



## Inicio

**FoxBarcode** y **FoxBarcodeQR** son parte de **VFPx**, un proyecto de la Comunidad Mundial de Visual FoxPro para crear complementos de código abierto para Visual FoxPro 9.0.

Las páginas principales de ambos proyectos **VFPx** son:

- **FoxBarcode**: <https://github.com/VFPX/FoxBarcode>
- **FoxBarcodeQR**: <https://github.com/VFPX/FoxBarcodeQR>



*En memoria de Guillermo Carrero (05/02/1961 - 14/01/2013)*

## FoxBarcode

**FoxBarcode** es una clase 100% **Visual FoxPro** que ofrece una herramienta para la generación de imágenes con distintas simbologías de códigos de barras, para ser usadas en informes y formularios de **VFP**, o exportadas a otras aplicaciones. Su uso y distribución es libre para toda la *Comunidad de Visual FoxPro*.

**Última versión FoxBarcode:** 1.19 lanzada el 21/05/2019 [Aquí](#)

### Características

**FoxBarcode** soporta las siguientes 18 simbologías lineales de códigos de barras: Código 128, Código 39, Código 39 Extendido (Full ASCII), Código 93, Código 93 Extendido (Full ASCII), Standard 2 de 5, Interleaved 2 de 5, EAN-8, EAN-13, UPC-A, UPC-E (EAN y UPC con suplementos de 2 y 5 dígitos), ITF-14, GS1/EAN/UCC-128, Codabar, Código 11, MSI / Plessey, Telepen, PostNet, RM4SCC, KIX-Code, One Track Pharmacode y Two Track Pharmacode. Ahora con el complemento **FoxBarcodeQR** se soporta el Código QR.

**FoxBarcode** soporta los siguientes tipos de imágenes: JPG, BMP, GIF, PNG y TIFF

**FoxBarcode** genera las imágenes con la clase **gplImage2** (<http://sites.google.com/site/gpimage2>) de Alexander Golovlev (Rusia) y Cesar Ch. (Brasil), basada en GDI+ y requiere gdiplus.dll.

**FoxBarcode** es compatible con todas las versiones de **Visual FoxPro** a partir de la versión 6.0

**FoxBarcode** permite un gran control en la generación de la imagen del código de barras al poder personalizar las siguientes PEMs:

## Propiedades

- **nImageHeight:** Altura en pixeles de la imagen.
- **nImageWidth:** Ancho en pixeles de la imagen. Ahora se puede configurar el ancho
- **nAlignBarcode:** Alineación del código de barra con respecto a la imagen. [0=Izquierda, 1=Centro y 2=Derecha]
- **nResolution:** Resolución de la imagen en DPI.
- **nRotation:** Rotación de la imagen. [0=0°, 1=90°, 2=180°, 3=270°]
- **cImageType:** Tipo de la imagen generada ["JPG", "GIF", "PNG", "BMP", "TIF"]
- **cImageFile:** Nombre del archivo de imagen generado. Si no se especifica, se genera un nombre de archivo aleatorio en la carpeta de archivos temporales de Windows.
- **cText:** Texto a codificar.
- **nAlignText:** Alineación del texto de la lectura humana. [0=Izquierda, 1=Centro, 2=Derecha]
- **IShowHumanReadableText:** .T. si se muestra la lectura humana.
- **IShowCheckDigit:** .T. si se muestra el dígito de control en la lectura humana.
- **IShowStartStopChars:** .T. si se muestra los caracteres de inicio y final en la lectura humana.
- **cFontName:** Nombre de la fuente de la lectura humana.
- **IFontBold:** .T. si la fuente de la lectura humana es Negrita.
- **IFontItalic:** .T. si la fuente de la lectura humana es Cursiva.
- **nFontSize:** Tamaño de la letra de la lectura humana. [Recomendado = 8 ó 9]
- **nFontColor:** Color de la fuente de la lectura humana. [Recomendado = Negro = RGB(0,0,0)].
- **nBackColor:** Color de fondo de la imagen [Recomendado = Blanco = RGB(255,255,255)].
- **nBarsColor:** Color de las barras [Recomendado = Negro = RGB(0,0,0)].
- **nBarcodeType:** Tipo de la simbología de código de barra. Vea la lista de códigos de barra soportados y su Id.
- **cSet128 Set:** Juego del Código 128. ["A", "B", "C" y "\*" (Auto)]
- **cSupplementalText:** Texto suplementario en códigos EAN y UPC.
- **IAddCheckDigit:** .T. si se calcula el dígito de control. En las simbologías en donde el dígito de control es obligatorio, no se tiene en cuenta esta propiedad.
- **nBearerBar:** Tipo del marco portante, solo en la simbología ITF-14. [0=Ninguno, 1=Rectángulo, 2=Superior e inferior]
- **IUseAppld:** .T. para que FoxBarcode interprete en el código GS1/EAN/UCC-128 a los Identificadores de Aplicación (encerrados entre paréntesis).
- **nFactor:** Factor de magnificación del código de barras. [1..9]
- **nMargin:** Margen alrededor del código de barras.
- **nRatio:** Relación entre las barras angostas y las barras anchas en las simbologías que lo permiten.
- **IDeleteTempFiles:** Permite elegir si borra o no la carpeta temporal con las imágenes temporales.

## Métodos

Para la generación de la imagen solamente se deberá invocar que genera la imagen del código de barra y retorna la ruta y nombre del archivo generado:

- **BarcodeImage(cTextToEncode, cFileNameImage, cPropertyList):** Codifica el texto y retorna la ruta y nombre de la imagen con el código de barra generado.

Todos los parámetros son opcionales.

- **cTextToEncode:** Texto a codificar.
- **cFileNameImage:** Nombre del archivo de imagen generado. Si no se especifica, se genera un nombre de archivo aleatorio en la carpeta de archivos temporales de Windows.
- **cPropertyList:** Éste tercer parámetro permite configurar todas las propiedades en una sola línea, separada por comas.

Actualmente con el método **BarcodeTest()** se puede validar el texto a codificar sin la necesidad de generar la imagen con el código de barras:

- **BarcodeTest(cTextToEncode, cFileNameImage, cPropertyList):** Valida la cadena de texto a codificar sin la generación de la imagen.

Todos los parámetros son opcionales y son los mismos que los del método BarcodeImage().

- **cTextToEncode:** Texto a codificar.
- **cFileNameImage:** Nombre del archivo de imagen generado. Si no se especifica, se genera un nombre de archivo aleatorio en la carpeta de archivos temporales de Windows.
- **cPropertyList:** Éste tercer parámetro permite configurar todas las propiedades en una sola línea, separada por comas.

## Lenguajes

Gracias a la Comunidad Mundial de VFP, se tradujeron los mensajes de **FoxBarcode** a los siguientes lenguajes:

- Inglés por VFPEncoding
- Español por VFPEncoding
- Checo por Martin Krivka
- Holandés por Koen Piller
- Alemán por Stefan Wuebbe
- Indonesio por Samir H.
- Portugués por Cesar Ch.
- Turco por Ugur Yilmaz
- Filipino por Glenn Gevero
- Francés por Samir H.
- Serbio por Michael Kopljan
- Italiano por Roberto Saccomanno
- Rumano por Vilhelm-Ion Praisach

## Ejemplos

El siguiente es un ejemplo de código para generar una imagen tipo "PNG" del código de barras "Código 128 C" con una altura de "100" pixeles y un factor de magnificación de "2":

```
*-- Crear el objeto
loFbc = CREATEOBJECT("FoxBarcode")
*-- Configurar las propiedades
WITH loFbc
.cImageType = "PNG"
.nBarcodeType = 110 && Code 128
.cSet128 = "C" && Set 128 C
.nImageHeight = 100
.nFactor = 2
ENDWITH
*-- Generar la imagen
lcImagen = loFbc.BarcodeImage("123456789012")
```

Desde la versión 0.11 se agregó un 3er. parámetro al método BarcodeImage() que permite configurar varias propiedades, separadas por comas. El mismo ejemplo de arriba, con el uso del 3er. parámetro.

```
*-- Creo el objeto
loFbc = CREATEOBJECT("FoxBarcode")
*-- Genero la imagen con sus propiedades
lcImagen = loFbc.BarcodeImage("", "", [cText="123456789012", cImageType="PNG", ;
nBarcodeType=110, cSet128="C", nImageHeight=100, nFactor=2])
```

la imagen generada se muestra a continuación



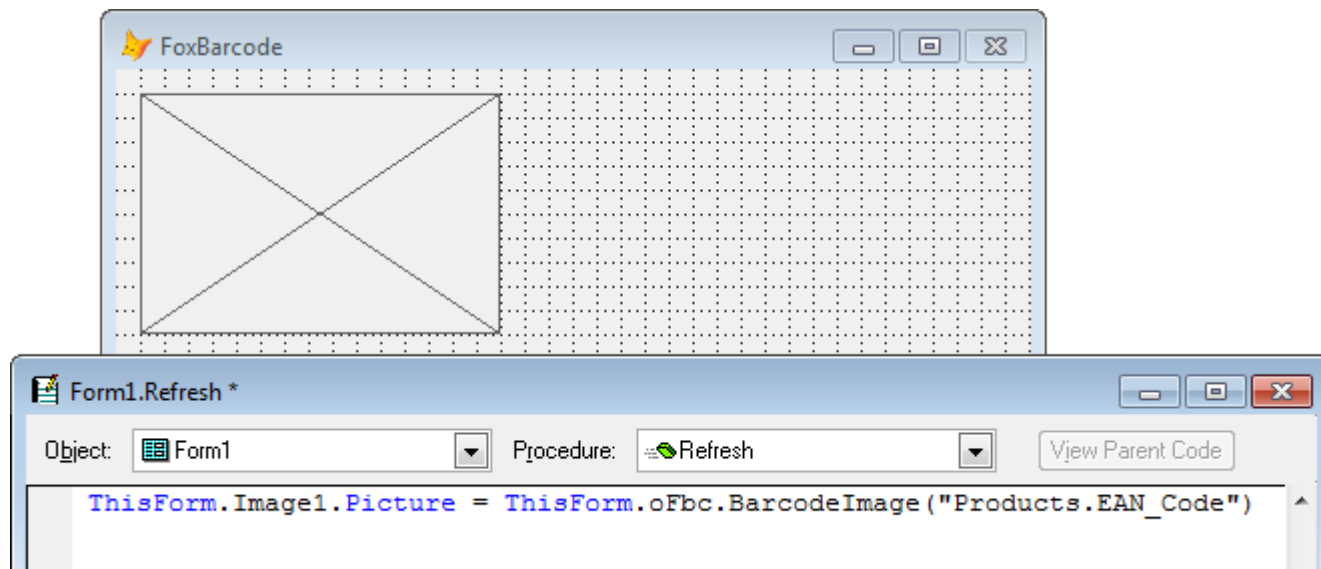
Otras imágenes de ejemplo cambiando solo algunas propiedades de la clase **FoxBarcode**



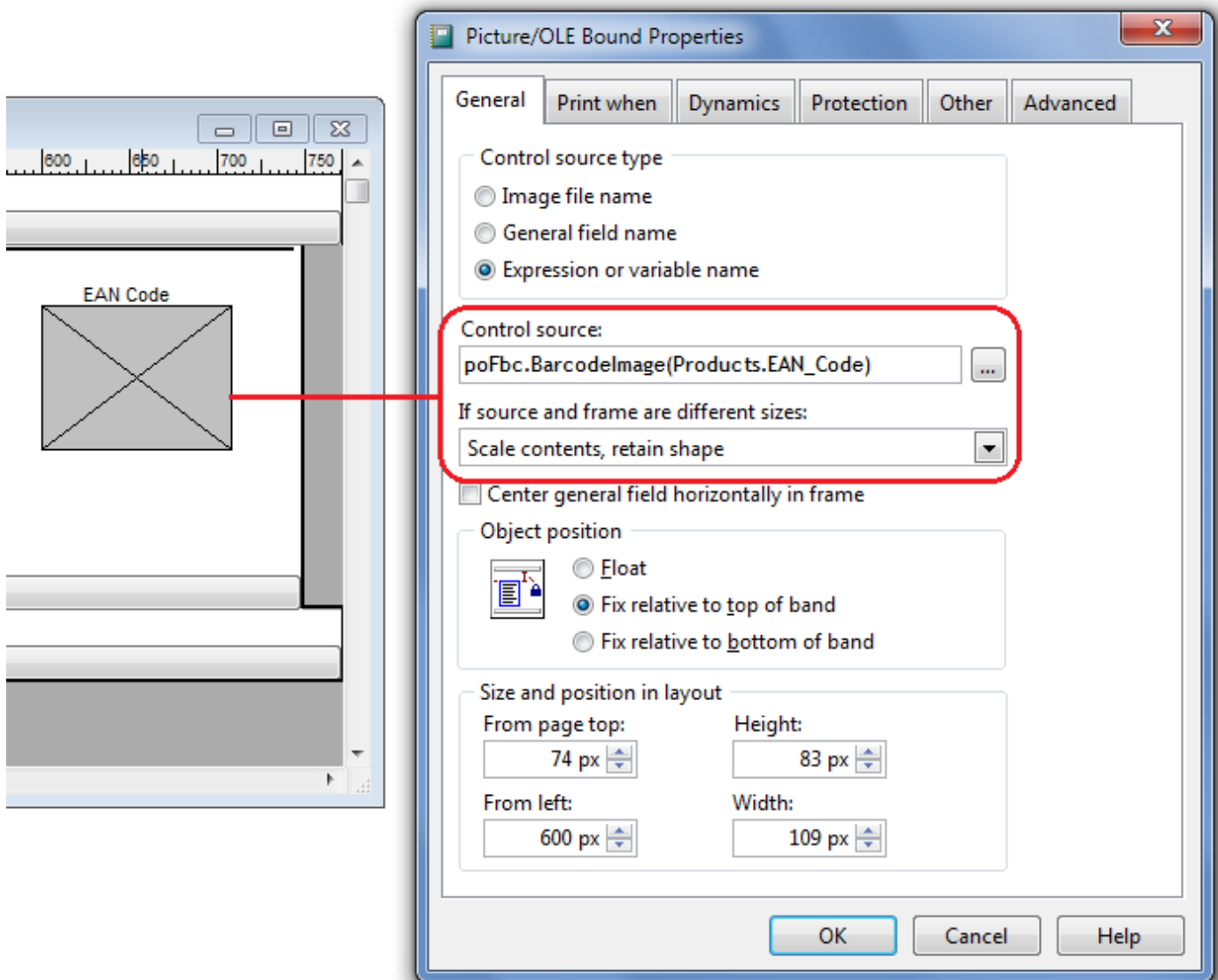
Ahora **FoxBarcode** soporta el código EAN/UCC/GS1 128. La versión 1.19 mejoró la codificación de los IAs (Identificadores de Aplicación) con los FNC1 en campos de longitud fija.



Para incluir un código de barras en un formulario, se debe insertar un objeto Image. Como la imagen del código de barras no existe en tiempo de diseño, la propiedad Picture tomará el nombre de la imagen cuando se llame al método BarcodeImage(), por ejemplo desde el método Refresh de Formulario, como se muestra en la siguiente figura:



Para incluir un código de barras en un Informe, se debe insertar un objeto Image y configurar la propiedad ControlSource con un llamado al método BarcodeImage() y se recomienda configurar "Scale contents, retain shape" si la imagen difiere de tamaño con el cuadro.



**NOTA:** Antes de ejecutar el informe y crear el objeto **FoxBarcode**, se debe declarar la variable como PRIVATE para que ésta tenga alcance en el informe, como se muestra a continuación:

```
PRIVATE poFbc
poFbc = CREATEOBJECT("FoxBarcode")
...
REPORT FORM MyReport
```

### **Distribución**

Los únicos archivos necesarios que se deben distribuir para que **FoxBarcode** funcione correctamente son:

- FoxBarcode.prg
- gpImage.prg

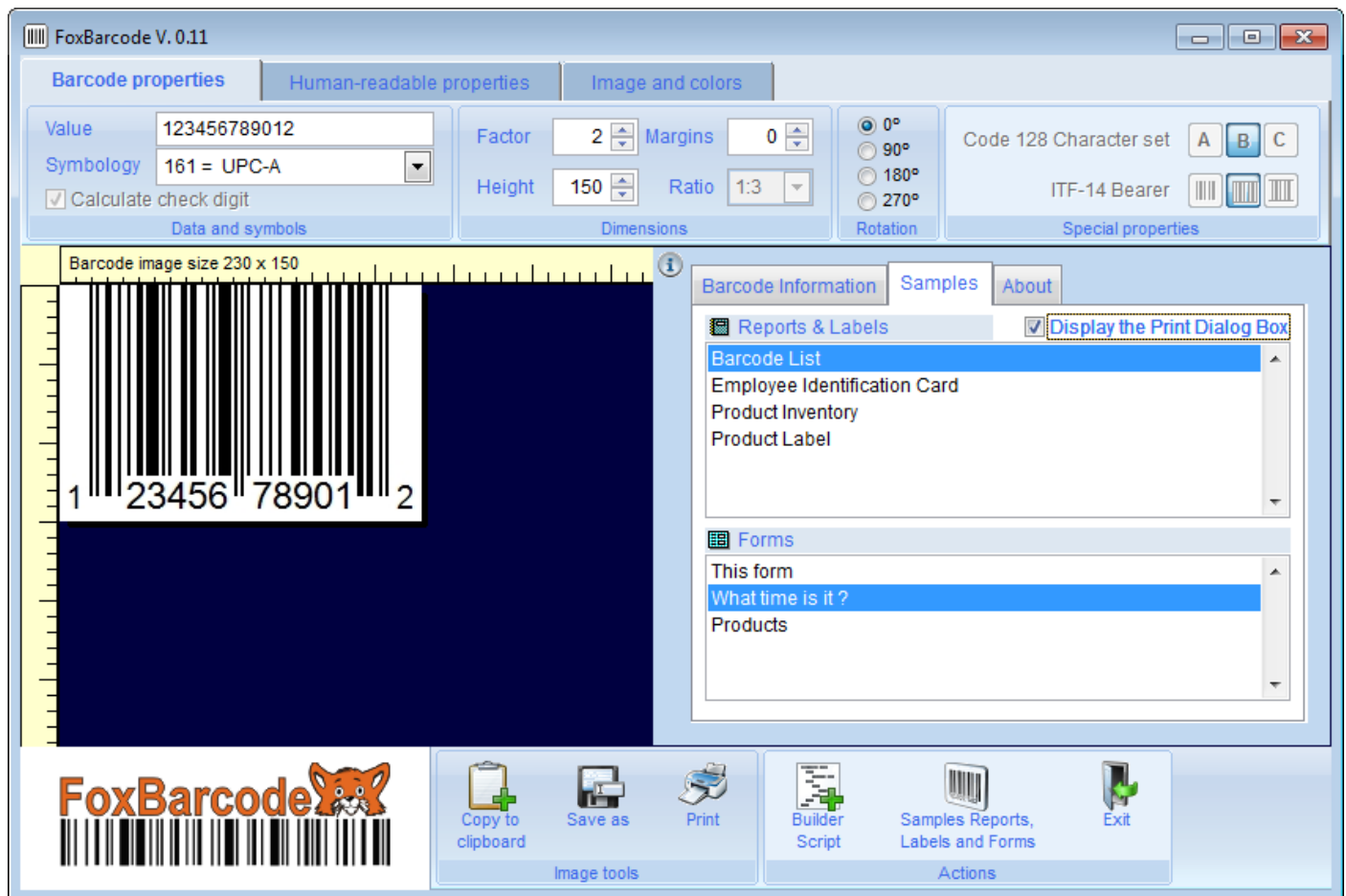
Estos archivos se encuentran en la carpeta \Source incluida en el archivo de la descarga.

Antes de instanciar la clase **FoxBarcode**, se deben ejecutar las siguientes sentencias:

```
SET PROCEDURE TO LOCFILE("FoxBarcode.prg"), LOCFILE("gpImage2.prg") ADDITIVE
```

## Demo

En los archivos de la descarga, está incluido un proyecto completo de una aplicación Demo que utiliza **FoxBarcode**. La siguiente es una captura de pantalla de la aplicación.



La aplicación permite:

- Configurar todas las propiedades de la clase **FoxBarcode** y mostrar la imagen generada automáticamente.
- Copiar al portapapeles la imagen generada.
- Guardar en un archivo la imagen.
- Imprimir el código de barras.
- Ejecutar diversos informes y formularios de ejemplo.
- Construir un Script con el código en **Visual FoxPro**.

En la aplicación también se muestra una breve descripción de cada simbología de códigos de barras, según se seleccione en la lista desplegable de las simbologías.

Puede leer mas sobre este demo en: [Generando códigos de barras con FoxBarcode](#)

## Descargas

Para descargar la última versión de **FoxBarcode**, visite la página de [Descargas](#) haciendo [clic aquí](#).

## Feedback

**FoxBarcode** se encuentra abierto a cualquier comentario que será bienvenido. Los comentarios, como así también los reportes de incidencias usando la clase **FoxBarcode** se deben enviar por correo electrónico a [vfpenencoding@gmail.com](mailto:vfpenencoding@gmail.com)

## ¿Quiénes somos?

**VFPEncoding** es un grupo de desarrolladores pertenecientes a la *Comunidad Hispana de Visual FoxPro* que creo **FoxBarcode**. Los miembros de VFPEncoding son:

- **Guillermo Carrero [QEPD]** (Barcelona, España)
- **Luis María Guayán** (Tucumán, Argentina)

## Agradecimientos

- A nuestras familias.
- A **Cesar Ch.** por sus continuas ideas y colaboraciones con sus clases **gplImage2** y **FoxyPreviewer**.

## ¿Qué hay de nuevo en FoxBarcode?

### v.1.19 - Liberación 2019.05.21

- Mejoras en la codificación del código GS1-128 con el uso de los Als (Identificadores de Aplicación) de longitud fija.

### v.1.17 - Liberación 2016.12.21

- Nuevo Lenguaje RUMANO. Traducido por Vilhelm-Ion Praisach

### v.1.10 - Liberación 2012.06.03

- Nuevo lenguaje: ITALIANO por Roberto Saccomanno.
- Cambio en la propiedad nImageWidth: Ahora se permite configurar el ancho de la imagen.
- Nueva propiedad nAlignBarcode: Permite alinear el código de barras a la izquierda, centro o derecha de la imagen.
- Se han solucionado algunos errores menores.

### v.1.00 - Liberación 2011.12.26

- Lanzamiento para producción.
- Verificación y corrección de todas las variables para soporte mDot.
- Se han solucionado algunos errores menores.

### v.0.16 - Liberación 2011.08.28

- Mensajes en múltiple lenguajes:
  - INGLES
  - ESPAÑOL por VFPEncoding
  - CHECO por Martin Krivka
  - HOLANDÉS por Koen Piller
  - ALEMÁN por Stefan Wuebbe
  - INDONESIA por Samir H.
  - PORTUGUÉS por Cesar Chalom
  - TURCO por Ugur Yilmaz



- FILIPINO por Glenn Gevero
- FRANCÉS por Samir H.
- SERBIO por Michael Kopljan
- Nuevas simbologías:
  - RM4SCC (Royal Mail Barcode - Reino Unido)
  - KIX-Code (Dutch Postal - Holanda)
- Nuevo método BarcodeTest() para validar la cadena codificar antes de generar la imagen.
- Se han solucionado algunos errores menores.

**v.0.15** - Liberación 2011.06.05

- Nuevas simbologías:
  - One Track Pharmacode (usado en la industria farmacéutica)
  - Two Track Pharmacode (usado en la industria farmacéutica)
- Mejoras en el código y los ejemplos.

**v.0.14** - Liberación 2011.02.19

- Corrección en la visualización correcta del dígito de control del código Interleaved 2 of 5. Gracias a Vishal Anand (India)
- Corrección en el nombre de variables en las funciones Float2Int() y Int2Float() . Gracias a Anatoly Mogylevets (Canadá)

**v.0.13** - Liberación 2011.01.04

- Mejoras en el algoritmo para la generación de la simbología EAN/UCC/GS1-128
- Corrección en la selección de fuentes compatibles con GDI+ para visualizar los mensajes de advertencia
- Se han solucionado algunos errores menores.

**v.0.12** - Liberación 2010.11.29

- Nueva simbología EAN/UCC/GS1-128
- Validación de familias y estilos de fuentes permitidas en GDI+. Gracias a Koen Piller (Holanda) y Cesar Chalom (Brasil)

**v.0.11** - Liberación 2010.11.22

- Agregado de un tercer parámetro al método BarcodeImage()
- Se han solucionado algunos errores menores.

**v.0.10** - Liberación 2010.11.19

- Simbologías de códigos de barra soportadas: Code 128, Code 39, Code 39 Extended, Code 93, Code 93 Extended, Standard 2 of 5, Interleaved 2 of 5, EAN-8, EAN-13, ITF-14, UPC-A, UPC-E, Codabar, Code 11, MSI / Plessey, PostNet y Telepen
- Tipos de imágenes soportadas: JPG, BMP, GIF, PNG and TIFF