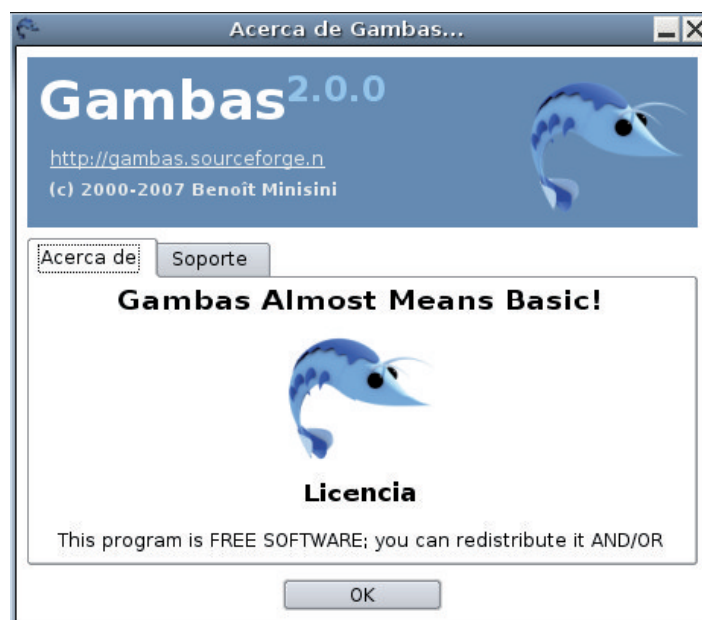


# GAMBAS EN UNA CLASE!

Para todo aquellos programadores de Visual Basic que aún no han encontrado la manera de desarrollar aplicaciones que corran en un sistema GNU/Linux, les esperamos buenas noticias:

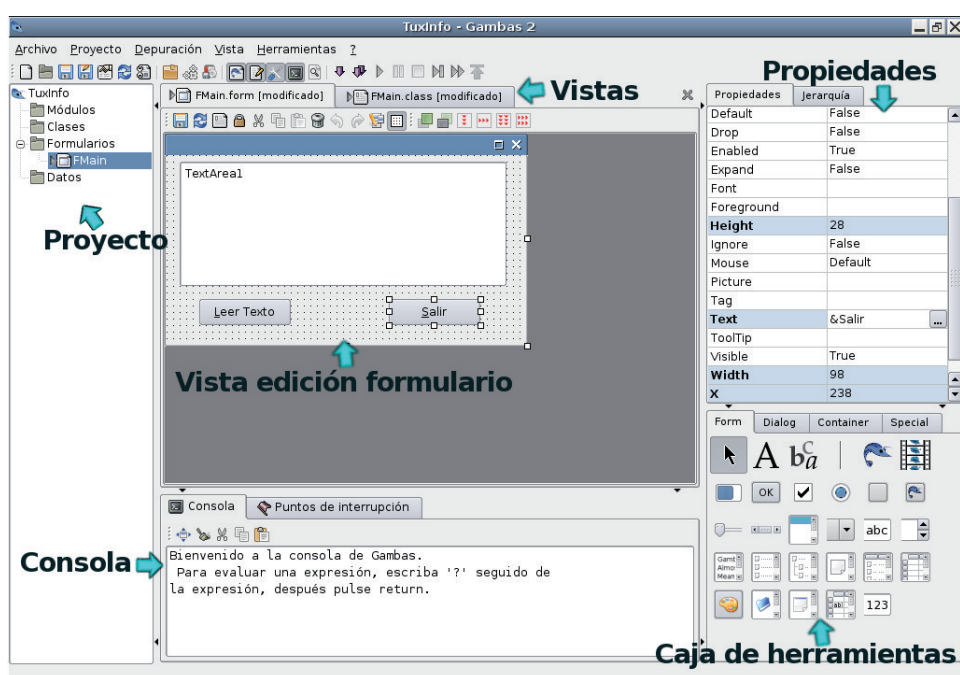
- 1- Existe Gambas
- 2- Gambas no es Visual Basic, es mejor que Visual Basic!



TuxInfo a lo largo de este artículo los invita a descubrir esta poderosa herramienta que les permitirá desarrollar aplicaciones que corran en las más variadas y populares distros de nuestro sistema GNU/Linux, sin limitaciones en cuanto al tipo de escritorio utilizado (KDE, Gnome, XFCE) y con perfecta integración a los principales motores de base de datos (MySQL, PostgreSQL, SQLite).

## ¿Qué es Gambas?

Gambas (acrónimo recursivo de Gambas Almost Means Basic, Gambas es casi como Basic) es un entorno de desarrollo de aplicaciones gráficas cuya licencia es nuestra querida GPL. Por medio de su IDE nos permite diseñar la interfaz gráfica de nuestra aplicación, programar los eventos que disparan los controles, introducir el código, realizar pruebas y depuración, entre otras cosas, para luego generar los paquetes de instalación de nuestro proyecto. Todo esto desde nuestro entorno de trabajo que se aprecia en la siguiente imagen:



## Características de Gambas:

- Basado en el lenguaje Basic.
- Su licencia es GPL.
- Sencilla creación de la GUI.
- Basado en componentes que añaden funcionalidad al lenguaje.
- Es interpretado, genera Bytes-codes.
- Una "máquina virtual" los interpreta (gambas-runtime), al mejor estilo java.
- Genera paquetes de instalación en formato .deb o .rpm.

## Instalación de Gambas

Todas las nuevas versiones se publican a través del sitio web oficial del proyecto <http://gambas.sourceforge.net>

Actualmente la versión estable es la 2.6.0, la cual podemos instalar a través del código fuente, paquetes precompilados o mediante repositorios que se encuentran disponibles en la sección download de la web oficial. Allí, hay una amplia documentación sobre como efectuar la instalación en cada una de las distribuciones.

## Nuestro Primer Programa: Gambas habla!

El fin de nuestro primer programa es que el usuario ingrese un texto, y Gambas se lo lea. Así es! Gambas leerá el texto que ingresemos. Para ello necesitamos tener instalado en nuestro sistema los dos siguientes paquetes que son los que realmente se encargan de hacer dicha tarea.

-alsa-aoss

-espeak

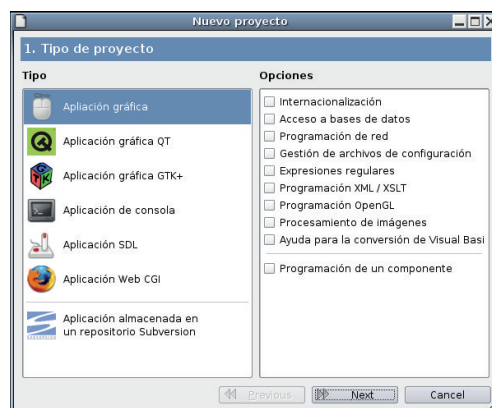
Los usuarios de Debian y sus derivados lo pueden instalar con estos dos comandos:

```
sudo apt-get install alsa-aoss
sudo apt-get install espeak
```

Cabe aclarar que estos paquetes no son parte de Gambas, más bien, son aplicaciones de consola que ejecutaremos desde el código de Gambas. Esto, sin dudas, nos abre un amplio abanico de posibilidades, el hecho de poder ejecutar los poderosos comandos de consola desde nuestra aplicación es una característica destacable de Gambas.

## Iniciando un nuevo proyecto

Al igual que en Visual Basic los programas que desarrollamos son denominados proyectos, y cada uno de los archivos componentes se alojan en una carpeta cuyo nombre es, justamente, el nombre del proyecto. Para iniciar uno nuevo ingresamos en Gambas y en la pantalla inicial seleccionamos nuevo proyecto, lo cual conduce a la siguiente ventana:



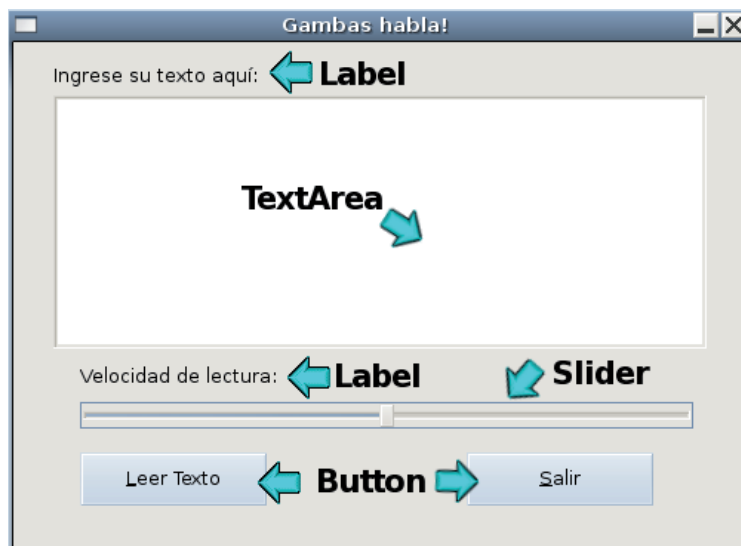
Aquí debemos elegir el tipo de proyecto a realizar (en este caso aplicación gráfica) y decidir con cuál de las dos clásicas librerías que dieron origen a KDE (librerías QT) y Gnome (librerías GTK+) se implementará nuestra interfaz gráfica. Esta elección no es excluyente, es decir, un programa creado en gambas con GTK+ puede funcionar en KDE y uno creado con QT hacerlo en Gnome, pero con el costo de perder homogeneidad en cuanto a la apariencia con respecto al resto de las aplicaciones.

Sin embargo, una de las virtudes de Gambas es que si el programa se creó con el componente GTK+ (gb.gtk), funcionará sin ningún problema si este componente es reemplazado por QT (gb.qt), y viceversa.

## Diseño del formulario

Una vez decidido el componente de nuestra interfaz gráfica, pasamos a diseñar la ventana de nuestra aplicación. Es importante que antes se tomen un tiempo para reconocer el IDE y las diferentes alternativas para mostrar cada una de las vistas (desde el menú principal Vistas) que son señaladas en la primer figura de este artículo.

Seguramente en este punto ya habrán comprendido también la utilidad de las flechitas que permiten contraer o expandir cada una de las vistas, por ahora alcanza con activar las vistas de Proyecto, Propiedades y Caja de herramientas, ya que son fundamentales para diseñar el formulario que a continuación propongo:



Como vemos, el formulario está compuesto por un control TextArea, un control Slider, dos Label y dos Button. Todos se incorporan desde la caja de herramientas. Para ello debemos seleccionar el formulario FMain desde la vista proyecto, logrando de esta manera visualizarlo en el diseñador de formulario. Luego nos dirigimos a la caja de herramientas e identificamos los controles ya mencionados (dejando el puntero del mouse sobre cada uno vemos su nom-

bre) y los arrastramos a nuestro formulario.

El control Label se utiliza para mostrar una leyenda en nuestro formulario, en este caso para darle claridad a nuestra interfaz gráfica de usuario.

El control TextArea permite ingresar un texto. Se diferencia del TextBox al permitir el uso de múltiples líneas, ideal para textos extensos.

El control Slider tiene la función de seleccionar un número dentro de un rango predefinido.

Los controles Button permiten al usuario ejecutar una determinada acción, en este caso efectuar la lectura o finalizar el programa.

## Ventana de propiedades

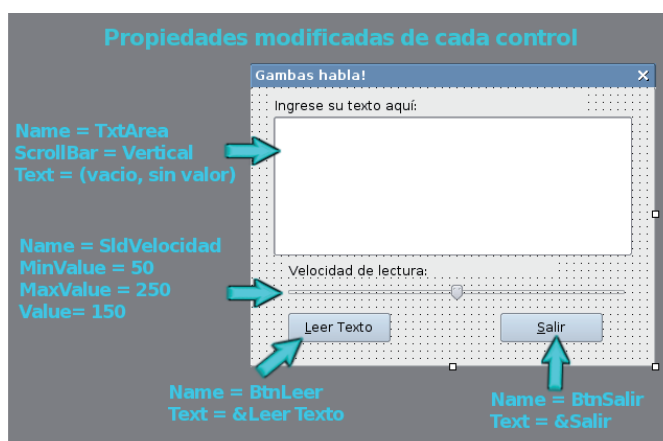
Ya contamos con los controles pero aún el formulario carece del aspecto deseado, por ello es necesario modificar las propiedades de nuestros controles y formulario. Esto se hace por medio de la ventana de propiedades, la cual permite modificar los atributos del control seleccionado.

Si deseamos ponerle un título a nuestra ventana, lo hacemos seleccionando el formulario, y colocando en la propiedad Text el texto Gambas habla!. De igual forma ingresamos la leyenda de cada Label y luego lo propio en cada Button. En el caso del control Button podemos elegir la tecla rápida para activarlo, tal cual se lo hacía en Visual Basic, anteponiendo el andpersand (&) a la letra deseada. La propiedad Background nos permite darle un color de fondo a cada uno de los controles.

Al control TextArea le eliminamos el valor de la propiedad Text para limpiar su contenido. La propiedad ScrollBars nos permite habilitar las barras de desplazamiento en caso que el texto sea muy extenso y no pueda visualizarse en forma completa. Seleccionamos aquí Vertical.

En el control Slider el rango se establece con las propiedades MinValue y MaxValue. El rango propuesto va de 50 a 250. Como valor inicial indicamos 150 a través la propiedad Value.

Una buena práctica de programación es establecer el nombre de cada uno de los controles (propiedad Name). Dicho nombre debería ser lo mas descriptivo posible del control y su función, para facilitar así la lectura de nuestro código. La convención más utilizada es reservar los tres primeros caracteres para identificar el tipo de control y el resto para representar la función que cumple. Los nombres y el resto de las propiedades propuestas para cada uno de los controles lo apreciamos en la siguiente imagen.



## El código Gambas

La programación en Gambas, al igual que en Visual Basic, es orientada a eventos. Los eventos son sucesos que disparan los controles al interactuar con el usuario. Nuestro programa responderá a dos eventos:

- El usuario pulsa el botón "Leer texto".
- El usuario pulsa el botón "Salir".

Para programar el primer evento hacemos doble click al control Button denominado BtnLeer. Esto nos conduce a la ventana de código que nos propone introducir la programación correspondiente al evento Click del botón, de allí su nombre PUBLIC SUB BtnLeer\_Click(). Ahora viene lo interesante, hacer hablar a la computadora. Dicha tarea, como les comentaba previamente, es realizada por los paquetes alsa-aoss y espeak.

Partiendo de que el siguiente comando de consola permite realizar lo planteado:

```
aoss espeak -ves -s150 "Hola lector de Tuxinfo"
```

Donde -v va seguido del idioma (es) y -s va seguido de la velocidad de lectura (150), y, sumado a la facilidad de Gambas de ejecutar comandos de consola mediante la función EXEC, el evento Click del botón leer queda programado de la siguiente manera:

```
PUBLIC SUB BtnLeer_Click()
EXEC ["aoss", "espeak", "-ves", "-s" & Str(SldVelocidad.Value), TxtArea.Text]
END
```

Tres cosas resaltan del código anterior:

- El nombre del comando y cada uno de sus parámetros se colocan en la función EXEC separados por coma.
- Cada parámetro es un String (cadena de caracteres).
- El valor de la velocidad se obtiene del control Slide (mediante la función STR se convierte el valor numérico a String), y el texto a leer se obtiene del TextArea.

Por último programamos el botón Salir con una orden que descarga el formulario de memoria:

```
PUBLIC SUB BtnSalir_Click()
ME.Close
END
```

¡Nuestro programa está terminado! Para probarlo, lo hacemos como en Visual Basic, mediante F5 o con el icono play.

A continuación, la programación completa de nuestro primer programa, con salida profesional y mensaje de bienvenida incluidos. Porque, como verán, ¡Gambas no es Visual Basic, es mejor que Visual Basic!

Pablo Mileti  
pablomileti@gmail.com