

## **AUTOMATIZACIÓN CARTOGRÁFICA CON VISUAL BASIC Y AUTOCAD**

Mora Navarro, J. Gaspar  
Profesor Colaborador  
Departamento de Ingeniería Cartográfica, Geodesia y Fotogrametría  
Universidad Politécnica de Valencia  
Camino de Vera, s/n 46022 Valencia (España)  
e-mail: joamona@cgf.upv.es

Femenia Ribera, Carmen  
Profesora Titular de Escuela Universitaria  
Departamento de Ingeniería Cartográfica, Geodesia y Fotogrametría  
Universidad Politécnica de Valencia  
Camino de Vera, s/n 46022 Valencia (España)  
e-mail: cfemenia@cgf.upv.es

**Palabras Clave:** Aplicaciones cartográficas, Visual Basic, AutoCAD, Microstation

### **Resumen**

Los programas informáticos, en general, están diseñados para satisfacer las necesidades de un gran número de usuarios; un programa de diseño asistido por ordenador se puede utilizar, tanto para dibujar todas las piezas de un motor como para realizar un mapa topográfico. Es claro que en ingeniería industrial se utilizan herramientas, procesos y cálculos muy distintos a los utilizados en topografía y que, en ninguno de los dos casos, el programa de dibujo utilizado proporciona todas las herramientas que los ingenieros proyectistas hubiesen deseado. Es por ello que las empresas de software, proporcionan, con los programas que comercializan, uno o varios lenguajes de programación con los que los usuarios pueden crear aplicaciones que se añaden a las herramientas del programa, y que resuelven los problemas específicos de cada disciplina.

En este documento se exponen varios ejemplos de nuevas herramientas, programadas en el lenguaje de programación Visual Basic y que se han añadido a los editores gráficos AutoCAD o Microstation, los cuales resuelven problemas cartográficos específicos. Como ejemplos se explican los siguientes: el dibujo automatizado de un fichero ASCII procedente de la restitución fotogramétrica; la extracción de la información útil para el alta gráfica catastral de urbana del proyecto de un edificio o la extracción de las coordenadas de elementos de un dibujo para su posterior replanteo.

## 1. Introducción

Visual Basic se utiliza para crear nuevas herramientas en gran cantidad de programas comerciales; como por ejemplo AutoCAD, Microstation, CorelDRAW, Microsoft Word, 3D Studio, etc...; en todos ellos el código se introduce en el mismo editor de Visual Basic.

Para poder interactuar con los programas comerciales como AutoCAD, Microstation, Word, etc... hay que aprender a manejar los *objetos* propios de cada uno de ellos. Cada objeto tiene unas características que definen su apariencia, llamadas *propiedades*, y unas operaciones que puede realizar, llamadas *métodos*. Si se aprende Visual Basic, lenguaje cuya característica más importante es su sencillez, toda la dificultad reside en saber manejar las propiedades y métodos de los objetos que vienen con el programa.

Trabajando con Visual Basic existen dos formas de personalizar un programa: una utilizando el editor de Visual Basic que se instala con el propio programa que se quiere automatizar, AutoCAD, Microstation, etc., y otra con el editor de Visual Basic 6, instalado como editor y compilador externo. En el primer caso los programas se denominan *macros* y en el segundo *programas* o *aplicaciones*. Cada una de las dos posibilidades tiene unas ventajas y unos inconvenientes.

- Si se trabaja con el editor interno de los programas:

*Ventajas:*

- No es necesario adquirir ninguna licencia de Visual Basic.

*Inconvenientes:*

- Solo se pueden ejecutar aplicaciones si se abre el programa, se abre el código correspondiente y se ejecuta.
- El código es visible y algún usuario podría modificarlo por accidente.

- Si se trabaja con un editor compilador externo:

*Ventajas:*

- Se pueden generar, además de programas que realicen tareas de personalización, cualquier ejecutable que funcione de forma independiente. Por ejemplo para conversión de formatos de ficheros, cálculos matemáticos, etc...

*Inconvenientes:*

- Es necesario comprar una licencia de Visual Basic 6.

En este documento se describen las dos formas de trabajo: en primer lugar se personaliza el programa AutoCAD realizando macros y, en segundo lugar, se automatiza el programa Microstation realizando aplicaciones ejecutables externas (ficheros \*.exe) mediante el compilador de Visual Basic 6.

Es importante resaltar que, como es lógico, antes de poder realizar macros o aplicaciones de automatización de programas, es necesario dominar el programa en cuestión. De esta forma es relativamente sencillo aprender a programar a través de los ejemplos que se encuentran en la ayuda, ya que los nombres de las funciones y objetos que se utilizan para realizar las acciones son muy descriptivos.

## 2. Programación de macros en Visual Basic para AutoCAD

Se muestra en este apartado un ejemplo de automatización con Autocad con el código de una subrutina que dibuja una línea. Se intenta plasmar la rapidez con la que se pueden realizar las tareas y la sencillez con la que interpreta el código.

En el caso de AutoCAD, el editor de Visual Basic aparece al seleccionar la opción:

Herramientas → Macro → Editor de Visual Basic

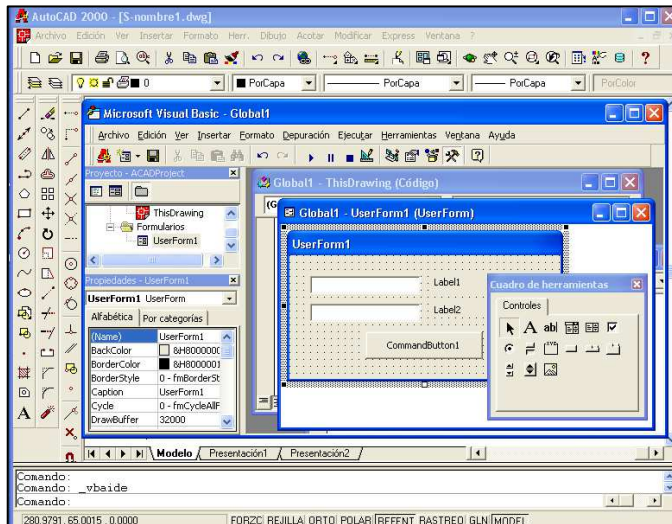


Figura 1: Editor de Visual Basic incluido en AutoCAD

El editor tiene todas las herramientas necesarias para la creación de formularios, botones, cuadros de texto, la introducción de código, depuración y ejecución de la aplicación, etc...

En la subrutina siguiente, se muestra la facilidad con la que es posible dibujar en AutoCAD programando en Visual Basic:

```

Sub EjemploDibujarLinea()
    'Este procedimiento añade una línea en el espacio modelo
    Dim objLinea As AcadLine 'Objeto línea que se creará
    Dim ptInicial(0 To 2) As Double 'Vectores coordenadas de la línea
    Dim ptFinal(0 To 2) As Double
    'Asignación de coordenadas inicial y final
    ptInicial (0) = 1#: ptInicial (1) = 1#: ptInicial (2) = 0#
    ptFinal (0) = 5#: ptFinal (1) = 5#: ptFinal (2) = 0#
    'Creación de la línea en el espacio modelo del dibujo activo
    Set objLinea = ThisDrawing.ModelSpace.AddLine(ptInicial, ptFinal)
    ZoomAll 'Realiza un zoom extensión
End Sub

```

Figura 2: Subrutina en Visual Basic que permite dibujar una línea, de coordenadas (1,1,0) a (5,5,0), en el espacio modelo

La forma más frecuente de ejecutar la subrutina es crear un formulario Windows e insertar un botón de comando, de forma que, al ejecutar la aplicación, si se presiona el botón, se ejecuta la subrutina y se dibuja la línea. Todo ello constituye la macro. Crear formularios, botones y asociar el código que se ejecuta con acciones sobre el formulario o sobre los botones, son operaciones realmente sencillas en Visual Basic y constituyen la principal ventaja frente a otros lenguajes de programación.

### 3. Ejemplos de aplicaciones realizadas en Visual Basic 6, interactuando con AutoCAD y que resuelven determinados problemas cartográficos

A continuación se explican dos nuevas herramientas, programadas en Visual Basic y añadidas a AutoCAD, las cuales resuelven problemas cartográficos específicos, como son:

- El dibujo automatizado de un fichero ASCII procedente de la restitución fotogramétrica.
- La extracción de la información útil para dar el alta gráfica catastral (en catastro de urbana) del proyecto de un edificio realizado por un arquitecto.

#### 3.1. Dibujo automatizado de un fichero ASCII procedente de restitución fotogramétrica

La *fotogrametría* es el conjunto de técnicas necesarias para obtener planos a partir de fotografías. Para realizar los planos, se colocan las fotografías en el restituidor, con el que se obtienen las coordenadas de los elementos a dibujar.

Toda esta información sobre las coordenadas se guarda en un archivo ASCII de cinco columnas: la primera y la última especifican qué tipo de elemento se ha de dibujar (línea, texto, etc...) y sobre qué capa. Las tres columnas restantes son las coordenadas de los puntos que componen cada elemento. El programa interpreta este archivo y dibuja los elementos en la capa correspondiente.

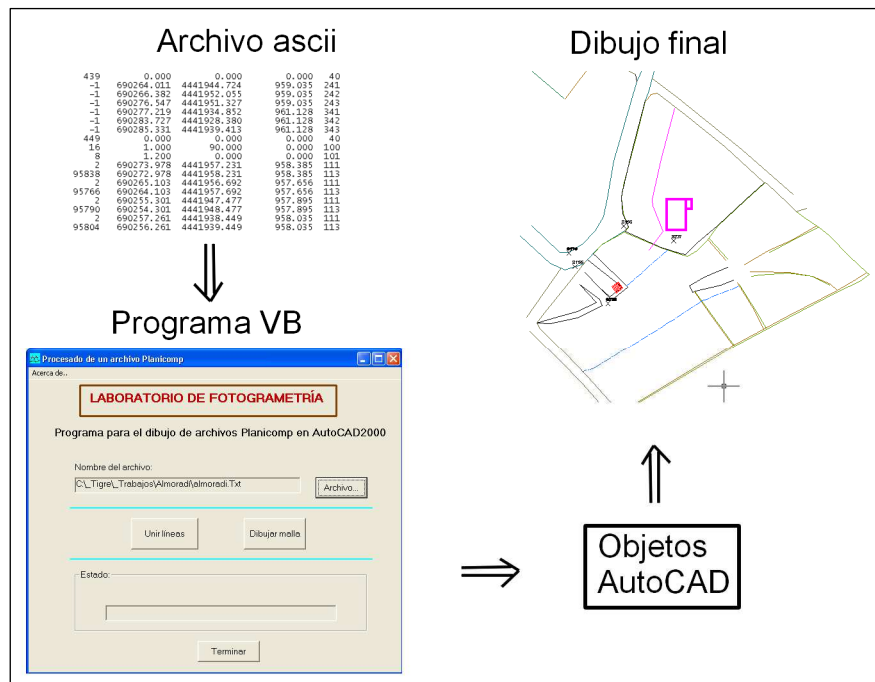


Figura 3: Esquema de creación de cartografía mediante la interpretación de un archivo ASCII

### 3.2. Extracción de la información útil para el alta gráfica catastral de urbana del proyecto de un edificio

Para realizar el alta gráfica catastral de urbana es necesario que el arquitecto entregue a la Gerencia Regional Del Catastro de Valencia (GRCV) los planos del proyecto fin de obra del edificio en cuestión. Si la entrega se realiza en ficheros de AutoCAD, la Gerencia se encuentra con gran cantidad de información que no necesita: red de agua y eléctrica, tabiquería, mobiliario, etc.... Para “extraer” solamente la información útil de los ficheros del proyecto se ha puesto a disposición de los proyectistas de la Comunidad Valenciana el **programa DECAUR**.

El programa DECAUR es una herramienta totalmente gratuita que es capaz de extraer las capas necesarias de uno o varios ficheros DWG, ficheros de AutoCAD 2000 ó 2002. Las capas extraídas son aquellas que contienen la información que necesita la GRCV y son copiadas a un nuevo fichero DWG. Utilizar esta herramienta aporta las siguientes ventajas:

- El proyectista trabaja como siempre lo ha hecho, no se modifica su forma de trabajo: nombres de ficheros, nombres de capas, contenidos, etc...
- Con el programa se seleccionan las capas útiles y se copian a otro fichero con nombres de capa estándar propuestos por la GRCV, nombres que no son modificables y que van incorporadas al programa (figura 4).
- El fichero resultante contiene muy poca información, en relación al fichero o ficheros originales del proyecto.
- No se ocasiona coste económico ni de pérdida de tiempo, al contrario, en unos minutos se tiene la información necesaria para el alta catastral.
- No hay pérdidas de precisión.
- Se puede entregar el fichero resultante a la Gerencia Regional en soporte informático.
- La interpretación del fichero resultante es sencilla.

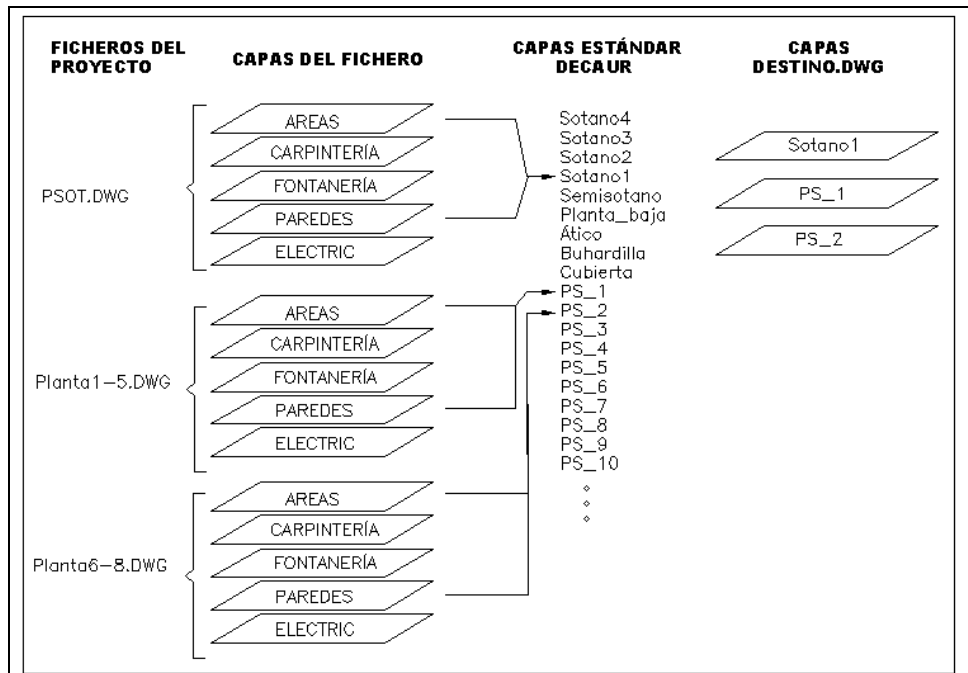


Figura 4: Copiado del contenido de las capas necesarias sobre capas estándar propuestas por la GRCV

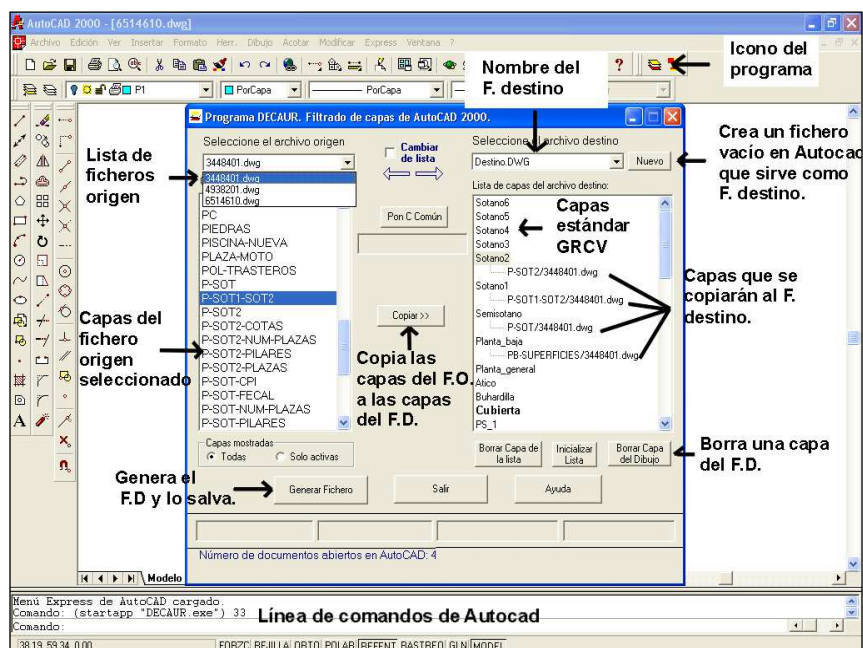


Figura 5: Interfaz del programa DECAUR

El **programa DECAUR** se puede obtener de la página web del Colegio Territorial de Arquitectos de Valencia<sup>1</sup>, entrando en el apartado de “declaraciones catastrales”:

<sup>1</sup> <http://www.ctav.es>

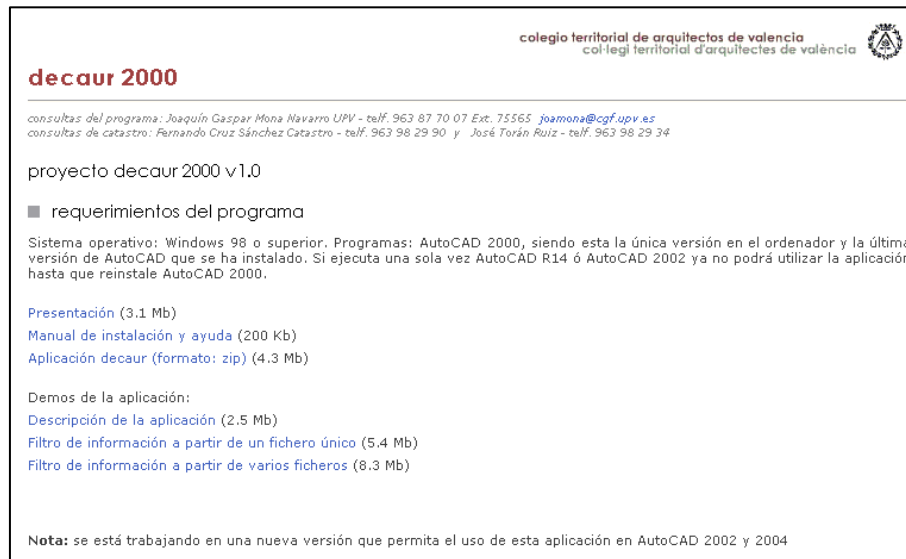


Figura 6: Programa DECAUR 2000 v1.0 en la página web del Colegio de Arquitectos

#### 4. Automatización cartográfica con Visual Basic 6 y Microstation 8.1.

En este apartado se detalla cómo realizar aplicaciones ejecutables externas a Microstation, utilizando Visual Basic 6 que realicen operaciones de automatización cartográfica. Para conseguir este objetivo son necesarias las siguientes condiciones:

- Tener instalado Microstation 8 o superior.
- Tener instalado Visual Basic 6.
- En el proyecto de la aplicación de automatización, dentro de Visual Basic 6, añadir el control *ActiveX* denominado *Bentley Microstation DGB 8.0 Object Library*.
- Mientras se esté interactuando con Microstation desde una aplicación de Visual Basic 6, un ejecutable externo, es necesario tener arrancado Microstation y, al menos, un documento abierto, es decir, un dibujo en pantalla.

El control *ActiveX* descrito se copia en el ordenador de forma automática al instalar Microstation. Este control contiene todos los *objetos* con los que se puede manejar Microstation, conocido también como *biblioteca de objetos o clases de Microstation*. Para añadir el control al proyecto de Visual Basic 6 se marca la biblioteca denominada *Bentley Microstation DGB 8.0 Object Library*, disponible en el cuadro de diálogo:

Proyecto → Referencias

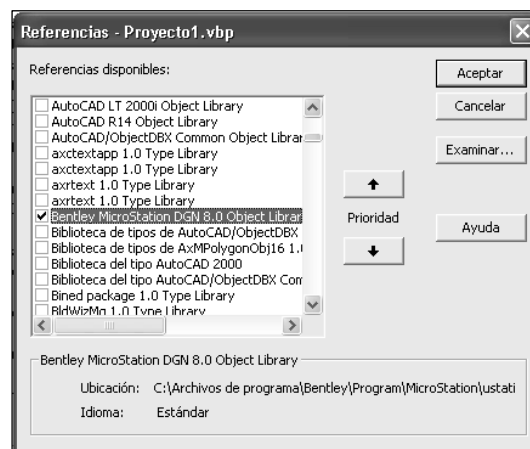


Figura 7:

Para ver los objetos que se pueden utilizar de la biblioteca de Microstation es necesario utilizar el *Examinador de Objetos* de Visual Basic. Para ello se selecciona la opción:

Ver → Examinador de objetos

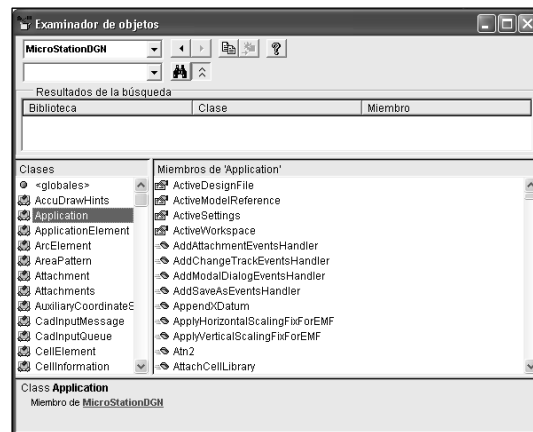


Figura 8:

En la figura anterior se observa que:

- A la izquierda aparecen todas las *clases* u *objetos* que tiene la biblioteca. De todas las clases se ha seleccionado la clase *Aplicacion*, que representa al programa Microstation. La ayuda dice que es un *miembro* de *MicrostationDGN*, es decir, la biblioteca seleccionada. Este es el objeto más importante, a través de él es posible interactuar con un dibujo de Microstation.
- A la derecha se presentan las *propiedades*, *métodos* y *eventos* que se pueden utilizar si se dispone de un objeto *Aplicacion*.
- Si se selecciona un método aparece en la parte de abajo una breve descripción de su sintaxis. Si se desea más información o ejemplos de código, pincha sobre el botón del interrogante (arriba a la derecha en el cuadro de diálogo anterior).

Con el mismo objetivo que cuando se describió la programación para AutoCAD, se implementa a continuación una subrutina que dibuja una línea en Microstation:

```
Public Sub EjemploDibujarLineaMS ()  
  'Este procedimiento añade una línea en el modelo activo  
  Dim pt1 As Point3d 'Punto inicial  
  Dim pt2 As Point3d 'Punto final  
  Dim oLinea As LineElement 'Objeto línea  
  pt1.X = 0 'Coordenadas del punto inicial  
  pt1.Y = 0  
  pt1.Z = 0  
  pt2.X = 200 'Coordenadas del punto final  
  pt2.Y = 200  
  pt2.Z = 0  
  Set oLinea = MS.CreateLineElement2(Nothing, pt1, pt2) 'Creación  
  del objeto línea  
  MS.ActiveModelReference.AddElement oLinea 'Se añade la línea al  
  modelo actual  
  oLinea.Redraw 'Se regenera la línea añadida  
End Sub
```

Figura 9: Subrutina en Visual Basic que permite dibujar una línea, de coordenadas (0,0,0) a (200,200,0)

En este caso los pasos son los siguientes: se crean el punto inicial y final de la línea, se crea la línea a partir de los puntos, y por último se añade la línea al modelo actual. Si se compara el código que realiza la misma tarea en AutoCAD, se puede observar que tiene características similares. En ambos casos se añade una línea al modelo abierto actual en el dibujo, aunque en Microstation se realizan más pasos previos.

## 5. Ejemplo de automatización topográfica con Visual Basic 6 y Microstation 8.1: Extracción de coordenadas de elementos de un dibujo para su replanteo

Para replantear los elementos de un proyecto en una obra son necesarias sus coordenadas, así como un croquis para saber cada coordenada a qué vértice del elemento pertenece. Con Visual Basic 6 y la metodología explicada es posible crear un programa externo a Microstation que realice esta tarea. El programa solicita un fichero de salida donde guardar las coordenadas que se van pinchando en Microstation. Cada vez que se pincha en la pantalla, el programa añade una nueva línea al fichero, donde aparece el número de punto, X, Y, Z y su código descriptivo.

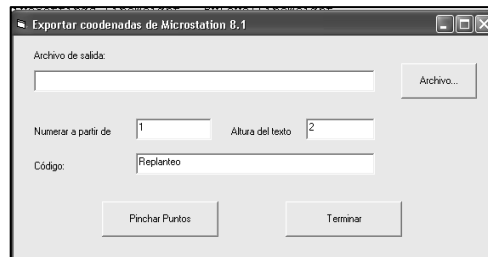


Figura 10: Cuadro de diálogo del programa

El mismo número de punto que se escribe en el fichero de salida se añade al dibujo, de forma que, si se imprime, se obtiene fácilmente un croquis de replanteo, tal como se aprecia en la figura siguiente:

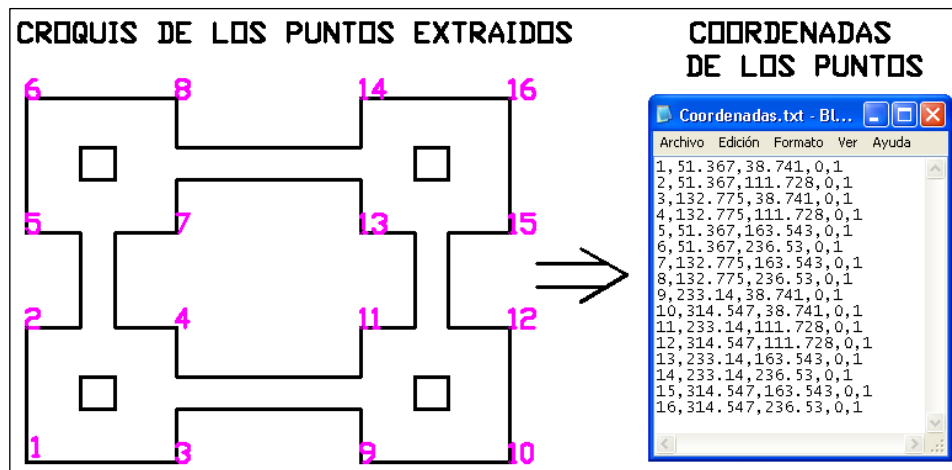


Figura 11: Extracción de coordenadas de un dibujo

Este programa se puede obtener de la página web del Departamento de Ingeniería Cartográfica, Geodesia y Fotogrametría de la Universidad Politécnica de Valencia<sup>2</sup>, entrando en el enlace de la unidad de Sistemas de Información Geográfica, en el apartado de “Aplicaciones”.

## 6. Ventajas e inconvenientes de utilizar Visual Basic 6 para programar en Autocad y en Microstation

### 6.1. Ventajas

- En muy poco tiempo de aprendizaje se puede dominar el lenguaje y los objetos.
- Posibilidad de realizar aplicaciones que resuelvan problemas complejos con poco esfuerzo.
- Con el mismo lenguaje de programación (Visual Basic) se pueden personalizar varios programas.

### 6.2. Inconvenientes:

- Lentitud en el manejo de los objetos de Autocad y Microstation: dibujando, copiando, moviendo. C es un lenguaje más rápido.

<sup>2</sup> <http://www.upv.es/cgf>