

iGambas! Mucho más que un plato de mariscos

Por: Carlos Armando López Solano
Ingeniero Civil. Universidad De La Salle
Tecnólogo en Informática. Corporación Universitaria Minuto de Dios
clopez@uniminuto.edu

Alejandro Moreno Pinilla
Tecnólogo en Informática. Corporación Universitaria Minuto de Dios
amoreni@uniminuto.edu

Introducción

Cuando escuchamos la palabra gambas una infinidad de imágenes traídas a nuestra mente por la semiótica tratan de buscar el significado a esta palabra, sin ir más lejos una gamba es un marisco o crustáceo también semejante al camarón o langostino. Pero, ¿a quien se le ocurre relacionar Gambas con un lenguaje de programación?, al señor Benoit Minisini quien ha denominado su obra con el acrónimo GAMBAS: Gambas Almost Means BASIC, es decir Gambas es casi como BASIC. Así que tendremos que pensar que este animalito de mar ahora hace parte del mundo de los lenguajes de programación.

GAMBAS es un entorno de desarrollo integrado para la creación de todo tipo de aplicaciones sobre sistemas GNU/Linux. Basado en el lenguaje de programación BASIC, ofrece facilidades para generar aplicaciones de todo tipo:

- Aplicaciones de texto.
- Aplicaciones gráficas basadas en GTK+/Gnome y QT/KDE.
- Servicios web.
- Gestión de bases de datos: MySQL, PostgreSQL, Firebird, SQLite, ODBC.
- Redes: sockets, HTTP, FTP.
- Trabajo con XML y XSLT.
- Diseño gráfico en 2D y 3D: SDL, OpenGL.
- Empaquetado y despliegue sencillo de aplicaciones en diferentes distribuciones.

Resumen

En el desarrollo de aplicaciones se busca siempre crear nuevas herramientas que faciliten el trabajo de los programadores; actualmente, los entornos de desarrollo son pieza fundamental en la creación de las mismas. Con la llegada de los entornos RAD (Rapid Development application) en el mundo del software privativo se produjo una gran aceleración en el desarrollo de aplicaciones ricas en funcionalidad y facilidades de uso. Como es de esperar, en el ámbito del software libre también contamos con varios entornos RAD de gran poder; sin embargo, uno de ellos se destaca, se trata de Gambas un entorno/lenguaje basado en Basic y Java.

Palabras Clave:

IDE, RAD, entornos de desarrollo, Visual Basic, Visual .NET, Software Libre, Lenguaje Gambas

·Internacionalización: soporte integrado de traducción de aplicaciones

GAMBAS es uno de los lenguajes más prometedores en el mundo del software libre ya que permite el desarrollo de aplicaciones por expertos de otras áreas diferentes a la programación, tal como ocurre con Visual Basic en la plataforma Windows, es decir, con GAMBAS, al igual que con visual Basic no se requieren grandes conocimientos en programación para lograr aplicaciones eficientes en corto tiempo, así que personas que requieran el desarrollo de aplicaciones en áreas como estadística, ingeniería o en otras donde se requiera una alta velocidad de desarrollo, encontrarán en GAMBAS una herramienta poderosa para la creación de aplicaciones en entornos diferentes a Windows.

Cabe destacar que este lenguaje actualmente es muy utilizado en el gran proyecto de Software Libre de Extremadura (España) en el ámbito empresarial, sustituyendo el uso de herramientas privativas en la creación de aplicaciones especializadas en áreas como la manufactura e ingeniería.

Ambiente de estudio

La comunidad de software libre de UNIMINUTO Arcacsl creó en febrero de 2007 el Grupo de Investigación y Desarrollo de Software Libre (GID-SL) integrado por estudiantes de diferentes semestres de los programas Tecnología en Informática y Tecnología en Redes, liderado por Armando López y Alejandro Moreno (egresados de UNIMINUTO). GID-SL asumió a GAMBAS como primer proyecto investigación debido a que además de ser una herramienta (lenguaje) que permite la creación de aplicaciones rápidas es desconocida en la comunidad estudiantil de UNIMINUTO, también porque cumple con los lineamientos del software libre, siendo este un entorno de desarrollo que se distribuye con la licencia GPL GNU (General Public License), esto significa que se distribuye siempre con el código fuente y respeta las cuatro libertades que define la Free Software Foundation y, a su vez, pone al desarrollador de software libre con las suficientes ventajas para competir en el negocio del software.

Orígenes de GAMBAS

El nombre BASIC corresponde a las siglas Beginner's All Purpose Symbolic Instruction Code (código para principiantes e instrucciones simbólicas con cualquier propósito). El lenguaje fue desarrollado en 1964, en el Dartmouth College, por los matemáticos John George Kemeny y Tom Kurtzas quienes intentaban cons-

truir un lenguaje de programación fácil de aprender para sus estudiantes de licenciatura.

Debido a su sencillez BASIC se hizo inmediatamente muy popular y se empezó a usar tanto en aplicaciones científicas como comerciales. Sin embargo, más tarde las interfaces gráficas de ventanas, que Apple popularizó y Microsoft adoptó con sucesivas versiones de Windows, se convirtieron en un estándar y BASIC no era un lenguaje preparado para estos entornos, por esta razón Microsoft creó su propia versión de BASIC y el 20 de marzo de 1991 se lanzó al mercado con el nombre de Visual Basic convirtiéndose con el tiempo en un éxito total. De esta manera fue evolucionando hasta la versión 6.0. En 2002, fue integrado en la plataforma de desarrollo .NET

Con la popularización de los sistemas operativos libres como GNU/LINUX se podía prever la aparición de un entorno de desarrollo equivalente a Visual Basic pero libre. Finalmente, Benoit Minisini un programador con experiencia e la escritura de compiladores comenzó a desarrollar un entorno para GNU/LINUX basado en BASIC y el 28 de febrero de 2002 publicó en Internet la primera versión de GAMBAS. Benoit eliminó del diseño del lenguaje bastantes de los errores que Visual Basic tenía, como la gestión de errores, y le añadió características comunes de los lenguajes más modernos, como la orientación a objetos y la propia estructura de los programas.

En enero de 2005, Benoit publicó la versión 1.0. en la que se incorporaban varios componentes desarrollados por otros programadores que colaboraban con la causa. A partir de esa fecha, comenzó la programación de la versión 2.0. Esta ya incluye algunas mejoras en el lenguaje, muchos más componentes y un nuevo modelo de objetos que permitirán usar GAMBAS en el futuro para el desarrollo de aplicaciones web, con la misma filosofía y facilidad que actualmente se usa para el desarrollo de aplicaciones de escritorio.

La ampliación del lenguaje BASIC alcanza con GAMBAS una gran potencia, profesionalidad y modernidad, sin abandonar la sencillez y claridad de este lenguaje de programación de alto nivel.

GAMBAS es un RAD ¿Qué es un RAD?

Rapid Application Development (RAD) es un proceso de desarrollo de software, (en inglés: software development process), desarrollado inicialmente por James Martín, en 1980. El método comprende el desarrollo interactivo, la construcción de prototipos y el uso de herramientas CASE (computer Aided Software Engi-

neering). Tradicionalmente, el desarrollo rápido de aplicaciones tiende a englobar también la usabilidad, utilidad y la rapidez de ejecución. Hoy en día, se suele utilizar para referirnos al desarrollo rápido de GUIs tal como Glade, o IDEs de desarrollo completas como Delphi, Foxpro o Anjuta. Uno de los lenguajes de programación más usados para hacer aplicaciones rápidamente es VisualBasic.NET

Ventajas y desventajas de RAD

El desarrollo rápido tiene dos ventajas primarias:

·Velocidad del desarrollo. Los aumentos de la velocidad son debido al uso de la herramienta CASE.

·Calidad. Según lo definido por el RAD, es el grado al cual un uso entregado resuelve las necesidades de usuarios, así como el grado al cual un sistema entregado tiene costes de mantenimiento bajos. El RAD aumenta calidad con la implicación del usuario en las etapas del análisis y del diseño (GRAPPLE).

Los RAD tienen dos desventajas primarias:

·Características reducidas.

·Escalabilidad reducida, debido a que el RAD se desarrolló como prototipo.

El desarrollo de aplicaciones bajo procesos RAD en sistemas operativos libres como GNU/Linux no ha tenido un gran auge debido a que no han habido herramientas de este tipo con las mismas comodidades que se tienen en otras plataformas como Windows o MAC. Sin embargo, la evolución de los procesos RAD han demostrado que la velocidad en la creación de aplicaciones aumenta considerablemente la posibilidad de adquirir colaboradores en el desarrollo de aplicaciones para un sistema operativo específico. Por ejemplo, en el caso de Microsoft Windows, del 70% al 80% de las aplicaciones que corren en ese sistema operativo corresponden a aplicaciones, creadas bajo procesos RAD usando para ello lenguajes como Visual Basic y VisualBasic.NET.

Al tener herramientas RAD para la creación de aplicaciones en GNU/Linux (tal es el caso de GAMBAS) se facilita la creación de aplicaciones especializadas y se puede aumentar la velocidad de creación de las mismas; sin embargo, el desconocimiento de la existencia de estas herramientas impide que esto ocurra.

GID-SL pescando gambas

En el ámbito del software libre la visión de la investigación se enfoca a la tradición de compartir conocimiento; por esto, muchos de los proyectos de software libre exitosos funcionan bajo un esquema

de cooperativismo, que proporciona un crecimiento acelerado de cada aplicación. El Grupo de Investigación y desarrollo de software Libre GID-SL, siguiendo los anteriores lineamientos, se dispuso a indagar sobre GAMBAS, el novedoso entorno y lenguaje de programación para plataformas GNU/LINUX.

Esta investigación se basa en la evolución de las herramientas de desarrollo de la plataforma GNU/Linux, buscando un verdadero RAD que facilite el trabajo a los programadores de dicha plataforma. Con este trabajo se busca conocer, utilizar y divulgar el lenguaje de programación GAMBAS, así como su entorno de desarrollo, con el fin de adquirir las capacidades necesarias para utilizarlo en la creación de aplicaciones basadas en procesos RAD.

La investigación estuvo comprendida por los siguientes puntos:

1.Lenguaje GAMBAS:

Historia del lenguaje

Configuración del lenguaje

Uso del paradigma de programación orientada a objetos a través de GAMBAS

Comprensión del IDE de GAMBAS

Análisis de la estructura de proyectos que se pueden crear con GAMBAS

2.Interfaces gráficas:

Uso del entorno de desarrollo de GAMBAS para la creación de Widgets

Análisis del uso de las librerías QT y GTK para la obtención de interfaces con el lenguaje GAMBAS

3.Desarrollo de aplicaciones:

Análisis de los procesos RAD

Aplicación de RAD en la creación de aplicaciones

Para estos fines se hizo uso de la investigación aplicada, este tipo de investigación se caracteriza porque busca la aplicación inmediata de los conocimientos que se adquieren y no se enfoca en el desarrollo de teorías. Estuvo caracterizada por los siguientes procesos:

1. Recopilación de información acerca de los procesos RAD.

Se buscó información acerca de la creación de aplicaciones rápidas, teniendo en cuenta las herramientas CASE que pueden estar involucradas en el desarrollo de aplicaciones libres.

2. Recopilación de información acerca del lenguaje GAMBAS.

Se recopiló la documentación existente acerca de

GAMBAS, tal como tutoriales y manuales de operación.

3. Análisis del entorno de desarrollo GAMBAS

Se analizó el funcionamiento del IDE de GAMBAS, teniendo en cuenta sus partes y la forma de uso de cada una de ellas. La información obtenida se recopiló, a fin de tener un análisis más preciso acerca de su funcionamiento y utilidad.

4. Comprensión del lenguaje GAMBAS (como dialecto de BASIC).

Se recopiló información acerca del lenguaje utilizado, teniendo en cuenta palabras reservadas y formas de codificación. Se generaron ejemplos que permitan comprenderlo mejor.

5. Análisis de uso.

Se le buscaron fines prácticos al desarrollo de aplicaciones con GAMBAS, se analizó la facilidad en la administración de bases de datos, graficación y acceso a Internet.

Como producto de esta investigación se produjo documentación concreta sobre GAMBAS, que explica detalladamente el proceso de instalación, el uso del lenguaje así como de su entorno de desarrollo. De la misma forma se explica la referencia del lenguaje para los siguientes aspectos:

- Definición de variables.
- Definición de funciones.
- Creación de clases.
- Definición de atributos.
- Definición de constructores.
- Definición de métodos.
- Creación de objetos.
- modificación de atributos y uso de métodos en un objeto.
- Eventos en los formularios.
- Uso de las bases de datos.
- Uso de diagramas.
- Compilación.
- Creación de paquetes.

Esta documentación se encuentra publicada y se puede descargar del sitio web del proyecto www.gidsl.col.nu y en la documentación de Arca-csl en www.arcacsl docs.col.nu; adicionalmente se puede encontrar en estos espacios virtuales recopilaciones de información como documentos y tutoriales de GAMBAS.

Como segunda instancia se crearon ejercicios prácticos que permiten definir elementos como: usabilidad, facilidad de aprendizaje, potencia del lenguaje y compatibilidad con otras plataformas.

Un ejemplo del funcionamiento de GAMBAS

Bien, he aquí un pequeño ejemplo de como crear una calculadora en GAMBAS con las funciones apenas básicas de suma, resta, multiplicación y división.

Para este ejercicio crearemos una clase llamada ClassCalculadora y un formulario llamado FormCalculadora; antes de empezar a codificar es importante tener en cuenta el diseño de nuestro programa, para esto usaremos solamente el diagrama de clases en UML

Únicamente necesitamos la siguiente clase:

Como se puede observar nuestra clase tiene 3 atributos que son las variables num1, num2 y res de tipo float, estas variables serán privadas, esto quiere decir que únicamente los métodos de nuestra clase pueden acceder a ellas.

```
Class Calculadora
num1: Float
num2: Float
res: Float

public set (float a, Float b)
public suma ()as Float
public resta ()as Float
public mult ()as Float
public div ()as Float
```

Tenemos 5 métodos:
public set(); se encargará de recibir por parámetros, 2 valores que serán asignados a num1 y num2
public suma(); realiza la operación suma
public resta(); realiza la

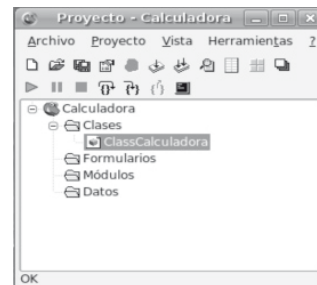
operación resta

public mult(); realiza la operación multiplicación

public div(); realiza la operación división

Manos a la obra:

Una vez dentro de GAMBAS se selecciona "Crear un proyecto gráfico" el nombre y título serán "Calculadora" . Seguido a esto en la ventana proyecto nos ubicamos sobre la carpeta "clases" click derecho->nuevo->clase y la llamamos "ClassCalculadora" ... se obtendrá lo siguiente:



Ahora sí a codificar nuestra clase:

(lo que está escrito con una comilla antes ' corresponde a los comentarios)

```
`GAMBAS class file
`atributos
PRIVATE num1 AS Float
```

```
PRIVATE num2 AS Float
PRIVATE res AS Float
'fin atributos
```

'métodos

'aquí tenemos el constructor que cumple con inicializar nuestros atributos en 0 (cero)

```
PUBLIC SUB _new()
num1 = 0
num2 = 0
res = 0
END
```

'aquí tenemos el método set() que recibe como parámetro a n1 y n2 de tipo float y los iguala a num1 y a num2

```
PUBLIC SUB set(n1 AS Float, n2 AS Float )
num1 = n1
num2 = n2
END
```

'aquí tenemos el método suma() realiza operación de suma y retorna un valor, en este caso "res" de tipo float, fijense la diferencia de esta función con la anterior, set() es una subrutina por esto se utiliza SUB y no retorna ningún valor mientras que suma() es una función y usa FUNCTION, retorna un valor de tipo float por lo cual se usa AS Float

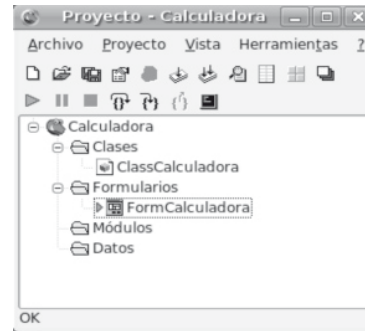
```
PUBLIC FUNCTION suma() AS Float
res = num1 + num2
RETURN res
END
PUBLIC FUNCTION resta() AS Float
res = num1 - num2
RETURN res
END
PUBLIC FUNCTION mult() AS Float
res = num1 * num2
RETURN res
END
```

'aquí tenemos el método div() tiene la misma estructura de el método suma() lo único que cambia es el operador + por /. en esta función debemos validar la división sobre cero pues no es valida, para esto se utiliza una condición con IF...THEN...ELSE

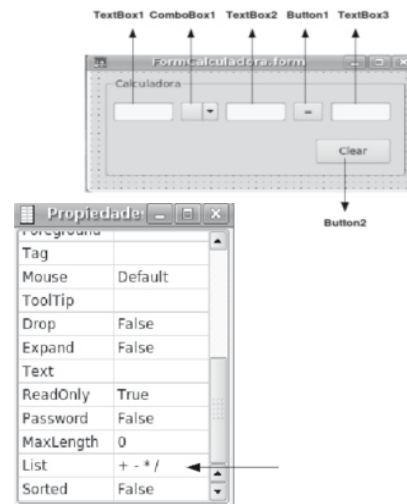
```
PUBLIC FUNCTION div() AS Float
IF (num2 = 0) THEN
Message.Warning("Error la división sobre 0 no es posible")
ELSE
res = num1 / num2
RETURN res
END IF
```

END 'fin métodos

Bueno, con lo anterior se ha creado la clase Class-Calculadora, ahora el paso a seguir es construir la interfaz gráfica. Para esto se crea un nuevo formulario, en la ventana proyecto nos ubicamos sobre la carpeta "formularios" click derecho->nuevo->formulario y lo llamamos "FormCalculadora" en las opciones se debe marcar la opción Clase de inicio y obtiene lo siguiente:



el paso siguiente es diseñar el formulario o la vista de nuestra calculadora, he aquí un diseño:



Es bastante sencillo consta de

TextBox1: es donde se digitará el primer número.

ComboBox1: aquí seleccionamos el tipo de operación a realizar (+, -, *, /).

TextBox2: es donde se digita el segundo número.

Button1: produce un evento en el cual se realiza la operación requerida.

TextBox3: visualiza el resultado de la operación.

Button2 se encarga de limpiar los TextBox.

Todos estos objetos se encuentran en la ventana Caja de herramientas.

Y, ¿cómo se le agrega al ComboBox1 la lista que deberá desplegar? ...

Dentro del formulario se selecciona el ComboBox1, dando click derecho seleccionamos propiedades y se cargará una nueva ventana de propiedades, allí

se encuentran los atributos de ComboBox1 en el atributo "List" se adiciona la lista de operadores(+ - * /), obteniendo lo siguiente:

Una vez terminado el formulario sólo hace falta programar los eventos de los botones Button1 y Button2; para esto se hace click derecho en Button1 seleccionando el menú de eventos -> Click y aparece una nueva ventana con el nombre "FormCalculadora. Class", en esta ventana se codifica el evento Click de Button1

```
PUBLIC SUB Button1_Click()
```

'a continuacion se crea la variable Calc de tipo ClassCalculadora

```
DIM Calc AS ClassCalculadora
```

'se establece que Calc será un objeto de la clase ClassCalculadora

```
Calc=NEW ClassCalculadora
```

'accedemos al método set() y enviamos los parámetros que no son nada mas que los valores que se ingresan en TextBox1 y TextBox2, estos valores siempre serán de tipo string por lo cual 'debemos convertirlos a valor numérico para poder hacer las operaciones, para este fin utilizamos "Val"

```
Calc.set(Val(TextBox1.Text) , Val(TextBox2.Text))
```

' aquí sencillamente decimos que de acuerdo al operador seleccionado en el ComboBox1 se 'realizará la operación adecuada

```
IF (ComboBox1.Text = "+") THEN
```

'como los TextBox solo reconocen valores de tipo string debemos convertir el float que devuelve la 'función suma en string y para esto usamos "CStr"

```
TextBox3.Text = CStr(Calc.suma())
```

```
ELSE IF (ComboBox1.Text = "-") THEN
```

```
TextBox3.Text = CStr(Calc.resta())
```

```
ELSE IF (ComboBox1.Text = "*") THEN
```

```
TextBox3.Text = CStr(Calc.mult())
```

```
ELSE IF (ComboBox1.Text = "/") THEN
```

```
TextBox3.Text = CStr(Calc.div())
```

```
ENDIF
```

```
END
```

bien ya tenemos programado el botón Button1, solo hace falta el botón Button2 procedemos igual que en con Button1 y agregamos este código que se encargara de limpiar los TextBox.

```
PUBLIC SUB Button2_Click()
```

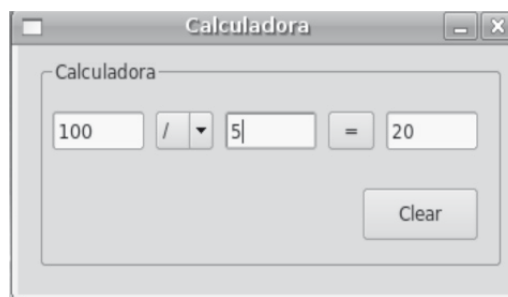
```
TextBox1.Text=""
```

```
TextBox2.Text=""
```

```
TextBox3.Text=""
```

```
END
```

Ahora al compilar y correr, se obtendrá algo como esto :



Conclusiones

Se puede comprender con facilidad el paradigma de programación orientada a objetos a través del lenguaje/entorno GAMBAS, gracias a su organización de componentes. Así que para fines académicos es muy útil a la hora de enseñar este paradigma.

Se ha determinado que GAMBAS puede ser utilizado para la creación de aplicaciones de forma rápida y segura.

La creación de una aplicación gráfica con GAMBAS es sencilla y basta con tener conocimientos básicos en programación. La difícil y tediosa utilización de las librerías QT y GTK toma otra cara, ya que este RAD realiza el trabajo sucio evitando dificultades al programador. Adicionalmente, GAMBAS cuenta con ayudas en el entorno de codificación tales como la generación de texto predictivo e identificación de clases; esto hace muy cómodo el trabajo en los elementos de la aplicación que requieren el uso de código.

Se puede implementar de forma sencilla el patrón Modelo vista Controlador.

Este patrón es uno de los más utilizados actualmente en el desarrollo de aplicaciones y en GAMBAS se utiliza muy fácilmente, gracias a que los formularios, tal como ocurre en lenguajes como Java o VisualBasic. Net, son instancias de clases (objetos) permitiendo separarlos de los procesos internos de la aplicación.

Referencias

Campos Fernández Daniel y Redrejo José Luis. (2003). GAMBAS, programación visual con Software Libre, EDIT LIN EDITORIAL S.L.

<http://gambas.sourceforge.net/>

<http://gambas.gnulinex.org/web/>

<http://gambas.magic-systems.net/cgi-bin/index.gambas>