

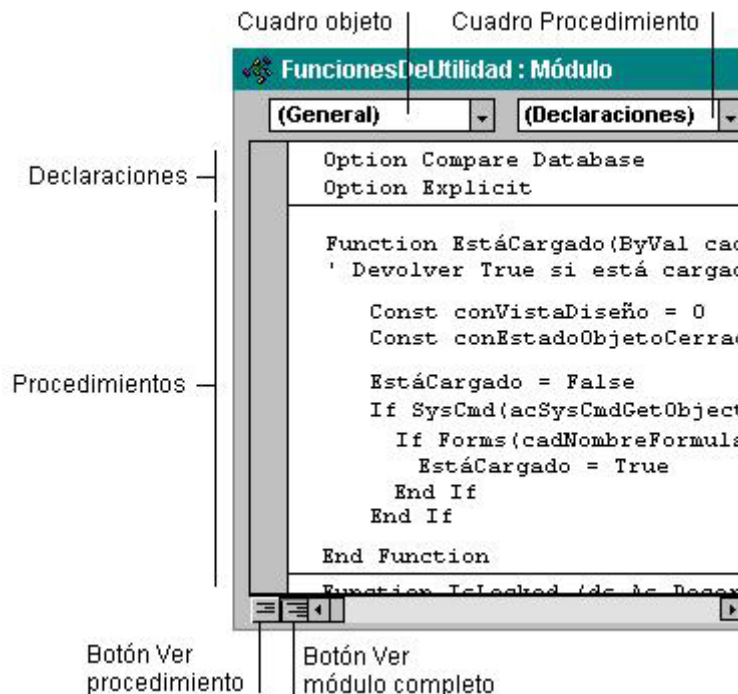
Introducción a los Módulos

Access nos permite realizar una programación, basada en el lenguaje VBA Visual Basic para Aplicaciones y esta programación se realiza a través de los denominados módulos.

Los módulos se asocian directamente a objetos tales como formularios o informes y estos se crean y se asignan a los eventos más comunes de los objetos mencionados. También se pueden crear módulos independientes, que estarán visibles en la ventana de la base de datos en el apartado de Módulos.

Qué es un módulo

Un módulo es una colección de declaraciones y procedimientos de Visual Basic para aplicaciones que se almacenan juntos como una unidad.

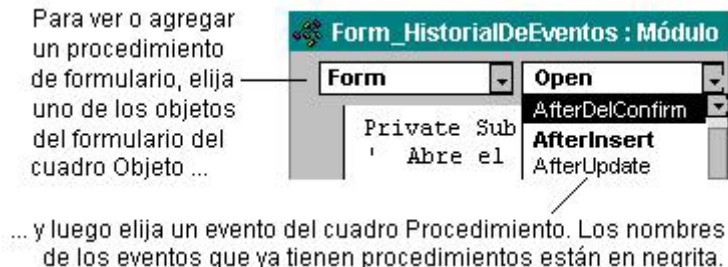


Hay dos tipos básicos de módulos: los módulos de clases y los módulos estándar. Cada procedimiento en un módulo puede ser un procedimiento **Function** o un procedimiento **Sub**.

Módulos de clases

Los módulos de **formularios** y los módulos de **informes** son módulos de clases que están asociados con un formulario o informe determinado. Los módulos de formularios y de informes contienen a menudo **procedimientos de evento** que se ejecutan en respuesta a un evento en un formulario o informe. Puedes usar los procedimientos de evento para controlar el comportamiento de los formularios e informes, y sus respuestas a acciones de los usuarios como por ejemplo hacer clic con el **mouse** en un botón de comando.

Cuando se crea el primer procedimiento de evento para un formulario o informe, Access crea automáticamente un módulo de formulario o un módulo de informe. Para ver el módulo para un formulario o informe, haz clic en **Código** en la barra de herramientas en la vista Diseño de informe o de formulario.

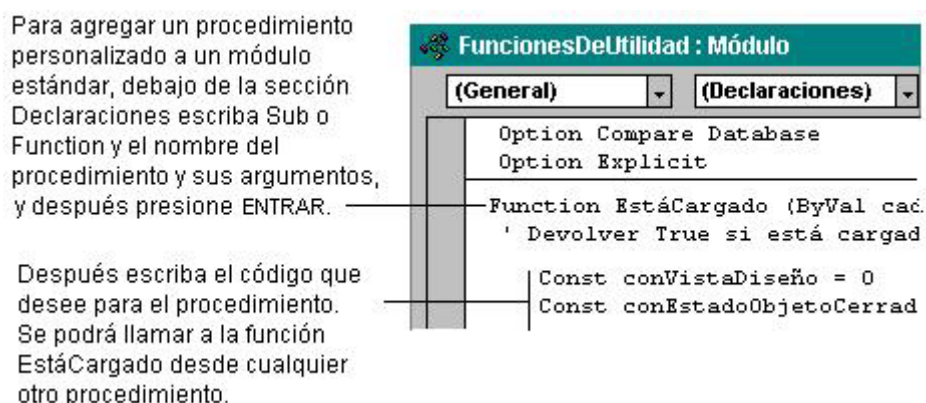


Los procedimientos en los módulos de formularios y de informes pueden llamar a procedimientos que haya agregado a los módulos estándar.

En Access 95, los módulos de clase existen únicamente en asociación con un formulario o informe. En Access 97 o versiones posteriores, los módulos de clase pueden existir también con independencia de un formulario o informe, y este tipo de módulo de clase se enumera en la opción **Módulos**, situada bajo la sección **Objetos** de la ventana Base de datos. Puede usar un módulo de clase de la opción **Módulos** para crear una definición para un objeto personalizado. Para averiguar más acerca de la programación con estos módulos de clase,

Módulos estándar

Los módulos estándar contienen procedimientos generales que no están asociados con ningún otro objeto y procedimientos usados con frecuencia que pueden ser ejecutados desde cualquier parte de la base de datos.



La lista de módulos estándar de la base de datos se puede ver haciendo clic en la opción **Módulos**, situada bajo la sección **Objetos** de la ventana Base de datos. Los formularios, informes y módulos estándar se enumeran también en el Examinador de objetos.

Qué es un procedimiento

Un procedimiento es una unidad de código de Visual Basic. Un procedimiento contiene una serie de instrucciones y métodos que realizan una operación o calculan un valor. Por ejemplo, el siguiente procedimiento de evento usa el método **OpenForm** para abrir el formulario Pedidos:

```
Private Sub AbrirPedidos_Click()
```

```
    DoCmd.OpenForm "Pedidos"
```

```
End Sub
```

Hay dos tipos de procedimientos:

- Los procedimientos **Sub** realizan una operación o una serie de operaciones pero no devuelven un valor. Puede crear sus propios procedimientos **Sub** o usar las plantillas de procedimientos de evento que Microsoft Access crea por usted.

Cada formulario o informe de la base de datos tiene un módulo de formulario o un módulo de informe incorporado que contiene plantillas de procedimientos de evento. Puede agregar código que se ejecute en respuesta a los eventos que se produzcan en el formulario, en el informe o en los controles del formulario o informe. Cuando Microsoft Access reconoce que se ha producido un evento en un formulario, informe o control, ejecuta de forma automática el procedimiento de evento nombrado para el objeto o evento. Por ejemplo, puede escribir un procedimiento de evento que establece el enfoque a un cuadro de texto Nombre esposo cuando el usuario selecciona la casilla de verificación Casado.

Sugerencia Muchos de los asistentes (por ejemplo, el Asistente para botones de comando) que crean objetos crean también procedimientos de evento para el objeto. Puede mirar estos procedimientos de evento para ver ejemplos de cómo funcionan estos procedimientos.

- Los procedimientos **Function** (llamados a menudo simplemente funciones) devuelven un valor, tal como el resultado de un cálculo. Microsoft Visual Basic incluye muchas funciones incorporadas; por ejemplo, la función **Now** devuelve la fecha y hora actual. Además de estas funciones incorporadas, puede crear sus propias funciones personalizadas.

Puesto que las funciones devuelven valores, puede usarlas en expresiones. Puede usar funciones en expresiones en muchos lugares de Microsoft Access, incluyendo en una instrucción o método de Visual Basic, en muchos valores de propiedades o en una expresión de criterios en un filtro o consulta.

Éste es un ejemplo de un procedimiento **Function**, `PrimerDíaDelMesSiguiente`, que devuelve la fecha del primer día del mes siguiente a la fecha actual:

Function PrimerDíaDelMesSiguiente()

```
PrimerDíaDelMesSiguiente = _  
    DateSerial(Year(Now), Month(Now) + 1, 1)
```

End Function

Esta función personalizada consiste en una única instrucción de asignación que asigna el resultado de una expresión (en el lado derecho del signo igual [=]) al nombre de la función, PrimerDíaDelMesSiguiente (en el lado izquierdo del signo igual). La función calcula el resultado usando las funciones incorporadas de Visual Basic **DateSerial**, **Year**, **Now** y **Month**.

Después de que haya creado esta función, puede usarla en una expresión prácticamente en cualquier parte de Microsoft Access. Por ejemplo, podría especificar que un cuadro de texto muestre el primer día del mes que sigue a la fecha actual como el valor predeterminado estableciendo la propiedad **ValorPredeterminado (DefaultValue)** del control del cuadro de texto a la siguiente expresión en la hoja de propiedades:

```
=PrimerDíaDelMesSiguiente()
```

Nota Para usar una función como valor de una propiedad, la función debe estar en el módulo del formulario o del informe, o en un módulo estándar. No se puede usar una función en un módulo de clase que no esté asociada con un formulario o informe como valor de propiedad de formulario o informe.

Los procedimientos **Function** y **Sub** pueden aceptar argumentos.

¿Qué es un argumento?

Una constante, variable, o expresión que proporciona información a una acción, evento, propiedad o procedimiento.

Un argumento ofrece información adicional a un procedimiento, instrucción o método. Cuando se crea un procedimiento **Function** o procedimiento **Sub**, se declaran argumentos si desea poder suministrar información al procedimiento cuando se le llama.

Cuando se llama a un procedimiento que tenga argumentos, se especifican los argumentos que se desea pasarle. También se pueden especificar argumentos para muchas instrucciones y métodos.

Dada la amplitud de posibilidades que nos ofrece este lenguaje, podemos decir que las líneas escritas no son más que una breve introducción a la programación de VBA y que éste podría ser motivo indiscutible del desarrollo de un curso específico de esta materia.