

TRABAJO ESPECIAL DE GRADO

PROGRAMA PARA EL ANALISIS, DISEÑO Y DIBUJO AUTOMATIZADO DE ESCALERAS DE CONCRETO ARMADO EN AMBIENTE DE TRABAJO CAD

Presentado ante la Ilustre
Universidad Central de
Venezuela para optar al Título
De Ingeniero Civil
Por el Br. Marín N. Robert M.

Caracas, Noviembre de 2003

TRABAJO ESPECIAL DE GRADO

PROGRAMA PARA EL ANALISIS, DISEÑO Y DIBUJO AUTOMATIZADO DE ESCALERAS DE CONCRETO ARMADO EN AMBIENTE DE TRABAJO CAD

TUTOR ACADEMICO: Prof. Tomás Osers

Presentado ante la ilustre
Universidad Central de
Venezuela para optar al Título
De Ingeniero Civil
Por el Br. Marín N. Robert M

Caracas, Noviembre de 2003

MARÍN N. ROBERT M.,

**PROGRAMA PARA EL ANALISIS, DISEÑO Y DIBUJO
AUTOMATIZADO DE ESCALERAS DE CONCRETO ARMADO EN
AMBIENTE DE TRABAJO CAD**

Tutor Académico: Prof. Tomás Osers.
Trabajo Especial de Grado. Caracas, U.C.V. Facultad de Ingeniería.
Escuela de Ingeniería Civil. Año 2003, 132 Pág.

Palabras Claves: Escaleras, EscalerasCAD, Concreto, VBA, AutoCAD®

Resumen. Se crea un programa que realiza el análisis, diseño y dibujo automatizado de escaleras de concreto armado en ambiente de trabajo CAD, específicamente en el entorno VBA (Visual Basic para Aplicaciones), que se encuentra dentro de AutoCAD®, con la finalidad de poder crear cualquier función, con las mismas herramientas de Visual Basic, dentro del programa. EscalerasCAD es el nombre de este programa, que hace el análisis y dibujo de tres escaleras de concreto armado, que son: escalera autoportante, escalera de un tramo con dos descansos, y escalera de dos tramos con tres descansos. Para realizar el dibujo de estas escaleras, se hizo uso, de las entidades gráficas de AutoCAD® (líneas, círculos, acotaciones...). Para realizar el cálculo de las solicitaciones necesarias en la determinación del refuerzo de acero, se basó en el método simplificado de Fuchsteiner, para la escalera autoportante, que sustituye la estructura de la escalera en una armadura espacial compuesta de barras lineales; y para las otras dos escaleras, la ejecución del cálculo como estructura plegada, que considera las aristas de la escalera como apoyos fijos. El refuerzo del acero se determina en base a la teoría clásica y de rotura. Todos estos métodos son llevados a los módulos del programa como procedimientos propios del mismo, para ser usados en todos los módulos creados en VBA. Se hace la recopilación de estos procedimientos en los módulos de formulario, para crear el entorno de la entrada de datos.

AGRADECIMIENTOS

Al Señor Todopoderoso, por estar siempre presente conmigo, en las buenas y en las malas, por la vida, por la salud, por todo lo que me ha pasado, que de seguro, fue lo mejor.

A mi familia, por cuidarme en los momentos más difíciles, por haberme tenido bastante paciencia, por apoyarme en todas las cosas que hago, en fin, a ellos le agradezco todo lo que han hecho por mí.

A mi hermana Jenny como agradecimiento especial, por ayudarme en todo, por haber sido la segunda persona que hizo este trabajo.

A Jorge Córdoba, por haber formado parte de la asistencia técnica del computador.

Al Prof. Tomás Osers, porque más que el Tutor de este trabajo, pude contar con él para cualquier problema, recomendación, duda, por tener paciencia, y por que este trabajo quedara lo mejor posible.

Al Prof. José Manuel Velásquez, por haberme ayudado en el cálculo del refuerzo, por haberme facilitado la bibliografía utilizada, y por darme algunas recomendaciones.

A todos mis compañeros, que con pequeños detalles marcaron grandes diferencias: Irina Bello, Luís Subero, Giovanni De Santis, Daniel Méndez, Álvaro Hernández, Sorelys Cabrera, Julio Du Bois, Felipe López.

INDICE

1	INTRODUCCION	4
2	OBJETIVOS	7
2.1	OBJETIVO GENERAL	7
2.2	OBJETIVOS ESPECIFICOS	7
3	MARCO TEORICO	8
3.1	INTRODUCCION	8
3.2	DEFINICIONES BASICAS	9
3.2.1	Programación.....	9
3.2.2	Programación orientada a objetos (OOP)	9
3.2.3	Objetos.....	10
3.2.3.1	Propiedades	10
3.2.3.2	Métodos y eventos	11
3.2.4	Visual Basic	13
3.2.5	Visual Basic para Aplicaciones (VBA)	13
3.2.6	Introducción a la tecnología AutoCAD® ActiveX.....	14
3.2.7	Objetos AutoCAD® ActiveX.....	14
3.2.8	Interface AutoCAD® Visual Basic For Applications(VBA).....	15
3.2.9	¿Cómo es implementado VBA en AutoCAD®?.....	15
3.3	INICIO DE UN PROGRAMA EN AutoCAD® VBA	16
3.3.1	Estructura de un proyecto en VB (VBA)	17
3.3.1.1	Módulos de formulario.....	17
3.3.1.2	Módulos de clase	17
3.3.1.3	Módulos estándar.....	17
3.3.1.4	Componentes	17
3.3.1.4.1	Controles ActiveX	17
3.3.1.4.2	Objetos insertables	18
3.3.1.4.3	Controles estándar	18
3.3.1.5	Estructura de un código en VBA.....	18
3.3.1.5.1	Declaración de variables	19
3.3.1.5.1.1	Declaración implícita	20
3.3.1.5.1.2	Declaración explícita	20
3.3.1.5.1.3	Declarar constantes	20
3.3.1.5.1.4	Tipos de Datos	23
3.3.1.5.1.5	Declarar variables con tipos de datos	23
3.3.1.5.1.6	Declarar matrices (de tamaño fijo)	25
3.3.1.5.1.7	Declarar variables de objeto.....	26
3.3.1.5.1.8	Asignar variables de objeto	26
3.3.1.5.2	Procedimientos	27
3.3.1.5.2.1	Procedimientos Sub	27
3.3.1.5.2.2	Procedimientos Function	28
3.3.1.5.3	Estructuras de control.....	29
3.3.1.5.3.1	Estructuras de decisión	29

3.3.1.5.3.1.1	If...Then	29
3.3.1.5.3.1.2	If...Then...Else	30
3.3.1.5.3.1.3	Select Case	30
3.3.1.5.3.2	Estructuras de bucle.....	31
3.3.1.5.3.2.1	Do...Loop.....	31
3.3.1.5.3.2.2	For...Next	33
3.3.1.5.3.2.3	For Each...Next	34
3.4	DESCRIPCION DE ENTIDADES INTERNAS (OBJETOS GRAFICOS DE AutoCAD®)	35
3.4.1	Introducción	35
3.4.2	Línea	37
3.4.3	Círculo	38
3.4.4	Polilínea	39
3.4.5	Dimensión	41
3.4.5.1	DimAligned Object (AcadDimAligned)	41
3.4.5.2	DimRotated Object (AcadDimRotated)	42
3.4.6	Texto	45
3.4.7	Línea auxiliar.....	47
3.4.8	Flecha (Leader).....	48
3.4.9	Sombreado (Hatch).....	50
3.4.10	Propiedades	52
3.4.10.1	Angle	52
3.4.10.2	Area	52
3.4.10.3	ArrowheadSize.....	53
3.4.10.4	ArrowheadType.....	53
3.4.10.5	AssociativeHatch	55
3.4.10.6	AttachmentPoint.....	55
3.4.10.7	BasePoint.....	57
3.4.10.8	Center	57
3.4.10.9	Closed	58
3.4.10.10	Color	58
3.4.10.11	Coordinate	59
3.4.10.12	EndPoint	60
3.4.10.13	Height.....	60
3.4.10.14	InsertionPoint.....	61
3.4.10.15	Layer	61
3.4.10.16	Length	62
3.4.10.17	Linetype	62
3.4.10.18	LinetypeScale	63
3.4.10.19	Measurement	64
3.4.10.20	SecondPoint.....	64
3.4.10.21	StartPoint	65
3.4.10.22	TextGap	65
3.4.10.23	VerticalTextPosition	65
3.4.11	Métodos.....	67
3.4.11.1	AppendOuterLoop.....	67
3.4.11.2	AppendVertex	67
3.4.11.3	Copy.....	68
3.4.11.4	Delete.....	68
3.4.11.5	IntersectWith	69
3.4.11.6	Move	70
3.4.11.7	Offset	70
3.4.11.8	Rotate	71
3.5	ESCALERAS	72
3.5.1	Especificaciones constructivas generales	72

3.5.2	Cálculo de escaleras de concreto armado	72
3.5.3	Escalera autoportante con descanso	75
3.6	CONSIDERACIONES PARA EL CÁLCULO DEL REFUERZO DE ACERO	78
4	METODOLOGIA.....	79
4.1	MÉTODOS DE CÁLCULO EN LA DETERMINACION DE LAS SOLICITACIONES DE LAS ESCALERAS.	79
4.1.1	Escalera autoportante.....	79
4.1.2	Escalera de un tramo con dos descansos y escalera de dos tramos con tres descansos ...	87
4.1.3	Cálculo del área de acero.....	89
4.2	INICIO DEL PROGRAMA.....	90
4.2.1	Inicio del código del programa	91
4.2.2	Creación de los procedimientos de dibujo.....	92
4.2.2.1	Escalera autoportante	92
4.2.2.1.1	DibujarConjuntoInferior	92
4.2.2.1.2	DibujarConjuntoSuperior	94
4.2.2.1.3	DibujarPlantaAutoportante.....	94
4.2.2.2	Para la escalera de un tramo con dos descansos	96
4.2.2.2.1	DibujarConjuntoDescansoEscaleraDescanso	96
4.2.2.2.2	DibujarPlantaDED.....	98
4.2.2.3	Escalera de dos tramos con tres descansos	100
4.2.2.3.1	DibujarConjuntoEscaleraDescansoEscalera	100
4.2.2.3.2	DibujarPlantaDEDED.....	104
4.2.3	Creación de los procedimientos de cálculo	105
4.2.4	Creación de los módulos de formularios.	108
4.2.4.1	FormaAceroL.....	109
4.2.4.2	FormaAceroT	111
4.2.4.3	FormaAcotacion	113
4.2.4.4	FormaApoyo.....	114
4.2.4.5	FormaAsDef	115
4.2.4.6	FormaAutoportante	116
4.2.4.7	FormaDED	119
4.2.4.8	FormaDEDED	120
4.2.4.9	FormaDescanso	121
4.2.4.10	FormaEscalera.....	123
4.2.4.11	FormaEspecificaciones.....	124
4.2.4.12	FormaOrientacion	125
5	CONCLUSIONES	126
6	RECOMENDACIONES.....	127
7	BIBLIOGRAFIA	128
8	ANEXOS.....	129

1 INTRODUCCION

Hoy en día el uso de los ordenadores ha logrado el ahorro de espacio, tiempo y dinero a la hora de realizar un trabajo que esté al alcance de los mismos; como presentaciones, informes, planos, cálculos, manipulación de base de datos, comunicaciones, entre muchas otras cosas, sin hablar de la capacidad de permitir el almacenamiento de información, bien sea en el mismo ordenador o en unidades especiales externas (diskettes, Zip, CD). Todas estas cosas pueden hacerse mediante la interacción hardware-software, que es el principio de un ordenador o computador.

El hardware es el soporte técnico del ordenador (parte física: monitor, teclado, mouse, CPU, etc.). El software es el soporte lógico del ordenador (parte operativa: programas, lenguajes, sistemas operativos, etc.). La parte operativa es el alma que le da vida al computador.

Los usuarios de los computadores se caracterizan por estar aptos para manipular programas que le permiten hacer las funciones que éstos deseen.

En algunas ocasiones se tiene la necesidad de crear componentes a la medida de las exigencias particulares, sean técnicos o lógicos; esto depende de la capacidad que se tenga para crear estas partes.

La parte más accesible para la mayoría de los usuarios (entre las dos partes anteriores) de computadores es la capacidad de crear programas, es por ello de la creación de lenguajes de programación para poder realizar estos programas. En el mercado existen diversos tipos de lenguajes que se adaptan a las necesidades que se requieran.

Sin embargo, muchos de los programas que existen actualmente, tienen incluidos accesos de lenguajes de programación que permiten crear funciones, rutinas, procedimientos dentro del entorno del mismo programa, es decir, especies de “pequeños lenguajes” que trabajan en el programa en cuestión. Esto es una

herramienta muy importante, porque permite el uso del programa tradicional, añadiendo todas aquellas funciones que puedan ser creadas para poder aplicarlas en el programa.

Uno de estos lenguajes de programación más común dentro de la mayoría de los programas es el conocido como Visual Basic para aplicaciones (VBA). Visual Basic para aplicaciones es un único entorno y lenguaje de escritura de comandos que los usuarios y los programadores pueden compartir entre sus escritorios de Windows. Visual Basic para aplicaciones se incluye en Microsoft Office y en otras aplicaciones de Microsoft. También se ha autorizado su uso a otros fabricantes de software y se incluye en una variada gama de productos, entre ellos está el sistema AutoCAD®.

El sistema AutoCAD® es un software destinado al diseño en dos y en tres dimensiones (2D y 3D, respectivamente). Su nombre surge de la unión de las siglas Auto, proveniente del nombre de la empresa que lo desarrolló (Autodesk, Inc. California, Estados Unidos de América) y CAD, siglas que significan Diseño Asistido por Computadora. Este sistema es usado por arquitectos, ingenieros, diseñadores gráficos, entre otros. Específicamente en el área de ingeniería civil, es usado en diversas disciplinas: estructuras, vialidad, topografía, geodesia...

En estas disciplinas es necesario manejar muchos datos, procesarlos, y presentarlos. Esto se logra; invirtiendo el menor tiempo, espacio, y dinero posible, a través de la automatización o programación de las funciones y procedimientos necesarios para la manipulación de estos datos, en los programas adecuados. En el sistema AutoCAD®, en la mayoría de los casos, se hacen: dibujos, planos a escala, presentaciones en 3D, cálculos, textos, entre otras aplicaciones. En este sistema estas aplicaciones pueden ser automatizadas a través de Visual Basic para Aplicaciones (VBA).

Es por ello que el principal objetivo de este trabajo es la creación de un sistema computacional que opere dentro del sistema AutoCAD®(Inside AutoCAD®), que realice el análisis, diseño y dibujo automatizado de escaleras de concreto

armado en ambiente de trabajo CAD). Se realizará un programa en VBA que realice el análisis y diseño de escaleras de concreto armado comunes. Específicamente en este trabajo se realizará el análisis de: escaleras autoportantes, escaleras con descansos en los extremos apoyados sobre vigas o sobre muros, y escaleras con dos tramos de escalones, tramo intermedio y descansos en los extremos con las mismas condiciones que la anterior. Se elaborarán módulos de graficación que operarán dentro del sistema AutoCAD®, y se creará un conjunto de detalles típicos de armados de estas escaleras.

2 OBJETIVOS

2.1 OBJETIVO GENERAL

Crear un sistema computacional que opere dentro del sistema AutoCAD® (Inside AutoCAD®) para realizar el ANALISIS, DISEÑO, Y DIBUJO AUTOMATIZADO DE ESCALERAS DE CONCRETO ARMADO EN AMBIENTE DE TRABAJO CAD

2.2 OBJETIVOS ESPECIFICOS

- a. Elaborar un programa en VBA que realice el Análisis y Diseño de Escaleras.
 - b. Elaborar módulos de graficación que operarán en ambiente AutoCAD®.
 - c. Crear un conjunto de dibujos estáticos, de detalles típicos de armado de escaleras.
-

3 MARCO TEORICO

3.1 INTRODUCCION

El marco teórico consta de tres partes fundamentales:

1. Definiciones básicas: Esta parte contiene todas aquellas definiciones consideradas útiles para el desarrollo de este trabajo. En concreto se trata de definir los conceptos básicos que van desde la misma programación, hasta llegar a la tecnología AutoCAD® ActiveX, pasando por la programación orientada a objetos (OOP), Visual Basic, y Visual Basic para aplicaciones (VBA).
 2. Inicio de un proyecto en AutoCAD® VBA: La estructura de un proyecto en AutoCAD® VBA es muy similar a la estructura de un proyecto en VB, es por ello que esta parte resume todo lo relacionado a un proyecto en VB: La estructura de un proyecto en VB y la estructura de un código en VB.
 3. Descripción de las entidades internas (objetos gráficos de AutoCAD®): Enumera y explica todos los objetos gráficos utilizados en el desarrollo del programa.
-

3.2 DEFINICIONES BASICAS

3.2.1 Programación

Programar es automatizar y definir una serie de procesos para resolver un problema y obtener un resultado final. Un programa es el conjunto de instrucciones que se le dan al ordenador para resolver un problema o tarea determinada. Consiste en proporcionar a un equipo un conjunto de instrucciones (o sentencias) que deben ser ejecutadas en orden, y que proporcionan una salida. Preparación de los datos previos indispensables para obtener la solución de un problema mediante las instrucciones codificadas de un ordenador. Los lenguajes de programación se utilizan para indicar al ordenador las acciones que ha de realizar para resolver un determinado problema. Básicamente los lenguajes de programación se componen de ordenes (en adelante llamadas instrucciones) que es lo que en si mismo le dice al ordenador lo que tiene que hacer. Un conjunto de esas instrucciones forman el programa.

3.2.2 Programación orientada a objetos (OOP)

La programación orientada a objetos es un sistema de programación que permite una jerarquía abstracta y modular, así como las características de polimorfismo, herencia y encapsulamiento

El polimorfismo es la posibilidad de tener métodos con el mismo nombre, pero distinto contenido, para clases relacionadas. El procedimiento que se usa lo determina en tiempo de ejecución la clase del objeto. Por ejemplo, objetos relacionados podrían tener métodos Draw. Un procedimiento, transferido como parámetros, puede llamar al método Draw sin necesidad de saber qué tipo de objeto es el parámetro.

La herencia es la posibilidad de que una subclase adopte las características de la clase en la que se basa. Si las características de la clase primaria cambian, la subclase en la que se basa heredará dichas características. Por ejemplo, si agrega

una nueva propiedad, IsBold, a un control de edición, todas las subclases basadas en su control también tendrán una propiedad IsBold.

El encapsulamiento es un término de programación orientada a objetos que se aplica a la posibilidad de contener y ocultar información acerca de un objeto, como las estructuras internas de datos y el código. El encapsulamiento aísla la complejidad interna del funcionamiento de un objeto del resto de la aplicación. Por ejemplo, cuando establece la propiedad Caption de un botón de comando, no necesita saber cómo se almacena la cadena.

3.2.3 Objetos

Un objeto es una combinación de código y datos que se puede tratar como una unidad. Un objeto puede ser una parte de una aplicación, como un control o un formulario. También puede ser un objeto una aplicación entera. La tabla siguiente describe ejemplos de tipos de objetos que puede usar en Visual Basic.

Ejemplo	Descripción
Botón de comando	Son objetos los controles de un formulario, como botones de comandos y marcos.
Formulario	Cada formulario de un proyecto de Visual Basic es un objeto distinto.
Base de datos	Las bases de datos son objetos y contienen otros objetos, como campos e índices.
Gráfico	Un gráfico de Microsoft Excel es un objeto.

Los formularios y los controles son objetos que se puede incluir en las aplicaciones. Se puede manipular estos objetos a través de propiedades, métodos y eventos

3.2.3.1 Propiedades

Un objeto tiene ciertas propiedades o atributos. Los objetos también tienen propiedades que están determinadas por la clase en la que se basa el objeto. Estas propiedades pueden establecerse en tiempo de diseño, que es el modo de funcionamiento cuando se crea una aplicación donde se generan los componentes

de la aplicación y se establece sus propiedades, o en tiempo de ejecución, que es el tiempo durante el cual se ejecuta una aplicación.

Propiedad	Descripción
Caption	Texto descriptivo que aparece junto a la casilla de verificación.
Enabled	Especifica si un usuario puede elegir o no la casilla de verificación.
ForeColor	Color del texto del título.
Left	Posición del extremo izquierdo de la casilla de verificación.
MousePointer	Apariencia del puntero del <i>mouse</i> (ratón) cuando está situado sobre la casilla de verificación.
Top	Posición de la parte superior de la casilla de verificación.
Visible	Especifica si la casilla de verificación es visible o no.

Por ejemplo, en la tabla siguiente se indican algunas propiedades que puede tener una casilla de verificación (CheckBox).

3.2.3.2 Métodos y eventos

Cada objeto reconoce y puede responder a determinadas acciones denominadas eventos. Un evento es una actividad específica y predeterminada, iniciada por el usuario o por el sistema. Los eventos, en la mayor parte de los casos, se generan por interacción del usuario. Por ejemplo, con un teléfono, se desencadena un evento cuando un usuario descuelga el receptor. Los eventos también se desencadenan cuando el usuario presiona los botones para efectuar una llamada.

Las acciones del usuario que desencadenan eventos incluyen clicks, movimientos del mouse y pulsaciones de teclas. Inicializar un objeto y encontrar una línea de código que produce un error son eventos iniciados por el sistema.

Los métodos son procedimientos asociados a un objeto. Los eventos pueden tener métodos asociados. Por ejemplo, si escribe código de método para el evento Click, ese código se ejecutará cuando se produzca el evento Click. Los métodos también pueden existir independientemente de los eventos. Se debe llamar a estos métodos de forma explícita en el código.

El conjunto de eventos es limitado, aunque amplio. No es posible crear nuevos eventos. Sin embargo, el conjunto de métodos puede ampliarse indefinidamente.

La tabla siguiente muestra algunos de los eventos asociados a una casilla de verificación (CheckBox):

Evento	Descripción
Click	El usuario hace clic en la casilla de verificación.
GotFocus	El usuario activa la casilla de verificación al hacer clic en ella o al llegar a ella a través de la tecla TAB.
LostFocus	El usuario selecciona otro control.

La tabla siguiente muestra algunos métodos asociados a una casilla de verificación (CheckBox):

Método	Descripción
Refresh	El valor de la casilla de verificación se actualiza para reflejar los cambios que se puedan haber producido en el origen de datos subyacente.
SetFocus	El enfoque se establece en la casilla de verificación como si el usuario hubiera presionado la tecla TAB hasta activar la casilla de verificación.

3.2.4 Visual Basic

La palabra "Visual" hace referencia al método que se utiliza para crear la interfaz gráfica de usuario (GUI). En lugar de escribir numerosas líneas de código para describir la apariencia y la ubicación de los elementos de la interfaz, simplemente puede agregar objetos prefabricados en su lugar dentro de la pantalla.

Este lenguaje nace del BASIC (Beginner's All-purpose Symbolic Instruction Code) que fue creado en su versión original en el Dartmouth College, con el propósito de servir a aquellas personas que estaban interesadas en iniciarse en algún lenguaje de programación. Luego de sufrir varias modificaciones, en el año 1978 se estableció el BASIC estándar. La sencillez del lenguaje ganó el desprecio de los programadores avanzados por considerarlo "un lenguaje para principiantes".

Primero fue GWBASIC, luego se transformó en QuickBASIC y actualmente se lo conoce como Visual Basic y la versión más reciente es la 6 que se incluye en el paquete Visual Studio 6 de Microsoft. Esta versión combina la sencillez del BASIC con un poderoso lenguaje de programación Visual que juntos permiten desarrollar robustos programas de 32 bits para Windows. Esta fusión de sencillez y la estética permitió ampliar mucho más el monopolio de Microsoft, ya que el lenguaje sólo es compatible con Windows, un sistema operativo de la misma empresa.

3.2.5 Visual Basic para Aplicaciones (VBA)

Visual Basic para Aplicaciones, es un lenguaje de programación adaptado del Visual Basic. Se emplea para automatizar procesos en Microsoft Office. Con este lenguaje de programación, se pueden ampliar y personalizar las opciones existentes de los programas de Office con nuevas funciones, elementos del menú o iconos.

3.2.6 Introducción a la tecnología AutoCAD® ActiveX

AutoCAD® ActiveX proporciona un mecanismo para manipular de forma programada AutoCAD® desde dentro o fuera del mismo. Esto lo hace exponiendo los objetos de AutoCAD® al "mundo externo" ("Outside World"). Una vez que estos objetos son expuestos, ellos pueden accederse por muchos entornos de lenguajes de programación diferentes y por otras aplicaciones como Microsoft® Word VBA o Excel VBA.

Hay dos ventajas a llevar a cabo una interfaz de ActiveX para AutoCAD®:

- El acceso programático a los dibujos de AutoCAD® se abre a muchos más entornos de programación
- Permite compartir datos con otras aplicaciones de Windows®, como Microsoft Excel® y Word®.

3.2.7 Objetos AutoCAD® ActiveX

Un objeto es el bloque del edificio principal de cualquier aplicación de ActiveX. Cada objeto expuesto representa una parte precisa de AutoCAD®. Hay muchos tipos diferentes de objetos en la interfaz de AutoCAD® ActiveX. Por ejemplo:

- Los objetos gráficos como las líneas, los arcos, texto, y dimensiones son objetos.
 - Las escenas de estilo como los tipos de líneas y estilos de la dimensión son objetos.
 - Las estructuras organizacionales como las capas, los grupos, y bloques son objetos.
 - Las vistas view y viewport son objetos.
 - Incluso el dibujo (drawing) y la aplicación de AutoCAD® son consideradas como objetos.
-

3.2.8 Interface AutoCAD® Visual Basic For Applications(VBA)

Microsoft VBA es un entorno de programación basada en objetos diseñado para proporcionar capacidades de desarrollo similares a los de Visual Basic (VB). La diferencia principal entre VBA y VB es que VBA se ejecuta en el mismo espacio AutoCAD®, proporcionando un entorno de programación AutoCAD®-inteligente y muy rápido.

VBA también proporciona otras aplicaciones VBA-activadas a la integración de la aplicación. Esto significa que AutoCAD®, mientras usa otras bibliotecas de objeto de aplicación, puede ser un controlador Automation para otras aplicaciones como Microsoft Word o Excel.

3.2.9 ¿Cómo es implementado VBA en AutoCAD®?

VBA envía los mensajes a AutoCAD® por la interface AutoCAD® ActiveX Automation. AutoCAD® VBA le permite al entorno VBA ejecutarse simultáneamente con AutoCAD® y proporciona control programático de AutoCAD® a través de la interfaz ActiveX Automation. Este acoplamiento de AutoCAD®, la Automatización de ActiveX, y VBA no sólo mantiene una interfaz sumamente poderosa manipulando objetos de AutoCAD®, sino que permite enviar o recuperar datos de otras aplicaciones.

Hay tres elementos fundamentales que definen ActiveX y VBA que programan en AutoCAD®. El primero es el propio AutoCAD® que tiene un juego provechoso de objetos que encapsulan entidades de AutoCAD®, datos, y órdenes. El segundo elemento es la interfaz AutoCAD® ActiveX Automation, que establece los mensajes (la comunicación) con los objetos de AutoCAD®. El tercer elemento es el entorno de programación de VBA que tiene su propio juego de objetos, palabras claves, las constantes, y tan adelante eso proporciona flujo del programa, mando, depuración, y ejecución.

3.3 INICIO DE UN PROGRAMA EN AutoCAD® VBA

Antes de iniciar un programa en AutoCAD® VBA, primero hay que hacer un recorrido de la estructura de una aplicación de Visual Basic para Aplicaciones y cómo ésta se relaciona con otras aplicaciones.

Para crear una aplicación con Visual Basic se trabaja con proyectos. Un proyecto es una colección de archivos que se usan para generar una aplicación.

Cuando se desarrolla una aplicación, se trabaja con un archivo de proyecto para administrar todos los diferentes archivos que se crean. Un proyecto consta de:

- Un archivo de proyecto que realiza el seguimiento de todos los componentes (.vbp)
- Un archivo para cada formulario (.frm).
- Un archivo de datos binarios para cada formulario que contiene datos sobre propiedades de controles del formulario (.frx). Estos archivos no se pueden modificar y los genera automáticamente cualquier archivo .frm que tenga propiedades en formato binario, como **Picture** o **Icon**.
- Opcionalmente, un archivo para cada módulo de clase (.cls).
- Opcionalmente, un archivo para cada módulo estándar (.bas).
- Opcionalmente, uno o más archivos con controles ActiveX (.ocx).
- Opcionalmente, un único archivo de recursos (.res).

El archivo de proyecto es simplemente una lista de todos los archivos y objetos asociados con el proyecto, así como información acerca de las opciones de entorno establecidas. Esta información se actualiza cada vez que se guarda el proyecto. Todos los archivos y objetos también se pueden compartir con otros proyectos.

3.3.1 Estructura de un proyecto en VB (VBA)

3.3.1.1 Módulos de formulario

Los módulos de formulario (extensión de nombre de archivo .frm) pueden contener texto descriptivo del formulario y sus controles, incluyendo los valores de sus propiedades. También pueden contener declaraciones de formulario de constantes, variables y procedimientos externos, así como procedimientos de evento y procedimientos generales.

3.3.1.2 Módulos de clase

Los módulos de clase (extensión de nombre de archivo .cls) son similares a los módulos de formulario, excepto en que no tienen interfaz de usuario visible. Puede usar módulos de clase para crear sus propios objetos, incluyendo código para métodos y propiedades.

3.3.1.3 Módulos estándar

Los módulos estándar (extensión de nombre de archivo .bas) pueden contener declaraciones públicas o a nivel de módulo de tipos, constantes, variables, procedimientos externos y procedimientos públicos.

3.3.1.4 Componentes

Además de archivos y módulos, también es posible agregar otro tipo de componentes a un proyecto. Esos componentes son:

3.3.1.4.1 Controles ActiveX

Los controles ActiveX (extensión de nombre de archivo .ocx) son controles opcionales que se pueden agregar al cuadro de herramientas y se pueden usar en formularios. Cuando instala Visual Basic, los archivos que contienen los controles incluidos en Visual Basic se copian a un directorio común. Existen controles ActiveX adicionales disponibles en diversas fuentes. También puede crear sus propios controles mediante las ediciones Profesional y Empresarial de Visual Basic.

3.3.1.4.2 *Objetos insertables*

Los objetos insertables, como un objeto Hoja de cálculo de Microsoft Excel, son componentes que se pueden usar como bloques para generar soluciones integradas. Una solución integrada puede contener datos en diferentes formatos, como hojas de cálculo, mapas de bits y texto, creados por diferentes aplicaciones.

3.3.1.4.3 *Controles estándar*

Los controles estándar los proporciona Visual Basic. Los controles estándar, como CommandButton (botón de comando) o Frame (marco), siempre están incluidos en el cuadro de herramientas, al contrario de lo que ocurre con los controles ActiveX y los objetos insertables, que se pueden agregar y quitar del cuadro de herramientas.

3.3.1.5 Estructura de un código en VBA

El código en Visual Basic se almacena en módulos. Hay tres tipos de módulos: de formulario, estándar y de clase.

Las aplicaciones sencillas pueden consistir en un único formulario y todo el código de la aplicación reside en ese módulo de formulario. A medida que sus aplicaciones vayan creciendo y siendo más sofisticadas, agregará formularios adicionales. A veces tendrá código común que deseará ejecutar en varios formularios. No querrá duplicar el código en ambos formularios, por lo que creará un módulo independiente que contenga un procedimiento que ejecute el código común. Este módulo independiente debe ser un módulo estándar. Con el tiempo, puede construir una biblioteca de módulos que contenga los procedimientos compartidos.

Cada módulo estándar, de clase y de formulario puede contener lo siguiente:

- Declaraciones. Puede colocar declaraciones de constantes, tipos, variables y procedimientos de bibliotecas de vínculos dinámicos (DLL) al nivel de módulo de formulario, de clase o estándar.

- Procedimientos. Un procedimiento **Sub**, **Function** o **Property** contiene partes de código que se pueden ejecutar como una unidad. Se describen en la sección

3.3.1.5.1 Declaración de variables

Declarar una variable es decirle al programa algo de antemano. Se declara una variable mediante la instrucción **Dim**, proporcionando un nombre a la variable:

```
Dim nombreVariable [As tipo]
```

Las variables que se declaran en un procedimiento mediante la instrucción **Dim** sólo existen mientras se ejecuta el procedimiento. Cuando termina el procedimiento, desaparece el valor de la variable. Además, el valor de una variable de un procedimiento es local de dicho procedimiento; es decir, no puede tener acceso a una variable de un procedimiento desde otro procedimiento. Estas características le permiten usar los mismos nombres de variables en distintos procedimientos sin preocuparse por posibles conflictos o modificaciones accidentales.

Debe ser única en el mismo alcance, que es el intervalo desde el que se puede hacer referencia a la variable: un procedimiento, formulario, etc.

La cláusula opcional **As** tipo de la instrucción **Dim** le permite definir el tipo de dato o de objeto de la variable que va a declarar. Los tipos de datos definen el tipo de información que almacena la variable. Algunos ejemplos de tipos de datos son **String**, **Integer** y **Currency**. Las variables también pueden contener objetos de Visual Basic u otras aplicaciones. Algunos ejemplos de tipos de objeto de Visual Basic, o clases, son **Object**, **Form1** y **TextBox**.

Hay otras formas de declarar variables:

- Declarar una variable en la sección Declaraciones de un módulo de formulario, estándar o de clase, en vez de en un procedimiento, hace que la variable esté disponible para todos los procedimientos del módulo.
-

- Declarar una variable mediante la palabra clave `Public` hace que esté accesible para toda la aplicación.
- Declarar una variable local mediante la palabra clave `Static` preserva su valor aunque termine el procedimiento.

3.3.1.5.1.1 Declaración implícita

No tiene por qué declarar una variable antes de usarla.

Visual Basic crea automáticamente una variable con ese nombre, que puede usar como si la hubiera declarado explícitamente. Aunque es cómodo, puede provocar errores sutiles en el código si se equivoca de nombre de variable. En este caso esta declaración no aplica al programa creado

3.3.1.5.1.2 Declaración explícita

Para evitar problemas al equivocarse de nombre en las variables, puede estipular que Visual Basic le avise siempre que encuentre un nombre que no se haya declarado explícitamente como una variable.

Para declarar variables de forma explícita

Incluya esta instrucción en la sección Declaraciones del módulo de clase, de formulario o estándar:

```
Option Explicit
```

Nota: La instrucción `Option Explicit` funciona sobre módulo a módulo; debe colocarse en la sección Declaraciones de todos los módulos de formulario, estándar o de clase en los que desee que Visual Basic obligue a declarar explícitamente las variables.

3.3.1.5.1.3 Declarar constantes

Una constante es un nombre significativo que sustituye a un número o una cadena que no varía. Aunque una constante recuerda ligeramente a una variable, no puede

modificar una constante o asignarle un valor nuevo como ocurre con una variable. Hay dos orígenes para las constantes:

Constantes intrínsecas o definidas por el sistema proporcionadas por aplicaciones y controles. Las constantes de Visual Basic se muestran en Visual Basic (VB) y Visual Basic para aplicaciones (VBA) y las bibliotecas de objetos en el Examinador de objetos. Otras aplicaciones que proporcionan bibliotecas de objetos, como Microsoft Excel y Microsoft Project, también proporcionan una lista de constantes que puede usar con sus objetos, métodos y propiedades. También se definen constantes en la biblioteca de objetos de cada control ActiveX.

Las constantes simbólicas o definidas por el usuario se declaran mediante la instrucción Const.

En Visual Basic, los nombres de constantes tienen un formato que combina mayúsculas y minúsculas, con un prefijo que indica la biblioteca de objetos que define la constante. Las constantes de las bibliotecas de objetos de Visual Basic y Visual Basic para aplicaciones tienen el prefijo "vb"; por ejemplo, vbTileHorizontal.

Se ha diseñado el sistema de prefijos para evitar colisiones accidentales en casos donde las constantes tienen nombres idénticos y representan valores diferentes. Incluso con prefijos, es posible que dos bibliotecas de objetos puedan contener constantes idénticas que representen valores diferentes. La constante a la que se hace referencia en cada caso depende de la biblioteca de objetos que tenga mayor prioridad.

Para estar totalmente seguro de evitar colisiones en los nombres de constantes, puede calificar las referencias a constantes con la sintaxis siguiente:

`[nombreBiblioteca.][nombreMódulo.]nombreConstante`

`nombreBiblioteca` suele ser el nombre de clase del control o la biblioteca, `nombreMódulo` es el nombre del módulo que define la constante y

`nombreConstante` es el nombre de la constante. Cada uno de estos elementos se define en la biblioteca de objetos y se puede ver en el Examinador de objetos.

La sintaxis para declarar una constante definida por el usuario es la siguiente:

```
[Public|Private] Const nombreConstante[As tipo] = expresión
```

El argumento `nombreConstante` es un nombre simbólico válido (las reglas son las mismas que para crear nombres de variable) y `expresión` está compuesta por constantes y operadores de cadena o numéricos; sin embargo, no puede usar llamadas a funciones en expresión.

Una instrucción `Const` puede representar una cantidad matemática o de fecha y hora:

```
Const conPi = 3.14159265358979
```

```
Public Const conMaxPlanetas As Integer = 9
```

```
Const conFechaSalida = #1/1/95#
```

Se puede usar también la instrucción `Const` para definir constantes de cadena:

```
Public Const conVersion = "07.10.A"
```

Puede colocar más de una declaración de constante en una única línea si las separa con comas:

A menudo, la expresión del lado derecho del signo igual (=) es un número o cadena literal, pero también puede ser una expresión que dé como resultado un número o una cadena (aunque la función no puede contener llamadas a funciones). Puede incluso definir constantes en términos de constantes previamente definidas:

```
Const conPi2 = conPi * 2
```

Una vez que defina las constantes, puede colocarlas en el código para hacerlo más legible.

3.3.1.5.1.4 Tipos de Datos

Las variables son marcadores de posición que se utilizan para almacenar valores; tienen nombre y un tipo de dato. El tipo de dato de la variable determina cómo se almacenan los bits que representan esos valores en la memoria del equipo. Cuando declare una variable, también puede proporcionarle un tipo de dato. Todas las variables tienen un tipo de dato que determina la clase de datos que pueden almacenar.

De forma predeterminada, si no proporciona un tipo de dato, la variable toma el tipo de dato `Variant`. El tipo de dato `Variant` puede representar diferentes tipos de datos en distintas situaciones.

Sin embargo, si sabe que una variable almacenará siempre un tipo de dato determinado, Visual Basic tratará de forma más eficiente los datos si declara la variable con ese tipo. Por ejemplo, se representa mejor una variable que va a almacenar nombres de personas como el tipo de dato `String`, ya que un nombre está siempre compuesto de caracteres.

Los tipos de datos se aplican a otras cosas además de a las variables. Cuando asigna un valor a una propiedad, dicho valor tiene un tipo de dato; los argumentos de las funciones tienen también tipos de datos. De hecho, todo lo relacionado con datos en Visual Basic tiene un tipo de dato.

3.3.1.5.1.5 Declarar variables con tipos de datos

Antes de usar una variable que no sea `Variant` debe usar las instrucciones `Private`, `Public`, `Dim` o `Static` para declararla `As` tipo.

La tabla siguiente muestra los tipos de datos compatibles, incluyendo el tamaño de almacenamiento y el intervalo.

Tipo de datos	Tamaño de almacenamiento	Intervalo
Byte	1 byte	0 a 255
Boolean	2 bytes	True o False
Integer	2 bytes	-32,768 a 32,767
Long (entero largo)	4 bytes	-2,147,483,648 a 2,147,483,647
Single (coma flotante/ precisión simple)	4 bytes	-3,402823E38 a –1,401298E-45 para valores negativos; 1,401298E-45 a 3,402823E38 para valores positivos
Double (coma flotante/ precisión doble)	8 bytes	-1,79769313486232E308 a -4,94065645841247E-324 para valores negativos; 4,94065645841247E-324 a 1,79769313486232E308 para valores positivos
Currency (entero a escala)	8 bytes	-922.337.203.685.477,5808 a 922.337.203.685.477,5807
Decimal	14 bytes	+/- 79.228.162.514.264.337.593.543.950.335 sin punto decimal; +/-7,9228162514264337593543950335 con 28 posiciones a la derecha del signo decimal; el número más pequeño distinto de cero es +/-0,00000000000000000000000000000001
Date	8 bytes	1 de enero de 100 a 31 de diciembre de 9999
Object	4 bytes	Cualquier referencia a tipo Object
String (longitud variable)	10 bytes + longitud de la cadena	Desde 0 a 2.000 millones
String (longitud fija)	Longitud de la cadena	Desde 1 a 65.400 aproximadamente
Variant (con números)	16 bytes	Cualquier valor numérico hasta el intervalo de un tipo Double
Variant (con caracteres)	22 bytes + longitud de la cadena	El mismo intervalo que para un tipo String de longitud variable
Definido por el usuario (utilizando Type)	Número requerido por los elementos	El intervalo de cada elemento es el mismo que el intervalo de su tipo de datos.

3.3.1.5.1.6 Declarar matrices (de tamaño fijo)

Hay tres formas de declarar una matriz de tamaño fijo, dependiendo del alcance que desee que tenga la matriz:

- Para declarar una matriz pública, utilice la instrucción `Public` en la sección Declaraciones de un módulo para declarar la matriz.
- Para crear una matriz a nivel de módulo, utilice la instrucción `Private` en la sección Declaraciones de un módulo para declarar la matriz.
- Para crear una matriz local, utilice la instrucción `Private` en un procedimiento para declarar la matriz.

Establecer los límites superior e inferior

Cuando declare una matriz, ponga a continuación del nombre de la matriz el límite superior entre paréntesis. El límite superior no puede exceder el intervalo de un tipo de dato `Long` (-2.147.483.648 a 2.147.483.647). Por ejemplo, estas declaraciones de matrices pueden aparecer en la sección Declaraciones de un módulo:

```
Dim Contadores(14) As Integer ' 15 elementos.
```

```
Public Contadores(14) As Integer
```

Para especificar el límite inferior, proporciónelo explícitamente (como tipo de dato `Long`) mediante la palabra clave `To`:

```
Dim Contadores(1 To 15) As Integer
```

```
Dim Sumas(100 To 120) As String
```

Matrices multidimensionales

En Visual Basic puede declarar matrices de varias dimensiones:

```
Static MatrizA(9, 9) As Double
```

```
Static MatrizA(1 To 10, 1 To 10) As Double
```

Puede hacer que tenga más de dos dimensiones. Por ejemplo:

```
Dim MultiD(3, 1 To 10, 1 To 15)
```

3.3.1.5.1.7 Declarar variables de objeto

Declare una variable de objeto de la misma forma en que declara las demás variables, con `Dim`, `Static`, `Private` o `Public`. Las únicas diferencias son la palabra clave opcional `New` y el argumento `clase`. La sintaxis es la siguiente:

```
{Dim | ReDim | Static | Private | Public} variable As [New]  
clase
```

También puede declarar una variable de objeto que pueda hacer referencia a cualquier formulario de la aplicación:

```
Dim anyForm As Form
```

De forma similar, puede declarar una variable de objeto que haga referencia a cualquier cuadro de texto de la aplicación:

```
Dim anyText As TextBox
```

También puede declarar una variable de objeto que pueda hacer referencia a un control de cualquier tipo:

```
Dim anyControl As Control ' Variable genérica de control.
```

3.3.1.5.1.8 Asignar variables de objeto

Asigne un objeto a una variable de objeto con la instrucción `Set`:

```
Set variable = objeto
```

Utilice la instrucción `Set` siempre que desee que una variable de objeto haga referencia a un objeto.

3.3.1.5.2 Procedimientos

Puede simplificar las tareas de programación si divide los programas en componentes lógicos más pequeños. Estos componentes, llamados procedimientos, pueden convertirse en bloques básicos que le permiten mejorar y ampliar Visual Basic.

En Visual Basic se utilizan varios tipos de procedimientos:

- Procedimientos `Sub` que no devuelven un valor.
- Procedimientos `Function` que devuelven un valor.
- Procedimientos `Property` que pueden devolver y asignar valores, así como establecer referencias a objetos.

En este caso, solamente se hará referencia a los procedimientos `Sub` y `Function`.

3.3.1.5.2.1 Procedimientos Sub

Un procedimiento `Sub` es un bloque de código que se ejecuta como respuesta a un evento. Al dividir el código de un módulo en procedimientos `Sub`, es más sencillo encontrar o modificar el código de la aplicación.

La sintaxis de un procedimiento `Sub` es la siguiente:

```
[Private|Public] [Static] Sub nombreProcedimiento (argumentos)
instrucciones
```

```
End Sub
```

Cada vez que se llama al procedimiento se ejecutan las `instrucciones` que hay entre `Sub` y `End Sub`. Se pueden colocar los procedimientos `Sub` en módulos estándar, módulos de clase y módulos de formulario. De forma predeterminada, los

procedimientos `Sub` son `Public` en todos los módulos, lo que significa que se les puede llamar desde cualquier parte de la aplicación.

Los `argumentos` de un procedimiento son como las declaraciones de variables; se declaran valores que se pasan desde el procedimiento que hace la llamada.

3.3.1.5.2.2 Procedimientos Function

Visual Basic incluye funciones incorporadas o intrínsecas, como `Sqr`, `Cos` o `Chr`. Además, puede usar la instrucción `Function` para escribir sus propios procedimientos `Function`.

La sintaxis de un procedimiento `Function` es la siguiente:

```
[Private|Public] [Static] Function nombreProcedimiento  
(argumentos) [As tipo]  
instrucciones  
  
End Function
```

Al igual que un procedimiento `Sub`, un procedimiento `Function` es un procedimiento diferente que puede tomar `argumentos`, realizar una serie de `instrucciones` y cambiar el valor de los `argumentos`. A diferencia de los procedimientos `Sub`, los procedimientos `Function` pueden devolver un valor al procedimiento que realiza la llamada. Hay tres diferencias entre los procedimientos `Sub` y `Function`:

Generalmente, se llama a una función incluyendo el nombre y los `argumentos` del procedimiento en la parte derecha de una instrucción o expresión mayor (`valorRetorno = función()`).

- Los procedimientos `Function` tienen tipos de datos, al igual que las variables. Esto determina el tipo del valor devuelto. (En ausencia de la cláusula **As**, el tipo es el tipo predeterminado `Variant`.)
-

- Se devuelve un valor asignándole al propio `nombreProcedimiento`. Cuando el procedimiento `Function` devuelve un valor, se puede convertir en parte de una expresión mayor.

3.3.1.5.3 Estructuras de control

Las estructuras de control le permiten controlar el flujo de ejecución del programa. Si no se controla mediante instrucciones de control de flujo, la lógica del programa fluirá por las instrucciones de izquierda a derecha y de arriba a abajo. Aunque se pueden escribir algunos programas sencillos con un flujo unidireccional y aunque se puede controlar parte del flujo mediante operadores para regular la precedencia de las operaciones, la mayor parte del poder y utilidad de un lenguaje de programación deriva de su capacidad de cambiar el orden de las instrucciones mediante estructuras y bucles.

3.3.1.5.3.1 Estructuras de decisión

3.3.1.5.3.1.1 If...Then

Use la estructura **If...Then** para ejecutar una o más instrucciones basadas en una condición. Puede usar la sintaxis de una línea o un *bloque* de varias líneas:

```
If condición Then instrucción
```

```
If condición Then  
instrucciones
```

```
End If
```

Condición normalmente es una comparación, pero puede ser cualquier expresión que dé como resultado un valor numérico. Visual Basic interpreta este valor como `True` o `False`; un valor numérico cero es `False` y se considera `True` cualquier valor numérico distinto de cero. Si condición es `True`, Visual Basic ejecuta todas las instrucciones que siguen a la palabra clave `Then`.

3.3.1.5.3.1.2 If...Then...Else

Utilice un bloque **If...Then...Else** para definir varios bloques de instrucciones, uno de los cuales se ejecutará:

```
If condición1 Then
[bloque de instrucciones 1]
[ElseIf condición2 Then
[bloque de instrucciones 2]] ...
[Else
[bloque de instrucciones n]]

End If
```

Visual Basic evalúa primero `condición1`. Si es `False`, Visual Basic procede a evaluar `condición2` y así sucesivamente, hasta que encuentre una condición `True`. Cuando encuentra una condición `True`, Visual Basic ejecuta el bloque de instrucciones correspondientes y después ejecuta el código que sigue a `End If`. Opcionalmente, puede incluir un bloque de instrucciones `Else`, que Visual Basic ejecutará si ninguna de las condiciones es `True`.

3.3.1.5.3.1.3 Select Case.

Visual Basic proporciona la estructura `Select Case` como alternativa a `If...Then...Else` para ejecutar selectivamente un bloque de instrucciones entre varios bloques de instrucciones. La instrucción `Select Case` ofrece posibilidades similares a la instrucción `If...Then...Else`, pero hace que el código sea más legible cuando hay varias opciones.

La estructura `Select Case` funciona con una única expresión de prueba que se evalúa una vez solamente, al principio de la estructura. Visual Basic compara el resultado de esta expresión con los valores de cada `Case` de la estructura. Si hay una coincidencia, ejecuta el bloque de instrucciones asociado a ese `Case`:

```
Select Case expresiónprueba
[Case listaexpresiones1
[bloque de instrucciones 1]]
[Case listaexpresiones2
[bloque de instrucciones 2]]
.
.
.
[Case Else
[bloque de instrucciones n]]

End Select
```

Cada `listaexpresiones` es una lista de uno o más valores. Si hay más de un valor en una lista, se separan los valores con comas. Cada bloque de instrucciones contiene cero o más instrucciones. Si más de un **Case** coincide con la expresión de prueba, sólo se ejecutará el bloque de instrucciones asociado con la primera coincidencia. Visual Basic ejecuta las instrucciones de la cláusula (opcional) **Case Else** si ningún valor de la lista de expresiones coincide con la expresión de prueba.

3.3.1.5.3.2 Estructuras de bucle

Las estructuras de bucle le permiten ejecutar una o más líneas de código repetidamente. Las estructuras de bucle que acepta Visual Basic son:

Do...Loop

For...Next

For Each...Next

3.3.1.5.3.2.1 Do...Loop

Utilice el bucle `Do` para ejecutar un bloque de instrucciones un número indefinido de veces. Hay algunas variantes en la instrucción `Do...Loop`, pero cada una evalúa

una condición numérica para determinar si continúa la ejecución. Como ocurre con `If...Then`, la condición debe ser un valor o una expresión que dé como resultado `False` (cero) o `True` (distinto de cero).

En el ejemplo de `Do...Loop` siguiente, las instrucciones se ejecutan siempre y cuando condición sea `True`:

```
Do While condición
instrucciones
```

```
Loop
```

Cuando Visual Basic ejecuta este bucle `Do`, primero evalúa condición. Si condición es `False` (cero), se salta todas las instrucciones. Si es `True` (distinto de cero), Visual Basic ejecuta las instrucciones, vuelve a la instrucción `Do While` y prueba la condición de nuevo.

Por tanto, el bucle se puede ejecutar cualquier número de veces, siempre y cuando condición sea distinta de cero o `True`. Nunca se ejecutan las instrucciones si condición es `False` inicialmente.

Otra variante de la instrucción `Do...Loop` ejecuta las instrucciones primero y prueba condición después de cada ejecución. Esta variación garantiza al menos una ejecución de instrucciones:

```
Do
instrucciones
```

```
Loop While condición
```

Hay otras dos variantes análogas a las dos anteriores, excepto en que repiten el bucle siempre y cuando condición sea `False` en vez de `True`.

Hace el bucle cero o más veces	Hace el bucle al menos una vez
Do Until condición instrucciones Loop	Do instrucciones Loop Until condición

3.3.1.5.3.2.2 For...Next

Los bucles **Do** funcionan bien cuando no se sabe cuántas veces se necesitará ejecutar las instrucciones del bucle. Sin embargo, cuando se sabe que se van a ejecutar las instrucciones un número determinado de veces, es mejor elegir el bucle **For...Next**. A diferencia del bucle **Do**, el bucle **For** utiliza una variable llamada contador que incrementa o reduce su valor en cada repetición del bucle. La sintaxis es la siguiente:

```
For contador = iniciar To finalizar [Step incremento]
instrucciones

Next [contador]
```

Los argumentos `contador`, `iniciar`, `finalizar` e `incremento` son todos numéricos.

Nota El argumento `incremento` puede ser positivo o negativo. Si `incremento` es positivo, `iniciar` debe ser menor o igual que `finalizar` o no se ejecutarán las instrucciones del bucle. Si `incremento` es negativo, `iniciar` debe ser mayor o igual que `finalizar` para que se ejecute el cuerpo del bucle. Si no se establece **Step**, el valor predeterminado de `incremento` es 1.

Al ejecutar el bucle `For`, Visual Basic:

1. Establece `contador` al mismo valor que `iniciar`.
2. Comprueba si `contador` es mayor que `finalizar`. Si lo es, Visual Basic sale del bucle. (Si `incremento` es negativo, Visual Basic comprueba si `contador` es menor que `finalizar`.)

3. Ejecuta instrucciones.
4. Incrementa contador en 1 o en instrucciones, si se especificó.
5. Repite los pasos 2 a 4.

3.3.1.5.3.2.3 For Each...Next

El bucle `For Each...Next` es similar al bucle `For...Next`, pero repite un grupo de instrucciones por cada elemento de una colección de objetos o de una matriz en vez de repetir las instrucciones un número especificado de veces. Esto resulta especialmente útil si no sabe cuántos elementos hay en la colección.

He aquí la sintaxis del bucle `For Each...Next`:

```
For Each elemento In grupo  
instrucciones
```

```
Next elemento
```

3.4 DESCRIPCION DE ENTIDADES INTERNAS (OBJETOS GRAFICOS DE AutoCAD®)

3.4.1 Introducción

En este capítulo se dará una descripción de todas aquellas entidades gráficas, u objetos gráficos de AutoCAD® en VBA utilizados en este programa. Esta descripción consta de dos partes:

La primera parte reúne los conceptos básicos de cada una de estas entidades gráficas, su declaración dentro de cualquier módulo de código, sus nombres de clase, y su sintaxis dentro del código para poder ejecutarlo, además de listar todas sus propiedades y métodos, y solamente resaltar aquellas usadas en el programa (resaltadas en forma de **Negrita**).

En la declaración de los objetos dentro de un código, ésta se hace mediante la siguiente estructura de declaración:

```
Dim Nombre_Objeto As Nombre_Clase
```

```
Public Nombre_Objeto As Nombre_Clase
```

Donde `Nombre_Objeto` es el nombre asignado al objeto declarado, y `Nombre_Clase` es el nombre de clase que representa al objeto. En el caso de una línea, por ejemplo, ésta sería su sintaxis:

```
Dim Linea As AcadLine.
```

```
Public Linea As AcadLine.
```

La segunda parte describe las propiedades y métodos utilizados en el programa, en orden alfabético. Esta descripción consta de: concepto, sintaxis, y definición de las variables usadas en la sintaxis. En las propiedades se puede destacar si la propiedad es solamente de lectura, de escritura, o de ambos.

Si la propiedad es de lectura, su sintaxis es como sigue:

```
Valor_Propiedad = Objeto.Propiedad
```

Si la propiedad es de escritura:

```
Objeto.Propiedad = Valor_Propiedad
```

Donde `Objeto` es la entidad cuya propiedad es aplicable, y `Valor_Propiedad` es una variable que en el caso de lectura, es una variable que almacena el valor de propiedad; en el caso de escritura, puede ser un valor o una variable. El tipo de variable de `Valor_Propiedad` dependerá del valor de la propiedad.

En el caso de las propiedades los objetos `DimAligned` y `DimRotated`, las propiedades se manejaron a través de las variables del sistema.

Esta segunda parte es como un glosario, ya que muchas de estas propiedades y métodos son comunes entre los objetos gráficos.

3.4.2 Línea

La línea es una entidad utilizada para dibujar sobre los espacios de AutoCAD® (modelo y papel) segmentos de línea recta, comprendido entre dos puntos bien definidos como punto inicial y punto final.

Este objeto se declara dentro del entorno AutoCAD® VBA, haciendo uso de su nombre de clase, `AcadLine`, y se crea según el método `AddLine`.

La sintaxis utilizada para crear la línea es la siguiente:

```
Set Linea = Objeto.AddLine(StartPoint, EndPoint)
```

Donde:

Linea: Objeto declarado como `AcadLine`

Objeto: `ModelSpace Collection`, `PaperSpace Collection`, o `Block`

`StartPoint`: Punto inicial del segmento. Variable de tipo `Variant` (matriz de 3 elementos de tipo `Double`).

`EndPoint`: Punto final del segmento. Variable de tipo `Variant` (matriz de 3 elementos de tipo `Double`).

PROPIEDADES	METODOS
Angle	<code>ArrayPolar</code>
<code>Application</code>	<code>ArrayRectangular</code>
Color	Copy
<code>Document</code>	Delete
<code>Delta</code>	<code>GetBoundingBox</code>
Endpoint	<code>GetExtensionDictionary</code>
<code>Handle</code>	<code>GetXData</code>
<code>HasExtensionDictionary</code>	<code>Highlight</code>
<code>Hyperlinks</code>	<code>IntersectWith</code>
Layer	<code>Mirror</code>
Length	<code>Mirror3D</code>
Linetype	Move
LinetypeScale	Offset
<code>LineWeight</code>	Rotate
<code>Normal</code>	<code>Rotate3D</code>

PROPIEDADES	MÉTODOS
ObjectID	ScaleEntity
OwnerID	SetXData
PlotStyleName	TransformBy
StartPoint	Update
Thickness	
Visible	

La línea, por ser un objeto gráfico de AutoCAD®, tiene métodos, propiedades y eventos que permiten manipularla

El objeto línea fue utilizado en la gran parte del dibujo de las escaleras: Escaleras, acotaciones, líneas de referencia. También fue utilizada como referencia para otras aplicaciones, es decir, se crea la línea y se aprovechan sus propiedades (Length, Angle, StartPoint, EndPoint) y métodos (Offset, Copy, Move, Rotate); al finalizar de utilizar la línea, se puede eliminar o no (con el método Delete).

3.4.3 Circulo

El círculo es una entidad que permite dibujar sobre los espacios en AutoCAD® (modelo y papel) circunferencias, con centro en un punto determinado, y un radio de valor definido.

Este objeto se declara dentro del entorno AutoCAD® VBA, haciendo uso de su nombre de clase, `AcadCircle`, y se crea según el método `AddCircle`.

La sintaxis utilizada para crear la línea es la siguiente:

```
Set Circulo = Objeto.AddCircle(Center, Radius)
```

Donde:

Circulo: Objeto declarado como `AcadCircle`

Objeto: `ModelSpace Collection`, `PaperSpace Collection`, o `Block`

Center: Punto que define el centro de la circunferencia. Variable de tipo Variant (matriz de 3 elementos de tipo Double).

Radius: Radio de la circunferencia, valor que debe ser positivo. Variable de tipo Double.

El objeto círculo fue utilizado para dibujar los aceros transversales de la escalera, y para dibujar los ejes de los nervios.

PROPIEDADES	METODOS
Application	ArrayPolar
Area	ArrayRectangular
Center	Copy
Circumference	Delete
Color	GetBoundingBox
Diameter	GetExtensionDictionary
Document	GetXData
Handle	Highlight
HasExtensionDictionary	IntersectWith
Hyperlinks	Mirror
Layer	Mirror3D
Linetype	Move
LinetypeScale	Offset
LineWeight	Rotate
Normal	Rotate3D
ObjectID	ScaleEntity
OwnerId	SetXData
PlotStyleName	TransformBy
Radius	Update
Thickness	
Visible	

3.4.4 Polilínea

La polilínea es una entidad que permite dibujar sobre los espacios en AutoCAD® (modelo y papel) una línea compuesta de segmentos de líneas y arcos circulares

Este objeto se declara dentro del entorno AutoCAD® VBA, haciendo uso de su nombre de clase, AcadPolyline, y se crea según el método AddPolyline.

La sintaxis utilizada para crear la línea es la siguiente:

```
Set Polilinea = Objeto.AddPolyline(VerticesList)
```

Donde:

Polilínea: Objeto declarado como AcadPolyline

Objeto: ModelSpace Collection, PaperSpace Collection, o Block

VerticesList: Lista de vértices. Esta lista contiene las coordenadas de los puntos en el orden en que se va a dibujar, es decir, si se declara una variable llamada ListaPuntos como un vector columna de 9 elementos:

```
Dim ListaPuntos (0 To 8) As Double
```

Las coordenadas del primer punto ocupan las tres primeras posiciones de la lista:

```
ListaPuntos(0) = X1
```

```
ListaPuntos(1) = Y1
```

```
ListaPuntos(2) = Z1
```

Las coordenadas del segundo punto ocupan las tres segundas posiciones de la lista:

```
ListaPuntos(3) = X2
```

```
ListaPuntos(4) = Y2
```

```
ListaPuntos(5) = Z2
```

Las coordenadas del tercer punto ocupan las tres segundas posiciones de la lista:

```
ListaPuntos(6) = X3
```

```
ListaPuntos(7) = Y3
```

```
ListaPuntos(8) = Z3
```

Entonces, para dibujar esta polilínea, su sintaxis es:

```
Set Polilínea = Objeto.AddPolyline(ListaPuntos)
```

El objeto polilínea fue utilizado para dibujar los aceros longitudinales de la escalera, y para determinar el área de la sección transversal de una escalera.

PROPIEDADES	MÉTODOS
Application	AppendVertex
Area	ArrayPolar
Closed	ArrayRectangular
Color	Copy
Coordinate	Delete
Coordinates	Explode
Document	GetBoundingBox
Elevation	GetBulge
Hyperlinks	GetExtensionDictionary
Handle	GetWidth
HasExtensionDictionary	GetXData
Layer	Highlight
Linetype	IntersectWith
LinetypeGeneration	Mirror
LinetypeScale	Mirror3D
LineWeight	Move
Normal	Offset
ObjectID	Rotate
ObjectName	Rotate3D
OwnerID	ScaleEntity
PlotStyleName	SetBulge
Thickness	SetWidth
Type	SetXData
Visible	TransformBy
	Update

3.4.5 Dimensión

Son entidades utilizadas para acotar dimensiones de cualquier objeto gráfico de AutoCAD® entre dos puntos (dimensiones lineales).

Las dimensiones lineales utilizadas en este programa son:

3.4.5.1 DimAligned Object (AcadDimAligned)

Esta entidad es usada para acotar dimensiones lineales en la dirección del elemento lineal, es decir, al acotar entre dos puntos, el valor de la dimensión es la distancia entre esos dos puntos.

Este objeto se declara dentro del entorno AutoCAD® VBA, haciendo uso de su nombre de clase, AcadDimAligned, y se crea según el método AddDimAligned.

La sintaxis utilizada para crear la línea es la siguiente:

```
Set DimAlineada = Objeto.AddDimAligned(ExtLine1Point,  
ExtLine2Point, TextPosition)
```

Donde:

DimAlineada: Objeto declarado como AcadDimAligned.

Objeto: ModelSpace Collection, PaperSpace Collection, o Block.

ExtLine1Point: Primer punto de la línea de referencia. Variable de tipo Variant (matriz de 3 elementos de tipo Double).

ExtLine2Point: Segundo punto de la línea de referencia. Variable de tipo Variant (matriz de 3 elementos de tipo Double).

TextPosition: Punto de ubicación del texto. Variable de tipo Variant (matriz de 3 elementos de tipo Double).

El objeto DimAligned fue utilizado para acotar longitudes (espesor del descanso, huella, contrahuella...) y distancias (aceros longitudinales, distancias de ejes...)

3.4.5.2 DimRotated Object (AcadDimRotated)

Esta entidad es usada para acotar dimensiones lineales en cualquier dirección, es decir, al acotar entre dos puntos, el valor de la dimensión es la distancia proyectada sobre la dirección de la acotación.

Este objeto se declara dentro del entorno AutoCAD® VBA, haciendo uso de su nombre de clase, AcadDimRotated, y se crea según el método AddDimRotated.

La sintaxis utilizada para crear la línea es la siguiente:

```
Set DimLineal = Objeto.AddDimRotated(ExtLine1Point,
ExtLine2Point, DimLineLocation, RotationAngle)
```

Donde:

DimLineal: Objeto declarado como AcadDimRotated.

Objeto: ModelSpace Collection, PaperSpace Collection, o Block.

ExtLine1Point: Primer punto de la línea de referencia. Variable de tipo Variant (matriz de 3 elementos de tipo Double).

ExtLine2Point: Segundo punto de la línea de referencia. Variable de tipo Variant (matriz de 3 elementos de tipo Double).

DimLineLocation: Punto de ubicación de la línea de cota. Variable de tipo Variant (matriz de 3 elementos de tipo Double).

RotationAngle: Ángulo de rotación de la línea de cota. Valor de tipo Double.

El objeto DimAligned fue utilizado para acotar distancias horizontales (luz libre de escalones) y distancias verticales (desniveles).

PROPIEDADES	PROPIEDADES
AltRoundDistance	Arrowhead2Block
AltSuppressLeadingZeros	Arrowhead2Type
AltSuppressTrailingZeros	ArrowheadSize
AltSuppressZeroInches	Color
AltSuppressZeroFeet	DecimalSeparator
AltTextPrefix	DimensionLineColor
AltTextSuffix	DimensionLineExtend
AltTolerancePrecision	DimensionLineWeight
AltToleranceSuppressLeadingZeros	DimLine1Suppress
AltToleranceSuppressTrailingZeros	DimLine2Suppress
AltToleranceSuppressZeroFeet	DimLineInside
AltToleranceSuppressZeroInches	Document
AltUnitsFormat	ExtensionLineColor
AltUnits	ExtensionLineExtend
AltUnitsPrecision	ExtensionLineOffset
AltUnitsScale	ExtensionLineWeight
Application	ExtLine1Point
Arrowhead1Block	ExtLine2Point

Arrowhead1Type	ExtLine1Suppress
PROPIEDADES	PROPIEDADES
ExtLine2Suppress	TextPrefix
Fit	TextRotation
ForceLineInside	TextStyle
FractionFormat	TextSuffix
Handle	ToleranceDisplay
HasExtensionDictionary	ToleranceHeightScale
HorizontalTextPosition	ToleranceJustification
Hyperlinks	ToleranceLowerLimit
Layer	TolerancePrecision
LinearScaleFactor	ToleranceSuppressLeadingZeros
Linetype	ToleranceSuppressTrailingZeros
LinetypeScale	ToleranceSuppressZeroInches
LineWeight	ToleranceSuppressZeroFeet
Measurement	ToleranceUpperLimit
Normal	UnitsFormat
ObjectID	VerticalTextPosition
OwnerId	Visible
PlotStyleName	METODOS
PrimaryUnitsPrecision	ArrayPolar
Rotation	ArrayRectangular
RoundDistance	Copy
ScaleFactor	Delete
StyleName	GetBoundingBox
SuppressLeadingZeros	GetExtensionDictionary
SuppressTrailingZeros	GetXData
SuppressZeroFeet	Highlight
SuppressZeroInches	IntersectWith
TextColor	Mirror
TextGap	Mirror3D
TextHeight	Move
TextInside	Rotate
TextInsideAlign	Rotate3D
TextMovement	ScaleEntity
TextOutsideAlign	SetXData
TextOverride	TransformBy
TextPosition	Update

Nota: Estos son los métodos y las propiedades de las dimensiones (DimAligned y DimRotated). Sin embargo, las propiedades se manipularon mediante las variables del sistema. AutoCAD® almacena operaciones y algunos comandos en variables del sistema. Estas variables tienen asociados un nombre y un valor, éste último depende de la variable. Para modificar las variables del sistema, se usa el método SetVariable:

Documento.SetVariable Name, Value

Donde:

Name: Nombre de la variable del sistema. Variable de tipo String

Value: Nuevo valor de la variable del sistema. Valor de tipo Variant (depende de la variable del sistema)

Si por ejemplo, se usa la propiedad TextHeight, que define la altura del texto de la dimensión, se tiene que el nombre de la variable del sistema asociado a esta propiedad es DIMTXT, y su valor puede ser cualquier número, por ejemplo, 0.30, entonces:

```
Documento.SetVariable DIMTXT, 0.30
```

Entonces, la variable del sistema DIMTXT, que se refiere a la propiedad TextHeight, tiene un valor de 0.30.

En el módulo Utilitarios, uno de los módulos usados en el programa, se encuentra el código que define las propiedades y métodos de estas formas de dimensiones, a partir de las variables del sistema que rigen estas propiedades y métodos, para crear estilos de dimensiones.

3.4.6 Texto

El texto es una entidad que permite insertar sobre los espacios en AutoCAD® (modelo y papel) texto en múltiples líneas, es decir, crea un espacio rectangular destinado a agregar caracteres alfanuméricos.

Este objeto se declara dentro del entorno AutoCAD® VBA, haciendo uso de su nombre de clase, AcadMText, y se crea según el método AddMText.

La sintaxis utilizada para crear la línea es la siguiente:

```
Set MTexto = Objeto.AddMText(InsertionPoint, Width, Text)
```

Donde:

MTexto: Objeto declarado como AcadMText

Objeto: ModelSpace Collection, PaperSpace Collection, o Block

InsertionPoint: Punto de inserción del texto. Variable de tipo Variant
(matriz de 3 elementos de tipo Double).

Width: ancho del espacio rectangular del texto. Variable de tipo Double

Text: Texto (Cadena de caracteres alfanuméricos). Variable de tipo String.

Esta entidad fue utilizada para identificar todo lo relacionado con el acero representado: longitud, espaciamiento, diámetro de la barra de refuerzo o cabilla, número de cabillas.

PROPIEDADES	METODOS
Application	ArrayPolar
AttachmentPoint	ArrayRectangular
Color	Copy
Document	Delete
DrawingDirection	GetBoundingBox
Handle	GetExtensionDictionary
HasExtensionDictionary	GetXData
Height	Highlight
Hyperlinks	IntersectWith
InsertionPoint	Mirror
Layer	Mirror3D
LineSpacingFactor	Move
LineSpacingStyle	Rotate
Linetype	Rotate3D
LinetypeScale	ScaleEntity
LineWeight	SetXData
Normal	TransformBy
ObjectID	Update
ObjectName	
OwnerID	
PlotStyleName	
Rotation	
StyleName	
TextString	
Visible	
Width	

3.4.7 Línea auxiliar

La línea auxiliar es una entidad que permite dibujar sobre los espacios en AutoCAD® (modelo y papel) líneas infinitas, usadas como auxiliares, definiendo su dirección a través de dos puntos.

Este objeto se declara dentro del entorno AutoCAD® VBA, haciendo uso de su nombre de clase, `AcadXLine`, y se crea según el método `AddXLine`.

La sintaxis utilizada para crear la línea es la siguiente:

```
Set LineaAuxiliar = Objeto.AddXLine(Point1, Point2)
```

Donde:

`LineaAuxiliar`: Objeto declarado como `AcadXLine`

`Objeto`: `ModelSpace Collection`, `PaperSpace Collection`, o `Block`

`Point1`: Primer punto de referencia. Este punto resulta ser el punto base o de ubicación de la línea auxiliar. Variable de tipo Variant (matriz de 3 elementos de tipo Double).

`Point2`: Segundo punto de referencia. Este punto resulta ser el punto que define la dirección de la línea auxiliar. Variable de tipo Variant (matriz de 3 elementos de tipo Double).

La línea auxiliar fue utilizada para definir líneas auxiliares horizontales, verticales, e inclinadas, con la finalidad principal de obtener puntos de intersección entre estas líneas.

PROPIEDADES	METODOS
<code>Application</code>	<code>ArrayPolar</code>
<code>BasePoint</code>	<code>ArrayRectangular</code>
<code>Color</code>	<code>Copy</code>
<code>DirectionVector</code>	<code>Delete</code>
<code>Document</code>	<code>GetBoundingBox</code>
<code>Handle</code>	<code>GetExtensionDictionary</code>

PROPIEDADES	METODOS
HasExtensionDictionary	GetXData
Hyperlinks	Highlight
Layer	IntersectWith
Linetype	Mirror
LinetypeScale	Mirror3D
LineWeight	Move
ObjectID	Offset
ObjectName	Rotate
OwnerID	Rotate3D
PlotStyleName	ScaleEntity
SecondPoint	SetXData
Visible	TransformBy
	Update

3.4.8 Flecha (Leader)

La flecha es una entidad que permite dibujar sobre los espacios en AutoCAD® (modelo y papel) una polilínea, en donde el primer punto de esa polilínea se encuentra la punta de la flecha. Se puede crear con o sin texto, bien sea para señalar (sin texto), o para hacer indicaciones (con texto).

Este objeto se declara dentro del entorno AutoCAD® VBA, haciendo uso de su nombre de clase, AcadLeader, y se crea según el método AddLeader.

La sintaxis utilizada para crear la línea es la siguiente:

```
Set Flecha = Objeto.AddLeader(ArrayPoints, Annotation,
Type)
```

Donde:

Flecha: Objeto declarado como AcadLeader

Objeto: ModelSpace Collection, PaperSpace Collection, o Block

ArrayPoints: Matriz de puntos que genera la polilínea (se crea de forma similar que en el dibujo de la polilínea). Los dos primeros puntos son obligatorios, del tercero en adelante, son opcionales.

Annotation: El texto que contiene la flecha (MText). Variable de tipo String.

Type: un AcLeaderType:

acLineNoArrow : Genera una polilínea sin punta de flecha

acLineWithArrow Genera una polilínea con punta de flecha

acSplineNoArrow Genera una polilínea suavizada sin punta de flecha

acSplineWithArrow Genera una polilínea suavizada con punta de
flecha

La flecha fue utilizada solamente para hacer las acotaciones transversales del
acero del escalón.

PROPIEDADES	METODOS
Annotation	ArrayPolar
Application	ArrayRectangular
ArrowheadBlock	Copy
ArrowheadSize	Delete
ArrowheadType	Evaluate
Color	GetBoundingBox
Coordinate	GetExtensionDictionary
Coordinates	GetXData
Document	Highlight
DimensionLineColor	IntersectWith
Handle	Mirror
HasExtensionDictionary	Mirror3D
Hyperlinks	Move
Layer	Rotate
Linetype	Rotate3D
LinetypeScale	ScaleEntity
LineWeight	SetXData
Normal	TransformBy
ObjectID	Update
OwnerID	
PlotStyleName	
ScaleFactor	
StyleName	
TextGap	
Type	
VerticalTextPosition	
Visible	

3.4.9 Sombreado (Hatch)

El sombreado es una entidad que permite crear sobre los espacios en AutoCAD® (modelo y papel) rellenos predeterminados, definidos por el usuario, personalizados. El relleno se crea a partir de una frontera, que puede ser cualquier objeto gráfico, o una combinación de ellos. Dentro de esa frontera es donde se crea el relleno.

Este objeto se declara dentro del entorno AutoCAD® VBA, haciendo uso de su nombre de clase, `AcadHatch`, y se crea según el método `AddHatch`.

La sintaxis utilizada para crear la línea es la siguiente:

```
Set Relleno = Objeto.AddHatch(PatternType, PatternName,  
Associativity)
```

Donde:

Relleno: Objeto declarado como `AcadHatch`

Objeto: `ModelSpace Collection`, `PaperSpace Collection`, o `Block`

PatternType: un `AcPatternType`:

`acHatchPatternTypePredefined` (Tipo de patrón predefinido): Selecciona un patrón de AutoCAD® predefinido. Estos patrones se almacenan en los archivos `acad.pat`.

`acHatchPatternTypeUserDefined` (Tipo de patrón definido por el usuario): Define un patrón de líneas basado en el tipo de línea actual del dibujo.

`acHatchPatternTypeCustomDefined` (Tipo de patrón personalizado): Selecciona un patrón definido de otro archivo `PAT` personalizado que el archivo `acad.pat`

PatternName: Establece el nombre del patrón. Variable de tipo String

Associativity: Indicador que define la asociatividad del contorno y el sombreado. Variable de tipo Boolean:

True: Crea un sombreado que se actualiza cuando se modifican los contornos.

False: Crea un sombreado independiente de los contornos.

El sombreado fue utilizado para crear rellenos sobre los círculos que representan a los aceros transversales.

PROPIEDADES	METODOS
Application	AppendInnerLoop
AssociativeHatch	AppendOuterLoop
Color	ArrayPolar
Document	ArrayRectangular
Elevation	Copy
Handle	Delete
HasExtensionDictionary	Evaluate
HatchStyle	GetBoundingBox
Hyperlinks	GetExtensionDictionary
ISOPenWidth	GetLoopAt
Layer	GetXData
Linetype	Highlight
LinetypeScale	InsertLoopAt
LineWeight	IntersectWith
Normal	Mirror
NumberOfLoops	Mirror3D
ObjectID	Move
OwnerID	Rotate
PatternAngle	Rotate3D
PatternDouble	ScaleEntity
PatternName	SetPattern
PatternScale	SetXData
PatternSpace	TransformBy
PatternType	Update
PlotStyleName	
Visible	

3.4.10 Propiedades

3.4.10.1 Angle

Propiedad que contiene el ángulo de una línea. Esta propiedad es solamente de lectura.

Sintaxis:

`Objeto.Angle`

Donde:

`Objeto`: Line.

`Angle`: Valor del ángulo en radianes. Este ángulo es medido desde el lado positivo eje X en el sentido contrario de las agujas del reloj. El valor del ángulo es de tipo Double.

El ángulo tiene como punto base el punto inicial de la línea.

3.4.10.2 Area

Propiedad que especifica el área encerrada de cualquier objeto, o combinación de ellos, cuya propiedad es aplicable (círculos, polilíneas, arcos de circunferencia...). En el caso de polilíneas abiertas, se calcula el área como si ésta se encerrada con un segmento de línea recta que pasa por el punto inicial y el punto final de la polilínea. Esta propiedad es solamente de lectura, para la polilínea, y de lectura y escritura para el círculo.

Sintaxis:

`Objeto.Area`

Donde:

`Objeto`: Circle, Polyline.

Area: Valor del área del objeto, en unidades cuadradas (m^2 , in^2). Este valor es de tipo Double.

3.4.10.3 ArrowheadSize

Propiedad que especifica el tamaño de la punta de la flecha del objeto el cual esta propiedad es aplicable. Esta propiedad es de lectura y escritura.

Sintaxis:

`Objeto.ArrowheadSize`

Donde:

Objeto: DimAligned, DimRotated, Leader.

ArrowheadSize: Tamaño de la punta de la flecha, este valor debe ser mayor que 0.0. Valor de tipo Double

3.4.10.4 ArrowheadType

Propiedad que establece el tipo de punta de la flecha del objeto cuya propiedad es aplicable. Esta propiedad es de lectura y escritura.

Sintaxis:

`Objeto.ArrowheadType`

Donde:

Objeto: Leader

ArrowheadType: Variable de tipo AcDimArrowheadType

`acArrowDefault`

`acArrowDot`



acArrowDotSmall	
acArrowDotBlank	
acArrowOrigin	
acArrowOrigin2	
acArrowOpen	
acArrowOpen90	
acArrowOpen30	
acArrowClosed	
acArrowSmall	
acArrowNone	
acArrowOblique	
acArrowBoxFilled	
acArrowBoxBlank	
acArrowClosedBlank	
acArrowDatumFilled	
acArrowDatumBlank	
acArrowIntegral	
acArrowArchTick	
acArrowUserDefined	

3.4.10.5 AssociativeHatch

Propiedad que especifica si el sombreado es asociativo o no con el contorno. Existe asociación cuando se modifica el sombreado al modificar el contorno; caso contrario, el sombreado y el contorno son independientes. Esta propiedad es solamente de lectura.

Sintaxis:

`Objeto.AssociativeHatch`

Donde:

Objeto: Hatch

`AssociativeHatch`: valor que indica la asociación del sombreado con el contorno. Variable de tipo Boolean:

`True`: Existe asociación entre el sombreado y el contorno.

`False`: No existe asociación entre el sombreado y el contorno.

3.4.10.6 AttachmentPoint

Propiedad que especifica la posición relativa del contorno del objeto MText con el punto de inserción del texto, y también establece la justificación relativa del texto. Este punto es independiente del punto de inserción, es decir, si cambia la posición relativa del contorno, cambiará el punto de inserción del objeto. Esta propiedad es de lectura y escritura.

Sintaxis:

`Objeto.AttachmentPoint`

Donde:

Objeto: Hatch

AttachmentPoint: es una variable de tipo acAttachmentPoint:

acAttachmentPointTopLeft: Especifica el punto de inserción en la parte superior izquierda del contorno del objeto MText.

acAttachmentPointTopCenter: Especifica el punto de inserción en la parte superior central del contorno del objeto MText.

acAttachmentPointTopRight: Especifica el punto de inserción en la parte superior derecha del contorno del objeto MText.

acAttachmentPointMiddleLeft: Especifica el punto de inserción en la parte intermedia izquierda del contorno del objeto MText.

acAttachmentPointMiddleCenter: Especifica el punto de inserción en la parte intermedia central del contorno del objeto MText.

acAttachmentPointMiddleRight: Especifica el punto de inserción en la parte intermedia derecha del contorno del objeto MText.

acAttachmentPointBottomLeft: Especifica el punto de inserción en la parte inferior izquierda del contorno del objeto MText.

acAttachmentPointBottomCenter: Especifica el punto de inserción en la parte inferior central del contorno del objeto MText.

acAttachmentPointBottomRight: Especifica el punto de inserción en la parte inferior derecha del contorno del objeto MText.

3.4.10.7 BasePoint

Propiedad que especifica el punto por donde pasa la línea auxiliar. Este punto resulta ser el punto inicial (Point1) cuando se crea la línea auxiliar. Esta propiedad es de lectura y escritura.

Sintaxis:

`Objeto.BasePoint`

Donde:

`Objeto: XLine`

`BasePoint`: Es una variable de tipo Variant (matriz de 3 elementos de tipo Double) que contienen las coordenadas del punto por donde pasa la línea auxiliar.

3.4.10.8 Center

Propiedad que contiene un punto con las coordenadas del centro de un círculo. Esta propiedad es de lectura y escritura.

Sintaxis:

`Objeto.Center`

Donde:

`Objeto: Circle.`

`Center`: Variable de tipo Variant (matriz de 3 elementos de tipo Double) que contiene las coordenadas del centro del círculo.

3.4.10.9 Closed

Propiedad que especifica si una polilínea está abierta o cerrada. Si la polilínea se especifica cerrada, la polilínea se cierra uniendo el punto inicial con el punto final de la polilínea. Esta propiedad es de lectura y escritura.

Sintaxis:

`Objeto.Closed`

Donde:

`Objeto`: Polyline

`Closed`: Variable de tipo Boolean:

`True`: La polilínea se encuentra cerrada.

`False`: La polilínea se encuentra abierta.

3.4.10.10 Color

Propiedad que especifica el color del objeto. Esta propiedad es de lectura y escritura.

Sintaxis:

`Objeto.Color`

Donde:

`Objeto`: Todos los objetos gráficos (Circle, DimAligned, DimRotated, Hatch, Line, Leader, MText, Polyline), Layer.

`Color`: Variable de tipo `acColor`. Esta variable puede ser los colores Standard en AutoCAD®, o un valor entre 0 y 256. Los colores Standard en AutoCAD® son:

- `acByBlock`: Color según bloque.

- `acByLayer`: Color según capa.
- `acRed`: Color rojo.
- `acYellow`: Color amarillo.
- `acGreen`: Color verde.
- `acCyan`: Color cyan.
- `acBlue`: Color azul.
- `acMagenta`: Color magenta.
- `acWhite`: Color blanco.

El color Standard `acByLayer` no es válido para el objeto Layer.

3.4.10.11 Coordinate

Propiedad que especifica o indica un vértice del objeto. El orden en que los vértices se colocaron para dibujar el objeto, les permite ubicarlos a través del índice que los identifica.

Sintaxis:

`Objeto.Coordinate (Index)`

Donde:

`Objeto`: Leader, Polyline.

`Coordinate`: Variable de tipo Variant (matriz de 3 elementos de tipo Double) que contiene las coordenadas del vértice especificado. Para polilíneas, representa las coordenadas X e Y, la coordenada Z está presente, pero no utilizada.

Index: Índice en la matriz de vértices del objeto que ubica al vértice en cuestión. El índice del primer punto es 0. Valor de tipo Integer.

3.4.10.12 EndPoint

Propiedad que especifica el punto final de un objeto cuya propiedad es aplicable. Esta propiedad es de lectura y escritura para la línea.

Sintaxis:

`Objeto.EndPoint`

Donde:

`Objeto: Line`

EndPoint: Variable de tipo Variant (matriz de 3 elementos de tipo Double) que contiene las coordenadas del punto final del objeto.

3.4.10.13 Height

Propiedad que especifica la altura del objeto donde esta propiedad es aplicable. Para el caso de los objetos Texto (MText), esta propiedad especifica la altura del texto en mayúsculas, y actúa como un factor de escala entre la altura y el ancho del texto. Esta propiedad es de lectura y escritura.

Sintaxis:

`Objeto.Height`

Donde:

`Objeto: MText`

Height: Valor de tipo Double, que contiene la altura del objeto, medida en unidades corrientes. Este valor debe ser positivo.

3.4.10.14 InsertionPoint

Propiedad que especifica el punto de inserción de un objeto cuya propiedad es aplicable. Para el objeto MText, especifica la ubicación del punto de inserción en función de la posición relativa del contorno del texto con este punto. Esta propiedad es de lectura y escritura.

Sintaxis:

`Objeto.InsertionPoint`

Donde:

`Objeto`: MText.

`InsertionPoint`: Variable de tipo Variant (matriz de 3 elementos de tipo Double) que contiene las coordenadas del punto de inserción.

3.4.10.15 Layer

Propiedad que especifica la capa de una entidad. Esta propiedad es de lectura y escritura.

Sintaxis:

`Objeto.Layer`

Donde:

`Objeto`: Todos los objetos gráficos (Line, Circle, Polyline, Leader, Hatch, DimAligned, DimRotated, MText).

`Layer`: Variable de tipo String, que contiene el nombre de la capa. La capa debe ser previamente creada para poder asignarla al objeto. Para crear una capa, se hace uso del método Add y se declara mediante su nombre de clase AcadLayer.

Sintaxis para crear una capa:

```
Set Capa = Objeto.Add(Name)
```

Donde:

Capa: Objeto declarado como AcadLayer.

Objeto: Layers Collection.

Name: Nombre de la capa a añadir a la colección Objeto. Variable de tipo String. Esta variable será, posteriormente, el nombre a definir en la variable Layer.

3.4.10.16 Length

Propiedad que contiene la longitud del segmento de línea, que es la distancia entre el punto inicial y el punto final de la línea.

La sintaxis es la siguiente:

```
Objeto.Length
```

Donde:

Objeto: Line

Length: valor de tipo Double, que contiene la longitud del segmento de línea

3.4.10.17 Linetype

Propiedad que especifica el tipo de línea de una entidad. Esta propiedad es de lectura y escritura.

Sintaxis:

```
Objeto.Linetype
```

Donde:

Objeto: Todos los objetos gráficos (Circle, DimAligned, DimRotated, Hatch, Line, Leader, MText, Polyline), Layer.

Linetype: Variable de tipo String, que contiene el nombre del tipo de línea. El tipo de línea debe ser cargado a AutoCAD® previamente antes de especificarla en una entidad cualquiera. Para cargar un tipo de línea, se hace uso de la siguiente sintaxis:

```
Objeto.Load LinetypeName, FileName
```

Donde:

Objeto: Linetypes Collection

LinetypeName: Nombre del tipo de línea a cargar. Valor de tipo String

FileName: Nombre del archivo donde se encuentra el tipo de línea cargado. Valor de tipo String.

3.4.10.18 LinetypeScale

Propiedad que especifica la escala de un tipo de línea de una entidad. Esta propiedad es de lectura y escritura.

Sintaxis:

```
Objeto.LinetypeScale
```

Donde:

Objeto: Todos los objetos gráficos (Circle, DimAligned, DimRotated, Hatch, Line, Leader, MText, Polyline).

`LinetypeScale`: Escala del tipo de línea, este valor debe ser positivo. Valor de tipo Double.

3.4.10.19 Measurement

Propiedad que contiene la medida de una acotación. Esta propiedad es solamente de lectura.

Sintaxis:

`Objeto.Measurement`

Donde:

`Objeto`: `DimAligned`, `DimRotated`

`Measurement`: Longitud de la medida de la acotación. Valor de tipo Double.

3.4.10.20 SecondPoint

Propiedad que especifica el segundo punto de una línea auxiliar. Este punto define la dirección de la línea auxiliar. Esta propiedad es de lectura y escritura.

Sintaxis:

`Objeto.SecondPoint`

Donde:

`Objeto`: `XLine`

`SecondPoint`: Variable de tipo Variant (matriz de 3 elementos de tipo Double) que representa las coordenadas del segundo punto de la línea auxiliar.

3.4.10.21 StartPoint

Propiedad que especifica el punto inicial de un objeto gráfico cuya propiedad es aplicable. Esta propiedad es de lectura y escritura para la línea.

Sintaxis:

`Objeto.StartPoint`

Donde:

`Objeto`: Line

`StartPoint`: Variable de tipo Variant (matriz de 3 elementos) que contiene las coordenadas del punto inicial del objeto.

3.4.10.22 TextGap

Propiedad que especifica la distancia entre la línea de cota y el texto dimensión cuando se interrumpe la línea de cota para acomodar el texto dimensión. Esta propiedad es de lectura y escritura.

Sintaxis:

`Objeto.TextGap`

Donde:

`Objeto`: DimAligned, DimRotated, Leader.

`TextGap`: Distancia vacía entre el texto dimensión y la línea de cota. Valor de tipo Double

3.4.10.23 VerticalTextPosition

Propiedad que especifica la posición vertical del texto con respecto a la línea de dimensión.

Sintaxis:

`Objeto.VerticalTextPosition`

Donde:

`Objeto`: `DimAligned`, `DimRotated`, `Leader`.

`VerticalTextPosition`: Variable de tipo `acDimVerticalJustification`:

`acVertCentered`: Centra el texto entre las dos partes de la línea de cota.

`acAbove`: Ubica el texto de cota sobre la línea de cota. La distancia desde la línea de cota hasta la línea de base de la línea más baja del texto es el intervalo de texto actual.

`acOutside`: Coloca el texto de cota al lado de la línea de cota situada más lejos del primer punto de definición.

`acJIS`: Coloca el texto de cota de forma que se adapte a una representación JIS (Japanese Industrial Standards, Normativa industrial japonesa).

3.4.11 Métodos

3.4.11.1 AppendOuterLoop

Método que añade un bucle externo al sombreado. Un bucle es una curva o una sucesión de curvas conectadas que definen un área en un plano con un contorno que no se interseca entre sí. Un bucle puede estar constituido por una combinación de líneas, polilíneas, círculos, arcos, elipses, arcos elípticos y polilíneas suavizadas. Los objetos que conforman los bucles deben ser objetos cerrados o formar áreas cerradas que compartan los puntos finales con otros objetos.

Sintaxis:

```
Objeto.AppendOuterLoop Loop
```

Donde:

Objeto: Hatch

Loop: matriz de objetos formando un contorno cerrado o bucle.

Luego de haber creado el sombreado usando el método AddHatch, se añade el bucle externo, que va a tener de relleno el sombreado creado, a partir de este procedimiento.

3.4.11.2 AppendVertex

Método que añade un vértice al final de un objeto creado, el cual este método es aplicable, generando sobre este objeto un nuevo segmento entre el punto final del objeto y el vértice añadido.

Sintaxis:

```
Objeto.AppendVertex Vertex
```

Donde:

Objeto: Polyline

Vertex: Variable de tipo Variant (matriz de tres elementos de tipo Double) que contiene las coordenadas del vértice a ser añadido en el objeto. La coordenada Z está presente, pero no utilizada.

3.4.11.3 Copy

Método que copia un objeto creado. El objeto resultante está sobre el objeto original.

Sintaxis:

```
Set CopiaObjeto = Objeto.Copy()
```

Donde:

Objeto: Todos los objetos gráficos (Circle, DimAligned, DimRotated, Hatch, Line, Leader, MText, Polyline).

CopiaObjeto: Variable de tipo Object. En la mayoría de los casos, el objeto copia es declarado como el objeto original, es decir, CopiaObjeto es declarado como haya sido declarado Objeto.

3.4.11.4 Delete

Método que borra un objeto.

Sintaxis:

```
Objeto.Delete
```

Donde:

Objeto: Todos los objetos gráficos (Circle, DimAligned, DimRotated, Hatch, Line, Leader, MText, Polyline), Layer.

3.4.11.5 IntersectWith

Método que obtiene los puntos comunes entre dos objetos. Básicamente, determina los puntos de intersección de un objeto sobre otro.

Sintaxis:

```
Puntos = Objeto.IntersectWith(IntersectObject,  
ExtendOption)
```

Donde:

Puntos: Valor de tipo Variant (matriz de elementos de tipo Double) que contiene los puntos comunes entre Objeto y IntersectObject.

Objeto: Todos los objetos gráficos (Circle, DimAligned, DimRotated, Hatch, Line, Leader, MText, Polyline), Layer.

IntersectObject: Variable de tipo Object. Puede ser uno de los objetos gráficos (Circle, DimAligned, DimRotated, Hatch, Line, Leader, MText, Polyline), Layer.

ExtendOption: Variable de tipo AcExtendOption:

acExtendNone: no se extienden ningún objeto

acExtendThisEntity: Extiende el objeto base (Objeto).

acExtendOtherEntity: Extiende el objeto pasado por argumento (IntersectObject).

acExtendBoth: Se extienden ambos objetos.

3.4.11.6 Move

Método que permite mover un objeto de un punto origen a un punto destino, es decir, mueve un objeto según un vector definido entre el punto origen y el punto destino. Básicamente, mueve un elemento una distancia determinada por la longitud del vector formado entre el punto origen y el punto destino y en un sentido determinado por el ángulo de ese mismo vector.

Sintaxis:

```
Objeto.Move FromPoint, ToPoint
```

Donde:

Objeto: Todos los objetos gráficos (Circle, DimAligned, DimRotated, Hatch, Line, Leader, MText, Polyline).

FromPoint: Punto de origen. Variable de tipo Variant (matriz de 3 elementos de tipo Double).

ToPoint: Punto de destino. Variable de tipo Variant (matriz de 3 elementos de tipo Double).

3.4.11.7 Offset

Método utilizado para crear un nuevo objeto, cuya propiedad sea aplicable, a una distancia de separación de la línea existente.

Sintaxis:

```
NuevoObjeto = Objeto.Offset(Distance)
```

Donde:

Objeto: Circle, Line, Polyline.

NuevoObjeto: Variable de tipo Variant (matriz de objetos), que contiene los nuevos objetos creados.

Distance: Distancia de separación entre la línea original y la línea creada. Esta distancia puede ser positiva o negativa, pero nunca puede ser nulo. Valor de tipo Double

3.4.11.8 Rotate

Método que permite girar o rotar la línea alrededor de un punto fijo con un ángulo de rotación

Sintaxis:

```
Objeto.Rotate BasePoint, RotationAngle
```

Donde:

Objeto: Todos los objetos gráficos (Circle, DimAligned, DimRotated, Hatch, Line, Leader, MText, Polyline).

BasePoint: Punto base de rotación de la línea. Variable de tipo Variant (matriz de tres elementos de tipo Double).

RotationAngle: Ángulo de rotación en radianes. Este ángulo es definido a partir del eje horizontal X en sentido contrario a las agujas del reloj. Valor de tipo Double.

3.5 ESCALERAS

Serie de escalones que sirven para subir y bajar, donde se denomina a los escalones como cada una de las partes en que se apoya el pie para subir o bajar.

3.5.1 Especificaciones constructivas generales

Los requisitos mínimos de una escalera varían de una norma a otra. Sin embargo, algunas de las más comunes son:

Se denomina con “*s*” a la contrahuella (altura del escalón), con “*a*” a la huella (profundidad del escalón) y se elige normalmente la relación:

$2s + a = 62 - 64\text{cm}$, y en casos especiales la relación $s + a = 48\text{cm}$. En escaleras monumentales, $s = 13 - 15\text{cm}$, en casas-habitación $s = 15 - 17.5\text{cm}$, y en escaleras secundarias o en casas económicas, $s_{\text{máx}} = 20\text{cm}$.

El ancho de la escalera suele hacerse en casas-habitación de 75-130 cm. Y en edificios públicos de 1.90 a 3.00 m. El número de escalones cada tramo no debe pasar de 15.

Las escaleras de concreto armado se caracterizan por ser resistentes al fuego, en el sentido de las normas correspondientes.

De acuerdo con la forma de la planta, se distinguen: escaleras rectas (de un único o varios tramos), o bien curvas, escalones rectangulares, en abanico (compensados) y curvados; descansos esquineros, descansos centrales e intermedios (rellanos)

3.5.2 Cálculo de escaleras de concreto armado

En general, las escaleras con sus descansos, al ser proyectadas sobre sus plantas, pueden ser transformadas en estructuras ideales constituidas por vigas y losas de eje horizontal, cuyo comportamiento estático es similar al de la estructura real.

Todas las cargas, dimensiones longitudinales, momentos, esfuerzos de corte, etc. se refieren por conveniencia a esta estructura ideal. Esto se consigue introduciendo como peso propio $g = \gamma d \cos \alpha$ para todas las losas inclinadas de un ángulo α , espesor d , y con peso específico γ ; utilizando además para el peso de los escalones $g' = \gamma s/2$.

La sobrecarga se refiere directamente a la proyección en planta.

Las reacciones de vínculo pueden ser absorbidas:

- Por medio de viga de descanso:

La viga de descanso es estáticamente el medio de apoyo más sencillo, pero actualmente casi no se emplea. El apoyo sobre una viga de descanso puede ser considerado rígido, lo cual es de importancia para los momentos y las reacciones de vínculo de la viga continua.

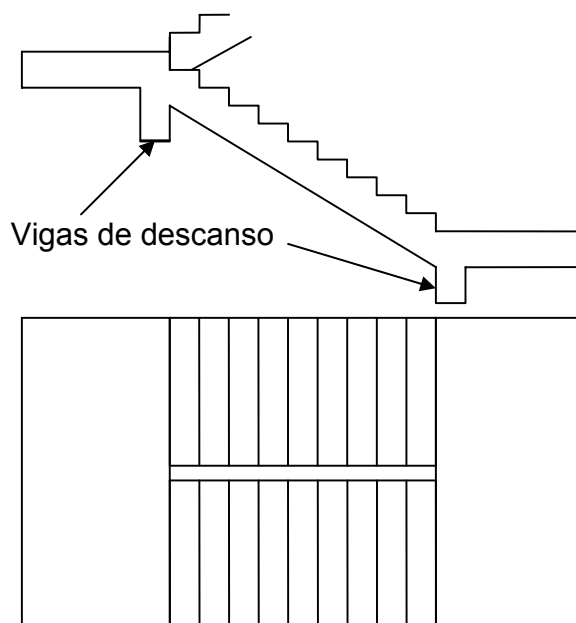


Fig. 1 Escalera con vigas de descanso

- Por medio de viga disimulada en el descanso:
-

En oposición a la viga visible de descanso la viga disimulada constituye un apoyo elástico. Por su deformación elástica disminuyen considerablemente tanto los momentos de apoyo M_B y M_C como también las reacciones de vínculo B y C

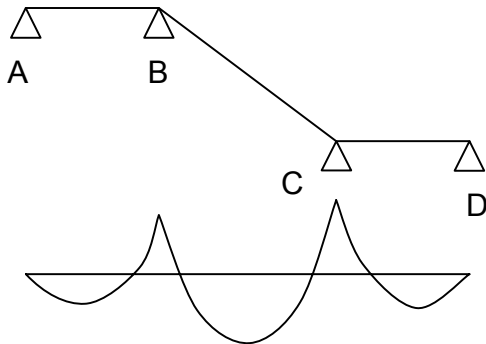


Fig. 2

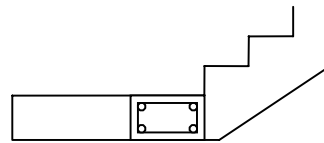


Fig. 3 Viga disimulada en el descanso

(Fig. 3)

- Por medio de la losa del descanso que actúa como una placa apoyada en tres de sus bordes y que recibe en su borde libre cargas adicionales transmitidas por el tramo:

Este método de cálculo puede ser empleado tanto para el tramo como para el descanso.

- Por el efecto espacial de estructura plegada de la escalera:

Si se considera a la estructura con descansos como estructura plegada (Faltwerk), las aristas constituyen apoyos rígidos. Descanso y tramo se calculan entonces en forma tal como si estuvieran apoyados en sus aristas, trabajando como una viga continua según Fig. 3 o Fig. 4. Las reacciones de vínculo en las aristas constituyen así las cargas de esta estructura plegada. Estas reacciones de vínculo B son absorbidas por efecto de lámina de los tramos y descansos (Scheibenwirkung). Se entiende como B a la carga uniforme equivalente a las verdaderas reacciones de vínculo.

El caso más simple lo constituye la escalera de la figura 15. La carga B se transmite a las placas y produce un efecto de estructura aporricada común con

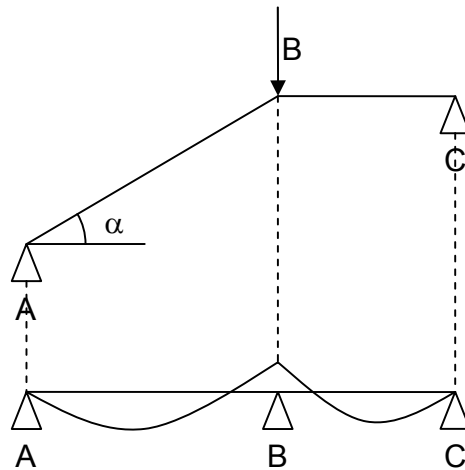


Fig. 4 Estructura plegada

los esfuerzos normales:

$$N_{AB} = B/\sin \alpha$$

$$N_{AC} = B \cot \alpha$$

Estos esfuerzos normales deben tenerse en cuenta al dimensionar y especialmente verificar su absorción por los apoyos A y B .

3.5.3 Escalera autoportante con descanso

Naturalmente, es factible construir escaleras sobre plantas de cualquier forma. También es verdad que no siempre resultará fácil encontrar la solución completa en base a fórmulas directas. Es casi imposible establecer fórmulas válidas de carácter general, y más bien será necesario aspirar a que con el tiempo se establezca una colección de fórmulas que abarque los posibles tipos de conformación de escaleras.

Un tipo de escalera autoportante muy usado en construcciones de carácter representativo es la escalera recta, con descansos, empotrada solamente en los extremos de sus dos tramos y sin otro apoyo adicional.

Se presenta un método directo de análisis de escaleras de descanso libre con partes superior e inferior iguales y sin apoyo en el descanso, para carga simétrica. El método se basa en la simplificación de Fuchsteiner de la estructura de la escalera en una armadura espacial compuesta de barras lineales. Las deformaciones son calculadas por integración con la aplicación del principio de trabajo mínimo, y las redundancias son determinadas resolviendo las ecuaciones elásticas. La selección de un sistema estáticamente determinado apropiado permite un alcance directo y elimina cálculos tediosos de momentos desconocidos mediante la superposición de diferentes condiciones de apoyo. Esta selección permite la consideración de condiciones de apoyo fijo y apoyo simple., en los niveles superior e inferior. Introduciendo dos simplificaciones menores, el método es además desarrollado para incluir fórmulas de trabajo para las redundancias.

Como aproximación se supone el ojo de la escalera reducido, y los momentos de inercia de tramo y descanso son idénticos.

Se supone también que la escalera está rígidamente apoyada en al arista de quiebre; en tal caso es posible calcular los momentos para el tramo y para el descanso. Por conveniencia se multiplica la carga del descanso q_p por el factor $(1 + m/2b)$ para tener en cuenta la parte de carga que actúa junto al ojo de la escalera. En este caso se origina en la arista de quiebre de cada tramo una reacción de vínculo B , la que distribuye sobre el ancho del tramo como carga lineal. Esta reacción de vínculo no existe en realidad y por ello hay que introducirla como carga adicional imaginaria, que actúa de arriba abajo en el sistema supuesto sin peso.

Para tener en cuenta el efecto de la carga B, se simplifica el sistema, prescindiendo de las esquinas del descanso, transformándolos en una placa semicircular, resulta entonces $r = \frac{1}{2}(b + m)$. Se hace el sistema estáticamente determinado por medio de un corte en el centro del descanso. Para carga simétrica quedan solamente dos magnitudes estáticamente indeterminadas.

En los métodos de cálculo para la escalera autoportante se presentan las ecuaciones que rigen este cálculo.

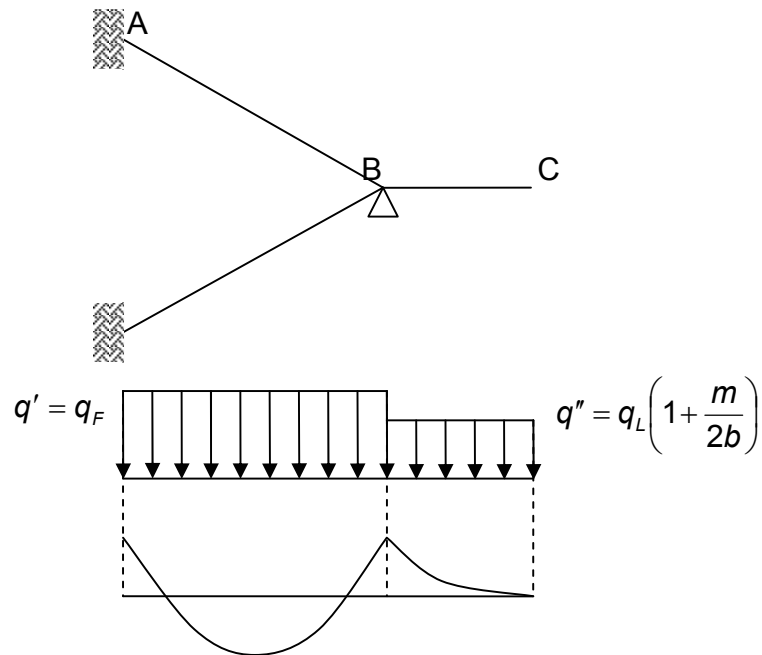


Fig. 5 Modelo estructural de la escalera autoportante

3.6 CONSIDERACIONES PARA EL CÁLCULO DEL REFUERZO DE ACERO

En la determinación de algunos parámetros para la realización de este programa, se toman en cuenta algunas consideraciones basadas en la norma COVENIN-MINDUR 1753-85:

Acero transversal, acero mínimo y acero para losas macizas de espesor uniforme:

Artículo 10.5.2: “Para losas macizas de espesor uniforme, el área de acero mínima y la separación de las armaduras principales serán las que se requieren por retracción y temperatura de acuerdo al artículo 7.12”

Artículo 7.12.1: “Para contrarrestar los esfuerzos de retracción y temperatura en losas estructuralmente armadas en una dirección, se proporcionará una armadura de repartición perpendicular a la principal.”

Artículo 7.12.2.1: “La relación del área de la armadura de retracción y temperatura respecto al área de la sección total del concreto no será menor de los valores siguientes:

- a. En losas donde se empleen barras estriadas de grado 28 0 35.....0.0020
- b. En losas donde se empleen barras estriadas o malla de alambre soldada (lisa o estriada), grado 42.....0.0018
- c. En losas donde se utiliza refuerzo cuya resistencia cedente sea mayor que 4200 kgf/cm^2 $0.018 \times 4200/f_y$

Pero, en ningún caso será menor que 0.0014.”.

4 METODOLOGIA

4.1 METODOS DE CÁLCULO EN LA DETERMINACION DE LAS SOLICITACIONES DE LAS ESCALERAS.

Antes de iniciar la elaboración de los procedimientos que formarán parte del programa, hay que definir cuales son los métodos de cálculo de las solicitaciones para cada una de las escaleras, que serán utilizados para realizar el análisis y diseño de las escaleras de concreto armado estudiadas.

A continuación se describen las ecuaciones utilizadas en el cálculo de las solicitaciones de interés correspondientes a cada una de las escaleras en cuestión.

4.1.1 Escalera autoportante

Según el método de Fuchsteiner, las ecuaciones generales para las solicitaciones son:

$$V_X = V_{X_o} + V_{X_1} X_1 + V_{X_2} X_2 \quad (1)$$

$$V_Y = V_{Y_o} + V_{Y_1} X_1 + V_{Y_2} X_2 \quad (2)$$

$$N = N_o + N_1 X_1 + N_2 X_2 \quad (3)$$

$$M_X = M_{X_o} + M_{X_1} X_1 + M_{X_2} X_2 \quad (4)$$

$$M_Y = M_{Y_o} + M_{Y_1} X_1 + M_{Y_2} X_2 \quad (5)$$

$$M_T = M_{T_o} + M_{T_1} X_1 + M_{T_2} X_2 \quad (6)$$

Donde:

1. Corte en dirección X.
 2. Corte en dirección Y.
 3. Fuerza Axial.
 4. Momento en X.
 5. Momento en Y.
 6. Momento de torsión.
-

Estas ecuaciones son el resultado de la superposición de dos sistemas estáticos, donde X_1 y X_2 son las redundantes de cada sistema estático.

$$\left. \begin{array}{l} X_1 = X_1' + X_1'' \\ X_2 = X_2' + X_2'' \end{array} \right\} \quad (7)$$

$$\left. \begin{array}{l} X_1' = -Aq'l^2 \\ X_2' = -Bq'l^2 \\ X_1'' = -Cq''l^2 \\ X_2'' = -Dq''l^2 \end{array} \right\} \quad (8)$$

Donde:

$$\left. \begin{array}{l} A = \frac{3\pi\gamma + 4\cos\alpha + 18\pi\gamma^3\cos^2\alpha}{4(6\pi\gamma + 8\cos\alpha + 9\pi\gamma^3\cos^2\alpha)} \\ B = \frac{3\gamma\cos\alpha}{6\pi\gamma + 8\cos\alpha + 9\pi\gamma^3\cos^2\alpha} \\ C = \frac{2\gamma^2(9\pi\gamma^2\cos^2\alpha(1+\gamma) - 3\pi\gamma - 4\cos\alpha)}{6\pi\gamma + 8\cos\alpha + 9\pi\gamma^3\cos^2\alpha} \\ D = \frac{8\gamma^2(2+3\gamma)\cos\alpha}{6\pi\gamma + 8\cos\alpha + 9\pi\gamma^3\cos^2\alpha} \end{array} \right\} \quad (9)$$

$$\begin{aligned} \gamma &= \frac{r}{l} \\ r &\cong \frac{b}{2}; m \cong 0 \\ \alpha &= \arctg\left(\frac{h}{l}\right) \end{aligned} \quad (10)$$

q' = Carga vertical sobre el tramo de escalera.

q'' = Carga vertical sobre el descanso de la escalera.

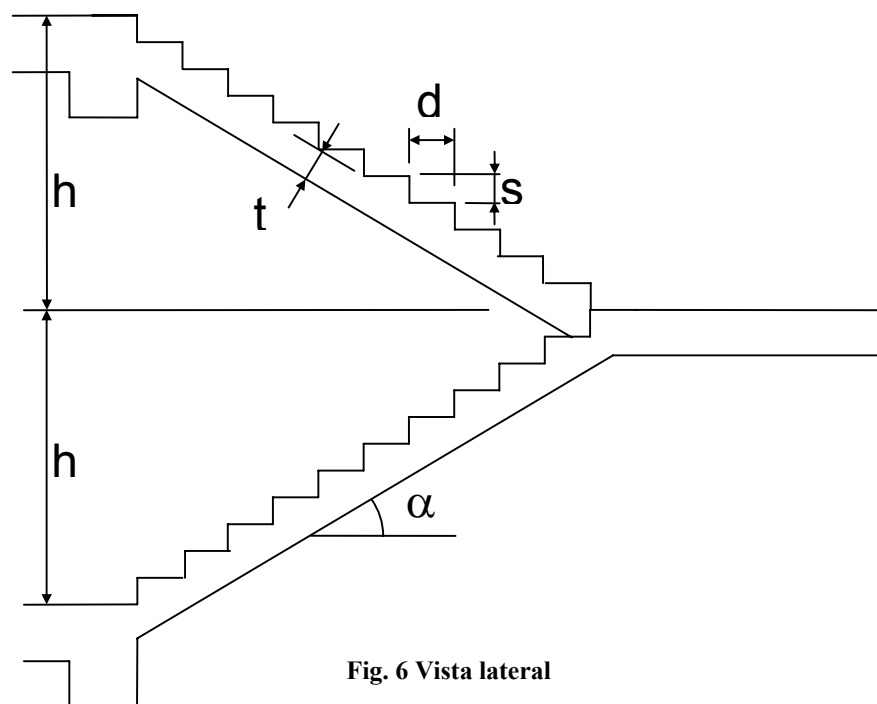


Fig. 6 Vista lateral

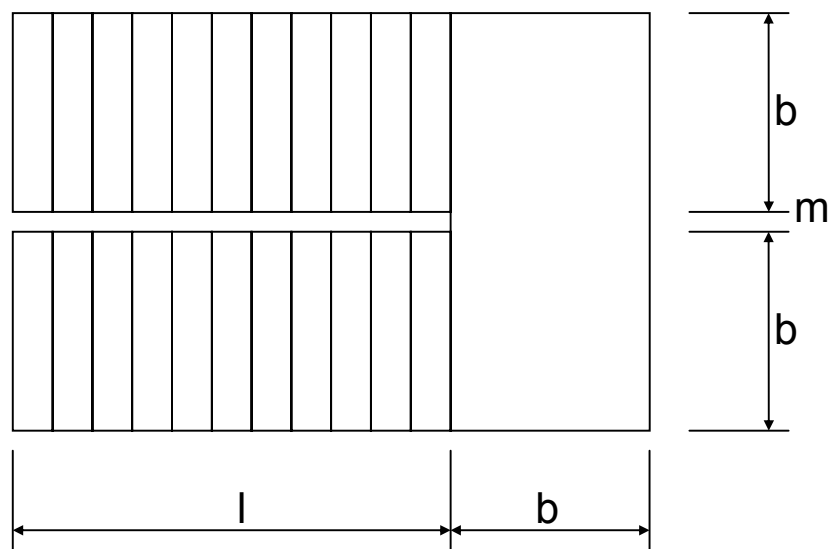


Fig. 7 Vista en planta

Momentos y fuerzas resultantes para $X_1 = 1$

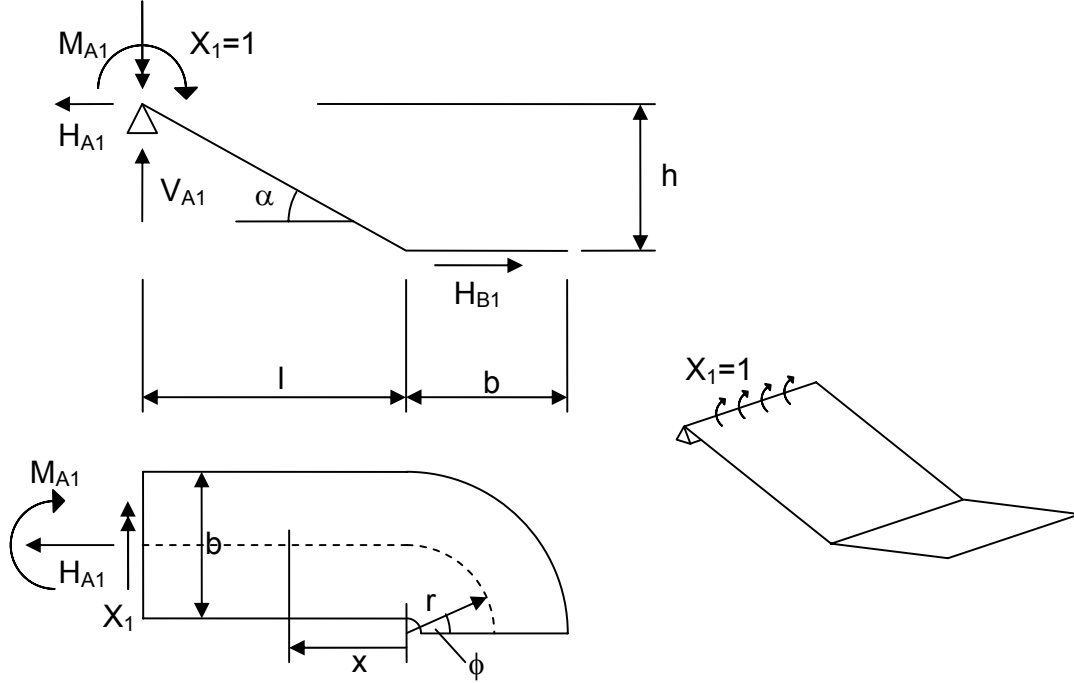


Fig. 8 Momentos y fuerzas resultantes para $X_1 = 1$

$$\left. \begin{aligned} H_{B_1} &= H_{A_1} = \frac{1}{h} \\ V_{A_1} &= 0 \\ M_{A_1} &= \frac{r}{h} \end{aligned} \right\} \quad (11)$$

En el descanso:

$$\left. \begin{aligned} Q_{X_1} &= 0 \\ Q_{Y_1} &= -H_{B_1} \cos \phi = \frac{-\cos \phi}{h} \\ N_1 &= \pm H_{B_1} \sin \phi = \frac{\pm \sin \phi}{h} \\ M_{X_1} &= 0 \\ M_{Y_1} &= \pm H_{B_1} r \sin \phi = \frac{\pm r}{h} \sin \phi \\ M_{T_1} &= 0 \end{aligned} \right\} \quad (12)$$

En el tramo de escalera:

$$\left. \begin{aligned} Q_{X_1} &= -H_{B_1} \sin \alpha = \frac{-\sin \alpha}{h} \\ Q_{Y_1} &= 0 \\ N_1 &= \pm H_{B_1} \cos \alpha = \frac{\pm \cos \alpha}{h} \\ M_{X_1} &= H_{B_1} \sin \alpha \frac{x}{\cos \alpha} = \frac{x}{h} - \tan \alpha \\ M_{Y_1} &= \pm H_{B_1} r \cos \alpha = \frac{\pm r}{h} \cos \alpha \\ M_{T_1} &= \pm H_{B_1} r \sin \alpha = \frac{\pm r}{h} \sin \alpha \end{aligned} \right\} \quad (13)$$

Momentos y fuerzas resultantes para $X_2 = 1$

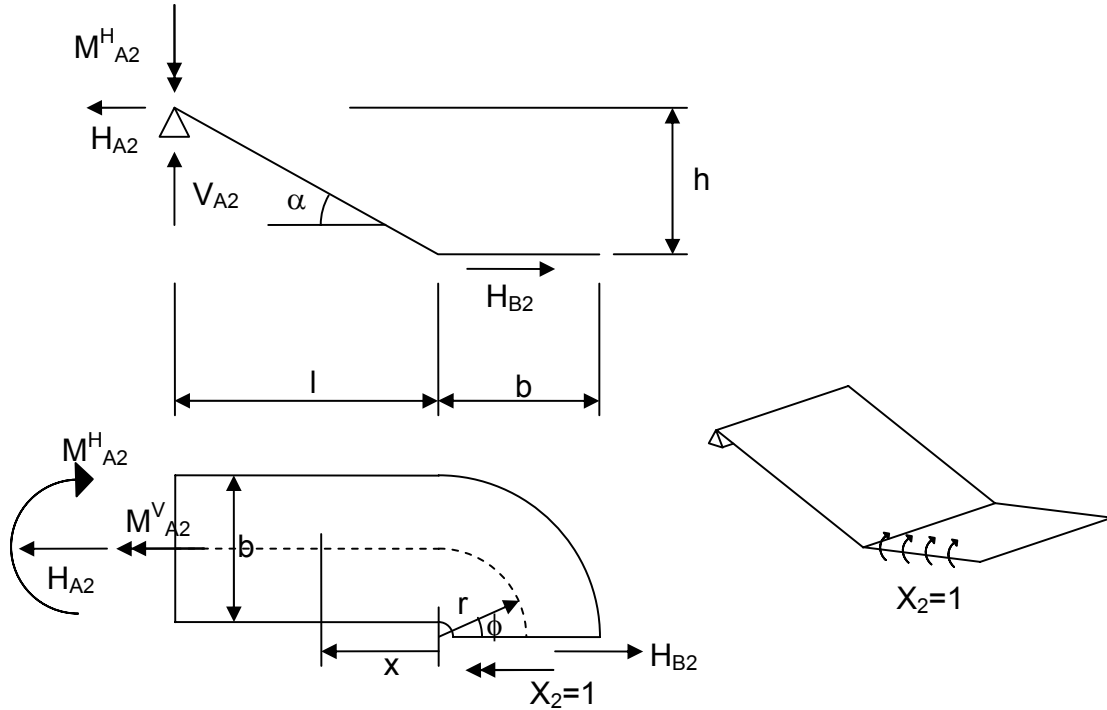


Fig. 9 Momentos y fuerzas resultantes para $X_2 = 1$

$$\left. \begin{aligned} H_{B_2} &= H_{A_2} = 0 \\ V_{A_2} &= 0 \\ M_{A_2}^H &= 0 \\ M_{A_2}^V &= 1 \end{aligned} \right\} \quad (14)$$

En el descanso:

$$\left. \begin{aligned} Q_{X_2} &= 0 \\ Q_{Y_2} &= 0 \\ N_2 &= 0 \\ M_{X_2} &= \cos \phi \\ M_{Y_2} &= 0 \\ M_{T_2} &= \pm \sin \phi \end{aligned} \right\} \quad (15)$$

En el tramo de escalera:

$$\left. \begin{aligned} Q_{X_2} &= 0 \\ Q_{Y_1} &= 0 \\ N_2 &= 0 \\ M_{X_2} &= 0 \\ M_{Y_2} &= \mp \sin \alpha \\ M_{T_2} &= \pm \cos \alpha \end{aligned} \right\} \quad (16)$$

Momentos y fuerzas resultantes para $q' = 1$

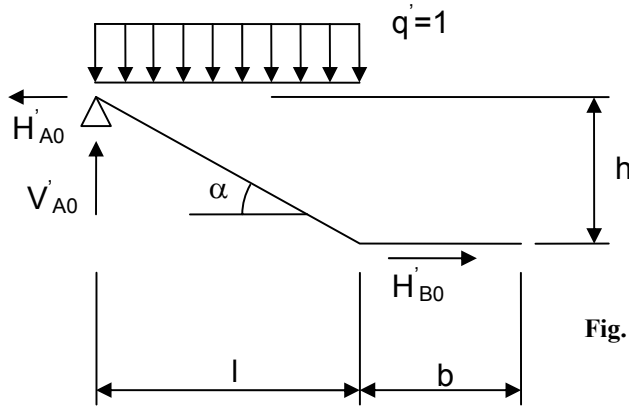


Fig. 10 Momentos y fuerzas resultantes para $q' = 1$

$$\begin{aligned} V'_{A_0} &= q'l \\ H'_{A_0} &= H'_{B_0} = \frac{q'l^2}{2h} \end{aligned} \quad (17)$$

En el descanso:

$$\begin{aligned} Q'_{X_0} &= 0 \\ Q'_{Y_0} &= -H'_{B_0} \cos \phi = -\frac{q'l^2}{2h} \cos \phi \\ N'_0 &= \pm H'_{B_0} \sin \phi = \pm \frac{q'l^2}{2h} \sin \phi \\ M'_{X_0} &= 0 \\ M'_{Y_0} &= \pm H'_{B_0} r \sin \phi = \pm \frac{q'l^2}{2h} r \sin \phi \\ M'_{T_0} &= 0 \end{aligned} \quad (18)$$

En el tramo de escalera:

$$\begin{aligned} Q'_{X_0} &= -H'_{B_0} \sin \alpha + q'x \cos \alpha = -\frac{q'l^2}{2h} \sin \alpha + q'x \cos \alpha \\ Q'_{Y_0} &= 0 \\ N'_0 &= \pm H'_{B_0} \cos \alpha \pm q'x \sin \alpha = \pm \frac{q'l^2}{2h} \cos \alpha \pm q'x \sin \alpha \\ M'_{X_0} &= H'_{B_0} x \tan \alpha - \frac{1}{2} q'x^2 = \frac{q'l^2}{2h} x \tan \alpha - \frac{1}{2} q'x^2 \\ M'_{Y_0} &= \pm H'_{B_0} r \cos \alpha = \pm \frac{q'l^2}{2h} r \cos \alpha \\ M'_{T_0} &= \pm H'_{B_0} r \sin \alpha = \pm \frac{q'l^2}{2h} r \sin \alpha \end{aligned} \quad (19)$$

Momentos y fuerzas resultantes para $q'' = 1$

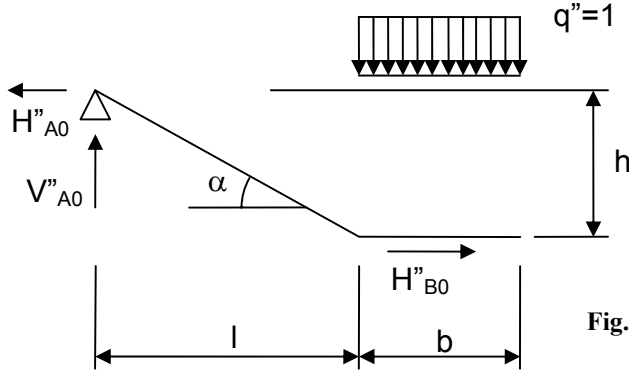


Fig. 11 Momentos y fuerzas resultantes para $q'' = 1$

En el descanso:

$$\left. \begin{aligned} V''_{A_0} &= q''b \\ H''_{A_0} &= H''_{B_0} = \frac{q''b}{2h}(2l+b) \end{aligned} \right\} \quad (20)$$

$$\left. \begin{aligned} Q'_{x_0} &= q''b \sin \phi \\ Q''_{y_0} &= -H''_{B_0} \cos \phi = -\frac{q''b}{2h}(2l+b) \cos \phi \\ N''_0 &= -H''_{B_0} \sin \phi = -\frac{q''b}{2h}(2l+b) \sin \phi \\ M''_{x_0} &= -\frac{q''b^2}{2} \sin \phi \\ M''_{y_0} &= \pm H''_{B_0} r \sin \phi = \pm \frac{q''b}{2h}(2l+b) r \sin \phi \\ M''_{T_0} &= 0 \end{aligned} \right\} \quad (21)$$

En el tramo de escalera:

$$\left. \begin{aligned} Q''_{x_0} &= -H''_{B_0} \sin \alpha + q''b \cos \alpha = -\frac{q''b}{2h}(2l+b) \sin \alpha + q''b \cos \alpha \\ Q''_{y_0} &= 0 \\ N''_0 &= \pm H''_{B_0} \cos \alpha \pm q''b \sin \alpha = \pm \frac{q''b}{2h}(2l+b) \cos \alpha \pm q''b \sin \alpha \\ M''_{x_0} &= H''_{B_0} x \tan \alpha - q''b \left(x + \frac{b}{2} \right) = \frac{q''b}{2h}(2l+b) x \tan \alpha - q''b \left(x + \frac{b}{2} \right) \\ M''_{y_0} &= \pm H''_{B_0} r \cos \alpha = \pm \frac{q''b}{2h}(2l+b) r \cos \alpha \\ M''_{T_0} &= \pm H''_{B_0} r \sin \alpha = \pm \frac{q''b}{2h}(2l+b) r \sin \alpha \end{aligned} \right\} \quad (22)$$

Entonces:

$$\begin{aligned}V_{x_0} &= M'_{x_0} + M''_{x_0} \\V_{y_0} &= V'_{x_0} + V''_{x_0} \\N_0 &= N'_0 + N''_0 \\M_{x_0} &= M'_{x_0} + M''_{x_0} \\M_{y_0} &= M'_{y_0} + M''_{y_0} \\M_{T_0} &= M'_{T_0} + M''_{T_0}\end{aligned} \quad (23)$$

4.1.2 Escalera de un tramo con dos descansos y escalera de dos tramos con tres descansos

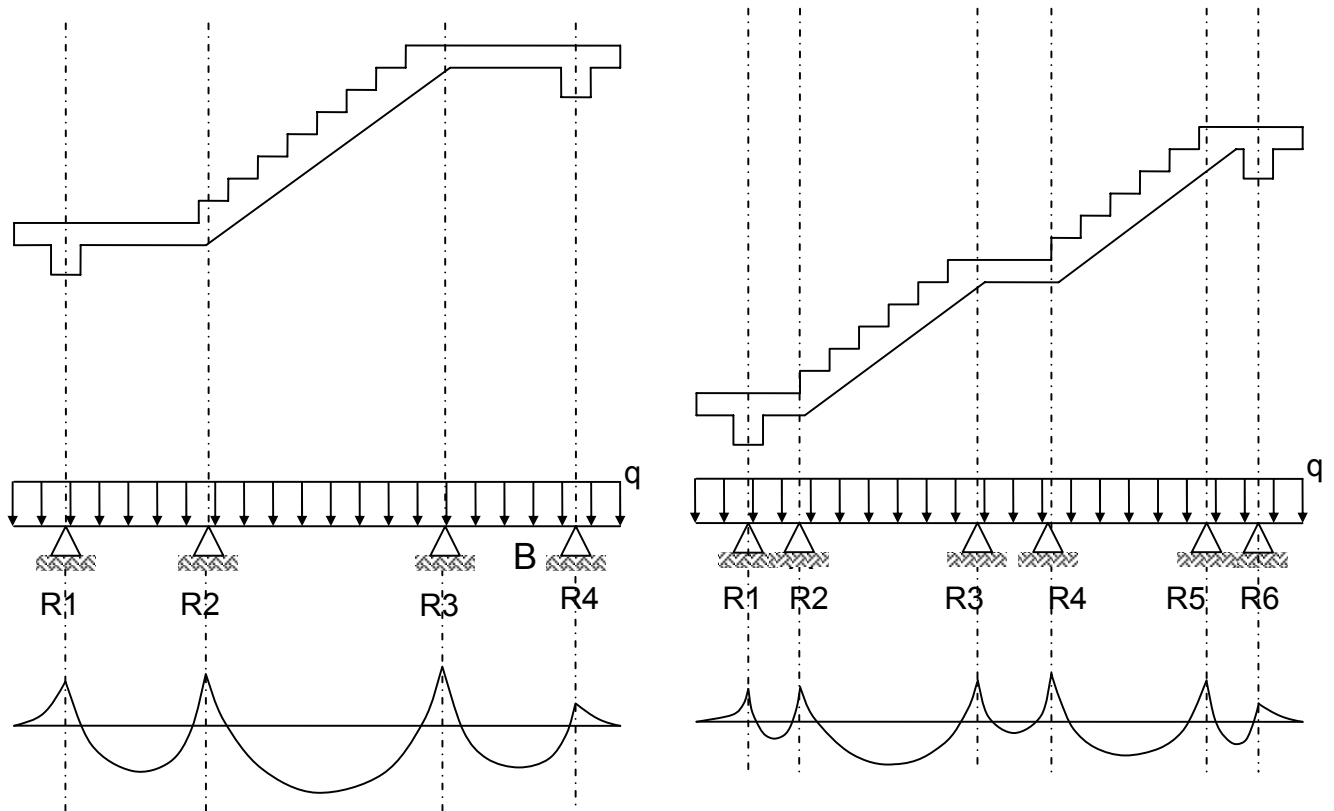


Fig. 12 Modelo hiperestático de las escaleras con apoyos

En la Fig. 12 se muestra el modelo estructural para el cálculo de las solicitaciones de estas escaleras, como una viga con múltiples apoyos fijos. En realidad, sólo hay dos apoyos fijos (los que pasan por las vigas), los demás apoyos son las reacciones de las aristas como apoyos fijos, recurriendo al método de cálculo de la escalera como una estructura plegada. Aquí se presenta dos estructuras hiperestática con articulaciones fijas como apoyos. La carga “q” es la combinación de la carga permanente y de la carga variable a la cual se encuentra sometida la escalera.

Los valores de las solicitaciones serán el resultado de resolver un sistema de ecuaciones de seis incógnitas, para la escalera “a”, y de ocho incógnitas, para la escalera “b”, de la Fig. 12. El sistema de ecuaciones lineales es:

$$\begin{bmatrix} 1 & X_1 & 0 & 0 & 0 & 0 \\ 1 & X_2 & \frac{(X_2 - X_1)^3}{6EI} & 0 & 0 & 0 \\ 1 & X_3 & \frac{(X_3 - X_1)^3}{6EI} & \frac{(X_3 - X_2)^3}{6EI} & 0 & 0 \\ 1 & X_4 & \frac{(X_4 - X_1)^3}{6EI} & \frac{(X_4 - X_2)^3}{6EI} & \frac{(X_4 - X_3)^3}{6EI} & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & L - X_1 & L - X_2 & L - X_3 & L - X_4 \end{bmatrix} \begin{bmatrix} y_0 \\ \theta_0 \\ R_1 \\ R_2 \\ R_3 \\ R_4 \end{bmatrix} = \begin{bmatrix} \frac{qX_1^2}{24EI} \\ \frac{qX_2^2}{24EI} \\ \frac{qX_3^2}{24EI} \\ \frac{qX_4^2}{24EI} \\ \frac{qL}{24EI} \\ \frac{qL^2}{2} \end{bmatrix}$$

$$\begin{bmatrix} 1 & X_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & X_2 & \frac{(X_2 - X_1)^3}{6EI} & 0 & 0 & 0 & 0 & 0 \\ 1 & X_3 & \frac{(X_3 - X_1)^3}{6EI} & \frac{(X_3 - X_2)^3}{6EI} & 0 & 0 & 0 & 0 \\ 1 & X_4 & \frac{(X_4 - X_1)^3}{6EI} & \frac{(X_4 - X_2)^3}{6EI} & \frac{(X_4 - X_3)^3}{6EI} & 0 & 0 & 0 \\ 1 & X_5 & \frac{(X_5 - X_1)^3}{6EI} & \frac{(X_5 - X_2)^3}{6EI} & \frac{(X_5 - X_3)^3}{6EI} & \frac{(X_5 - X_4)^3}{6EI} & 0 & 0 \\ 1 & X_6 & \frac{(X_6 - X_1)^3}{6EI} & \frac{(X_6 - X_2)^3}{6EI} & \frac{(X_6 - X_3)^3}{6EI} & \frac{(X_6 - X_4)^3}{6EI} & \frac{(X_6 - X_5)^3}{6EI} & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & L - X_1 & L - X_2 & L - X_3 & L - X_4 & L - X_5 & L - X_6 \end{bmatrix} \begin{bmatrix} y_0 \\ \theta_0 \\ R_1 \\ R_2 \\ R_3 \\ R_4 \\ R_5 \\ R_6 \end{bmatrix} = \begin{bmatrix} \frac{qX_1^2}{24EI} \\ \frac{qX_2^2}{24EI} \\ \frac{qX_3^2}{24EI} \\ \frac{qX_4^2}{24EI} \\ \frac{qX_5^2}{24EI} \\ \frac{qX_6^2}{24EI} \\ \frac{qL}{24EI} \\ \frac{qL^2}{2} \end{bmatrix}$$

(24)

Estos sistemas de ecuaciones son resueltos en el módulo Standard Matrices, a través de eliminación Gaussiana con pivoteo.

Teniendo todas estas variables, se aplica las ecuaciones generales para la resolución de vigas hiperestáticas, con N , el número de reacciones o vínculos (estas ecuaciones están condicionadas teniendo sólo carga en dirección vertical como única carga presente en el modelo estructural, no se consideran movimientos de apoyo, las condiciones iniciales indican que $V = 0$ y $M = 0$ para $X = 0$):

$$\begin{aligned} V(X) &= \sum_{k=1}^N R_k \langle X - X_k \rangle^0 - qX \\ M(X) &= \sum_{k=1}^N R_k \langle X - X_k \rangle - \frac{qX^2}{2} \end{aligned} \quad (25)$$

Para los momentos máximos, se realiza el cálculo de este momento, tomando en cuenta los apoyos que pasan por los ejes, es decir, el modelo estructural es una viga simplemente apoyada, sin considerar las aristas como apoyos fijos.

4.1.3 Cálculo del área de acero.

Luego de obtenidos los momentos máximos y los momentos mínimos, se realiza el cálculo del área de acero para cada uno de los momentos en las secciones críticas.

Para la escalera autoportante, se determina el área de acero en:

1. Los apoyos.
2. La sección donde el momento es máximo en el tramo de escalera
3. Las secciones en el descanso donde el momento es máximo o mínimo (mitad del descanso, entre el descanso y la escalera).

Para las otras escaleras, se determina el área de acero en los apoyos y en la sección donde el momento es máximo.

Después de definir los momentos actuantes a usar, se aplica el cálculo según la teoría clásica, para la escalera autoportante, y la teoría de rotura, para las otras escaleras.

4.2 INICIO DEL PROGRAMA

Luego de haber definido las ecuaciones y los parámetros para el cálculo del refuerzo de acero, se procede a realizar el programa. El primer paso a ejecutar es la creación de los diversos códigos de procedimientos que se usarán en el programa. Estos procedimientos se crean dentro de uno o más módulos; esto dependerá de considerar agrupar los procedimientos, según la función de los mismos; es decir, colocar en un módulo todos los procedimientos relacionados con cálculos, otro módulo con los procedimientos que se encarguen de dibujar, etc.

Los procedimientos globales, que se usan para todos los módulos, se colocaron en módulos Standard.

Los módulos Standard creados son:

1. **CalculoAreaAcero:** Este módulo contiene los procedimientos relacionados con el cálculo del área de acero para los tres tipos de escaleras: Cálculo del área de acero por la teoría clásica y rotura, cálculo del área de acero para las tres escaleras, área de acero por cabilla...
 2. **CalculoValoresAutoportantes:** Contiene los procedimientos que calculan los parámetros que definen a las solicitaciones de la escalera autoportante: Momentos en dirección X, Y, Momento de torsión...
 3. **CalculoValoresEstatica:** Contiene los procedimientos que calculan las solicitaciones de la escalera con un tramo y dos descansos, y de la escalera de dos tramos con tres descansos: Momentos en los apoyos, momentos máximos, momentos mínimos...
 4. **DibujoConcretoAcero:** Este módulo contiene todos aquellos procedimientos relacionados con el dibujo de las escaleras, en sección y en planta, y del dibujo del acero longitudinal y transversal
 5. **Matrices:** este módulo contiene los procedimientos que hacen la resolución de un sistema de ecuaciones.
 6. **Procedimientos:** Contiene todos los procedimientos auxiliares, o generales: máximo, mínimo, acotaciones, dibujo del nervio...
-

Existe un módulo adicional llamado Utilitarios., que contiene otros procedimientos auxiliares: estilos de textos, de dimensión, dibujo de líneas, polilíneas, textos, círculos...

4.2.1 Inicio del código del programa.

Antes de comenzar a crear todos los procedimientos, se crea uno que contiene las variables relacionadas con la aplicación AutoCAD®, llamado **Inicializar**. Este procedimiento hace que se almacene en unas variables bien definidas, todos los objetos que envuelven la aplicación AutoCAD®: Aplicación, Documento activo, y los espacios modelo y papel.

Las variables se declararon así:

```
Public Aplicacion As AcadApplication
Public Documento As AcadDocument
Public Modelo As AcadModelSpace
Public Papel As AcadPaperSpace
```

Y el código del procedimiento Inicializar es:

```
Set Aplicacion = GetObject(, "AutoCAD®.Application")
Set Documento = Aplicacion.ActiveDocument
Set Modelo = Documento.ModelSpace
Set Papel = Documento.PaperSpace
```

Entonces, esto permite hacer uso de las variables de aplicación para crear los objetos dentro de la aplicación AutoCAD®. Si se dibujara una línea en el espacio papel, basta con escribir dentro del código:

```
Set Linea = Papel.AddLine(Punto1, Punto2)
```

En lugar de:

```
Set Linea = ThisDrawing.PaperSpace.AddLine(Punto1, Punto2)
```

4.2.2 Creación de los procedimientos de dibujo

Los procedimientos que a continuación se describen, son los procedimientos utilizados para ejecutar el programa. Estos procedimientos son recopilaciones de otros procedimientos creados en el mismo módulo, o en otro módulo.

4.2.2.1 Escalera autoportante

4.2.2.1.1 *DibujarConjuntoInferior*

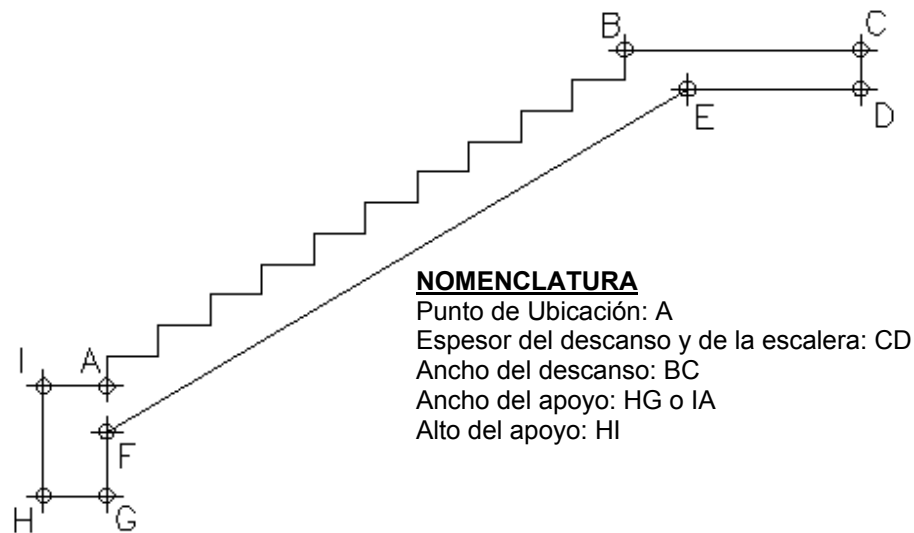


Fig. 13 Sección transversal inferior de la escalera autoportante

Este procedimiento dibuja la sección inferior de la escalera autoportante (Fig. 13).

Pasos:

1. Dibuja los escalones mediante el procedimiento DibujarEscalon, que los dibuja a partir de dos puntos (A y B, de la Fig. 13), donde comienzan los escalones y donde terminan. El punto inicial del dibujo de los escalones resulta ser el punto de ubicación del dibujo de la escalera. Este mismo procedimiento acota la huella, la contrahuella, la luz libre y el desnivel. Este procedimiento es aplicable para los otros tipos de escaleras.
-

2. Dibuja el descanso y la losa de la escalera a través del procedimiento `DibujarDescansoLosaEscalera`, que se ejecuta a partir del punto final del dibujo de los escalones (B, de la Fig. 13), con los espesores de la escalera y del descanso, y el ancho del descanso. También acota el espesor del descanso y el espesor de la losa de la escalera, así como el ancho del descanso.
 3. Dibuja el nervio o viga de apoyo con el procedimiento `DibujarNervioSeccion`, que además de dibujar el apoyo con su ancho y su alto, dibuja el eje del apoyo, con el respectivo nombre del eje, dentro de una circunferencia, que es la representación común de los ejes. El dibujo del eje del apoyo se ejecuta con el procedimiento `DibujoEjeNervio`.
 4. Dibuja el acero longitudinal principal mediante el procedimiento `DibujarAceroPrincipalLongitudinal`, que lo hace usando el ancho del descanso, los espesores de la escalera y el descanso, el recubrimiento de cálculo, el espaciamiento, y un punto de referencia, que resulta ser el punto final del dibujo de los escalones.
 5. `DibujarAceroDescansoSup` dibuja el acero longitudinal superior del descanso, con el recubrimiento de cálculo, con los espesores de la losa de escalera y descanso, el ancho del descanso, el espaciamiento, y el punto final del dibujo del escalón como referencia.
 6. Se hace el dibujo del acero longitudinal inferior del descanso con el procedimiento `DibujarAceroDescansoInf`, que se ejecuta con los mismos datos que el acero longitudinal superior del descanso.
 7. Se comienzan a dibujar los aceros transversales, con el dibujo del acero del escalón mediante el procedimiento `DibujarAceroEspaciadoEscalon`, y principal con el procedimiento `DibujarAceroEspaciadoPrincipal`. Estos procedimientos utilizan como datos los mismos que se hacen para el dibujo de los aceros longitudinales respectivos. En realidad el procedimiento que hace el dibujo del acero transversal es el llamado `DibujoAceroEspaciado`, solo que está dentro de cada procedimiento que realiza el dibujo del acero transversal, y los puntos que definen a este procedimiento depende del tipo de acero, es por ello que existen distintos procedimientos para realizar el dibujo del acero transversal. En la
-

mayoría de los casos, estos puntos corresponden a los mismos puntos con que se crearon los aceros longitudinales. Esto es aplicado para todos los aceros transversales

8. Se dibujan los aceros transversales del descanso, superior e inferior, con `DibujarAceroEspaciadoDescanso` y `DibujarAceroEspaciadoDescansoInf`, respectivamente, como procedimientos. Aplican los mismos datos para los casos de los aceros longitudinales del descanso.

4.2.2.1.2 *DibujarConjuntoSuperior*

Este procedimiento dibuja la sección superior de la escalera autoportante en sentido normal y en sentido simétrico. Los pasos son los mismos que los de la sección

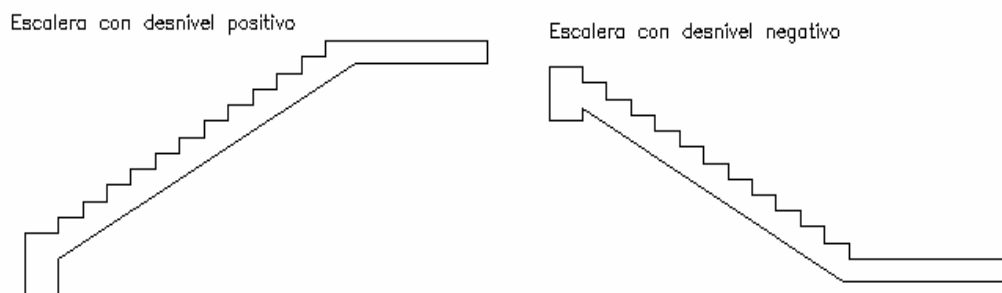


Fig. 14 Relación entre las secciones transversales inferior y superior de la escalera autoportante

inferior, variando el sentido del dibujo, modificando el signo del desnivel (Fig. 14):

4.2.2.1.3 *DibujarPlantaAutoportante*

Este procedimiento ejecuta el dibujo de la planta de la escalera autoportante (Fig. 15).

Pasos:

1. Dibuja los escalones del tramo inferior de la escalera a través del procedimiento `DibujarEscalonesPlanta`; este procedimiento dibuja los escalones a partir de un punto de ubicación, la luz libre, el ancho del escalón, y el número de escalones.
-

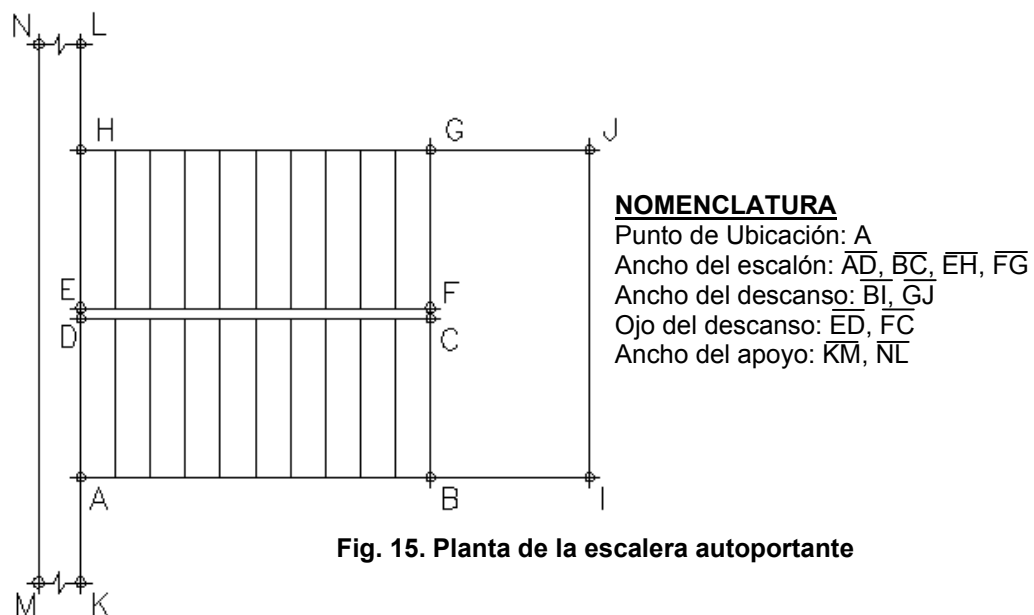


Fig. 15. Planta de la escalera autoportante

Este procedimiento consiste en dibujar, con una estructura de control For...Next, las líneas horizontales que definen la huella del escalón, y las líneas verticales que representan al ancho del escalón. El punto de ubicación dependerá del signo del ancho del escalón y de la luz libre. El punto de ubicación será (Fig. 15):

- A, si la luz libre y el ancho de escalón son positivos.
- B, si la luz libre es positiva y el ancho de escalón es negativo.
- C, si la luz libre y el ancho del escalón son negativos.
- D, si la luz libre es negativa y el ancho de escalón es positivo.

Con este procedimiento se dibujan, en planta, los escalones superiores e inferiores de la escalera autoportante. Este mismo procedimiento acota la huella del escalón, esta acotación es ubicada dependiendo del porcentaje de ubicación establecido.

2. Acota el ancho del escalón con el procedimiento AcotarLinea. Este procedimiento toma como dato la línea para definir el punto inicial y el punto final, y crea la acotación con estos puntos. En este caso, acota la última línea creada en el procedimiento DibujarEscalonesPlanta. Si se toma al punto A o al punto B de la Fig. 15, como punto de ubicación, la última línea creada es la que pasa por los

puntos C y D, cuya longitud es el ancho del escalón. Esta línea es llamada LineaEscalon.

3. Se repiten los pasos a. y b. para el caso del tramo superior de la escalera.
4. Se dibuja el descanso a través del procedimiento DibujarDescansoPlanta, que no es más que dibujar un rectángulo con las dimensiones en planta del descanso, a partir de un punto de ubicación definido de la misma forma que el dibujo de los escalones en planta (punto B de la Fig. 15), con el ancho y el largo del descanso. Este procedimiento también acota el ancho y el largo del descanso.
5. DibujarNervioPlanta es un procedimiento que dibuja el apoyo en planta, con el ancho del apoyo, dibujando las líneas MN y KL de la Fig. 15, y las líneas de interrupción (o Zig-Zag). Esta última se dibuja con el procedimiento ZigZag, que crea esta línea a partir de los pares de puntos N, L, y M, K de la Fig. 15. En este mismo procedimiento se dibuja el eje del apoyo con el procedimiento DibujoEjeNervio, explicado para el dibujo del eje del apoyo en las secciones.
6. Acota la luz libre del escalón con el procedimiento AcotarLinea. Primero se crea una línea de referencia que pasa por los puntos A y D (Fig. 15), luego se llama al procedimiento AcotarLinea, y por último la línea creada se borra.
7. Se acota el ojo de la escalera (en caso de tenerlo) de la misma forma que la luz libre del escalón, con la diferencia de usar el punto C del escalón inferior, y el punto F del escalón superior (Fig. 15) para crear la línea de referencia.

4.2.2.2 Para la escalera de un tramo con dos descansos

4.2.2.2.1 *DibujarConjuntoDescansoEscaleraDescanso*

Este procedimiento dibuja la sección transversal de la escalera de un tramo con dos descansos (Fig. 16).

Pasos:

1. Dibuja los escalones mediante el procedimiento DibujarEscalon, entre los puntos A y B de la Fig. 16.
 2. Dibuja la losa de la escalera, el descanso y el apoyo del tramo superior con el procedimiento DibujarDescansoLosaEscaleraNervio, que dibuja el descanso, la
-

losa de la escalera, y el apoyo ubicado a una cierta distancia del borde exterior

NOMENCLATURA

Punto de Ubicación: A

Ancho del descanso superior: $\overline{BC}$

Ancho del descanso inferior: $\overline{AP}$

Espesor del descanso superior: $\overline{CD}$

Espesor del descanso inferior: $\overline{OP}$

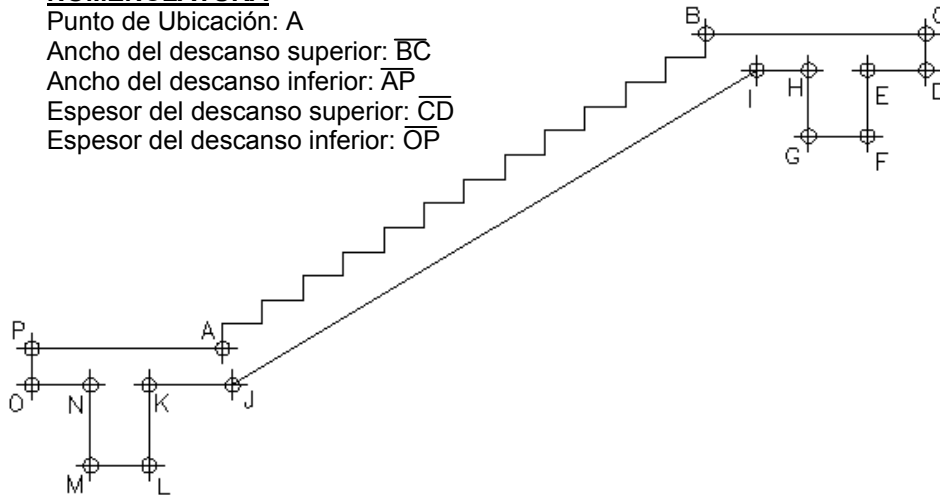


Fig. 16. Sección transversal de la escalera de un tramo con dos descansos

del descanso (distancia horizontal entre el punto D y el punto medio del segmento FG). También acota el alto y el ancho del apoyo, el ancho y el espesor del descanso, y el espesor de la losa de la escalera.

3. Dibuja el acero principal longitudinal del tramo, con el procedimiento DibujarAceroPrincipalLongitudinal, que dibuja el acero longitudinal principal de ese tramo de escalera como si fuera de un solo descanso, el superior en este caso, pero no forma parte del acero longitudinal de esta escalera con los dos descansos; solo es utilizado como referencia para crear el acero longitudinal original: se toma el acero creado y se almacena en otra variable para diferenciarlo del acero creado para el tramo superior.
4. Se repite el punto 2. para el tramo inferior.
5. Se repite el punto 3. para el tramo inferior.
6. Con el acero en el tramo superior y en el tramo inferior, se crea el acero longitudinal definitivo, tomando los puntos comunes entre los aceros de los tramos, y el acero resultante. Posteriormente, los aceros de los tramos se eliminan, y queda el acero principal original.
7. Se dibuja el acero del descanso superior e inferior con los procedimientos DibujarAceroDescansoSupSup (acero superior del descanso inferior y acero

inferior del descanso superior) y DibujarAceroDescansoInf (acero superior del descanso superior y acero inferior del descanso inferior). Este dibujo está condicionado por el espesor del descanso y de la losa de la escalera, del ancho del descanso donde el acero se esté dibujando, y el punto de ubicación para hacer el dibujo ("A" para el descanso inferior, y "B" para el descanso superior, de la Fig. 16)

8. Se realiza el dibujo del acero transversal de la misma forma en que se realiza para el tramo de escalera y el descanso de la escalera autoportante, para el tramo de escalera, y para los tramos de descansos.
9. Se realiza, en caso de ser seleccionado como apoyo, el acero del muro, tanto longitudinal como transversal, con los procedimientos DibujarAceroMuroLongitudinal y DibujarAceroEspaciadoMuro, respectivamente. Estos aceros se dibujan tomando en cuenta el alto, el ancho, y la ubicación del apoyo (en este caso, el muro). Es importante señalar que el acero del muro no es considerado en este trabajo, sin embargo es representado según el acero mínimo para acero transversal. Este punto aplica para la escalera de dos tramos con tres descansos.

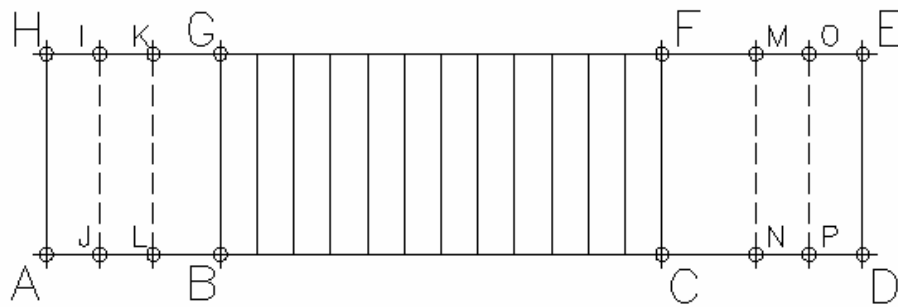
4.2.2.2.2 DibujarPlantaDED

Consiste en dibujar la planta de la escalera de un tramo con dos descansos (Fig. 17).

Pasos:

1. Dibuja los escalones de la misma forma como se hace para los escalones de la escalera autoportante, con la variante que se hace una vez, con el procedimiento DibujarEscalonesPlanta, tomando como punto de ubicación el punto B de la Fig. 17.
 2. Dibuja el descanso superior e inferior, en planta, con el procedimiento DibujarDescansoPlanta, que dibuja el descanso con el ancho del descanso superior e inferior, respectivamente, y con el ancho del escalón, a partir de un punto de ubicación. Este procedimiento trabaja con las mismas convenciones de signos que los escalones (Fig. 15 y Fig. 17) Para el descanso superior, se toma
-

como el punto de ubicación el punto C de la Fig. 17, y toma el ancho del descanso positivo; caso contrario ocurre con el descanso inferior, que toma al punto B de la Fig. 17, y toma el ancho del descanso inferior negativo. Este mismo procedimiento acota los anchos de los descansos superior e inferior, dibuja el apoyo y su eje respectivo para cada descanso, con el procedimiento `DibujarNervioPlanta`, representado por los puntos I, J, K y L para el apoyo inferior, y los puntos M, N, O y P para el apoyo superior. También acota la distancia entre el borde exterior del descanso y el eje del apoyo, es decir, de la Fig. 17, se acota entre el punto D y el punto medio del segmento NP para el descanso superior; en forma análoga, se acota entre el punto A y el punto medio del segmento JL para el descanso inferior.



NOMENCLATURA

Punto de Ubicación: B
 Ancho del descanso superior: $\overline{CD}$, $\overline{EF}$
 Ancho del descanso inferior: $\overline{AB}$, $\overline{GH}$
 Luz libre de escalones: $\overline{BC}$
 Ancho del escalón: AH, BG, CF, ED
 Ancho del apoyo superior: NP, MO
 Ancho del apoyo inferior: IK, JL

Fig. 17 Planta de la escalera de un tramo con dos descansos

3. Se acota el ancho del escalón con el procedimiento `AcotarLinea`, que acota un segmento de recta, en este caso, un segmento que pasa por los puntos A y H, o los puntos D y E de la Fig. 17.

4.2.2.3 Escalera de dos tramos con tres descansos

4.2.2.3.1 DibujarConjuntoEscaleraDescansoEscalera

Este procedimiento dibuja la sección transversal de la escalera de dos tramos con tres descansos (Fig. 18).

Pasos:

1. Dibuja los escalones del tramo inferior con el procedimiento DibujarEscalon, que dibuja los escalones entre el punto de ubicación y un punto definido por la luz libre del tramo inferior y el desnivel entre los descansos inferior e intermedio, con el número de escalones de ese tramo, a partir del punto de ubicación, en otras palabras, dibuja los escalones entre el punto A y el punto B de la Fig. 18.
2. Dibuja un segmento de recta horizontal (segmento BC de la Fig. 18), de longitud

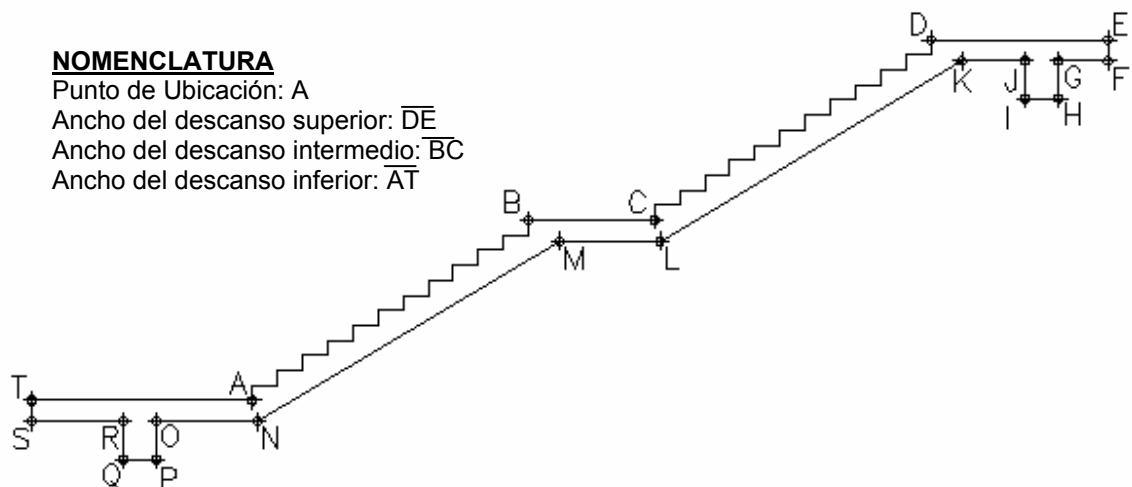


Fig. 18 Sección transversal de la escalera de dos tramos con tres descansos

el ancho del descanso intermedio que representa la parte superior del descanso intermedio.

3. Dibuja los escalones del tramo superior con el procedimiento DibujarEscalon, que dibuja los escalones entre el punto C, que es el punto final del segmento BC, y el punto D de la Fig. 18; éste último obtenido añadiendo a las coordenadas del punto de ubicación la suma de las luces de los tramos superior e inferior y el ancho del descanso intermedio, para la coordenada horizontal del punto A, y la

suma de los desniveles de los tramos superior e inferior para la coordenada vertical del punto A.

4. Dibuja la losa de la escalera, el descanso y el apoyo del tramo superior con el procedimiento `DibujarDescansoLosaEscaleraNervio`, que dibuja el descanso, la losa de la escalera, y el apoyo ubicado a una cierta distancia del borde exterior del descanso (distancia horizontal entre el punto F y el punto medio del segmento HI). También acota el alto y el ancho del apoyo, el ancho y el espesor del descanso, y el espesor de la losa de la escalera.
 5. Dibuja la losa de la escalera, el descanso y el apoyo del tramo inferior de la misma forma que en el tramo superior; dibuja el descanso, la losa de la escalera, y el apoyo ubicado a una determinada distancia del borde exterior del descanso (distancia horizontal entre el punto S y el punto medio del segmento PQ). También acota el alto y el ancho del apoyo, el ancho y el espesor del descanso, y el espesor de la losa de la escalera.
 6. Se determinan los puntos L y M de la Fig. 18, para definir la parte inferior del descanso intermedio, creando líneas auxiliares inclinadas que pasen por los puntos inicial y final de los segmentos respectivos que definen a los espesores de las losas de las escaleras superior e inferior. Luego se define una línea auxiliar horizontal separada del segmento BC el espesor del descanso intermedio. La intersección de la línea auxiliar inclinada que pasa por el segmento que define al espesor de la losa superior con la línea auxiliar horizontal, es el punto L. En forma análoga, la intersección de la línea auxiliar inclinada que pasa por el segmento que define al espesor de la losa inferior con la línea auxiliar horizontal, es el punto M.
 7. Se hace el dibujo del acero principal longitudinal del tramo superior combinando las funciones de `DibujarAceroPrincipalLongitudinal` con algunos procedimientos que realizaban el dibujo de las longitudes horizontal e inclinada del acero, tomando como punto de ubicación el punto D de la Fig. 18, y todos los elementos del tramo superior de la escalera con los descansos intermedio y superior. Se hace el despiece con el procedimiento `CopiarYAcotarAcero`, que, descrito con anterioridad, realiza una copia del acero longitudinal seleccionado y se mueve
-

esta copia a una cierta distancia y ángulo definido, donde también acota la copia del acero y señala el diámetro, la longitud del acero y el número de aceros de ese tipo.

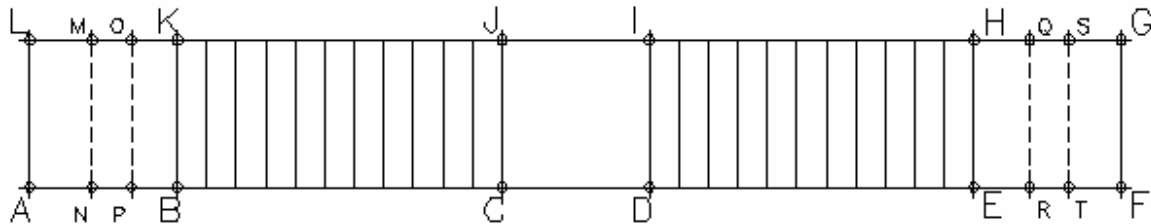
8. Hace el dibujo del acero principal longitudinal del tramo inferior con el procedimiento `DibujarAceroPrincipalLongitudinal` con los elementos del tramo inferior y los descansos intermedio e inferior, similar a la representación del dibujo del acero principal longitudinal del tramo superior.
 9. Se dibuja el acero longitudinal superior e inferior del descanso inferior con dos procedimientos distintos: para el acero superior se utilizó el procedimiento `DibujarAceroDescansoInf`, que hace el dibujo de este acero en función de la geometría del descanso inferior y de la longitud inclinada en la misma dirección del tramo de la escalera inferior.
 10. Se hace el dibujo del acero longitudinal superior e inferior del descanso intermedio, creando, para el acero superior, una línea auxiliar horizontal que esté separada del segmento BC de la Fig. 18 una distancia igual al recubrimiento de cálculo, para las longitudes inclinadas se trazan líneas auxiliares inclinadas paralelas a los segmentos KL y MN, separadas de éstos una distancia igual al espesor de la escalera del tramo inferior menos el recubrimiento, para el segmento KL, y una distancia igual al recubrimiento, para el segmento MN. Luego se toman los puntos de intersección entre las líneas auxiliares y a partir de estos puntos se definen las longitudes inclinadas con dirección igual a la de las líneas auxiliares respectivas. Para el acero inferior se utilizó el procedimiento `DibujarAceroDescansoSupSup` que hace el dibujo usando los elementos del descanso intermedio y del espesor de la escalera del tramo superior, con la longitud horizontal, obviamente, en dirección BC, y longitud inclinada en dirección KL de la Fig. 18.
 11. Dibuja el acero superior e inferior del descanso superior con los siguientes procedimientos: para el acero superior, se hizo uso del procedimiento `DibujarAceroDescansoSupSup2`, que trabaja de la misma forma para el descanso inferior e intermedio, variando los valores para el descanso superior. El descanso inferior se realizó con el procedimiento `DibujarAceroDescansoInf`, realizando el
-

dibujo de la misma forma que el acero superior del descanso inferior, con la variante de tomar los valores que definen al descanso superior, cambiando el sentido de los mismos.

12. El dibujo del acero longitudinal de los escalones se hace con el procedimiento DibujarAceroEscalon, para los tramos inferior y superior, tomando respectivamente los espesores de las escaleras de los tramos inferior y superior, y los pares de puntos utilizados para el dibujo de los escalones (A y B para el tramo inferior, y C y D, para el tramo superior, de la Fig. 18)
 13. Se hace el dibujo del acero principal transversal para el tramo superior e inferior con el procedimiento DibujarAceroEspaciadoPrincipal, que hace este dibujo tomando como datos los puntos AB y CD, respectivamente, y a partir de estos puntos se definen los puntos donde serán ubicados los aceros transversales. Este procedimiento, al igual que cualquiera que realice el dibujo del acero transversal, hace las acotaciones respectivas donde se incluye el diámetro del acero de refuerzo, el espaciamiento y el número de cabillas existentes para cada tramo.
 14. El dibujo de acero transversal para los descanso inferior, intermedio y superior se hizo con dos procedimientos: el primero de estos procedimientos dibujaba el acero del descanso en la parte superior, este procedimiento se llama DibujarAceroEspaciadoDescansoInf, que toma un punto de ubicación para hacer el dibujo (A, B y D, respectivamente) y el ancho y espesor respectivo de cada descanso.
 15. DibujarAceroEspaciadoEscalon hace el dibujo el acero transversal del escalón para los tramos inferior y superior de la escalera. Este procedimiento toma los puntos que definen al acero longitudinal del escalón para hacer el acero transversal (AB para el tramo inferior, y CD, para el tramo superior). Este procedimiento ya ha sido descrito en el dibujo del acero transversal del escalón de las escaleras anteriores.
 16. Se hace el acero del muro, en caso de ser seleccionado, de la misma forma que para la escalera de un tramo con dos descansos.
-

4.2.2.3.2 DibujarPlantaDEDED

Este procedimiento dibuja la planta de la escalera de dos tramos con tres descansos (Fig. 19).



NOMENCLATURA

Ancho del descanso inferior: AB, KL
Ancho del descanso intermedio: CD, IJ
Ancho del descanso superior: EF, GH
Ancho del escalón: AL, FG
Luz libre del tramo inferior: BC, JK
Luz libre del tramo superior: DE, HI
Ancho del apoyo inferior: MO, NP
Ancho del apoyo superior: QS, RT

Fig. 19 Planta de escalera de dos tramos con tres descansos

Pasos:

1. Dibuja los escalones de los tramos inferior y superior con el procedimiento DibujarEscalonPlanta, de la misma forma que se hace con las otras escaleras: se toma un punto de ubicación para dibujar los escalones (C para el tramo inferior, y D para el tramo superior, de la Fig. 19) y se toma la luz libre y el ancho del escalón, dibuja los escalones tantas veces como número de escalones tenga en cada tramo.
 2. Dibuja los descansos inferior, intermedio y superior con el procedimiento DibujarDescansoPlanta, procedimiento ya antes descrito, que hace el dibujo del escalón tomando como puntos de ubicación, respectivamente, B, C, y E y tomando los anchos de los descansos respectivos, y el ancho del escalón.
 3. Dibuja los apoyos en planta para los descansos inferior y superior, representados por las líneas segmentadas en la Fig. 19, con el procedimiento DibujarNervioPlanta, que lo hace de la misma forma que se hace para la escalera de un tramo con dos descansos, se toman en forma análoga los puntos A y F
-

como referencia para ubicar los apoyos. También hace los acotamientos descritos en la escalera de un tramo con dos descansos, y el dibujo de los ejes de los apoyos.

4.2.3 Creación de los procedimientos de cálculo

Los procedimientos que realizan el cálculo del área de acero se encuentran sobre el módulo Standard `CalculoAreaAcero`. Estos procedimientos se encargan de determinar el área de acero para cada tipo de acero, y también define el diámetro mínimo de la cabilla a utilizar, con una función denominada `IndiceAsSuperiorCalculo`, que, a partir del área de acero por cabilla obtenido por cualquier procedimiento que se encargue de determinarlo, determina el calibre mínimo de la cabilla que satisface el área de acero resultante, es decir, esta función compara el área de acero de entrada con el área de acero de cada uno de los calibres, y extrae el mínimo calibre que cubre el área de acero de entrada. El cálculo del área de acero se hace aplicando los principios básicos de la teoría clásica y de rotura, para la flexión pura. Más adelante, en los métodos de cálculo para las escaleras, se hace un resumen más detallado de la metodología utilizada para calcular el área de acero por las teorías antes mencionadas. Las funciones que definen el cálculo del área de acero por estas teorías son:

1. `CalculoAsTeoriaClasica`: esta función calcula el área de acero de la sección, a partir de la teoría clásica para momentos flectores, y toma el área de acero según la posición del acero en la sección: superior o inferior.
2. `CalculoAsTeoriaRotura`: esta función calcula el área de acero de la sección, a partir de la teoría de rotura para momentos flectores, y toma el área de acero según la posición del acero en la sección: superior o inferior.

A continuación se describen los procedimientos (funciones) utilizados para determinar el área del acero longitudinal por cabilla. El acero transversal se calcula de otra forma, descrita al final de la descripción del cálculo del acero longitudinal.

Para la escalera autoportante:

1. **CalculoAsL_Autoportante:** Esta función calcula el área de acero para los tramos de escalera y los tramos en el descanso, tomando como variables de entrada las que definen el cálculo del área de acero para la teoría clásica, y el tipo de sección donde se determina el área de acero: sección en los apoyos, sección donde el momento de flexión sea máximo tanto en el descanso como en el tramo de escalera, sección donde se unen el tramo de la escalera y el descanso, sección media del descanso; y la posición del acero en la sección: superior o inferior.

Para la escalera de un tramo con dos descansos

2. **CalculoAsL_DED:** Esta función calcula el área de acero de la escalera de un tramo con dos descansos para las secciones donde el momento flector sea máximo o mínimo, es decir, hace el cálculo para las secciones donde se encuentran los apoyos (solicitud mínima) y donde sea máxima la solicitud, tomando como variables de entrada las que definen el cálculo del área de acero para la teoría de rotura, el tipo de sección donde se determina el área de acero, y la posición del acero en la sección: superior o inferior.

Para la escalera de dos tramos con tres descansos:

3. **CalculoAsL_DEDED:** Esta función calcula el área de acero de la escalera de dos tramos con tres descansos para las secciones donde el momento flector sea máximo o mínimo, es decir, hace el cálculo para las secciones donde se encuentran los apoyos (solicitud mínima) y donde sea máxima la solicitud, tomando como variables de entrada las que definen el cálculo del área de acero para la teoría de rotura, el tipo de sección donde se determina el área de acero, y la posición del acero en la sección: superior o inferior. En pocas palabras, trabaja de la misma forma que la escalera de un tramo con dos descansos, con la diferencia que el volumen obtenido de concreto para las escaleras en cuestión son distintas, variando la carga permanente en cada escalera.

Para el cálculo del refuerzo transversal (a excepción del descanso en la escalera autoportante), se determina en base a la relación entre el área de acero y el área de

la sección total del acero (cuantía relativa de la armadura, ρ), donde este valor de cuantía se puede fijar en el formulario de especificaciones. Es decir, el área de acero para el cálculo del acero transversal, se obtiene según:

$$A_s = \rho * b * d \quad (26)$$

Donde ρ es la cuantía a usar, b es el ancho del escalón, y d la altura útil de la sección, que es la diferencia entre el espesor y el recubrimiento.

Para el cálculo del acero transversal del descanso en la escalera autoportante, se toma en cuenta la sección ubicada en la mitad del descanso, es decir, el acero transversal del descanso es el acero longitudinal para esa sección (mitad del descanso).

4.2.4 Creación de los módulos de formularios.

Una vez realizado todos los procedimientos para el dibujo y el cálculo, se procede, entonces, a crear los módulo de formularios que serán los módulos de entrada de datos para hacer la recopilación de todos los procedimientos ya realizados, y manipular los datos de entrada a los procedimientos. Para crear estos módulos de formularios se hizo uso de los objetos insertables y los controles Standard de VBA. Para que el desarrollo de estos módulos de entrada sea eficiente, se crea una serie de funciones que permiten determinar que los datos de entrada, antes de ser usados en los procedimientos, pase por estas funciones y verifique que el valor del dato en los módulos de formularios sea el adecuado.

Los módulos de formularios creados para el programa son:

- FormaAceroL
 - FormaAceroT
 - FormaAcotacion
 - FormaApoyo
 - FormaAsDef
 - FormaAutoportante
 - FormaDED
 - FormaDEDED
 - FormaDescanso
 - FormaEscalera
 - FormaEspecificaciones
 - FormaOrientacion
-

4.2.4.1 FormaAceroL

Este módulo de formulario fue diseñado para colocar todos los datos que relacionan el dibujo del acero longitudinal. Este módulo modifica las propiedades del acero seleccionado (espaciamiento, longitudes, datos del despiece). En la Fig. 20 se seleccionó un tipo de acero, y se ajustó los campos en función de este acero

Geometría del acero longitudinal

Espaciamiento: 0.25

Longitudes

Horizontal:

Inclinada: 0.45

Inclinada2:

Despiece:

Distancia: 1

Ángulo: 270

Distancia de texto: 0.05

- Acero principal de la sección inferior
- Acero superior del descanso de la sección inferior
- Acero inferior del descanso de la sección inferior
- Acero del escalón de la sección inferior
- Acero del arranque de la sección inferior**
- Acero del nervio de la sección inferior
- Acero principal de la sección superior
- Acero superior del descanso de la sección superior
- Acero del escalón de la sección superior
- Acero del arranque de la sección superior
- Acero del nervio de la sección superior

D: Distancia
LI: Longitud inclinada

Aceptar Cancelar

Fig. 20 FormaAceroL

seleccionado.

Esta forma consta de:

1. Cuadro de listas:

- 1.1. Lista de aceros longitudinales: .esta lista contiene los aceros que forman parte de la escalera. Al seleccionar cualquier elemento de esta lista, los demás controles ajustan los datos en función de esta selección.

2. Cuadros de textos:

- 2.1. Espaciamiento: Este cuadro contiene el espaciamiento del acero seleccionado en la lista de aceros longitudinales.
- 2.2. Longitudes:
- 2.2.1. Horizontal: Este cuadro contiene la longitud horizontal del acero seleccionado en la lista de aceros longitudinales. Este cuadro se habilita en caso de existir esta longitud en el acero seleccionado.
- 2.2.2. Inclínada: Este cuadro contiene la primera longitud inclinada del acero seleccionado en la lista de aceros longitudinales. Este cuadro se habilita en caso de existir esta longitud en el acero seleccionado.
- 2.2.3. Inclínada2: Este cuadro contiene la segunda longitud inclinada del acero seleccionado en la lista de aceros longitudinales. Este cuadro se habilita en caso de existir esta longitud en el acero seleccionado.

3. Despiece

- 3.1. Distancia: Este cuadro contiene la distancia para el despiece del acero seleccionado en la lista de aceros longitudinales.
- 3.2. Angulo: Este cuadro contiene el ángulo para el despiece del acero seleccionado en la lista de aceros longitudinales.
- 3.3. Distancia de texto: Este cuadro contiene la distancia entre el texto que define al acero y el acero dibujado, para el despiece del acero seleccionado en la lista de aceros longitudinales.
-

4.2.4.2 FormaAceroT

Este módulo de formulario fue diseñado para colocar todos los datos que relacionan el dibujo del acero transversal. Este módulo modifica las propiedades del acero seleccionado (espaciamiento, distancias, número de acotaciones). En la Fig. 21 se seleccionó un tipo de acero, y se ajustó los campos en función de este acero

Acero Transversal

Acero superior del descanso inferior
Acero inferior del descanso inferior
Acero superior del descanso intermedio
Acero inferior del descanso intermedio
Acero superior del descanso superior
Acero inferior del descanso superior
Acero principal del tramo inferior
Acero del escalón del tramo inferior
Acero principal del tramo superior
Acero del escalón del tramo superior
Acero lateral izquierdo del muro superior
Acero lateral derecho del muro superior

Espaciamiento: 0.25

Acotación:
Distancia Horizontal: 0.2
Distancia Vertical: 0.3
Numero de Líneas: 3

Porcentaje de Ubicación:
50

Orientación
☐ Arriba e Izquierda
☐ Abajo e Izquierda
☒ Arriba y Derecha
☐ Abajo y Derecha

DH: Distancia horizontal
DV: Distancia vertical

0 <----- % Ubicación -----> 100

Aceptar Cancelar

Fig. 21 FormaAceroT

seleccionado.

Esta forma consta de:

1. Cuadro de listas:

- 1.1. Lista de aceros transversales: .esta lista contiene los aceros que forman parte de la escalera. Al seleccionar cualquier elemento de esta lista, los demás controles ajustan los datos en función de esta selección.

2. Cuadros de textos:

- 2.1. Espaciamiento: Este cuadro contiene el espaciamiento del acero seleccionado en la lista de aceros transversales.
- 2.2. Acotación:
- 2.2.1. Distancia Horizontal: Este cuadro contiene la distancia horizontal de la acotación del acero seleccionado en la lista de aceros transversales.
- 2.2.2. Distancia Vertical: Este cuadro contiene la distancia vertical de la acotación del acero seleccionado en la lista de aceros transversales.
- 2.2.3. Número de líneas: Este cuadro contiene el número de líneas para las acotaciones del acero seleccionado en la lista de aceros transversales.
- 2.2.4. Porcentaje de ubicación: Este cuadro contiene la relación entre el punto de ubicación de la acotación del acero transversal y la longitud total del acero transversal.
3. Botones de selección:
- 3.1. Orientación: Este marco contiene los botones que definen el sentido de la acotación del acero transversal.
-

4.2.4.3 FormaAcotacion

Este módulo de formulario fue diseñado para colocar todos los datos que relacionan la posición y las distancias para las acotaciones de la escalera. Este módulo modifica las propiedades del tipo de acotación seleccionado (distancias, porcentaje de ubicación). En la Fig. 22 se seleccionó un tipo de acotación, y se ajustó los campos

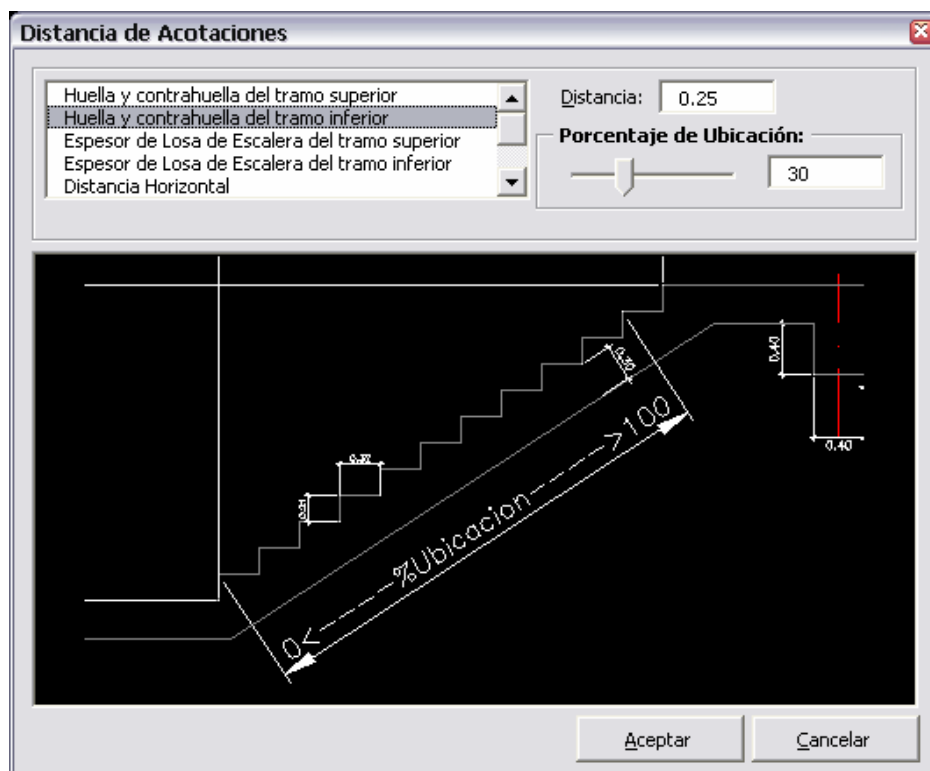


Fig. 22 FormaAcotacion

en función de este tipo de acotación seleccionado.

Esta forma consta de:

1. Cuadro de lista:

- 1.1. Lista de tipos de acotaciones: Esta lista contiene los tipos de acotación que forman parte de la escalera. Al seleccionar cualquier elemento de esta lista, los demás controles ajustan los datos en función de esta selección.

2. Cuadros de textos:

- 2.1. Distancia: Este cuadro contiene la distancia del tipo de acotación seleccionado en la lista de tipos de acotaciones.
- 2.2. Porcentaje de ubicación: Este cuadro contiene la relación entre el punto de ubicación de la acotación y la longitud total de los puntos posibles de acotación. Este cuadro se habilita en caso de existir el porcentaje de ubicación.

4.2.4.4 FormaApoyo

módulo de formulario creado para introducir los datos del tipo de apoyo seleccionado

Nervio

Dimensiones del Nervio

Ancho: 0.4

Alto: 0.7

Distancia al eje: 1

Dibujo del Eje del Nervio

Nombre: X

Distancia Superior: 1

Distancia Inferior: 0.5

NI, NS, DS, DI

DS: Distancia superior
DI: Distancia inferior
NS: Nombre del eje superior
NI: Nombre del eje inferior

Aceptar Cancelar

Fig. 23 FormaApoyo

(viga o muro). Este módulo consta de (Fig. 23):

1. Cuadros de textos:

- 1.1. Ancho: este cuadro contiene el ancho del apoyo.
 - 1.2. Alto: este cuadro contiene el alto del apoyo.
-

- 1.3. Distancia al eje: este cuadro contiene la distancia que hay del borde exterior del descanso al eje del apoyo.
- 1.4. Nombre: este cuadro contiene el nombre del eje del apoyo.
- 1.5. Distancia Superior: este cuadro contiene la distancia DS de la Fig. 23.
- 1.6. Distancia Inferior: este cuadro contiene la distancia DI de la Fig. 23.

4.2.4.5 FormaAsDef

Módulo de formulario creado para definir el diámetro del acero a utilizar en el dibujo

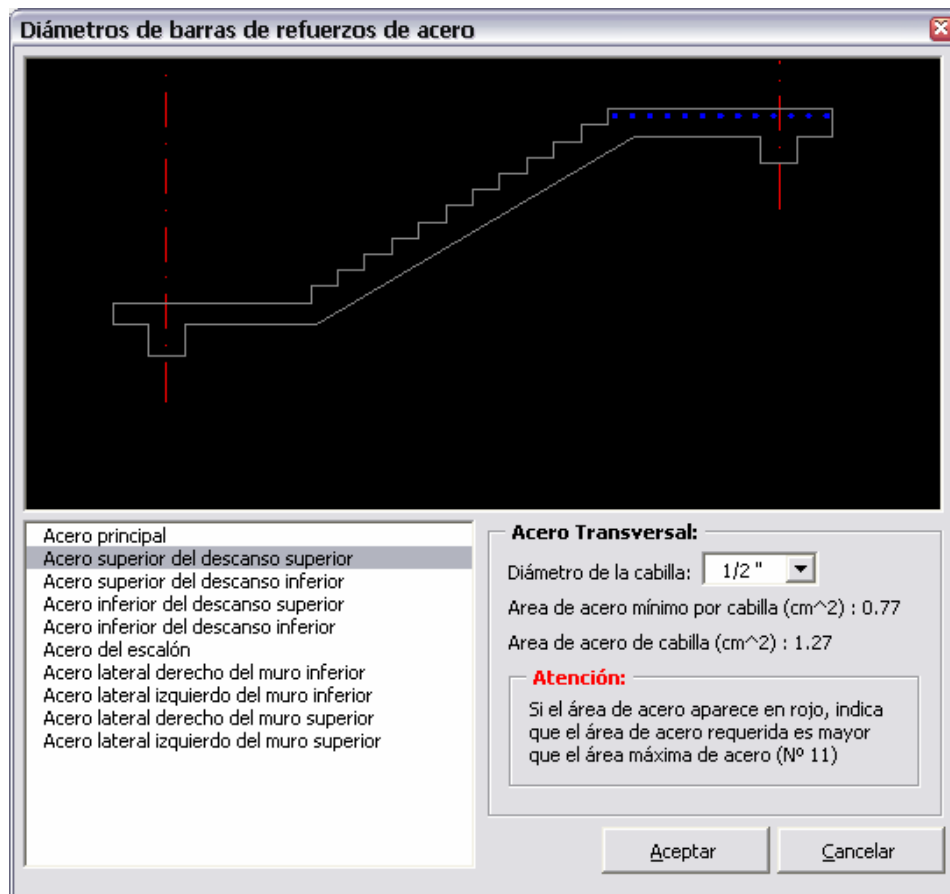


Fig. 24 FormaAsDef

del acero longitudinal o transversal (Fig. 24). Este módulo consta de:

- 1. Cuadro de lista:

- 1.1. Lista de aceros: .esta lista contiene los aceros que forman parte de la escalera. Al seleccionar cualquier elemento de esta lista, los demás controles ajustan los datos en función de esta selección.
2. Cuadro de lista desplegable:
 - 2.1. Diámetro de la cabilla: Este cuadro contiene una lista de los diámetros de la cabilla que se pueden disponer para el acero seleccionado en la lista de aceros.

4.2.4.6 FormaAutoportante

Este módulo fue diseñado para introducir todos los datos relacionados con el dibujo de la escalera autoportante (Fig. 25). Este módulo recopila los otros módulos de formularios que corresponda a cada caso. Por ejemplo, en la Fig. 25 aparece el

Fig. 25 FormaAutoportante

botón Acotaciones, este botón llama al módulo FormaAcotacion con los datos de las acotaciones de la escalera autoportante. Este es el módulo inicial de la escalera autoportante.

Este módulo consta de:

1. Cuadro de textos:

- 1.1. Luz libre de escalones: cuadro que contiene la luz entre el primer y el último escalón del tramo de escalera (distancia “L” de la Fig. 25).
- 1.2. Desnivel de entrepisos: cuadro que contiene el desnivel de entrepisos, es decir, la distancia que existe entre las partes superiores de los apoyos (distancia “DE” de la Fig. 25).
- 1.3. Número de escalones (por sección): cuadro que contiene el número de escalones por sección, es decir, el número de escalones para la sección superior, y para la sección inferior.
- 1.4. Ancho del escalón: cuadro que contiene el ancho del escalón para cada sección y, a su vez, contiene el ancho del descanso (distancia “A” de la Fig. 25).
- 1.5. Ojo del descanso: este cuadro contiene el espacio central de la escalera circundado por los escalones y el descanso (distancia “m” de la Fig. 25).
- 1.6. Espesor de la losa de la escalera: este cuadro contiene el espesor de la losa de la escalera para las secciones inferior y superior, y, a su vez, contiene el espesor del descanso (distancia “e” de la Fig. 25).
- 1.7. Ancho del nervio: cuadro que contiene el ancho de los nervios inferior y superior (distancia “AN” de la Fig. 25).
- 1.8. Alto del nervio superior: cuadro que contiene el alto del nervio superior (distancia “AS” de la Fig. 25).
- 1.9. Alto del nervio inferior: cuadro que contiene el alto del nervio inferior (distancia “AI” de la Fig. 25).
- 1.10. Nombre del eje del nervio: cuadro que contiene el nombre del eje de los nervios superior e inferior.

2. Botones:

- 2.1. Acero Longitudinal:
 - 2.1.1. Detallado...: este botón llama al módulo de formulario FormaAceroL (Fig. 20), con los datos correspondientes a los diversos tipos de aceros longitudinales para la escalera autoportante.
 - 2.1.2. Cálculo...: Este botón llama al formulario FormaAsDef (Fig. 24) con los datos correspondientes a los diversos tipos de acero longitudinal de la escalera autoportante.
- 2.2. Acero Transversal:
 - 2.2.1. Detallado...: este botón llama al módulo de formulario FormaAceroT (Fig. 21), con los datos correspondientes a los diversos tipos de aceros transversales para la escalera autoportante.
 - 2.2.2. Cálculo...: Este botón llama al formulario FormaAsDef (Fig. 24) con los datos correspondientes a los diversos tipos de acero longitudinal de la escalera autoportante.
- 2.3. Especificaciones...: Este botón llama al formulario FormaEspecificaciones (Fig. 30), con los datos generales que indican las variables de cálculo y dibujo en general.
- 2.4. Acotaciones...: Botón que llama al formulario FormaAcotacion (Fig. 22), con los datos correspondientes a la acotación de la escalera autoportante
- 2.5. Orientación...: Botón que llama al formulario FormaOrientacion (Fig. 31).
- 2.6. Dibujar Sección Inferior: Botón que solicita un punto en la pantalla para ejecutar el procedimiento DibujarConjuntoInferior.
- 2.7. Dibujar Sección Superior: Botón que solicita un punto en la pantalla para ejecutar el procedimiento DibujarConjuntoSuperior.
- 2.8. Dibujar Planta: Botón que solicita un punto en la pantalla para ejecutar el procedimiento DibujarPlantaAutoportante.
- 2.9. Salir: Descarga los formularios utilizados y guarda los cambios efectuados en los formularios FormaAutoportante y FormaEspecificaciones.

Estos botones son utilizados para las otras escaleras (excepto los botones de Dibujar Sección Inferior y Dibujar Sección Superior) en sus respectivos formularios y con sus respectivos datos.

4.2.4.7 FormaDED

Este módulo fue diseñado para introducir todos los datos relacionados con el dibujo de la escalera de un tramo con dos descansos (Fig. 26). Este módulo recopila los otros módulos de formularios que corresponda a cada caso. Este es el módulo inicial

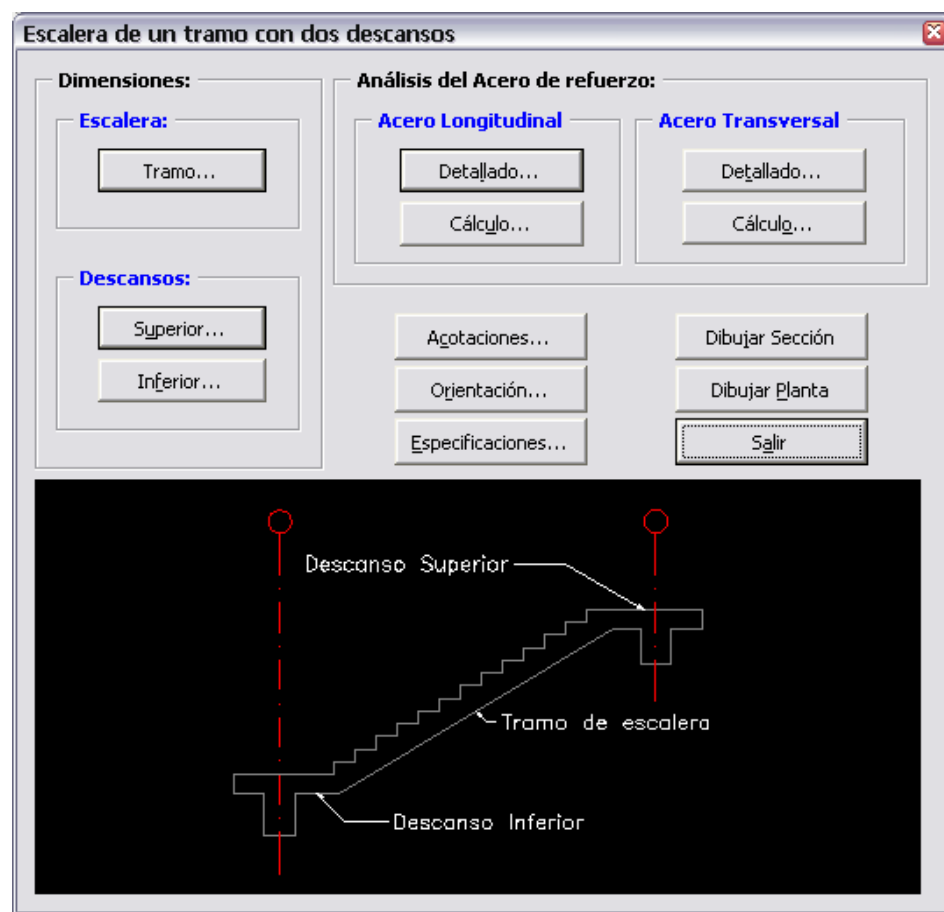


Fig. 26 FormaDED

de la escalera de un tramo con dos descansos.

Este formulario consta de:

1. Botones:

1.1. Escalera:

1.1.1. Tramo...: este boton llama al formulario FormaEscalera con los datos del tramo de escalera para la escalera de un tramo con dos descansos.

1.2. Descansos:

1.2.1. Superior...: Llama al formulario FormaDescanso con los datos del descanso superior.

1.2.2. Inferior...: Llama al formulario FormaDescanso con los datos del descanso inferior.

1.2.3. Dibujar Sección: este boton solicita un punto de ubicación para ejecutar el procedimiento DibujarConjuntoDescansoEscaleraDescanso.

1.2.4. Dibujar Planta: Botón que solicita un punto para ejecutar el procedimiento DibujarPlantaDED.

Los demás botones aplican de la misma forma que se describen en la escalera autoportante.

4.2.4.8 FormaDEDED

Este módulo fue diseñado para introducir todos los datos relacionados con el dibujo de la escalera de dos tramos con tres descansos (Fig. 27). Este módulo recopila los otros módulos de formularios que corresponda a cada caso. Este es el módulo inicial de la escalera de dos tramos con tres descansos.

Este formulario consta de:

1. Botones:

1.1. Tramos de escalera:

1.1.1. Superior...: Este botón llama al formulario FormaEscalera (Fig. 29), con los datos que corresponden al tramo de escalera superior.

1.1.2. Inferior...: Este botón llama al formulario FormaEscalera (Fig. 29), con los datos que corresponden al tramo de escalera inferior.

1.2. Descansos:

1.2.1. Superior...: Llama al formulario FormaDescanso, con los datos del descanso superior.

1.2.2. Intermedio...: Llama al formulario FormaDescanso, con los datos del descanso intermedio.

- 1.2.3. Inferior...: Llama al formulario FormaDescanso, con los datos del descanso inferior.
- 1.3. Dibujar Sección: Solicita un punto para ejecutar el procedimiento DibujarConjuntoEscaleraDescansoEscalera.
- 1.4. Dibujar Planta: Botón que solicita un punto para ejecutar el procedimiento DibujarPlantaDEDED.

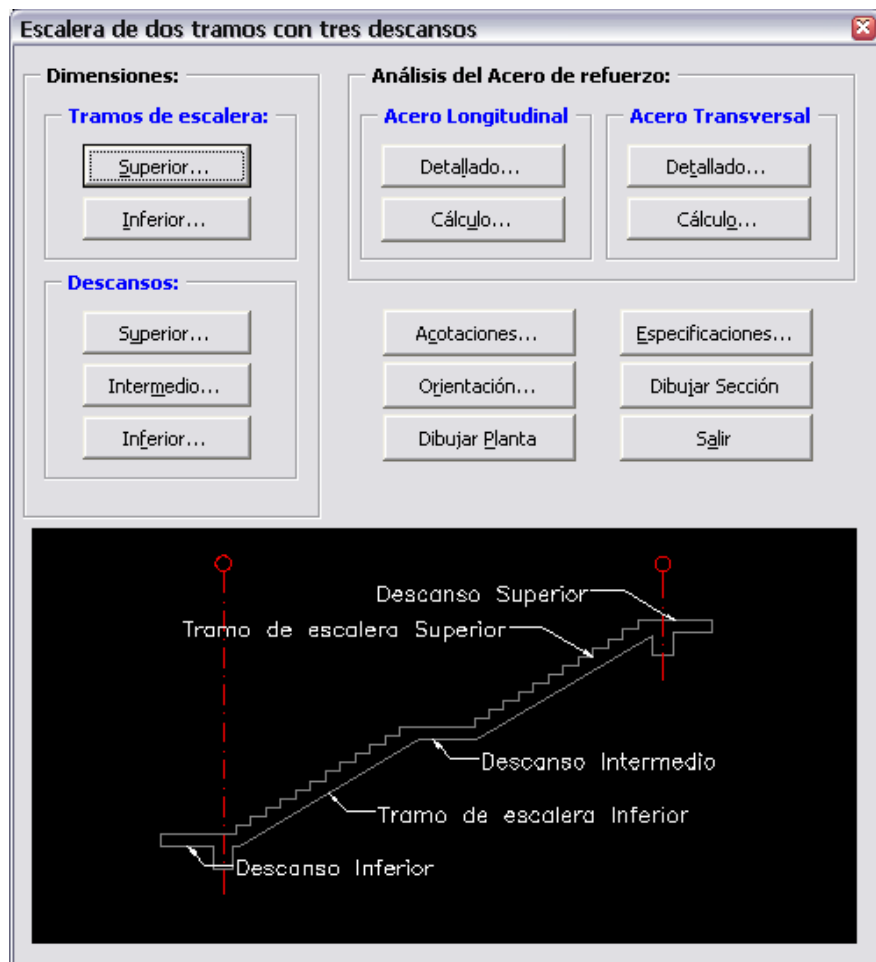


Fig. 27 FormaDEDED

4.2.4.9 FormaDescanso

Este módulo de formulario se diseñó para introducir los datos del descanso que corresponda (Fig. 28).

Este formulario consta de:

1. Cuadros de textos:

1.1. Ancho: cuadro que contiene al ancho del descanso.

1.2. Espesor: cuadro que contiene al espesor del descanso.

2. Botones de opción:

2.1. Nervio (o viga): Llama al formulario FormaApoyo con los datos del nervio, y a su vez, selecciona al nervio como el tipo de apoyo a representar.

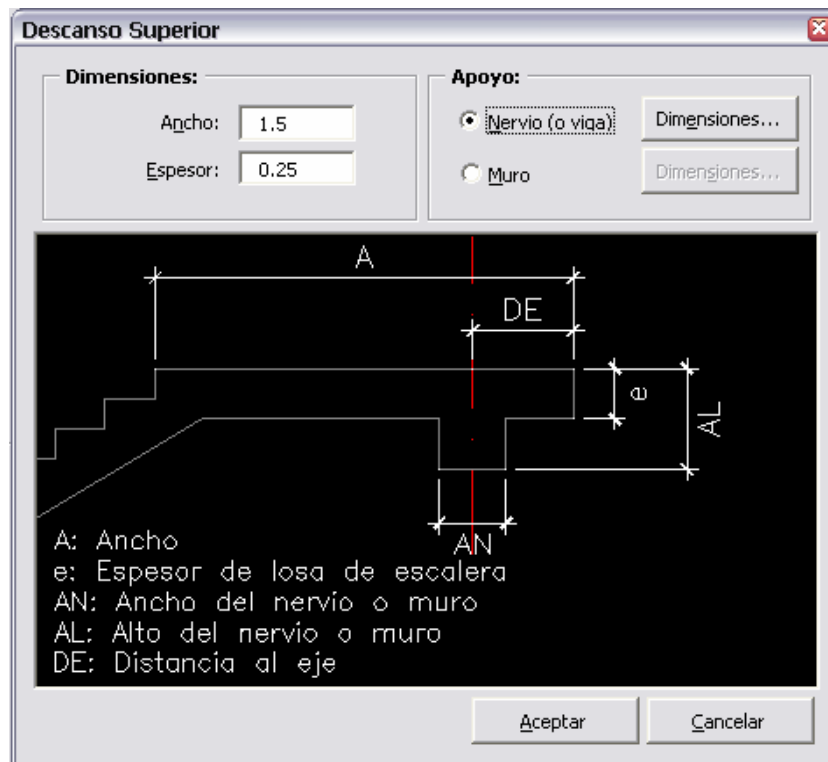


Fig. 28 FormaDescanso

2.2. Muro: Llama al formulario FormaApoyo (Fig. 23), con los datos del muro, y a su vez, selecciona al muro como el tipo de apoyo a representar.

3. Botones:

3.1. Dimensiones...: Este botón, dependiendo del tipo de apoyo, llama al formulario FormaApoyo (Fig. 23), con los datos del tipo de apoyo seleccionado.

4.2.4.10 FormaEscalera

Formulario creado para introducir los datos del tramo de la escalera que corresponda (Fig. 29) Formulario utilizado para las escaleras de un tramo con dos descansos y la escalera de dos tramos con tres descansos.

Esta forma consta de:

1. Cuadros de textos:

- 1.1. Luz libre de escalones: cuadro de texto que contiene la distancia entre el primer y el último escalón.

The image shows a software dialog box titled "Tramo de escalera". It has a tab labeled "Tramo Superior". Inside the tab, there are five input fields with their respective values: "Luz libre de los escalones" (3.3), "Desnivel entre descansos" (2.15), "Espesor de la losa de la escalera" (0.25), "Número de escalones" (11), and "Ancho del escalón" (1.5). Below these fields is a diagram of a staircase. The diagram shows a series of steps with a horizontal line at the top labeled "L" (Luz Libre), a vertical line on the left labeled "D" (Desnivel), and a small vertical line on a step labeled "e" (Espesor de losa de escalera). A legend below the diagram defines these labels: "L: Luz Libre", "D: Desnivel", and "e: Espesor de losa de escalera". At the bottom of the dialog box are two buttons: "Aceptar" and "Cancelar".

Fig. 29 FormaEscalera

- 1.2. Desnivel entre descansos: Como su nombre lo indica, contiene el desnivel entre descansos consecutivos.
- 1.3. Espesor de la losa de la escalera: contiene el espesor de la losa del tramo de escalera.
-

- 1.4. Número de escalones: Contiene el número de escalones del tramo de la escalera.
- 1.5. Ancho del escalón: Cuadro que contiene el ancho del escalón, y a su vez, el ancho de toda la escalera.

4.2.4.11 FormaEspecificaciones

Formulario diseñado para introducir los datos relacionados con las variables de cálculo (cargas permanentes, variables, resistencia del concreto, del acero...) y

Especificaciones

Cálculo | Dibujo

Variables del diseño:

Resistencia especificada del concreto en compresión (f'_c , kg/cm²): 250

Resistencia cedente especificada en armadura (f_y , kg/cm²): 4200

Recubrimiento de cálculo (r_c , m): 0.05

Factor de minoración de resistencia (ϕ): 0.9

Cuantía mínima de acero transversal (ρ): 0.0018

Módulo de elasticidad del acero (E_s , kg/cm²): 2100000

Cargas:

Permanentes:

Peso Unitario del concreto armado (γ , kg/m³): 2500

Factor de mayoración de carga: 1.4

Cargas adicionales (kg/m²): 1400

Variables:

Carga distribuida variable (kg/m²): 100

Factor de mayoración de carga: 1.7

Aceptar Cancelar

Fig. 30 FormaEspecificaciones

dibujo (escalas, estilos, dimensiones mínimas y máximas...).

4.2.4.12 FormaOrientacion

Formulario diseñado para definir la orientación del dibujo de la escalera, tanto en la sección transversal, como en planta. Esta forma contiene dos botones de opción para seleccionar la orientación de la escalera al dibujar (Fig. 31).

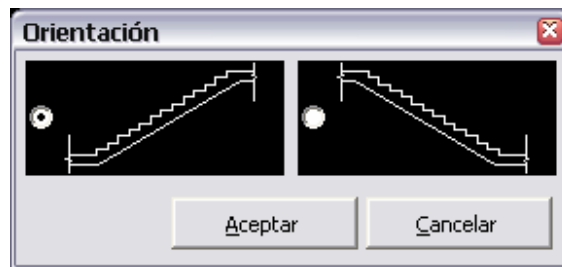


Fig. 31 FormaOrientacion

5 CONCLUSIONES

Después de haber realizado el programa que realiza el dibujo de escaleras de concreto armado, acompañado de su análisis y diseño, se puede decir que el programa cumple con los requisitos establecidos en los objetivos.

Se hizo la comparación del armado de una escalera ya hecho con el armado realizado con el programa, y los resultados estuvieron dentro del rango que establece el programa.

Se requiere de mucho tiempo para crear un programa completo, que reúna todas las variables que condicionan a la resolución de un problema.

Los módulos creados se pueden utilizar en cualquier otro programa de AutoCAD VBA, y algunos de éstos, en otros programas externos.

El cálculo de estructuras de concreto armado en muchos casos resulta complejo, y se invierte mucho tiempo para realizarlo, con el riesgo de cometer algún error, es aquí donde cobra importancia la programación para minimizar el tiempo de trabajo.

6 RECOMENDACIONES

A los usuarios del programa:

- En vista del gran volumen de datos que tiene el programa, fijar todos los datos referidos al dibujo de la escalera, para que solamente se haga la parte de cálculo de la escalera. Esto es permisible ya que el programa usa los datos de un archivo tipo texto, se ejecuta el programa, y al salir, guarda las modificaciones en el mismo archivo.
- Las especificaciones es un formulario creado para colocar todos los datos relacionados con las especificaciones. Solamente los resultados que arroje el programa serán confiables siempre y cuando los valores de las especificaciones estén acorde con valores comunes, por ejemplo, el programa permite colocar la luz mínima y máxima, puede estar en el orden de 2 a 4 m respectivamente, esto origina que al colocar una luz fuera de este intervalo en cualquiera de las escaleras, genere un error. Sin embargo si este intervalo se coloca entre 0 y 1000 m, el programa aceptará el valor que esté entre 0 y 1000 m y realizará el cálculo y dibujará la escalera, pero estos resultados no son aceptables.

A los usuarios de VBA (o AutoCAD® VBA):

- Establecer la relación entre el tiempo a invertir en el trabajo a realizar y el tiempo en hacer un programa que realice este trabajo, tomando en cuenta la frecuencia en que se use el programa. De esto dependerá realizar el programa o no.

A futuros trabajos especiales de grado:

- Si se realizara un trabajo similar pero con otras estructuras, y se maneja la representación del acero, crear una biblioteca de tipos de armados de acero, para seleccionar el tipo de acero que se desee; y no solamente para tipos de acero, sino para cualquier parámetro que tenga muchas posibilidades de presentar cualquier forma.
-

7 BIBLIOGRAFIA

Sauter, Franz, Free-Standing Stairs, ACI Publications, 1964.

Beton-Kalender, Manual teórico práctico del hormigón, Editorial “El Ateneo”, Buenos Aires, 1957.

Ministerio de Desarrollo Urbano, (COVENIN-MINDUR 1753-85) Estructuras de concreto armado para edificaciones. Análisis y diseño, Caracas, 1985.

Grossman, Stanley, Álgebra lineal, McGraw-Hill, 1996.

Oser, Rodolfo, Flujogramas para el cálculo de concreto armado.

<http://www.lawebdelprogramador.com>

Microsoft Corp., MSDN Library Visual Studio, EE.UU., 1991.

8 ANEXOS

EscalerasCAD Versión 1.0

Manual del usuario

INDICE:

1	INSTALACIÓN:	3
2	COMENZAR A USAR:	4
2.1	Escalera Autoportante:	4
2.2	Escalera de un tramo con dos descansos.	13
2.3	Escalera de dos tramos con tres descansos.	15
2.4	Especificaciones:	17
2.5	Acerca de.	17

1 Instalación:

Para la instalación de EscalerasCAD, sobre el entorno de AutoCAD, primero debe cargar el archivo ESCALERAS.lsp, que genera un menú adicional sobre la barra de menú de AutoCAD, que se llama Escaleras. Este archivo se carga con la función `Load Application...`, que muestra un cuadro de diálogo donde se puede cargar o descargar aplicaciones. La ruta de acceso para ubicar esta función es: `Tools>Load Application...`; también puede tener acceso a esta función escribiendo sobre la línea de comando `appload`.

Luego de ubicar el cuadro de diálogo, se busca el archivo ESCALERAS.lsp, que está en la carpeta `C:\EscalerasCAD`, se selecciona y se carga presionando `Load`. En el cuadro de diálogo debe aparecer un mensaje así: `ESCALERAS.lsp successfully loaded`, que indica que el archivo fue cargado sin problemas. Salga del cuadro de diálogo presionando `Close`.

En la línea de comando, escribe `escalerascad`, y aparecerá en la barra de menú de AutoCAD, un menú adicional titulado Escaleras, y el programa está listo para ser ejecutado.

2 Comenzar a usar:

Ya instalado el programa, ya puede comenzar a utilizar el programa EscalerasCAD, seleccionando la escalera de su interés, y aparecerá un cuadro de diálogo donde usted podrá dibujar la planta de la escalera seleccionada, y las secciones transversales de esa escalera. Cada cuadro de diálogo es distinto, es por ello que a continuación se hará una breve descripción del funcionamiento del programa, a través de cada tipo de escalera.

2.1 Escalera Autoportante:

La escalera autoportante se ubica en Autoportante..., donde aparece un cuadro de

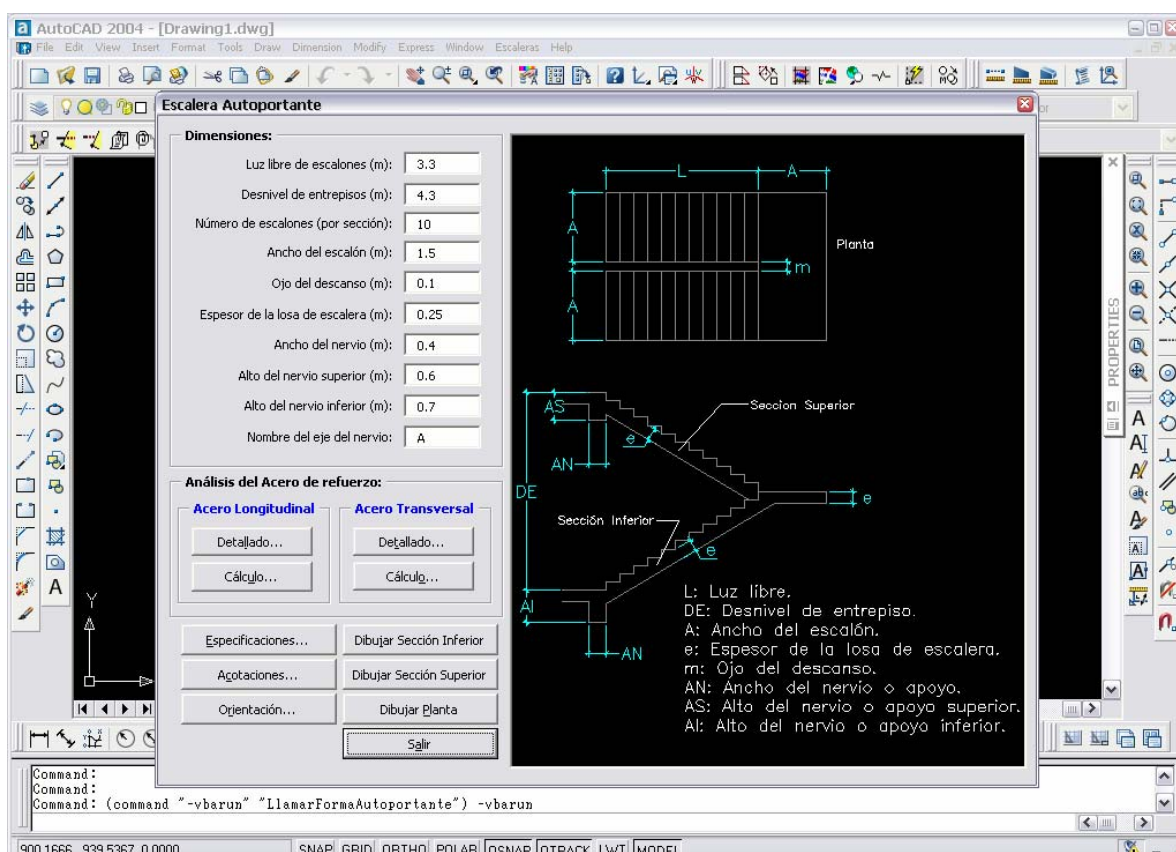
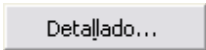


Fig. 1 Escalera Autoportante

diálogo como el de la Fig. 1:

Esta figura muestra un cuadro de diálogo donde se pueden modificar los datos de la escalera autoportante. En la figura aparece una imagen auxiliar a la derecha que define que es cada uno de los datos de la escalera autoportante, ya que posee una leyenda de lo que significa cada una de las variables; por ejemplo, el desnivel de entresijos está denotado en la imagen con las siglas “DE” que aparece en la leyenda. Es importante señalar que cada una de las escaleras tiene una imagen auxiliar que definen los parámetros de cada una de las escaleras.

Ahora se define cada uno de los botones:

El botón es rectangular con un fondo gris claro y el texto "Detallado..." en un color gris oscuro.

Este botón, para el acero longitudinal, abre un cuadro de diálogo que reúne las variables del acero longitudinal (Fig. 2). Este cuadro permite seleccionar en la lista “a” de la Fig. 2, un tipo de acero, y modificar sus propiedades en los marcos “b” y “c”. Al igual que los cuadros de diálogos de las escaleras, este cuadro también tiene imágenes auxiliares acorde con el tipo de acero seleccionado. El espaciamiento es la separación entre barras de acero, este dato es importante al momento de hacer el cálculo. En el marco “b” se colocan las longitudes horizontales y/o inclinadas existentes para cada tipo de acero, donde su longitud mínima depende del cálculo. Es por ello que antes de hacer cualquier modificación, debe hacer el cálculo. Estas longitudes aparecen, para cada tipo de acero, en la imagen auxiliar. En el marco “c”, se colocan las distancias y los ángulos en que se va a dibujar el despiece. El acero en el despiece se puede dibujar a una cierta distancia y un cierto ángulo. La distancia de texto es la separación entre el texto del acero y el dibujo del acero.

Para el acero transversal, aparece un cuadro de diálogo donde se pueden ubicar las variables del acero transversal (Fig. 3). En este cuadro se pueden colocar los datos que permiten dibujar el refuerzo transversal. Al igual que para el acero longitudinal, se selecciona un tipo de acero en la lista “a” de la Fig. 3 y se puede modificar sus propiedades en los campos “b” y “c”. El espaciamiento es la separación entre barras de refuerzo, requisito para el cálculo del acero transversal. En el marco “b”, aparecen los campos necesarios para hacer las acotaciones del acero transversal; la distancia horizontal y la distancia vertical, son las que se encuentran representadas en la

imagen superior de la Fig. 3, denotadas por “DH” y “DV”, respectivamente. El porcentaje de ubicación es la relación entre la ubicación de la acotación y la longitud total del acero, y también está representada en la imagen superior de la Fig. 3. El campo “c” contiene la orientación que recibirá el acero seleccionado

NOTA: La acotación es independiente de la orientación en que se dibuje la escalera, es decir, no cambia la orientación de la acotación transversal si se cambia la orientación en que se dibuja la escalera.

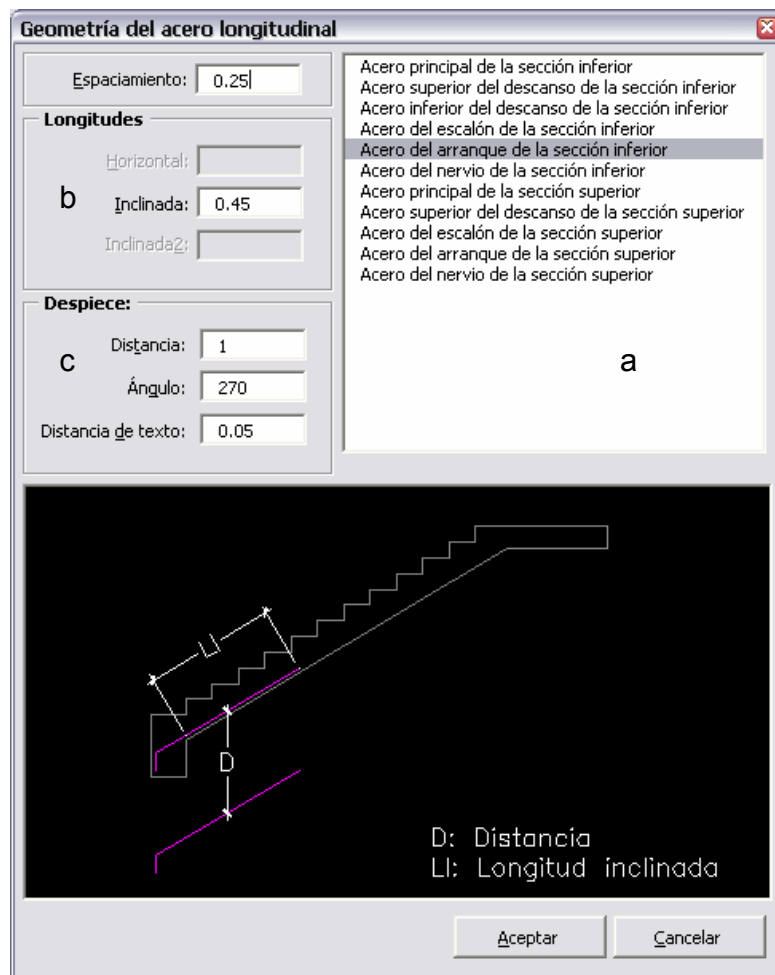


Fig. 2 Cuadro de diálogo para acero longitudinal

Cálculo...

Al presionar este botón aparece un cuadro de diálogo destinado a definir el diámetro del acero para cualquiera de los casos longitudinal y transversal (Fig. 4). En este cuadro se selecciona el tipo de acero en la lista “a” de la Fig. 4, y en el campo “b” se selecciona el diámetro de la cabilla. Este campo es una lista que

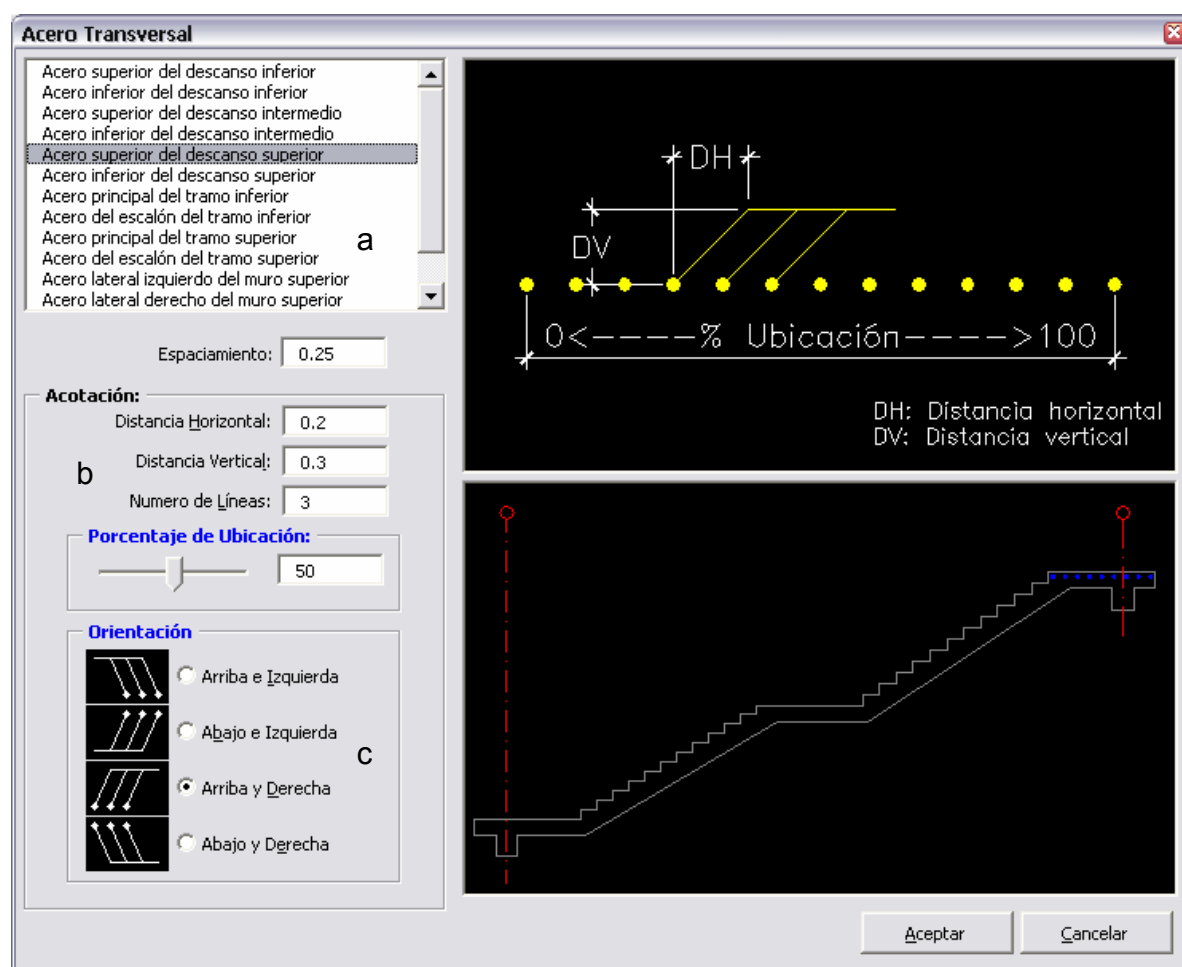


Fig. 3 Cuadro de diálogo para acero transversal

contiene todos los diámetros desde el mínimo por cálculo hasta el acero de calibre máximo (Nº 11) y adicionalmente aparece el área de acero mínimo requerido y el área de acero de la cabilla. Es importante resaltar la nota que aparece en el campo “b” que indica que si el área de acero está por encima del acero Nº 11, el área de acero cambia de color de negro a rojo, ya que se podría apreciar el área de acero requerido (superior al calibre Nº 11). Por defecto, si ocurre esta situación, en el dibujo se coloca el acero máximo (Nº 11). La selección del tipo de acero influye en el detallado de los aceros longitudinales, sobre todo en las longitudes horizontales e

inclinadas. Es por ello que, al iniciar el programa, debe hacerse primero el cálculo, y luego el detallado. Si desea modificar el espaciamiento de algún tipo de acero longitudinal, debe realizar el cálculo antes de iniciar el dibujo; de lo contrario, el dibujo no estará acorde con el cálculo. En resumen, cada vez que modifique el espaciamiento, debe realizar el cálculo.

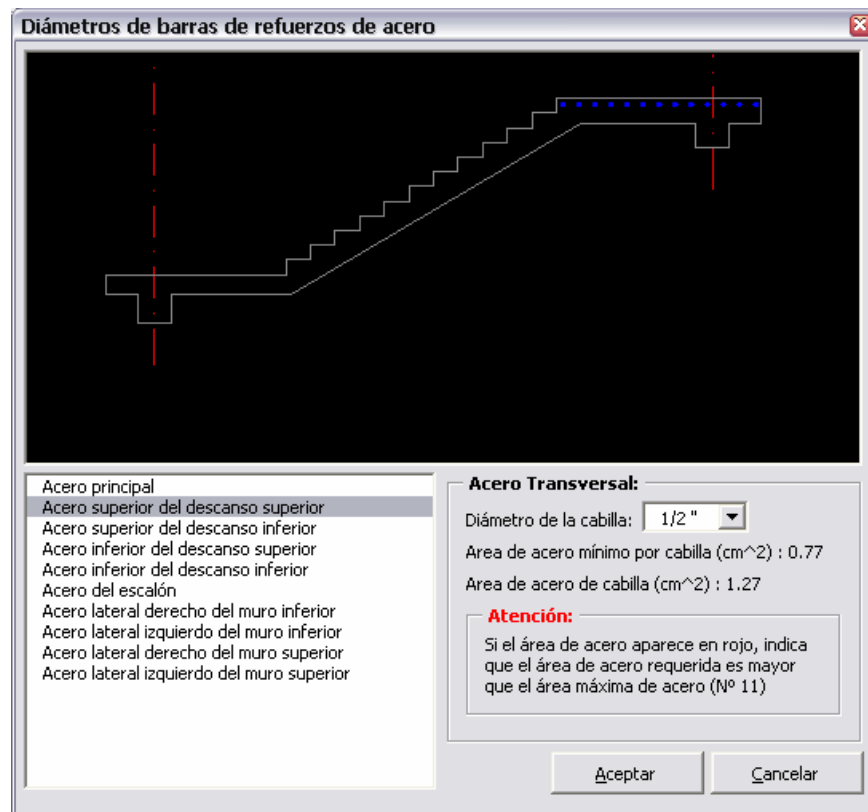


Fig. 4 Cuadro de diálogo para el cálculo del refuerzo

Especificaciones...

Este botón, al presionarlo, aparece el cuadro de diálogo de las especificaciones para el cálculo y dibujo de la escalera (Fig.5). Este cuadro se compone de dos partes:

1. Cálculo: En esta parte se introducen todos los datos relacionados con el cálculo del refuerzo de acero, tanto longitudinal, como transversal (Fig. 5a). Estos datos son:

Especificaciones (a) Cálculo

Variables del diseño:

Resistencia especificada del concreto en compresión (f'_c , kg/cm ²):	250
Resistencia cedente especificada en armadura (f_y , kg/cm ²):	4200
Recubrimiento de cálculo (r_c , m):	0.05
Factor de minoración de resistencia (ϕ):	0.9
Cuantía mínima de acero transversal (ρ):	0.0018
Módulo de elasticidad del acero (E_s , kg/cm ²):	2100000

Cargas:

Permanentes:

Peso Unitario del concreto armado (γ , kg/m ³):	2500
Factor de mayoración de carga:	1.4
Cargas adicionales (kg/m ²):	1400

Variables:

Carga distribuida variable (kg/m ²):	100
Factor de mayoración de carga:	1.7

Especificaciones (b) Dibujo

Variables Geométricas (Valores límites):

Luz libre:	Entre: 0.5	y: 3.9	m.
Desnivel:	Entre: 1	y: 3	m.
Ancho del descanso:	Entre: 0.5	y: 1.5	m.
Espesor de la losa de la escalera:	Entre: 0.15	y: 0.25	m.
Número de escalones:	Entre: 3	y: 15	m.
Ancho del escalón:	Entre: 0.5	y: 1.5	m.

Variables de impresión

Escala de Impresión: 1: 100

Estilos:

Estilo de dimensionamiento:	D100
Estilo de acotación de cabillas:	C100
Estilo de Texto:	L100
Estilo de texto del eje del nervio:	L300

Otras variables:

Separaciones:

Horizontal (m):	0.0125
Vertical (m):	0.0125

Precisión:

Múltiplo:	5
Decimales:	2

Radio (m): 0.01

Fig. 5 Cuadro de diálogo para especificaciones: a) Especificaciones de cálculo. b) Especificaciones de dibujo.

1.1.1. Variables del diseño: esta parte contiene los datos necesarios para hacer el cálculo del área del refuerzo de acero: la resistencia especificada del concreto a compresión (f'_c), resistencia cedente especificada en armadura (f_y), recubrimiento de cálculo (r_c), factor de minoración de resistencia (ϕ), cuantía mínima de acero transversal (ρ), Módulo de elasticidad del acero (E_s).

- 1.2. Cargas: Esta parte contiene las variables que condicionan a la carga permanente y a la sobrecarga que está sobre la escalera.

1.2.1. Para la carga permanente se tiene el peso unitario del concreto (γ) para determinar el volumen de concreto de la escalera, factor de mayoración de carga, y cargas adicionales, como peso de las barandas, tabiquería, pavimentos etc.

1.2.2. Para la carga variable se tiene la carga distribuida y su factor de mayoración.

2. Dibujo: En esta parte se introducen los datos que relacionan al dibujo (Fig. 5b).

Estas variables son:

- 2.1. Variables geométricas (Valores límites): En esta parte se colocan los valores mínimos y máximos de las dimensiones de la escalera.
- 2.2. Variables de impresión: Esta parte incluye la escala de impresión en que se desea realizar el dibujo, los estilos de dimensionamiento, los estilos de acotación de cabillas y de texto, tanto general como para el estilo del eje del nervio.
- 2.3. Otras variables: Esta parte incluye valores como: la separación que debe haber entre aceros longitudinales al dibujar, en sus dos direcciones; el múltiplo y los decimales a considerar para hacer los redondeos de los valores representados en el dibujo, como aceros longitudinales, ganchos; y el radio de las circunferencias del acero transversal,

Acotaciones...

Aparece un cuadro de diálogo que reúne los datos relacionados con ciertas acotaciones de la escalera (Fig. 6), como la cota de ha huella, la contrahuella, el espesor de la losa de la escalera, y las acotaciones horizontales y verticales. Esta parte depende de la escalera que se esté dibujando, es por ello que este cuadro de diálogo, en forma análoga a otros cuadros de diálogo, tiene una lista "a" de la Fig. 6, donde se selecciona el tipo de acotación y se modifican sus propiedades en el marco "b". Este marco contiene la distancia entre el elemento que se acota y el punto de ubicación de la acotación; también incluye, al igual que para el acero transversal, el porcentaje de ubicación, que es la relación entre la longitud en que se desea acotar y la longitud total donde se puede acotar.

Orientación...

Al presionar este botón, aparece un cuadro de diálogo que determina la orientación en que se dibujará la escalera (Fig. 7). Este cuadro es independiente del tipo de escalera que se esté dibujando, y contiene dos botones de

opción, el botón “a” de la Fig. 7 define la escalera que se dibuja por defecto, el botón “b” dibuja la parte simétrica de la escalera por defecto.

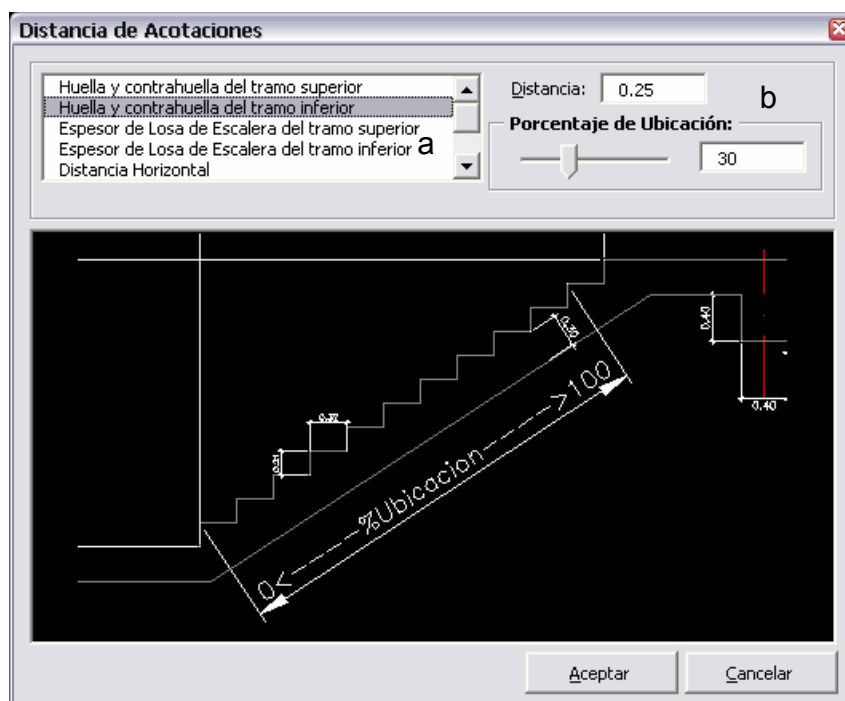


Fig. 6 Cuadro de diálogo para las acotaciones de la escalera

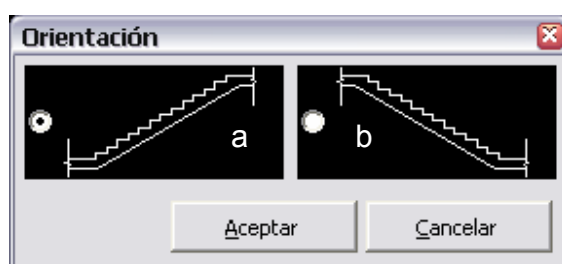
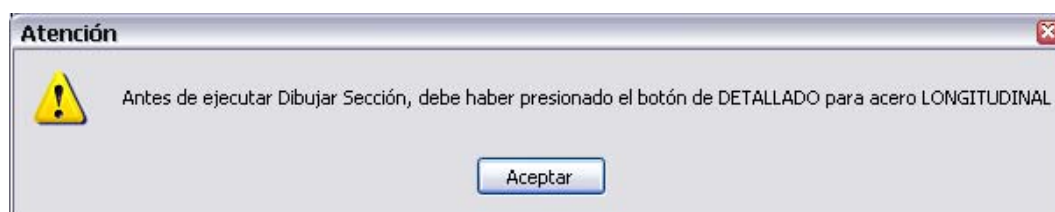


Fig. 7 Cuadro de diálogo para la orientación de la escalera

Dibujar Sección Inferior

Aquí procede a dibujar la sección inferior de la escalera autoportante, solicitando un punto de ubicación para la el dibujo de la escalera. A continuación solicita un punto para la ubicación del título (Sección Transversal Inferior).

Nota: Esta ejecución se realiza siempre y cuando haya presionado previamente Detallado y Cálculo para acero longitudinal, y solamente Cálculo para acero transversal, de lo contrario, antes de iniciar el dibujo, aparece un mensaje de advertencia como el que sigue:

**Dibujar Sección Superior**

Este botón realiza las mismas operaciones del botón Dibujar Sección Inferior, sólo que estas operaciones corresponden al dibujo de la sección superior de la escalera: solicita un punto de ubicación para el dibujo de la sección superior de la escalera autoportante, y a continuación solicita un punto para ubicar el título (Sección Transversal Superior). También aplica la condición de que debe presionar previamente el Detallado y el Cálculo para acero longitudinal, y sólo Cálculo para acero transversal.

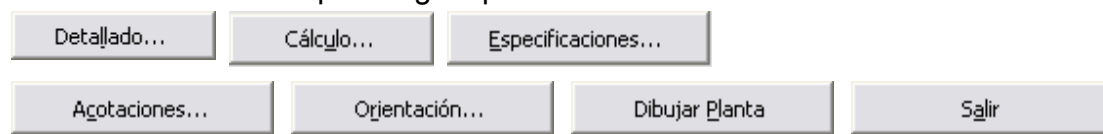
Dibujar Planta

Este botón solicita un punto para ubicar la planta de la escalera autoportante y solicita un punto de ubicación para ubicar el título (Planta).

Salir

Este botón sale del cuadro de diálogo escalera autoportante. Al salir, si ha hecho cualquier modificación, éstos cambios serán guardados en un archivo tipo texto, para poder utilizarlos en otra ocasión (solamente los parámetros que no requieren ser calculados, como dimensiones, desnivel, espesor...).

Nota: Estos botones aplican igual para las otras escaleras:



2.2 Escalera de un tramo con dos descansos.

Esta escalera aparece en la barra de menú como Un tramo y dos descansos...

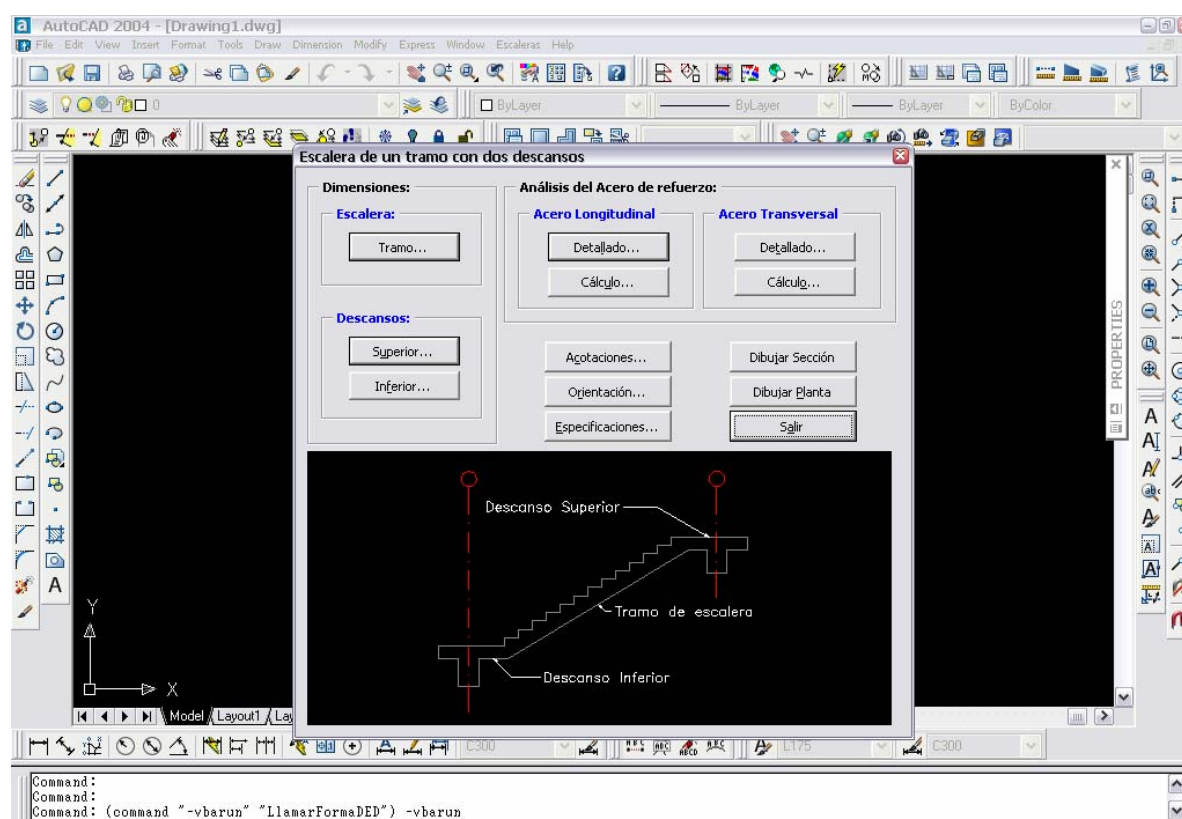


Fig. 6 Escalera de un tramo con dos descansos

donde aparece el cuadro de diálogo de la Fig. 8

Este cuadro de diálogo contiene una serie de botones, muchos de éstos son comunes para las otras escaleras.

Tramo... Aparece un cuadro de diálogo que contiene los datos de la escalera (Fig. 9). Este cuadro contiene la luz libre de los escalones (L), desnivel entre

descansos consecutivos (D), espesor de la losa de la escalera (e), número de los

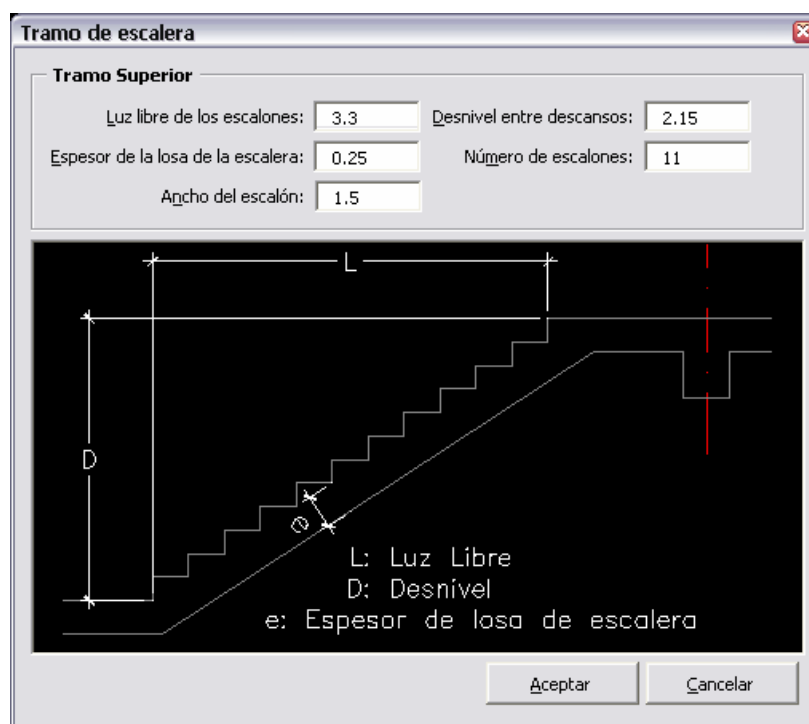


Fig. 7 Cuadro de diálogo para escalera

escalones, y el ancho del escalón.

Para los descansos:

Superior... Abre un cuadro de diálogo que contiene el ancho y el espesor del descanso superior (Fig. 10), y además, se puede especificar el tipo de apoyo que se desee: nervio (o viga, y muro). Las dimensiones del apoyo seleccionado se pueden ubicar presionando **Dimensiones...**, que muestra un cuadro de dialogo (Fig. 11) que contiene, además de las dimensiones y ubicación del eje del apoyo, el nombre del eje del apoyo, y las distancias superior (DS) e inferior (DI) del eje del nervio.

Nota: De no seleccionar el tipo de apoyo, por defecto se dibujará el nervio o viga

Inferior... Abre el mismo cuadro de diálogo del descanso (Fig. 11), con la variante que contiene el ancho y espesor del descanso inferior.

Dibujar Sección

Hace el dibujo de la sección transversal de la escalera de un tramo con dos descansos, solicitando un punto de ubicación para el dibujo, y un punto de ubicación para el título (Sección Transversal).

2.3 Escalera de dos tramos con tres descansos.

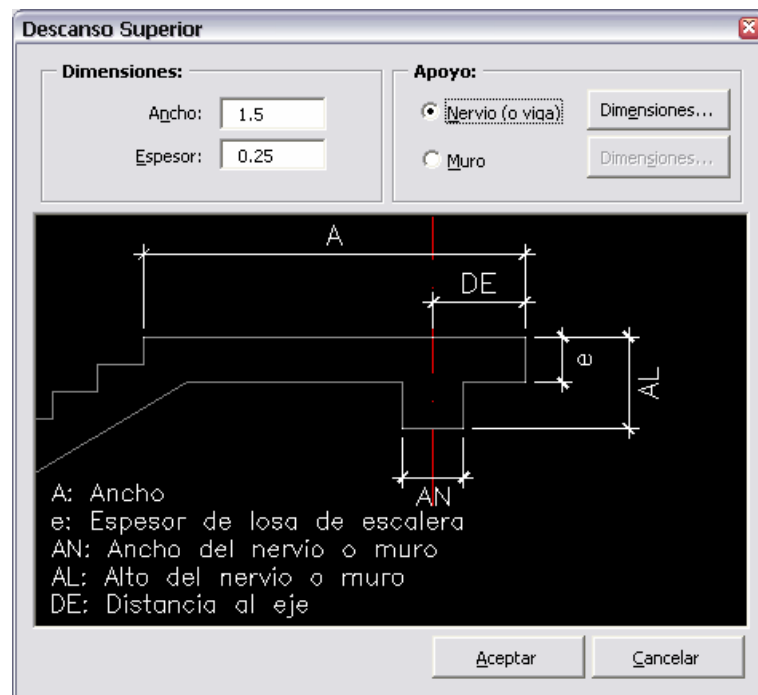


Fig. 8 Cuadro de diálogo para descansos

Aparece un cuadro de diálogo (Fig. 12) al presionar, en la barra de menú, Dos tramos y tres descansos....

Este cuadro contiene una serie de botones, similares a la de la escalera de un tramo con dos descansos.

Para el tramo de escalera:

Superior...

Abre el cuadro de diálogo para la escalera (Fig. 9), con los datos del tramo superior de la escalera.

Inferior...

Abre el cuadro de diálogo para la escalera (Fig. 9), con los datos del tramo inferior de la escalera.

Para el descanso:

Superior e inferior: similar al de la escalera de un tramo con dos descansos.

Intermedio...

Abre el cuadro de diálogo del descanso con los datos del descanso intermedio. En este caso, aparecen desactivados los botones de apoyo.

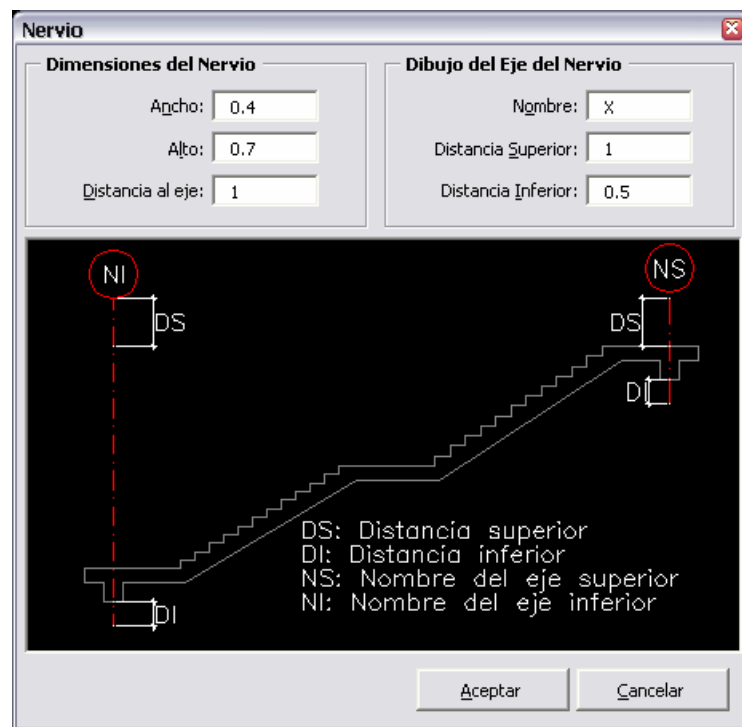


Fig. 9 Cuadro de diálogo para el apoyo del descanso

2.4 Especificaciones:

En el menú Escaleras, también se puede tener acceso a las especificaciones (Fig. 5) que se puede acceder en cada uno de los cuadros de diálogos, presionando Especificaciones... en el menú de escaleras.

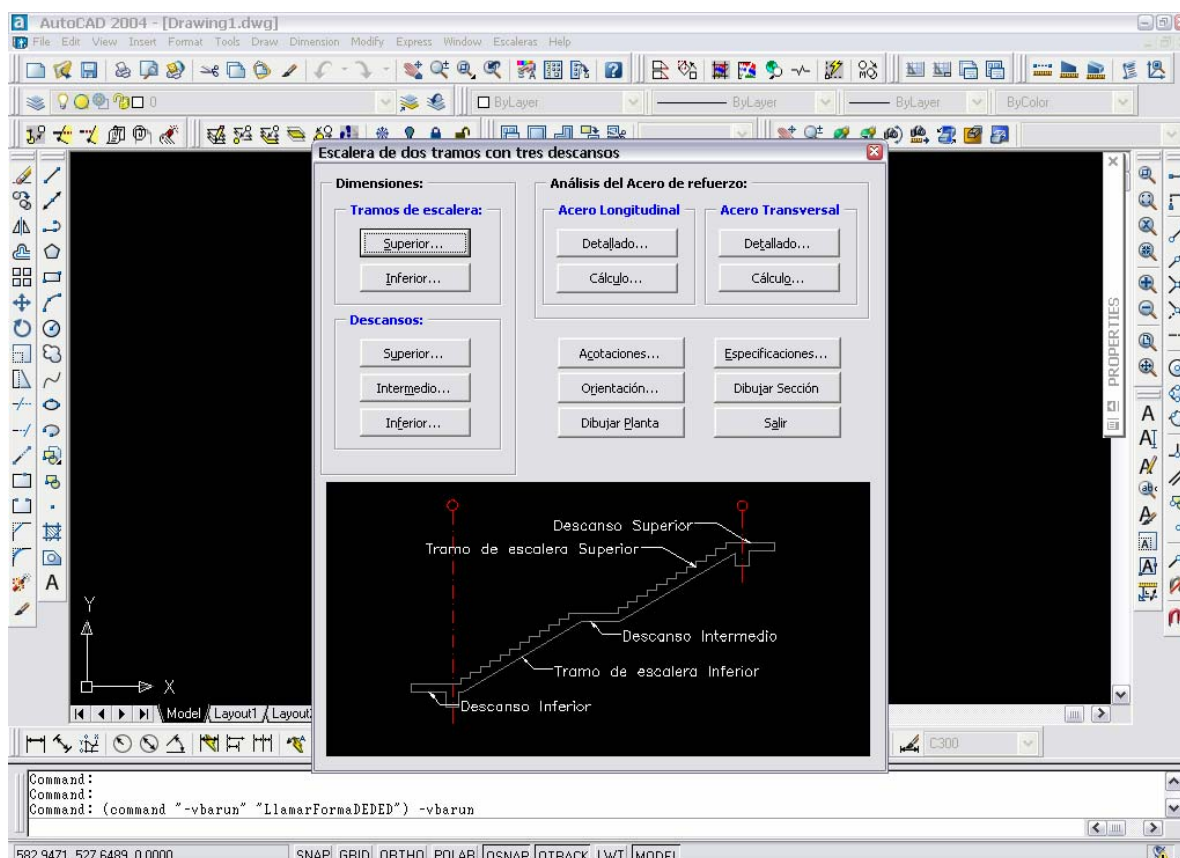


Fig. 10 Escalera de dos tramos con tres descansos.

2.5 Acerca de...

Este cuadro contiene información acerca del programa: nombre, versión, autor, descripción, correo electrónico para indicar cualquier problema, sugerencia, comentarios.