HISTORIAL DE VERSIONES

|  |  |  |  |
| --- | --- | --- | --- |
| Versión | Observaciones | Autor | Fecha |
| 01 | Se creó un documento para la investigación | ZRCA | 05/03/2019 |

# DATOS GENERALES

|  |  |
| --- | --- |
| **Lugar** | Centro de Bachillerato Tecnológico Agropecuario No.20 |
| **Fecha** | 05/03/2019 |
| **Objetivo** | Mostrar investigación para saber cómo generar un código de barras en c#. |

MIEMBROS

|  |  |
| --- | --- |
| No. | Nombre |
| 1 | Marisol Galvan Soto |
| 2 | Christian Axel |

PUNTOS DE LA INVESTIGACIÓN

**La generación de código de barras**

En lugar de escribir un generador de código de barras Code39 que imita el algoritmo para esa representación de código de barras, mi idea fue utilizar una de las fuentes de código de barras disponibles libremente para producir el código de barras (http://www.squaregear.net/fonts/free3of9.shtml).

Así que mi enfoque fue simple:

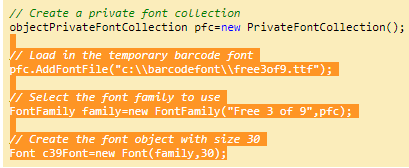
* Cargar la fuente de código de barras
* Crear un objeto de imagen.
* Dibuje una cadena en esa imagen usando una fuente de código de barras code39
* Devuelve esa imagen serializada.

**Usando la fuente Code39...**

La forma de usar una fuente en Windows es simple, todo lo que tiene que hacer es instalarla (copiándola en c: \ WINDOWS \ Fonts - bajo XP) y simplemente utilizarla.

Desafortunadamente, el contexto gráfico ASP.NET no le permite usar ninguna fuente (por ejemplo, free3of9.ttf) porque .NET GDI solo usa / enumera fuentes OpenType. Entonces, lo que tienes que hacer es crear un objeto de fuente temporal.

Este método es muy sencillo, como puede ver en el ejemplo de código a continuación:

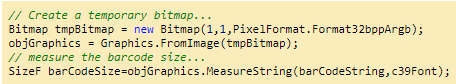


Con esta forma sencilla, obtiene un objeto de fuente asignado a la fuente de código de barras para que pueda crear el código de barras.

**Creación del contenedor de imágenes de código de barras**

La creación de la imagen es muy simple. Las clases .NET te permiten generar imágenes sobre la marcha. Entonces, para crear una imagen lo suficientemente grande como para acomodar el código de barras, primero debe determinar el tamaño que ocupará el dibujo de la cadena de código, utilizando la fuente del código de barras.

Puedes hacerlo utilizando el MeasureStringmétodo:



El tipo devuelto barCodeSizetiene el ancho y la altura que ocupará el dibujo de la cadena de código.

**Dibuja el código de barras**

Así que ahora tenemos que dibujar el código de barras. Usaremos el objeto de fuente de código de barras code39 instanciado anteriormente.

Suponiendo que la variable de código de barras contiene la cadena de código de barras, el código requerido es:

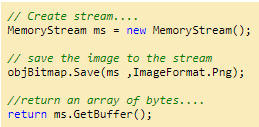


Tenga en cuenta que, por lo general, los códigos de barras code39 se representan concatenando el carácter (\*) al principio y al final de la cadena de códigos de barras... lo que significa que el código 123456 debe escribirse como \* 123456 \*. Pero voy a dejar eso a tu experiencia.

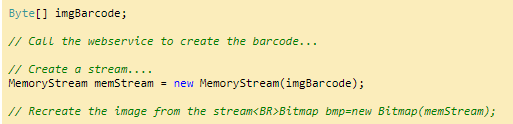
**Serializar / Deserializar la imagen**

Para devolver la imagen desde el método del servicio web, ahora tiene que serializar la imagen, lo que significa que su método web debe devolver una matriz de bytes.

De esta manera, debe crear una secuencia a partir de la imagen de mapa de bits y devolverla como una matriz de bytes. Una vez más, el marco .NET nos facilita la tarea:



En el otro extremo (el lado del cliente) cuando está consumiendo el servicio web, debe poder deserializar la matriz de bytes de nuevo a la imagen:



Una nota final sobre el código de ejemplo adjunto...

Después de crear la aplicación web de código de barras que será su servicio web, debe configurar el archivo web.config para especificar dónde se encuentra la fuente del código de barras. Busque la siguiente sección y realice los cambios correspondientes.

