos memoria reserva. Como no va a cambiar su valor lo declaro como constante

Public Ycc As Single

Public Yfc As Single

Public Es As Variant 'Módulo de Young o Módulo de elasticidad del acero Es (N/mm2)(escojo el tipo de dato byte pues es de los posibles el que menos memoria reserva). Como no va a cambiar su valor lo de claro como constante.

Public xc As Single

Public xl As Single

Public EA22 As Single

Public EA23 As Single

Public EA24 As Single

Public EA24a As Single

Public EA25 As Single

Public EA12 As Single

Public EA13 As Single

Public EA14 As Single

Public EA14a As Single

Public EA15 As Single

Public TA22 As Single

Public TA23 As Single

Public TA24 As Single

Public TA24a As Single

Public TA25 As Single

Public TA12 As Single

Public TA13 As Single

Public TA14 As Single

Public TA14a As Single

Public TA15 As Single

Public NdporeA12 As Single

Public NdporeA13 As Single

Public NdporeA14 As Single



Public NdporeA24a As Single

Public NdporeA25 As Single

Public Nd2 As Single

Public Nd3 As Single

Public Nd4 As Single

Public Nd4a As Single

Public Nd5 As Single

Public eA2 As Single

Public eA3 As Single

Public eA4 As Single

Public eA4a As Single

Public eA5 As Single

Public ee2 As Single

Public ee3 As Single

Public ee4 As Single

Public ee4a As Single

Public ee5 As Single

Public Md2 As Single

Public Md3 As Single

Public Md4 As Single

Public Md4a As Single

Public Md5 As Single

'B) PROCEDIMIENTOS FUNCTION (FUNCIONES NO PREDEFINIDAS):Una función definida por el usuario devuelve un resultado no suelen afectar a su entorno como suelen hacer las subrutinas.Tienen las siguientes estructuras básicas: Function NombreProcedimiento (arg1,arg2,..)

[sentencias VBA]

NombreProcedimiento= ValorDevuelto

End Function

'B1. Función para el cálculo del canto útil en mm:

Function Calculard(h As Variant, r As Variant) As Single

Calculard = h - r

End Function

'B2. Función para el cálculo de X crítica en mm:

Function XCrítica(d As Single) As Single

XCrítica = 0.259 * d

End Function



```
'B3. Función para el cálculo de x límite en mm:
Function XLímite(d As Single) As Single
    XLímite = (3.5 * d) / (3.5 + 1.74)
End Function

'B4. Función para el cálculo de Yfc (Dominio 2,3,4, y 4a) en mm:
Function CalcularYfc(X As Variant) As Single
    CalcularYfc = 0.8 * X
End Function

'B5.Función para el cálculo de Ycc (Dominio 5) en mm:
Function CalcularYcc(h As Variant, X As Variant) As Single
    CalcularYcc = h - ((0.2 * h ^ 2) / X)
End Function

'B6.Función para el cálculo de Fcd (Resistencia de cálculo del hormigón (N/mm2)):
Function CalcularFcd(Fck As Variant, Yc As Variant) As Double
    CalcularFcd = Fck / Yc
End Function

'B5.Función para el cálculo de Fyd (Resistencia de cálculo del acero (N/mm2)):
Function CalcularFyd(Fyk As Variant, Ys As Variant) As Double
    CalcularFyd = Fyk / Ys
End Function

'B6.Función para el cálculo de EA22 (%)(Deformación en A2 para el dominio 2):
Function CalcularEA22(X As Variant, r As Variant, d As Single) As Single
    CalcularEA22 = ((10 * (X - r)) / (d - X)) * 0.001
End Function

'B7.Función para el cálculo de EA23 (‰)en tantos por uno (Deformación en A2 para el
dominio 3):
Function CalcularEA23(X As Variant, r As Variant) As Single
    CalcularEA23 = ((3.5 * (X - r)) / X) * 0.001
End Function

'B8.Función para el cálculo de EA24 (‰) en tantos por uno (Deformación en A2 para el
dominio 4):
Function CalcularEA24(X As Variant, r As Variant) As Single
    CalcularEA24 = ((3.5 * (X - r)) / X) * 0.001
End Function

'B9. Función para el cálculo de EA24a en tantos por uno (Deformación en A2 para el
dominio 4a)
Function CalcularEA24a(X As Variant, r As Variant) As Single
```




```
        CalcularEA24a = ((3.5 * (X - r)) / X) * 0.001
End Function
'B10. Función para el cálculo de EA25 (‰) en tantos por uno (Deformación en A2 para el dominio 5 en tantos por mil):
Function CalcularEA25(X As Variant, r As Variant, h As Variant) As Single
        CalcularEA25 = ((2 * (X - r)) / (X - ((3 / 7) * h))) * 0.001
End Function
'B11. Función para el cálculo de EA12 (‰) en tantos por uno (Deformación en A1 para el dominio 2)
Function CalcularEA12() As Single
        CalcularEA12 = 10 * 0.001
End Function
'B12. Función para el cálculo de EA13 (‰) en tantos por uno (Deformación en A1 para el dominio 3)
Function CalcularEA13(d As Single, X As Variant) As Variant
        ' xc = XCrítica(d)
        ' xl = XLímite(d)
        ' Ys = Ys_TextBox
        ' Yc = Yc_TextBox
        ' Fck = Fck_ComboBox
        ' Fyk = Fyk_ComboBox
        ' Es = Young_TextBox.Value
        CalcularEA13 = ((3.5 * (d - X)) / X) * 0.001
End Function
'B13. Función para el cálculo de EA14 (‰) en tantos por uno (Deformación en A1 para el dominio 4)
Function CalcularEA14(d As Single, X As Variant) As Single
        CalcularEA14 = ((3.5 * (d - X)) / X) * 0.001
End Function
'B14. Función para el cálculo de EA14a(‰) en tantos por uno (Deformación en A1 para el dominio 4a)
Function CalcularEA14a(X As Variant, d As Single) As Single
        CalcularEA14a = ((3.5 * (X - d)) / X) * 0.001
End Function
'B15. Función para el cálculo de EA15(‰) en tantos por uno (Deformación en A1 para el dominio 5)
Function CalcularEA15(X As Variant, d As Single, h As Variant) As Single
```



```
CalcularEA15 = ((2 * (X - d)) / (X - ((3 / 7) * h))) * 0.001
```

```
End Function
```

'B16. Función para el cálculo de TA22 (Tensión en la armadura A2 para el dominio 2 (N/mm²)):

```
Function CalcularTA22(EA22 As Single, Es As Variant) As Single
```

```
CalcularTA22 = EA22 * Es
```

```
End Function
```

'B17. Función para el cálculo de TA23 (Tensión en la armadura A2 para el dominio 3 (N/mm²)):

```
Function CalcularTA23(EA13, Es) As Single
```

```
CalcularTA23 = EA13 * Es
```

```
End Function
```

'B18. Función para el cálculo de TA24 (Tensión en la armadura A2 para el dominio 4 (N/mm²)):

```
Function CalcularTA24(Fyd) As Single
```

```
CalcularTA24 = Fyd
```

```
End Function
```

'B19. Función para el cálculo de TA24a (Tensión en la armadura A2 para el dominio 4a (N/mm²)):

```
Function CalcularTA24a(Fyd) As Single
```

```
CalcularTA24a = Fyd
```

```
End Function
```

'B20. Función para el cálculo de TA25 (Tensión en la armadura A2 para el dominio 5 (N/mm²)):

```
Function CalcularTA25(Fyd) As Single
```

```
CalcularTA25 = Fyd
```

```
End Function
```

'B21. Función para el cálculo de TA12 (Tensión en la armadura A1 para el dominio 2 (N/mm²)):

```
Function CalcularTA12(Fyd) As Single
```

```
CalcularTA12 = Fyd
```

```
End Function
```

'B22. Función para el cálculo de TA13 (Tensión en la armadura A1 para el dominio 3 (N/mm²)):

```
Function CalcularTA13(Fyd) As Single
```

```
CalcularTA13 = Fyd
```

```
End Function
```



```
'B23. Función para el cálculo de TA14 (Tensión en la armadura A1 para el dominio 4
(N/mm2)):
Function CalcularTA14(EA14 As Single, Es As Variant) As Single
    CalcularTA14 = EA14 * Es
End Function

'B24. Función para el cálculo de TA14a (Tensión en la armadura A1 para el dominio 4a
(N/mm2)):
Function CalcularTA14a(EA14a As Single, Es As Variant) As Single
    CalcularTA14a = EA14a * Es
End Function

'B25. Función para el cálculo de TA15 (Tensión en la armadura A1 para el dominio 5
(N/mm2)):
Function CalcularTA15(EA15 As Single, Es As Variant) As Single
    CalcularTA15 = EA15 * Es
End Function

'B26. Función para el cálculo de NdporeA1 dominio2:
Function CalcularNdporeA12(TA22 As Single, A As Variant, d As Single, r As Variant,
Fcd As Variant, b As Variant, Yfc As Single) As Single
CalcularNdporeA12 = ((TA22 * A * (d - r)) + (Fcd * b * d * Yfc) - (0.5 * Fcd * Yfc ^ 2)) *
0.001
End Function

'B27. Función para el cálculo de NdporeA1 dominio3:
Function CalcularNdporeA13(TA23 As Single, A As Variant, d As Single, r As Variant,
    Fcd As Variant, b As Variant, Y As Single) As Single
CalcularNdporeA13 = ((TA23 * A * (d - r)) + (Fcd * b * d * Y) - (0.5 * Fcd * b * Y ^ 2)) *
0.001
End Function

'B28. Función para el cálculo de NdporeA1 dominio4:
Function CalcularNdporeA14(TA24 As Single, A As Variant, d As Single, r As Variant,
Fcd As Variant, b As Variant, Y As Single) As Single
CalcularNdporeA14 = ((TA24 * A * (d - r)) + (Fcd * b * d * Y) - (0.5 * Fcd * b * Y ^ 2)) *
0.001
End Function

'B29. Función para el cálculo de NdporeA2 dominio4a:
Function CalcularNdporeA24a(TA14a As Single, A As Variant, d As Single, r As
Variant, Fcd As Variant, b As Variant, Y As Single) As Single
```



```
CalcularNdporeA24a = ((TA14a * A * (d - r)) - (Fcd * b * r * Y) + (0.5 * Fcd * b * Y ^ 2)) *  
0.001  
End Function  
'B29. Función para el cálculo de NdporeA2 dominio5:  
Function CalcularNdporeA25(TA15 As Single, A As Variant, d As Single, r As Variant,  
Fcd As Variant, b As Variant, Y As Single) As Single  
CalcularNdporeA25 = ((TA15 * A * (d - r)) - (Fcd * b * r * Y) + (0.5 * Fcd * b * Y ^ 2)) *  
0.001  
End Function  
'B30. Función para el cálculo de Nd dominio 2:  
Function CalcularNd2(TA12 As Single, A As Variant, TA22 As Single, Fcd As Variant, b  
As Variant, Y As Single) As Single  
CalcularNd2 = (-TA12 * A) + (TA22 * A) + (Fcd * b * Y)  
End Function  
'B31. Función para el cálculo de Nd dominio 3:  
Function CalcularNd3(TA13 As Single, A As Variant, TA23 As Single, Fcd As Variant, b  
As Variant, Y As Single) As Single  
CalcularNd3 = (-TA13 * A) + (TA23 * A) + (Fcd * b * Y)  
End Function  
'B32. Función para el cálculo de Nd dominio 4:  
Function CalcularNd4(TA14 As Single, A As Variant, TA24 As Single, Fcd As Variant, b  
As Variant, Y As Single) As Single  
CalcularNd4 = (-TA14 * A) + (TA24 * A) + (Fcd * b * Y)  
End Function  
'B33. Función para el cálculo de Nd dominio 4a:  
Function CalcularNd4a(TA14a As Single, A As Variant, TA24a As Single, Fcd As  
Variant, b As Variant, Y As Single) As Single  
CalcularNd4a = (TA14a * A) + (TA24a * A) + (Fcd * b * Y)  
End Function  
'B34. Función para el cálculo de Nd dominio 5:  
Function CalcularNd5(TA15 As Single, A As Variant, TA25 As Single, Fcd As Variant, b  
As Variant, Y As Single) As Single  
CalcularNd5 = (TA15 * A) + (TA25 * A) + (Fcd * b * Y)  
End Function  
'B35. Función para el cálculo de eA del dominio 2:  
Function CalcularA2(NdporeA12 As Single, Nd2 As Single) As Single  
CalcularA2 = (NdporeA12 / Nd2)
```



End Function

'B36. Función para el cálculo de eA del dominio 3:

Function CalcularA3(NdporeA13 As Single, Nd3 As Single) As Single

CalcularA3 = (NdporeA13 / Nd3)

End Function

'B36. Función para el cálculo de eA del dominio 4:

Function CalcularA4(NdporeA14 As Single, Nd4 As Single) As Single

CalcularA4 = (NdporeA14 / Nd4)

End Function

'B36. Función para el cálculo de eA del dominio 4a:

Function CalcularA4a(NdporeA24a, Nd4a) As Single

CalcularA4a = (NdporeA24a / Nd4a)

End Function

'B37. Función para el cálculo de eA del dominio 5:

Function CalcularA5(NdporeA25 As Single, Nd5 As Single) As Single

CalcularA5 = (NdporeA25 / Nd5)

End Function

'B38. Función para el cálculo de ee del dominio 2:

Function CalcularAee2(eA2 As Single, h As Variant, r As Variant) As Single

CalcularAee2 = eA2 - ((h / 2) - r)

End Function

'B39. Función para el cálculo de ee del dominio 3:

Function CalcularAee3(eA3 As Single, h As Variant, r As Variant) As Single

CalcularAee3 = eA3 - ((h / 2) - r)

End Function

'B40. Función para el cálculo de ee del dominio 4:

Function CalcularAee4(eA4 As Single, h As Variant, r As Variant) As Single

CalcularAee4 = eA4 - ((h / 2) - r)

End Function

'B41. Función para el cálculo de ee del dominio 4a:

Function CalcularAee4a(eA4a As Single, h As Variant, r As Variant) As Single

CalcularAee4a = -eA4a + ((h / 2) - r)

End Function

'B42. Función para el cálculo de ee del dominio 5:

Function CalcularAee5(eA5 As Single, h As Variant, r As Variant) As Single

CalcularAee5 = -eA5 + ((h / 2) - r)

End Function



```
'B43. Función para el cálculo de Md del dominio 2:  
Function CalcularMd2(N2 As Single, ee2 As Single)  
CalcularMd2 = Nd2 * ee2  
End Function  
  
'B44. Función para el cálculo de Md del dominio 3:  
Function CalcularMd3(Nd3 As Single, ee3 As Single)  
CalcularMd3 = Nd3 * ee3  
End Function  
  
'B45. Función para el cálculo de Md del dominio 4:  
Function CalcularMd4(Nd4 As Single, ee4 As Single)  
CalcularMd4 = Nd4 * ee4  
End Function  
  
'B46. Función para el cálculo de Md del dominio 4a:  
Function CalcularMd4a(Nd4a As Single, ee4a As Single)  
CalcularMd4a = Nd4a * ee4a  
End Function  
  
'B47. Función para el cálculo de Md del dominio 5:  
Function CalcularMd5(Nd5 As Single, ee5 As Single)  
CalcularMd5 = Nd5 * ee5  
End Function
```

2.2.7. Se realiza una fase a prueba de fallos

Una vez programada la aplicación comenzó una fase de “puesta a prueba”, en la que se sometió a la aplicación a uso continuo intentando encontrar algún tipo de fallo que pud ocurrir. De hecho este proceso es un proceso dinámico que va sucediéndose también a lo largo del proceso de diseño y programación del mismo.

2.2.8. Depuración del código y mejora

Una vez terminada la fase a prueba de fallos, ya se obtuvo una aplicación que funcionaba correctamente y que hacía todo lo que se esperaba de ella. Sin embargo se comenzó una fase en la que se intentó optimizar el código. Esta fase es importante, sobre todo en aplicaciones complejas que requieren gran cantidad de recursos del sistema, ya que si se logra una buena optimización puede suponer una mayor velocidad del programa y un menor consumo de los recursos internos del ordenador.



3. Resultados

3.1. Estructura del software

El software consta de 2 partes fundamentalmente. Una primera parte creada sin el uso de programación (Hoja de cálculo de Excel) y por otro lado una segunda parte que si ha sido creada mediante programación en VBA.

a) La hoja de cálculo de Excel:

Esta parte no se creó mediante programación en VBA, sino manualmente. El objetivo para el que se realizó, como ya hemos comentado anteriormente, fue el de ver las limitaciones del programa a la hora de resolver el problema que pretendíamos, por la forma convencional.

Esta parte consiste en 5 hojas de cálculo (Hoja 1, 2, 3, 4, y 5), para los dominios 2, 3, 4, 4a y 5 respectivamente; con 24 columnas cada una, (ver figura 9). En dicha parte, con el uso de las fórmulas predefinidas de Excel y las introducidas, se consiguió resolver la secuencia de cálculo planteada en el punto 2.2.3 Secuencia de cálculo de este trabajo.

SECUENCIA DE CÁLCULO									
A ₁ =A _x (mm ²)	Dominio	X (mm)	Y (mm)	ε _{Ax}	σ _x (N/mm ²)	ε _{Ax}	σ _x (N/mm ²)	ΣM _{Ax} =0 -> Nd*ε _{Ax} (kN.mm)	ΣF _x =0 -> N _k (kN)
452.4	2	0.1	0.08	-0.00111	-221.7	0.010	347.8	-31926.4	-257.2
452.4	2	0.2	0.16	-0.00111	-221.2	0.010	347.8	-31682.1	-256.5
452.4	2	0.3	0.24	-0.00110	-220.7	0.010	347.8	-31437.8	-255.8
452.4	2	0.4	0.32	-0.00110	-220.2	0.010	347.8	-31193.5	-255.1
452.4	2	0.5	0.4	-0.00110	-219.7	0.010	347.8	-30949.2	-254.4
452.4	2	0.6	0.48	-0.00110	-219.3	0.010	347.8	-30704.9	-253.7
452.4	2	0.7	0.56	-0.00109	-218.8	0.010	347.8	-30460.6	-253.0
452.4	2	0.8	0.64	-0.00109	-218.3	0.010	347.8	-30216.3	-252.3
452.4	2	0.9	0.72	-0.00109	-217.8	0.010	347.8	-29972.0	-251.6
452.4	2	1	0.8	-0.00109	-217.3	0.010	347.8	-29727.7	-250.8
452.4	2	1.1	0.88	-0.00108	-216.8	0.010	347.8	-29483.4	-250.1
452.4	2	1.2	0.96	-0.00108	-216.3	0.010	347.8	-29239.1	-249.4
452.4	2	1.3	1.04	-0.00108	-215.8	0.010	347.8	-28994.8	-248.7
452.4	2	1.4	1.12	-0.00108	-215.3	0.010	347.8	-28750.5	-248.0
452.4	2	1.5	1.2	-0.00107	-214.8	0.010	347.8	-28506.1	-247.3
452.4	2	1.6	1.28	-0.00107	-214.3	0.010	347.8	-28261.8	-246.6
452.4	2	1.7	1.36	-0.00107	-213.8	0.010	347.8	-28017.5	-245.9
452.4	2	1.8	1.44	-0.00107	-213.3	0.010	347.8	-27773.2	-245.2
452.4	2	1.9	1.52	-0.00106	-212.8	0.010	347.8	-27528.9	-244.5
452.4	2	2	1.6	-0.00106	-212.3	0.010	347.8	-27284.6	-243.8
452.4	2	2.1	1.68	-0.00106	-211.8	0.010	347.8	-27040.2	-243.1
452.4	2	2.2	1.76	-0.00106	-211.3	0.010	347.8	-26795.9	-242.4

Fig. 9 Vista de la Hoja 1 de la hoja de cálculo. Fuente: Elaboración propia



b) Parte creada a partir de VBA:

Esta es la parte del software que se ha programado por medio de VBA. Visualmente para el usuario, consta principalmente de 2 formularios (Userforms). Por un lado un primer formulario de seguridad (Ver figura 10), que solicita al usuario que introduzca un nombre y una contraseña para poder acceder a la aplicación; le da 3 oportunidades antes de cerrarse la aplicación. En segundo lugar, una vez introducidos correctamente los datos en el anterior formulario y solo si esto se cumple, el formulario de contraseña desaparece y se abre el segundo formulario y realmente importante, que es el formulario donde el usuario podrá introducir sus parámetros (datos característicos del caso particular que quiere resolver), es decir es una ventana que sirve para comunicar al usuario con la aplicación. En total la aplicación le solicita 10 parámetros distintos al usuario (ver figura 11) antes de poder realizar los cálculos (el botón calcular permanece inactivo hasta que todos estos datos han sido completados). Los datos son introducidos mediante cajas de texto (TextBox) y mediante combos (ComboBox). Los parámetros que necesita la aplicación son los vistos en la secuencia de cálculo. En la parte derecha del formulario, la aplicación devuelve resultados numéricos al usuario una vez realizados los cálculos (parte superior derecha) y también datos de tipo gráfico, estos últimos solo si son solicitados haciendo click en el botón generar gráfico (abajo derecha).

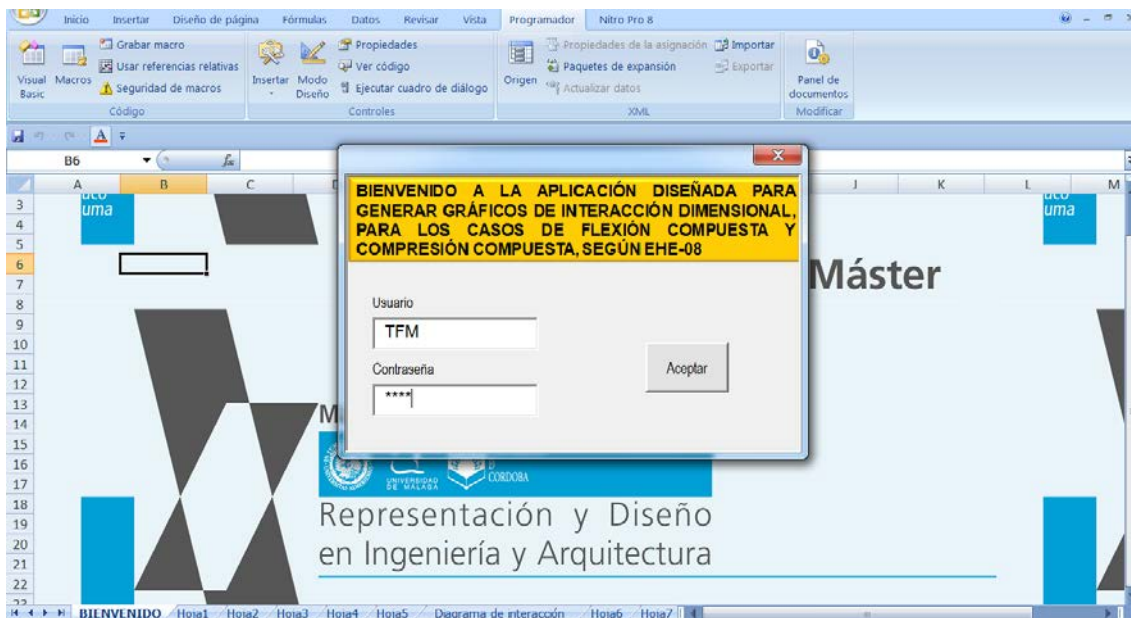


Fig. 10 Vista del Formulario que solicita las claves. Fuente: Elaboración propia



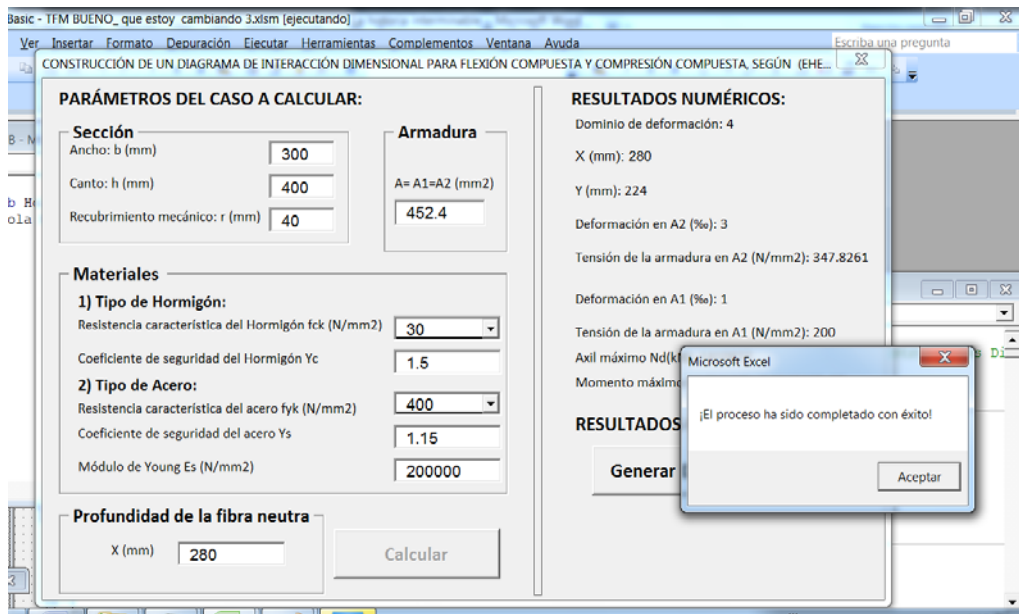


Fig. 11 Aspecto del formulario de solicitud de parámetros. Fuente: elaboración propia

La gráfica obtenida tanto en la opción de la hoja de Excel creada manualmente como en la aplicación creada mediante VBA es exactamente la misma, como no podía ser de otra forma. Obteniéndose el diagrama de interacción esperado, con todos sus datos (ver figura 12).

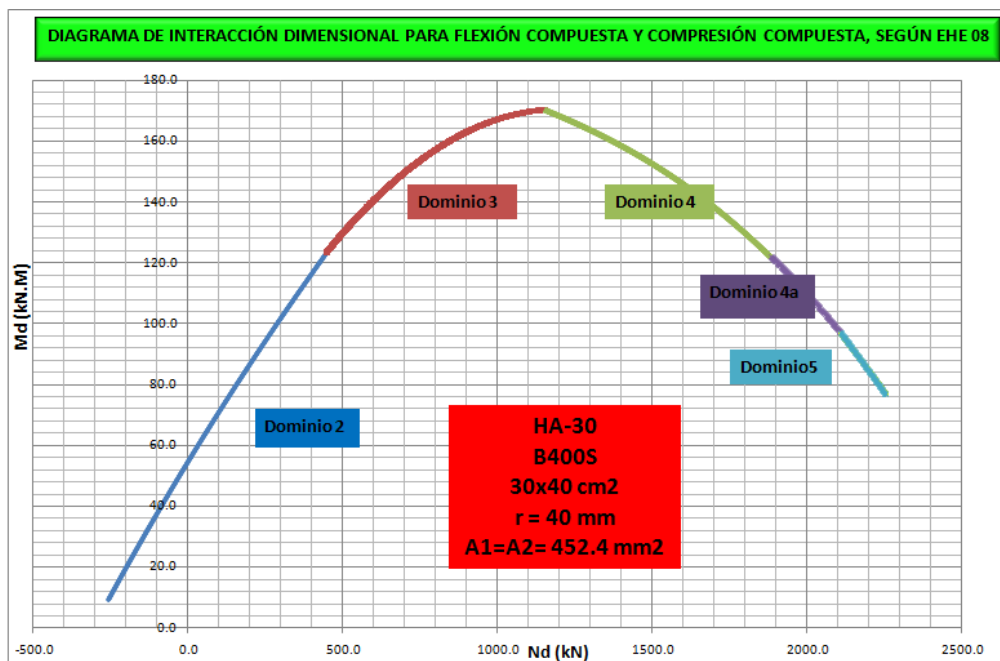


Fig. 12 Aspecto del diagrama de interacción generado. Fuente: elaboración propia

4. Discusión

a) La hoja de cálculo de Excel:

Esta parte presentó una limitación importante; como es que solo permitía resolver el enunciado del problema para una X determinada en cada instante y por hoja. Es decir, que para poder generar el diagrama, hubo que hacerse una hoja completa para cada dominio, con el trabajo que ello conllevó y luego pedirle a Excel que fuera uniendo los datos de cada hoja hasta formar el gráfico. Además la dificultad de moverse por la hoja cuando éstas son amplias y poseen muchas columnas y filas se hacía notable en estas hojas de cálculo de tamaño intermedio. Era difícil encontrar un resultado, pues visualmente se pierde la vista entre tanta celda, fila y columna, y se hacía lento y pesado trabajar con ellas. Por lo cual solventaba el problema del formato (que no se van a ensuciar, perder, romper a pie de obra, etc.) con respecto al sistema de diagramas de interacción en papel, pero poco más. Para nada puede decirse que resuelva todas las limitaciones de uso práctico que presentaba el sistema clásico en papel.

b) Parte creada a partir de VBA:

Esta parte demostró ser rápida y eficiente a la hora de realizar los cálculos. Por otro lado demostró que sirve para calcular cualquier ejemplo aunque cambien sus datos y no solo para uno como ocurría con los diagramas de papel y las hojas Excel. Además se hace muy fácil de usar y de ver los resultados gracias al diseño que otorgan los Userforms, con lo cual es muy fácil encontrar un resultado para el ejemplo que esté calculando en ese momento (pues lo reflejará en una pequeña ventana que tiene el usuario a la vista en todo momento). La aplicación cumple sobradamente con las expectativas e incluso va más allá, ya que la programación abre las puertas a introducir en la aplicación otras facilidades o análisis de datos que podrían ayudar aún más al usuario. De cara al futuro sería muy interesante seguir trabajando en esta línea de investigación, e intentar implantar la aplicación en un dispositivo más pequeño y manejable que un ordenador, como puede ser un móvil, lo cual abriría aún más el abanico de posibilidades y ventajas. El formato digital que se consigue en los datos de salida abre la puerta a poder trabajar con los datos obtenidos y otros programas distintos que Excel, lo que seguro que puede interesar.



5. Conclusiones

- La aplicación genera los diagramas de interacción para características variables de las secciones, e incluso para secciones distintas, mientras que los diagramas a papel son únicamente válidos para el ejemplo concreto para el que se dibujaron.
- La aplicación puede cubrir todos los casos posibles y los diagramas a papel no, pues es imposible dibujar las infinitas posibilidades.
- Mediante programación se puede crear y diseñar tanto como se quiera de forma que resulte infinitamente más intuitivas y fácil de manejar para el usuario los informes de usuario, que los diagramas a papel.
- La aplicación genera el diagrama de interacción en menos de 1 minuto, mientras que hacer un diagrama a mano para un caso que no se halle en bibliografía puede llevar mucho tiempo.
- El formato digital permite el fácil manejo a pie de obra de los diagramas de interacción, sin riesgo de que se estropeen.
- El formato digital permite el intercambio de archivos de forma rápida con otros puestos de trabajo o personas que se encuentren lejos de forma rápida vía internet, mientras que este tipo de comunicación en papel es mucho más difícil.



6. Referencias

Bibliografía

- Jiménez Montoya, P., 1971. Hormigón Armado, 1ª ed. Gustavo Gili, Vol.II., Barcelona.
- Jiménez Montoya, P., García Meseguer, A., Morán Cabré, F., 2001. Ábacos y diagramas para el cálculo de secciones. En: Hormigón Armado, 14ª ed. Gustavo Gili, Barcelona, pp. 629-781.
- McFedries, P., 2007. Excel 2007: Fórmulas y Funciones, 1ª ed. Anaya-Multimedia, Madrid.
- McFedries, P., 2007. Office 2007 con VBA, 1ª ed. Anaya-Multimedia, Madrid.

Webgrafía

- aulaClic, 2013. Curso de iniciación a la programación con Visual Basic. NET, <http://www.aulaclic.net/cursos/enlace.php?idp=514&id=24&texto=VisualBasic&clase=c>. (Accesible 29-sept-2013)
- Asurmendi, D., 2011. Programación en Excel 2010, VBA. Módulos y Procedimientos, sesión 01 [Archivo de video]. Recuperado de http://www.youtube.com/watch?feature=player_detailpage&v=XIGbc4X_bYA(Accesible 29-sept-2013)
- Asurmendi, D., 2011. Programación en Excel 2010, VBA. Módulos y Procedimientos, sesión 02 [Archivo de video]. Recuperado de http://www.youtube.com/watch?feature=player_detailpage&v=uYvedCDAuLc (Accesible 29-sept-2013)
- Asurmendi, D., 2011. Programación en Excel 2010, VBA. Primeros Pasos, sesión 01 [Archivo de video]. Recuperado de http://www.youtube.com/watch?feature=player_detailpage&v=xXFe3_YsXr0 (Accesible 29-sept-2013)
- Asurmendi, D., 2011. Programación en Excel 2010, VBA. Primeros Pasos, sesión 02 [Archivo de video]. Recuperado de http://www.youtube.com/watch?feature=player_detailpage&v=aD5n_alqpug(Accesible 29-sept-2013)
- Asurmendi, D., 2011. Programación en Excel 2010, VBA. Todo son objetos. Vídeo 1 de 1 [Archivo de video]. Recuperado de http://www.youtube.com/watch?feature=player_detailpage&v=E0nZxXUMJ5Y(Accesible 29-sept-2013)



- Gómez, J., 2012. VBA Excel. Introducción. Vídeo 1 [Archivo de video] Recuperado de http://www.youtube.com/watch?list=PLF79C9D65E1EC4807&v=6urgg0Eczho&feature=player_detailpage (Accesible 29-sept-2013)
- Gómez, J., 2012. VBA Excel. Vídeo 2 [Archivo de video] Recuperado de http://www.youtube.com/watch?feature=player_detailpage&list=PLF79C9D65E1EC4807&v=c-KP9hqsZO5 (Accesible 29-sept-2013)
- Gómez, J., 2012. VBA Excel. Vídeo 3 [Archivo de video] Recuperado de http://www.youtube.com/watch?feature=player_detailpage&v=VD1WOVrgfKs (Accesible 29-sept-2013)
- Gómez, J., 2012. VBA Excel. Vídeo 4 [Archivo de video] Recuperado de http://www.youtube.com/watch?feature=player_detailpage&v=NEQEzWtSwzk (Accesible 29-sept-2013)
- Gómez, J., 2012. VBA Excel. Ámbito de las variables Declaración de constantes Vídeo 5 [Archivo de video] Recuperado de http://www.youtube.com/watch?list=PLF79C9D65E1EC4807&v=TTUsELjfvIU&feature=player_detailpage (Accesible 29-sept-2013)
- Gómez, J., 2012. VBA Excel. Matrices de una dimensión Vídeo 6 [Archivo de video] Recuperado de http://www.youtube.com/watch?feature=player_detailpage&v=ibzYwq30CRw&list=PLF79C9D65E1EC4807 (Accesible 29-sept-2013)
- Gómez, J., 2012. VBA Excel. Matrices de dos dimensiones I Vídeo 7 [Archivo de video] Recuperado de http://www.youtube.com/watch?list=PLF79C9D65E1EC4807&v=t29X-z14j8M&feature=player_detailpage (Accesible 29-sept-2013)
- Gómez, J., 2012. VBA Excel. Variables objeto Vídeo 8 [Archivo de video] Recuperado de http://www.youtube.com/watch?list=PLF79C9D65E1EC4807&v=J_u2sj4HYp0&feature=player_detailpage (Accesible 29-sept-2013)
- Gómez, J., 2012. VBA Excel. Funciones predefinidas I Vídeo 9 [Archivo de video] Recuperado de http://www.youtube.com/watch?v=lx4FOiA3tJU&feature=player_detailpage&list=PLF79C9D65E1EC4807 (Accesible 29-sept-2013)
- Gómez, J., 2012. VBA Excel. Funciones predefinidas II Vídeo 10 [Archivo de video] Recuperado de http://www.youtube.com/watch?list=PLF79C9D65E1EC4807&v=-JZb1d0v11g&feature=player_detailpage (Accesible 29-sept-2013)
- Gómez, J., 2012. VBA Excel. Condicional If I Vídeo 11 [Archivo de video] Recuperado de http://www.youtube.com/watch?list=PLF79C9D65E1EC4807&feature=player_detailpage&v=1CblesPbSjA (Accesible 29-sept-2013)



- Gómez, J., 2012. VBA Excel. Condicional If II Vídeo 12 [Archivo de video] Recuperado de http://www.youtube.com/watch?feature=player_detailpage&list=PLF79C9D65E1EC4807&v=ZTVkWPE7Qzg (Accesible 29-sept-2013)
- Gómez, J., 2012. VBA Excel. Select Case Vídeo 13 [Archivo de video] Recuperado de http://www.youtube.com/watch?feature=player_detailpage&v=GNKmhWU7eps&list=PLF79C9D65E1EC4807 (Accesible 29-sept-2013)
- Gómez, J., 2012. VBA Excel. Bucle While Vídeo 14 [Archivo de video] Recuperado de http://www.youtube.com/watch?feature=player_detailpage&v=jSm8GoLPKMQ&list=PLF79C9D65E1EC4807 (Accesible 29-sept-2013)
- Gómez, J., 2012. VBA Excel. Bucle Do-Loop Vídeo 18 [Archivo de video] Recuperado de http://www.youtube.com/watch?v=2NGxN3Xvkjw&list=PLF79C9D65E1EC4807&feature=player_detailpage (Accesible 29-sept-2013)
- Gómez, J., 2012. VBA Excel. Trabajo con procedimientos I Vídeo 19 [Archivo de video] Recuperado de http://www.youtube.com/watch?list=PLF79C9D65E1EC4807&v=KhPtWP1OuIM&feature=player_detailpage (Accesible 29-sept-2013)
- Gómez, J., 2012. VBA Excel. Trabajo con procedimientos II Vídeo 20 [Archivo de video] Recuperado de http://www.youtube.com/watch?feature=player_detailpage&v=cPTPfpWp5zA&list=PLF79C9D65E1EC4807 (Accesible 29-sept-2013)
- Gómez, J., 2012. VBA Excel. Trabajos con procedimientos III Vídeo 21 [Archivo de video] Recuperado de http://www.youtube.com/watch?list=PLF79C9D65E1EC4807&v=L4JHI5LvXkQ&feature=player_detailpage (Accesible 29-sept-2013)
- Gómez, J., 2012. VBA Excel. Trabajo con procedimientos IV Vídeo 22 [Archivo de video] Recuperado de http://www.youtube.com/watch?list=PLF79C9D65E1EC4807&v=2sDk0WbrJ9A&feature=player_detailpage (Accesible 29-sept-2013)
- Gómez, J., 2012. VBA Excel. Procedimientos Function I Vídeo 28 [Archivo de video] Recuperado de http://www.youtube.com/watch?list=PLF79C9D65E1EC4807&v=qLpRY1AA8tk&feature=player_detailpage (Accesible 29-sept-2013)
- Gómez, J., 2012. VBA Excel. Procedimientos Function II Vídeo 29 [Archivo de video] Recuperado de http://www.youtube.com/watch?feature=player_detailpage&list=PLF79C9D65E1EC4807&v=12FkSC3HTs (Accesible 29-sept-2013)



- Gómez, J., 2012. VBA Excel. Técnicas I Trabajo rangos Vídeo 30 [Archivo de video] Recuperado de http://www.youtube.com/watch?feature=player_detailpage&v=hmLa5s2ozE0&list=PLF79C9D65E1EC4807(Accesible 29-sept-2013)
- Gómez, J., 2012. VBA Excel. Técnicas II Inserción de datos Vídeo 31 [Archivo de video] Recuperado de http://www.youtube.com/watch?list=PLF79C9D65E1EC4807&v=uZAI1w6xDh0&feature=player_detailpage (Accesible 29-sept-2013)
- Gómez, J., 2012. VBA Excel. Técnicas III Tipo datos y eliminación filas Vídeo 32 [Archivo de video] Recuperado de http://www.youtube.com/watch?v=xr7BwDRHVal&feature=player_detailpage&list=PLF79C9D65E1EC4807(Accesible 29-sept-2013)
- Gómez, J., 2012. VBA Excel. Formulario de Usuarios UserForms I Vídeo 34 [Archivo de video] Recuperado de http://www.youtube.com/watch?v=i_29gVPQis&feature=player_detailpage&list=PLF79C9D65E1EC4807 (Accesible 29-sept-2013)
- Gómez, J., 2012. VBA Excel. Formulario de Usuarios UserForms II Vídeo 35 [Archivo de video] Recuperado de http://www.youtube.com/watch?list=PLF79C9D65E1EC4807&v=cWzFqLdMdy8&feature=player_detailpage (Accesible 29-sept-2013)
- Gómez, J., 2012. VBA Excel. Formulario de Usuarios UserForms III Vídeo 36 [Archivo de video] Recuperado de http://www.youtube.com/watch?feature=player_detailpage&list=PLF79C9D65E1EC4807&v=mTo6PMpBd2c (Accesible 29-sept-2013)
- Gómez, J., 2012. VBA Excel. Eventos I Vídeo 37 [Archivo de video] Recuperado de http://www.youtube.com/watch?list=PLF79C9D65E1EC4807&feature=player_detailpage&v=Dqgm7hxbwWM(Accesible 29-sept-2013)
- Gómez, J., 2012. VBA Excel. Eventos II Vídeo 38 [Archivo de video] Recuperado de http://www.youtube.com/watch?list=PLF79C9D65E1EC4807&feature=player_detailpage&v=QdOWdLx08Rs (Accesible 29-sept-2013)
- Gómez, J., 2012. VBA Excel. Gráficos I Vídeo 39 [Archivo de video] Recuperado de http://www.youtube.com/watch?feature=player_detailpage&v=K33LMSRSSTc&list=PLF79C9D65E1EC4807 (Accesible 29-sept-2013)
- Gómez, J., 2012. VBA Excel. Gráficos II Vídeo 40 [Archivo de video] Recuperado de http://www.youtube.com/watch?list=PLF79C9D65E1EC4807&v=fytctejMVLl&feature=player_detailpage(Accesible 29-sept-2013)



- Ministerio de Fomento, 2008, EHE-08 Art. 42.1.2 Hipótesis básicas. Recuperado de <http://www.fomento.gob.es/NR/rdonlyres/DDB78AD3-25AB-40FE-A92F-8429F5EE21D7/37465/CAPITULOXborde.pdf> (Accesible 29-sept-2013)
- Ministerio de Fomento, 2008, EHE-08 Art. 42.1.3 Dominios de deformación. Recuperado de <http://www.fomento.gob.es/NR/rdonlyres/DDB78AD3-25AB-40FE-A92F-8429F5EE21D7/37465/CAPITULOXborde.pdf> (Accesible 29-sept-2013)
- Ministerio de Fomento, 2008, EHE-08 Art. 39.5.a Diagrama tensión-deformación de cálculo del hormigón. Diagrama parábola-rectángulo. Recuperado de <http://www.fomento.gob.es/NR/rdonlyres/64FF2359-E749-46C1-95D7-1F7B53B0E3FD/37463/CAPITULOVIIIborde.pdf> (Accesible 29-sept-2013)
- Ministerio de Fomento, 2008, EHE-08 Art. 38.4 Diagrama tensión- deformación de cálculo del acero en las armaduras pasivas. Recuperado de <http://www.fomento.es/NR/rdonlyres/64FF2359-E749-46C1-95D7-1F7B53B0E3FD/37463/CAPITULOVIIIborde.pdf> (Accesible 29-sept-2013)
- Ministerio de Fomento, 2008, EHE-08 Art. 15.3. Coeficientes parciales de seguridad de los materiales. Recuperado de <http://www.fomento.gob.es/NR/rdonlyres/B7AE915D-9A95-4B73-950F-74E6D8FD6BD4/37459/CAPITULOIVborde.pdf> (Accesible 29-sept-2013)
- Walkenbach, J., 2010. Excel 2010 Power Programming with VBA. Recuperado de <http://excelvba.ru/UsefulBooks/book9.pdf>. (Accesible 29-sept-2013)







Resumen:

Las estructuras de hormigón armado son elementos que están sujetos a complejos cálculos. La vía más utilizada para la simplificación de estos cálculos es la de los llamados diagramas de interacción. Sin embargo, éstos, aunque útiles, presentan ciertas limitaciones de tipo práctico. El objetivo de este trabajo fue solventar dichos problemas desarrollando para ello una aplicación informática apoyada en Excel mediante el lenguaje de programación Visual Basic For Applications y para la que seguimos, como base de cálculo, la normativa actual vigente en España, la Instrucción Española del Hormigón Estructural del año 2008. La aplicación se halla estructurada en formularios (Userforms) que comunican al usuario con el programa de forma que introduciendo los parámetros pertinentes para que la aplicación realice el cálculo requerido se obtengan los resultados que la aplicación le devuelve. La aplicación es capaz de realizar los cálculos para un caso práctico determinado, así como generar un diagrama de interacción general. Se ha obtenido así una aplicación cuyo formato atiende a las necesidades actuales facilitando su uso, a la vez que aumenta en versatilidad, rapidez y eficiencia, con respecto al empleo de diagramas con los formatos actuales.

