

# Guía detallada para la utilización de código de técnicas de ingeniería inversa en diagramas de UML con Microsoft Visio 2000

<http://www.microsoft.com/spanish/msdn/articulos/archivo/030801/voices/RevEngCode.asp>

<http://www.lawebdelprogramador.com>

Visimation, Inc.

Socio con licencia de Microsoft

Este artículo se aplica a:

Microsoft Visio 2000 Enterprise Edition

Microsoft Visio 2000 Professional Edition

Resumen: las ediciones Professional y Enterprise de Microsoft Visio 2000 incluyen la tecnología necesaria para la utilización de técnicas de ingeniería inversa desde código de Microsoft Visual C++, Microsoft Visual Basic y Microsoft Visual J++ a modelos de diagramas de clase de UML (Lenguaje unificado de modelado). En este artículo se exponen los pasos necesarios para la utilización de técnicas de ingeniería inversa en cada lenguaje. (9 páginas impresas.)

## Contenido

Introducción

Descripción general

Paso 1: Personalizar el entorno de desarrollo

Paso 2: Abrir el proyecto de código para utilizar las técnicas de ingeniería inversa

Paso 3: Seleccionar el botón Reverse Engineer en la barra de herramientas del complemento UML de Visio

Paso 4: Crear diagramas de UML propios en la solución de diagramas de modelo de Visio

## Introducción

Tanto la edición Enterprise como la edición Professional de Microsoft® Visio® 2000 incluyen soluciones para la utilización de código de técnicas de ingeniería inversa en diagramas de estructura estática de UML. Ambas soluciones admiten tres lenguajes: Microsoft® Visual C++® 6.0, Microsoft® Visual Basic® 6.0 y Microsoft® Visual J++® 6.0. La versión del estándar de UML empleada por la solución de diagramas de modelo de Visio es la 1.2.

La tecnología de ingeniería inversa empleada varía de un lenguaje a otro. En el caso de J++ y Visual Basic, la solución utiliza el modelo de objetos del entorno de desarrollo para recopilar la información de la estructura estática. Por el contrario, en el caso de C++, la solución emplea el archivo de información de exploración (.BSC) como el origen de la información de la estructura. Aunque las técnicas de ingeniería inversa de Visual Basic y J++ utilizan el modelo de objetos del entorno de desarrollo, el modelo de objetos de J++ expone más información y proporciona un modelo de UML de técnicas de ingeniería inversa más completo. El archivo de información de exploración en C++ es el que contiene menos información de los tres; por tanto, los modelos de UML resultantes no serán tan completos como los que se obtienen de las técnicas de ingeniería inversa de J++ y Visual Basic.

Este artículo presenta los pasos necesarios para utilizar técnicas de ingeniería inversa en la solución de diagramas de modelo de Visio.

**Nota** La solución de diagramas de modelo de Visio admite la utilización de técnicas de ingeniería inversa exclusivamente en el caso de Microsoft Visual C++ 6.0, Microsoft Visual J++ 6.0 y Microsoft Visual Basic 6.0. Será preciso instalar Service Pack 3.0 para Microsoft® Visual Studio® para poder utilizar el complemento UML de Visio.

## Descripción general

Los pasos necesarios para la utilización de código de técnicas de ingeniería inversa en la solución de diagramas de modelo de Visio son los siguientes:

1. Personalizar el entorno de desarrollo de Visual Studio
2. Abrir el proyecto para utilizar técnicas de ingeniería inversa en Visual Studio
3. Seleccionar el botón Reverse Engineer en la barra de herramientas del complemento UML de Visio de Visio
4. Crear diagramas de clase de UML en la solución de diagramas de modelo de Visio

El resto del artículo facilita instrucciones detalladas de cada uno de los pasos, incluyendo información específica para cada lenguaje de programación. Por ejemplo, se incluye una sección sobre personalización del entorno de desarrollo en cada uno de los lenguajes: Visual C++, Visual Basic y Visual J++.

### Paso 1: Personalizar el entorno de desarrollo

Para poder utilizar técnicas de ingeniería inversa el primer paso es personalizar el entorno de desarrollo agregando la barra de herramientas del complemento UML de Visio. Las instrucciones siguientes muestran cómo se puede agregar esta barra de herramientas a los distintos entornos de Visual C++, Visual Basic y Visual J++.

#### Para personalizar Microsoft Visual C++ 6.0

#### Para personalizar Microsoft Visual C++ 6.0

1. En Visual C++, seleccione **Customize** en el menú **Tools**.
2. En el cuadro de diálogo **Customize**, seleccione la ficha **Add-ins and Macro Files**.
3. En la lista **Add-ins and macro files**, seleccione **Visio UML Add-in**. Asegúrese de activar la casilla de verificación correspondiente a esta opción y haga clic en **Close**.
4. Aparecerá la barra flotante de cinco botones del complemento UML de Visio.



**Sugerencia** Si desea que la barra de herramientas se encuentre disponible cada vez que abra Microsoft Visual Studio, acople la barra arrastrándola al área de barras de herramientas.

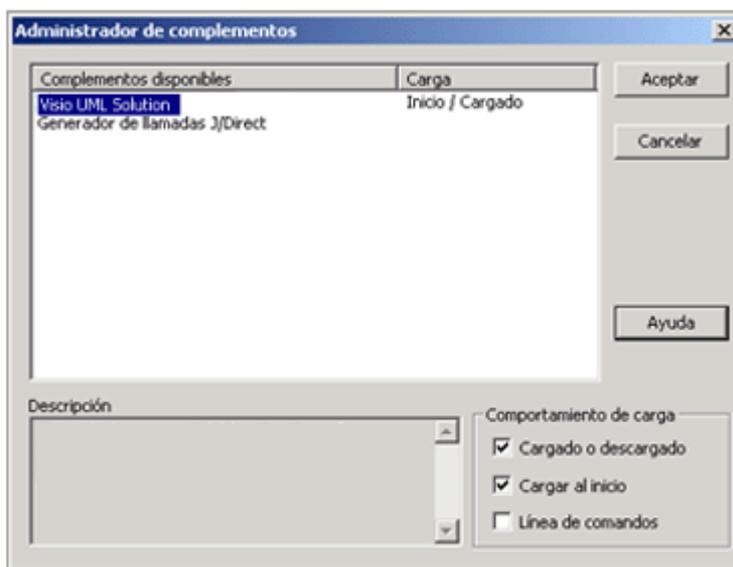
#### Para personalizar Microsoft Visual Basic 6.0

1. En Visual Basic, seleccione **Administrador de complementos** en el menú **Complementos**.
2. En el cuadro de diálogo **Administrador de complementos**, seleccione **Visio UML Add-in** en la lista Complementos disponibles. En Comportamiento de carga, active las casillas **Cargado/Descargado** y **Cargar al iniciar** y, a continuación, haga clic en **Aceptar**.
3. Aparecerá la barra flotante de cuatro botones del complemento UML de Visio.

**Sugerencia** Cuando el complemento UML de Visio crea la barra de herramientas, también agrega comandos al menú Tools. Podrá localizarlos seleccionando **Tools** en **Visio UML Solution**.

### Para personalizar Microsoft Visual J++ 6.0

1. En Visual J++, seleccione **Administrador de complementos** en el menú **Herramientas**.
2. En el cuadro de diálogo **Administrador de complementos**, seleccione **Visio UML Solution** en la lista Complementos disponibles. En Comportamiento de carga, active las casillas **Cargado o descargado** y **Carga al inicio** y, a continuación, haga clic en **Aceptar**.



3. Aparecerá la barra flotante de cuatro botones del complemento UML de Visio.



**Sugerencia** Cuando el complemento UML de Visio crea la barra de herramientas, también agrega comandos al menú **Herramientas**. Podrá localizarlos seleccionando **Visio UML Solution** en el menú **Herramientas**.

### Paso 2: Abrir el proyecto de código para utilizar las técnicas de ingeniería inversa

El proceso de utilización de técnicas de ingeniería inversa se inicia desde el entorno de desarrollo de cada uno de los lenguajes. Una vez se ha personalizado el entorno agregando la barra de herramientas del complemento UML de Visio, el paso siguiente consiste en abrir el proyecto en el que se desean utilizar las técnicas de ingeniería inversa.

### Pasos adicionales para Visual C++ 6.0

En el caso de Visual C++ existen varios pasos adicionales necesarios para utilizar las técnicas de

ingeniería inversa. El primero consiste en generar un archivo de información de exploración (.BSC) para el proyecto. Las instrucciones para su generación se incluyen más abajo. El segundo consiste en determinar cuáles son las bibliotecas de código que se deben incluir en las técnicas de ingeniería inversa. Las instrucciones para configurar estas opciones aparecen después de las instrucciones relativas al archivo de exploración. Por último, será preciso generar el proyecto antes de utilizar técnicas de ingeniería inversa.

### **Para generar un archivo de información de exploración para Visual C++ 6.0**

1. En Visual C++, seleccione **Settings** en el menú **Project**.
2. En el cuadro de diálogo **Project Settings**, seleccione el tipo de configuración de generación deseado y haga clic en la ficha **C/C++**; a continuación, active la casilla de verificación **Generate browse info**.
3. Haga clic en la ficha **Browse Info** para especificar el nombre y la ubicación del archivo de información de exploración, active la casilla de verificación **Build browse info file** y haga clic en **OK**.

### **Para seleccionar las bibliotecas que se deben incluir con las técnicas de ingeniería inversa de Visual C++ 6.0**

1. Haga clic en el botón **Visio UML Options** en la barra de herramientas del complemento UML para establecer las bibliotecas que se van a incluir con las técnicas de ingeniería inversa de Visual C++.
2. Aparecerá la pantalla **Visio UML Options**. Si desea que en las técnicas de ingeniería inversas se incluya información sobre **MFC Classes**, **ATL Classes** o **Standard Include Files**, simplemente active la casilla de verificación de la opción deseada y haga clic en **OK**.

### **Paso 3: Seleccionar el botón Reverse Engineer en la barra de herramientas del complemento UML de Visio**

El paso siguiente para la utilización de técnicas de ingeniería inversa consiste en seleccionar el botón **Reverse Engineer** en la barra de herramientas del complemento UML de Visio.

En Visual Basic y Visual J++ esta operación abrirá Visio 2000 en la solución de diagramas de modelo UML con la información de técnicas de ingeniería inversa en el modelo. La solución de diagramas de modelo UML se abrirá con un diagrama de estructura estática y una vista de árbol en **UML Navigator**, explorador en el que se incluirá la información sobre el modelo recopilada en el proyecto.

En el caso de Visual C++, puede que aparezcan dos pantallas adicionales antes de que se abra Visio 2000, mostrando la información de las técnicas de ingeniería inversa del proyecto en la solución de diagramas de modelo UML. Si existiera más de un proyecto en el espacio de trabajo de Visual C++, seleccione el proyecto en el que desea aplicar las técnicas de ingeniería inversa en el cuadro de diálogo **Select Project** y, a continuación, haga clic en **OK**.

Si la jerarquía del proyecto mostrara más de un archivo de exploración, seleccione el archivo deseado en el cuadro de diálogo **Select Browse File** y haga clic en **OK**.

#### **Paso 4: Crear diagramas de UML propios en la solución de diagramas de modelo de Visio**

Ahora la información del proyecto debe aparecer en Visio 2000. Para poder crear los diagramas de clase de UML, arrastre los elementos del modelo de **UML Navigator** y colóquelos en el diagrama **Static Structure**. Las relaciones de clases (generalización, realización, enlace) se mostrarán automáticamente. Con la solución de diagramas de modelo de Visio podrá crear artefactos de UML adicionales para especificar, construir, visualizar y documentar su proyecto de software.

