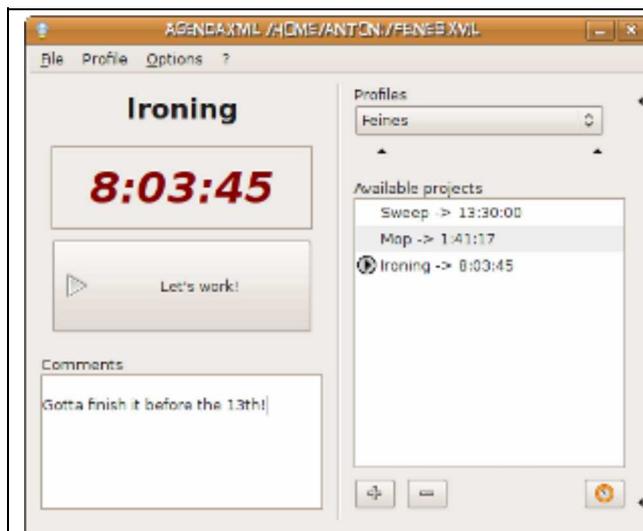




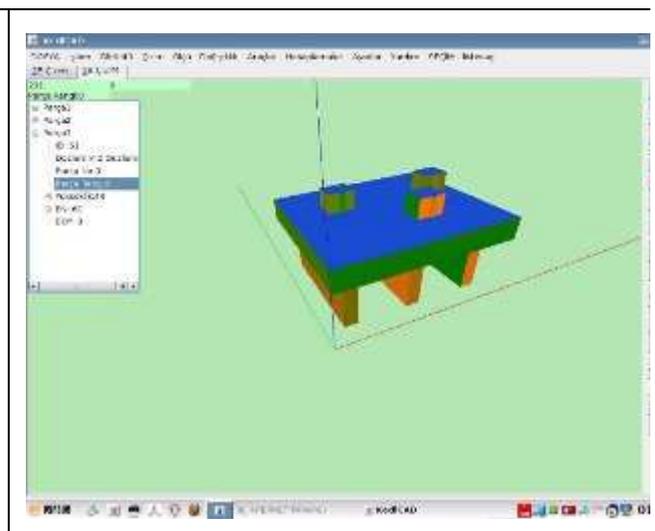
No es la primera vez que aparece este concepto: programación orientada a objetos (POO en español, OOP en inglés). Este paradigma de la programación convierte la creación de aplicaciones en un "juego de niños" cuando se le añade el diseño mediante una interfaz gráfica.

El uso de ventanas y botones es intuitivo y fácilmente comprensible para cualquier usuario. Desde este punto de vista, es un verdadero chollo. ¿Recuerdas a nuestro querido Sr. Computer? En su oficina usan una aplicación de gestión para introducir pedidos, enviar comunicaciones a los clientes, ... Una farmacia, una pequeña tienda de muebles, un hospital, la agencia tributaria, necesitan aplicaciones de gestión que pueda entender y usar un amplio y variado número de usuarios.



Img 1. Agenda XML

Aplicación disponible en [Gamba Application Repository](#)



Img 2. KediCAD

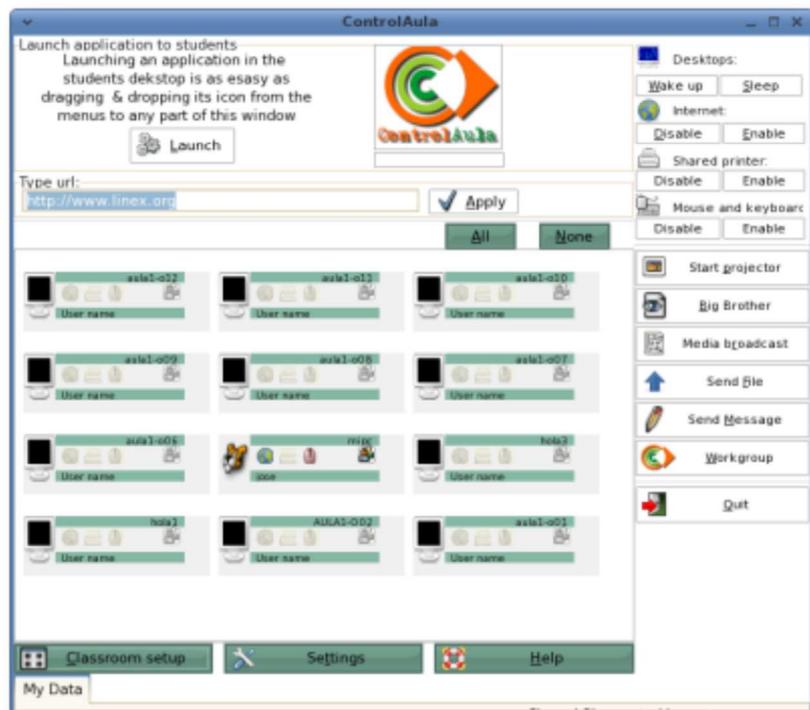
Aplicación disponible en [Gamba Application Repository](#)



Img 3. Logo ControlAula

Img 4. Pantalla de ControlAula

Un ejemplo en el campo de la educación es la aplicación CONTROLAULA creada para permitir a un profesor controlar los ordenadores de sus alumnos en un aula informatizada.



1. Programación orientada a objetos



Programar es conseguir que una máquina resuelva un problema. La orientación a objetos es una forma de programar (un paradigma) que intenta representar la realidad con la que se trabaja.

Para ello se busca el modelo que queremos trabajar (llamado **clase**) y, cuando el programa se ejecutó se crearán **objetos** de ese modelo.

Vamos a intentar explicar la orientación a objetos mediante el siguiente ejemplo. Tenemos que programar la gestión de urgencias de un hospital.

¿Por dónde empezamos? Primero diseño la clase paciente. Defino los campos que necesito tratar, almacenar o compartir. Podrían ser su número de historial, sus datos personales, su historial y su tratamiento.

Defino además los métodos que necesito para trabajar con el paciente, por ejemplo: diagnosticar, comprobar la evolución, dar el alta, etc.

Y ya sólo falta que ocurran **eventos**, el primero de ellos es que con cada persona que entre a urgencias se creará su ficha, un objeto de la clase paciente. Gracias a los métodos que tenemos programados podemos ir actualizando su historial, consultar su estado, darle el alta,....

Y además podemos establecer más clases, incluso heredadas. Así puedo tener la clase diabético, que es un tipo de paciente con sus atributos y métodos específicos.

Veamos la explicación formal.



Importante

Un **objeto** es una estructura de datos que tiene asignadas propiedades y que puede generar eventos y métodos. Un objeto es una instancia de una clase, es decir, un caso concreto de una clase.

Una **clase** describe el comportamiento de un objeto mediante las propiedades y los métodos.

Una **propiedad** es un atributo o una característica de un objeto. Almacenan información sobre el estado del objeto. El conjunto de valores de cada una de las propiedades de un objeto definen el estado de dicho objeto.

Un **evento** es una acción desencadenada por un objeto.

Estos conceptos son complejos, explicarlos es una tarea ardua que supera los objetivos del tema. No obstante, es necesario hacer una serie de aclaraciones:

- ▶ cada objeto tiene asignada una dirección en tiempo de ejecución que puede ser mostrada en la ventana de observación y un estado que puede ser modificado o consultado.
- ▶ cada clase está asociada a un objeto y puede estar definida en el programa o ser definida por el programador como pública o privada.



Para saber más

El concepto de programación orientada a objetos y, especialmente, el concepto de objeto es difícil de definir. Por eso, es mejor ilustrar la definición con los ejemplos que encontrarás en esta web: [ejemplos_objetos](#).

La sintaxis de un objeto y su propiedad es:

NombreDelObjeto.Propiedad



Autoevaluación

Resuelve el siguiente crucigrama:

1. Instancia de una clase.
2. Describe el comportamiento de un objeto.
3. Son los atributos de un objeto.

			1.			2.			
			<input type="text"/>						
			<input type="text"/>			<input type="text"/>			
			<input type="text"/>			<input type="text"/>			
			<input type="text"/>			<input type="text"/>			
			<input type="text"/>			<input type="text"/>			
			<input type="text"/>			<input type="text"/>			
3.	<input type="text"/>								

Comprobar

Las posibilidades de creación de aplicaciones con este tipo de lenguajes son tan grandes que los lenguajes más conocidos son lenguajes orientados a objetos.

- ▶ C++ y ADA son lenguajes textuales.
- ▶ Visual BASIC y Gambas son lenguajes basados en entornos gráficos.
- ▶ Java y VB.NET son lenguajes orientados a internet.

La característica que va a marcar el tema va a ser el desarrollo de aplicaciones en entornos gráficos. En este campo, sin duda ha destacado el lenguaje Visual BASIC. Pero los desarrolladores de software libre han creado el lenguaje Gambas para cubrir esta necesidad en los sistemas operativos GNU/Linux.



Para saber más

Visual BASIC versus Gambas

La polémica está servida. Este artículo es una traducción de una comparativa entre ambos lenguajes. Aunque no entiendas todos los conceptos, seguro que más adelante lo entenderás.

[Visual BASIC vs Gambas](#)



Importante

Beginner's **A**ll-purpose **S**ymbolic **I**nstruction **C**ode

Código de Instrucciones Simbólicas para Principiantes con Cualquier Propósito.

Desde su nacimiento en 1964 hasta alcanzar su mayoría de edad en 1975 el "abuelo" del Visual BASIC, de nombre BASIC, fue creciendo con el propósito de servir de herramienta a estudiantes y programadores principiantes. Su única pretensión era permitir a los programadores realizar programas en varios terminales en tiempo compartido.

PRINT "Hola mundo"

Sintaxis del programa Hola Mundo en Basic



Actividad de lectura

El acrónimo BASIC no es una casualidad. En su origen, este lenguaje se basaba en los siguientes principios:

- ▶ Facilidad e interactividad.
- ▶ No necesita conocer el hardware de la computadora.
- ▶ Rapidez de ejecución.
- ▶ Buena comunicación de los errores.

Por supuesto su éxito se debe a la capacidad para incorporar opciones de programación avanzadas.

Sus hermanos mayores (versiones anteriores) llevaban los nombres de GWBASIC y QuickBASIC. Y, por fin, en 1991 nace Visual BASIC, propiedad de la empresa Microsoft.

El auge del sistema operativo Windows ha hecho de este programa uno de los más utilizados para la creación de aplicaciones orientadas a entornos gráficos. El uso de ventanas, botones y otros elementos gráficos ha sido fundamental, tanto en las más sofisticadas (simulación de sistemas) como en las caseras (gestión de pequeños negocios) e incluso en los juegos.



Img 7. Buscaminas

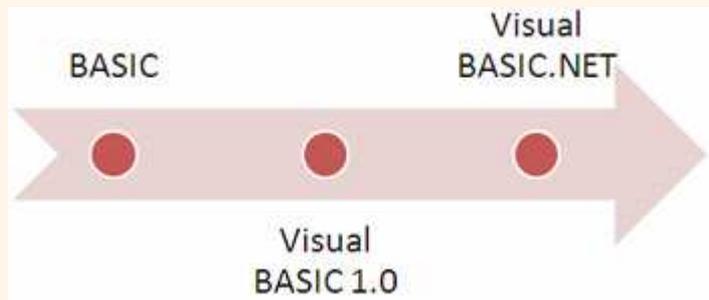
Juego de Windows creado con Visual BASIC.



Para saber más

Evolución

Ya sabes que nos encantan las líneas del tiempo. Son instrumentos útiles y gráficos para describir cualquier evolución. En esta dirección puedes encontrar la evolución de Visual Basic



Si quieres una historia más completa visita esta página web: historia_de_visual_basic



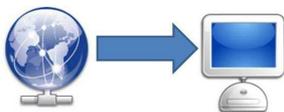
Importante

Visual BASIC (Visual Studio) es un entorno de desarrollo integrado (IDE), es decir un conjunto de elementos que, partiendo de un lenguaje de programación, permite realizar todas las operaciones necesarias desde la creación del código fuente hasta el programa ejecutable.

Incluye los siguientes elementos:

- ▶ un editor de código que permite escribir el código fuente.
- ▶ una interfaz gráfica o GUI para realizar la aplicación usando directamente elementos gráficos.
- ▶ un depurador que detecta, corrige y comunica errores.
- ▶ un compilador que convierte el código fuente a código máquina.

Instalación



Aunque la aplicación pertenece a la empresa Microsoft, existen versiones denominadas Express Edition de descarga gratuita.

Puedes descargarlas desde el sitio oficial en:
[Visual_Studio_Express_Edition_2008](#)



Autoevaluación

a) Visual BASIC fue creado por la empresa Microsoft, pero actualmente se puede descargar gratuitamente bajo licencia de software libre.

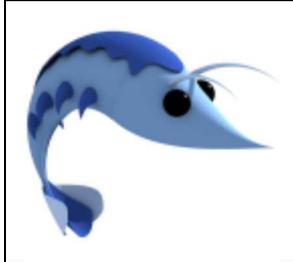
Verdadero Falso

b) Visual Studio.NET es un IDE multilenguaje que permite el diseño de aplicaciones web.

Verdadero Falso

c) Visual BASIC ha sido el lenguaje más extendido debido a que incluso en el diseño de aplicaciones complicadas sigue manteniendo sus características de RAD.

Verdadero Falso



Img. 8. Logo Gambas
Licencia Creative Commons

La alternativa de software libre con licencia GNU/Linux son el lenguaje de programación y el entorno de desarrollo denominados Gambas.

La primera versión se lanzó en 1999 y desde 2008 se distribuye la versión 2.0.



Importante

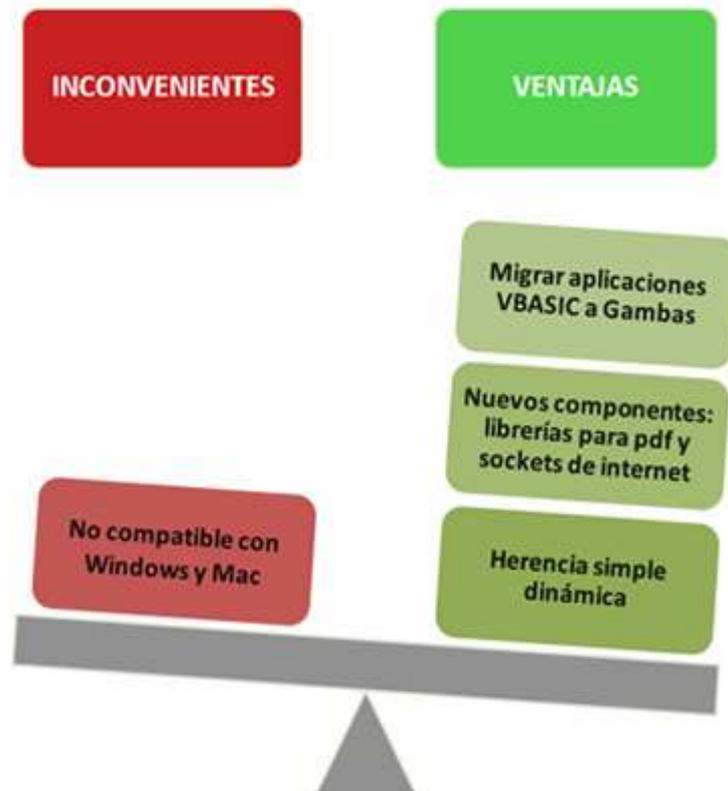
Gambas **A**lmost **M**eans **BAS**IC

Gambas Casi Significa BASIC.

Gambas es un lenguaje de programación orientado a objetos basado en BASIC. Su principal característica es que permite el desarrollo rápido de aplicaciones de cualquier complejidad. La ejecución de éstas requiere un conjunto de librerías intérpretes previamente instaladas (Gambas Runtime) para la ejecución (característica que comparte con Java). También permite el enlace con bases de datos MySQL.

Gambas posee un entorno de desarrollo integrado IDE que incluye una interfaz gráfica de usuario (GUI), un compilador y un intérprete.

Utiliza una biblioteca multiplataforma QT, lo que le hace compatible con casi todas las distribuciones GNU/Linux. Sin embargo, las aplicaciones creadas con Gambas no son válidas en Microsoft Windows ni en Mac OS X.



Las principales características de Gambas son:

- es un IDE, entorno de desarrollo integrado, que permite al programador trabajar mediante una GUI, depurar, compilar y crear el ejecutable de un programa.
- es una herramienta RAD, desarrollo rápido de aplicaciones, que permite al programador realizar tareas básicas de forma casi automática.



Autoevaluación

¿Nuevos acrónimos?

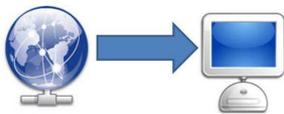
Así es, comenzaste la asignatura con el acrónimo de su nombre TIC y la terminarás con SQL. No esperarías que este tema fuese distinto.

Generalmente los acrónimos se corresponden con su terminología en inglés, pero tienen su significado en español. Comprueba que has tomado buena nota y escribe los acrónimos de estos términos:

- Código de instrucciones para programadores principiantes	<input type="text"/>
- Gambas casi significa BASIC	<input type="text"/>
- Entorno de desarrollo integrado	<input type="text"/>
- Interfaz gráfica de usuario	<input type="text"/>
- Desarrollo rápido de aplicaciones	<input type="text"/>

Comprobar

Instalación



Como todas las aplicaciones de software libre, puedes descargarlo mediante los repositorios oficiales de Synaptic, en el centro de soporte o desde el sitio web oficial [última_version_estable_gambas](#).

Una vez instalado lo encontrarás en el menú Aplicaciones → Programación → Gambas2.



Autoevaluación

a) Gambas es la versión de software libre de Visual BASIC.

Verdadero Falso

b) Los programas implementados utilizando Gambas y Visual BASIC no son compatibles.

Verdadero Falso

c) Es un lenguaje de programación orientada a objetos y, por tanto, cumple el principio de herencia.

Verdadero Falso

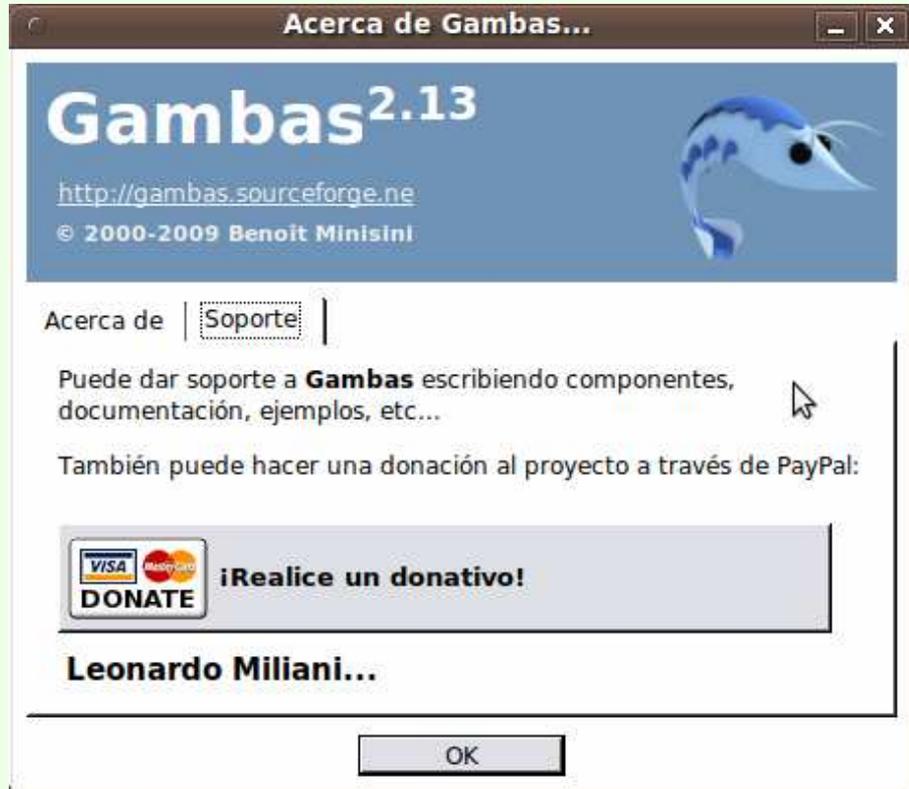


Curiosidad

Ya sabes que Gambas es un lenguaje de programación de software libre. Como en todos los programas GNU/Linux se ofrece la posibilidad de participar en su desarrollo mediante donaciones. Sin embargo, en el menú Acerca de... aparece esta pantalla en la que llama la atención la expresión © 2000-2009 Benoît Minisini. Investiga este nombre, las fechas y el porqué de este símbolo.

Img 9. Pantalla de Gambas

Menú Acerca de...



2. Trabajar con Gambas



En el desarrollo de este apartado aprenderás a realizar pequeñas aplicaciones con Gambas. No pretendemos que te conviertas en un experto en programación. Como ya has instalado el programa, despegamos.

View more [presentations](#) from [ecedenilla](#).



Autoevaluación

1) Gambas guarda los proyectos, por defecto, en el directorio:



- system
- media
- home

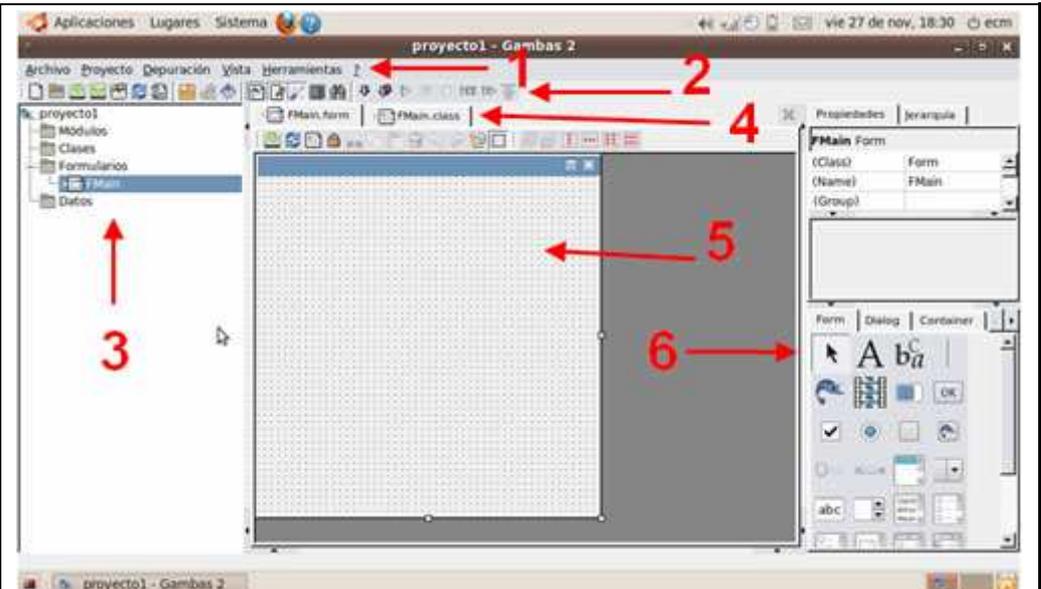
2) El archivo ejecutable se denomina:

- Formulario
- Igual que el proyecto
- Class

3) El formulario principal se llama:

- FMain
- Nombre del proyecto
- Form

En los siguientes apartados vas a realizar un recorrido por todos los elementos de Gambas. No pierdas de vista las distintas zonas de la pantalla de trabajo.

<ol style="list-style-type: none">1. Barra de menús desplegables2. Barra de iconos3. Carpetas4. Vistas5. Formulario activo6. Botones y controles	 <p>The screenshot shows the Gambas 2 IDE interface. Red arrows point to the following elements: 1. The menu bar at the top; 2. The toolbar below the menu bar; 3. The project tree on the left; 4. The view toolbar above the form canvas; 5. The active form canvas in the center; 6. The control palette on the right.</p>
<p>Img 10. Pantalla de trabajo de Gambas</p>	



Curiosidad

El formulario inicial se denomina, por defecto, FMain en Gambas.

Consulta su nombre en Visual BASIC.

¡No intentes establecer comparaciones entre Visual BASIC y Gambas! Ambos lenguajes tienen coincidencias y diferencias.

El trabajo en Gambas, como en cualquier lenguaje orientado a objetos se hace siguiendo estos pasos:



Tras este proceso se creará el archivo ejecutable, es decir, la aplicación en sí.

2.1. Formularios



El desarrollo de programas en entornos gráficos consiste en ofrecer al usuario aplicaciones basadas en el uso de ventanas. El trabajo en modo consola, es decir, con comandos escritos, no se usa en aplicaciones de usuario.

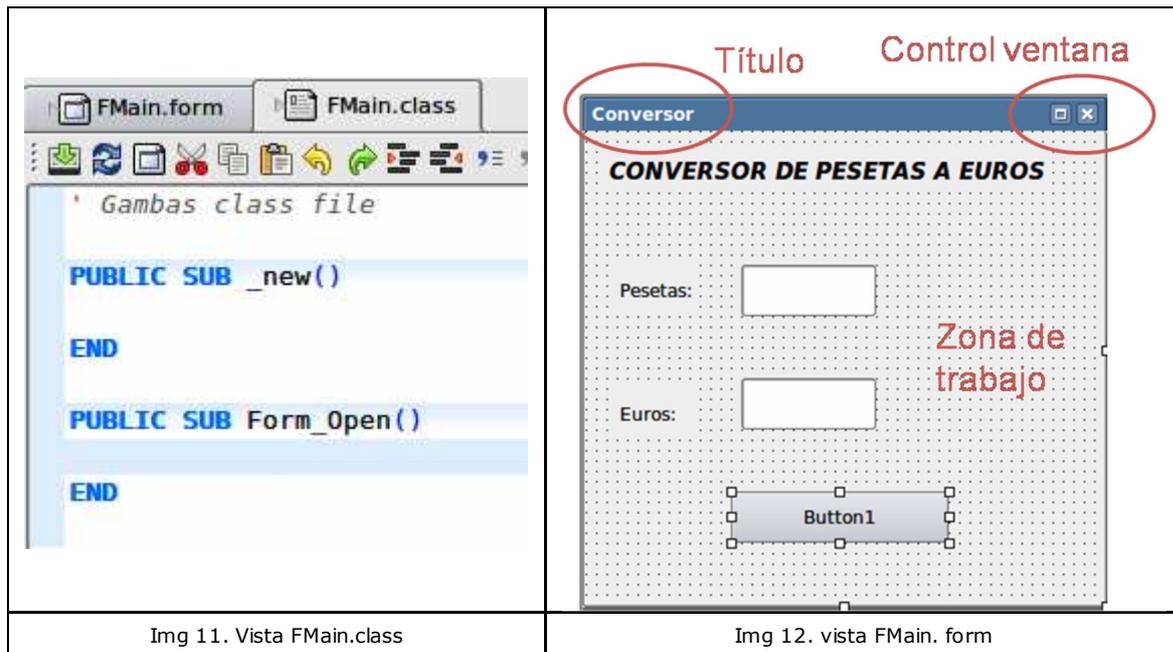
Las ventanas son denominadas formularios.



Importante

Un **formulario** es una ventana en la que se insertan los controles. Cada formulario dispone de controles propios (cerrar, maximizar y minimizar) y propiedades que se pueden modificar (tamaño, título).

Ya sabes que el formulario inicial se denomina FMain. Cuando se crea, Gambas genera el código automáticamente en la ventana class.



Las dos propiedades en que se actúa inicialmente son:

- ▶ Text.- cambia el título del formulario que aparecerá en la barra azul superior.
- ▶ Border.- permite mantener fijo el tamaño (fixed) o redimensionable según el tamaño de la ventana del escritorio (resizeable). Usa esta segunda opción para evitar que los controles puedan aparecer descolocados, el texto cortado, ...



Autoevaluación

1) El título del formulario inicial es el mismo que el del proyecto en Gambas.

Verdadero Falso

2) El tamaño del formulario está definido por defecto con dimensiones variables.

Verdadero Falso

3) Una aplicación está formada por un formulario o ventana con información de entrada y salida.

Verdadero Falso



Para saber más

Cuando se crea un nuevo formulario Gambas nos ofrece la posibilidad de seleccionar varias propiedades como *Constructor* y *Destructor*.

Estas opciones se usan cuando queremos realizar alguna acción con el formulario antes de abrirlo o cerrarlo (limpiar la pantalla, guardar el valor de algún control, ...).

Aunque por el momento no vas a utilizarlas, crea nuevos formularios en Gambas para investigar ambas propiedades.

2.2. Controles



Una vez creado el proyecto (ya tiene nombre, título y una carpeta en la que guardarse), dispones de un formulario vacío.

¿Qué elementos podemos colocar en él?

★ Importante

Los **controles** son las representaciones de los objetos en forma de botones, cuadros de texto o listas. Según su función pueden ser:

- ▶ de entrada.- permiten al usuario desencadenar una acción.
- ▶ de salida.- ofrecen el resultado de una operación o procedimiento.
- ▶ de diálogo.- ofrecen interactividad.
- ▶ contenedores.- los controles insertados en ellos están relacionados y pertenecen a un mismo grupo.

Img 13. Cuadro de controles



? Autoevaluación

¡Ahora te toca a tí!

Pasa el ratón sobre la barra de herramientas de controles y completa los espacios en blanco con el nombre de cada uno (recuerda que están en inglés).

	<input type="text"/>		<input type="text"/>		<input type="text"/>
	<input type="text"/>		<input type="text"/>		<input type="text"/>
	<input type="text"/>		<input type="text"/>		<input type="text"/>

Comprobar

Controles de diálogo

Permiten al usuario interactividad con la aplicación mediante el uso de cuadros de diálogo.

Seleccionar el color, la fecha, un directorio o un fichero son los más comunes.

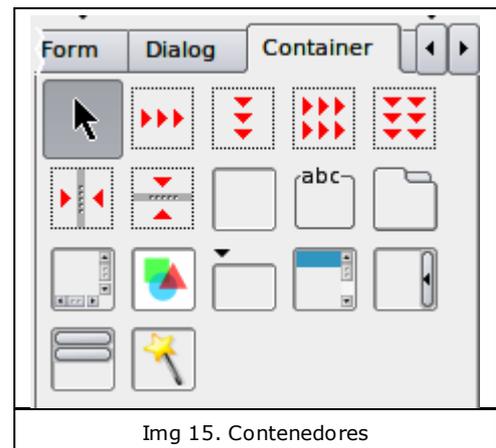


Img 14. Controles de diálogo

Contenedores

Son un tipo especial de control que puede contener varios controles, bien excluyentes (sólo se puede seleccionar uno de ellos), bien pertenecientes a un mismo grupo de opciones.

Uno de los objetivos de la utilización de contenedores es optimizar el espacio en el formulario facilitando el uso de la aplicación.



Img 15. Contenedores

Marcos, paneles laterales, paneles de herramientas, barras de herramientas y hasta la ya conocida varita mágica (wizard) son contenedores.

Para saber más

Controles especiales

Existe una categoría de controles denominados especiales que realizan tareas propias de aplicaciones relacionadas con internet como barras de estatus y de información propias de los navegadores.

Otro de los controles es un "bocadillo" o un "globo" para mostrar ayudas o comentarios.



Img 16. Controles especiales



Para saber más

En la página de documentación sobre Gambas con licencia CC aparecen todos los elementos de la librería gb.form (Gambas.Formulario) que puedes insertar en un formulario. Pincha en cada uno de ellos para ver un ejemplo de su apariencia en el formulario.

[gambas_doc](#)

2.2.1. Programa de ejemplo



La única forma de aprender un lenguaje de programación es hacer un programa. En el siguiente ejercicio vas a utilizar los controles más básicos.

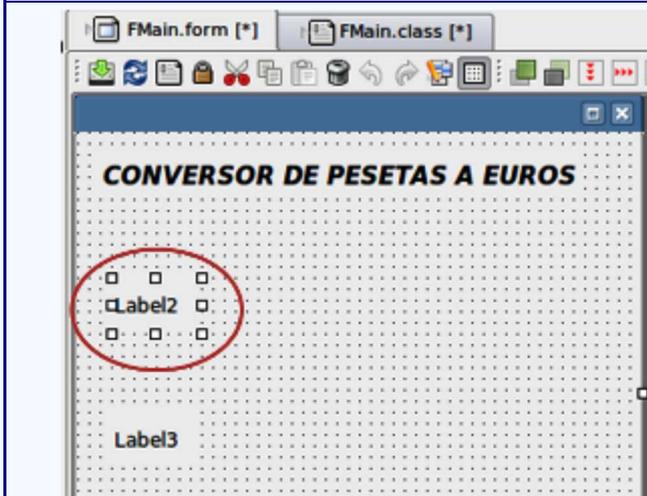
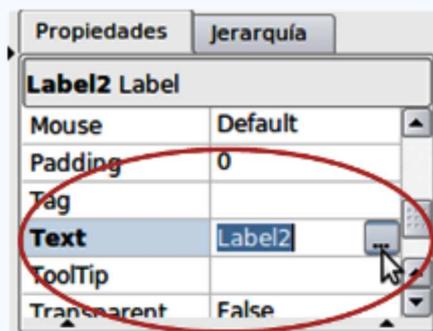
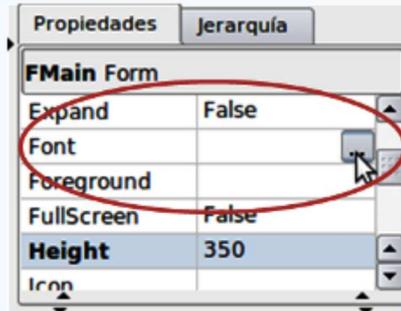
Autoevaluación

Te proponemos como primer ejercicio un conversor de peseta a euros.

Para hacerlo sigue los siguientes pasos:

1. En el formulario principal FMain cambia la propiedad Text asignando el nombre "Conversor". De momento el resto de propiedades no es necesario modificarlas.
2. Inserta una etiqueta de texto (TextLabel) para escribir el título del programa. Cambia las propiedades Text y Font para escribir el contenido y cambiar el tipo, el tamaño y el color de la letra. Escribe "CONVERSOR DE PESETAS A EUROS" en negrita, cursiva y con un tamaño mayor.
3. Inserta dos nuevas etiquetas de texto con la denominación de las monedas: "Pesetas" y "Euros".
4. Necesitarás dos etiquetas más para escribir la abreviatura de cada moneda: "pts" y "€".

Primera fase completada, ya tienes la disposición de los elementos fijos en el formulario.

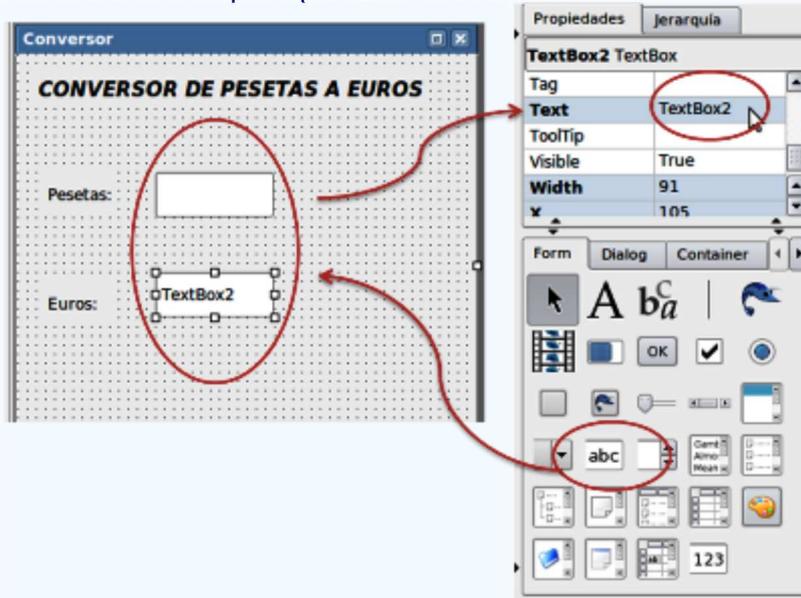




Autoevaluación

En la segunda fase colocarás los controles interactivos que te permitirán introducir los datos y mostrar el resultado.

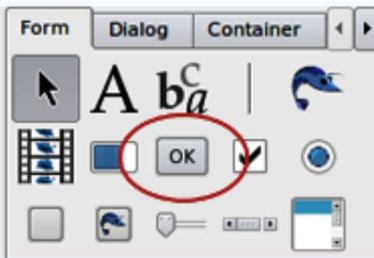
1. Inserta dos cuadros de texto (TextBox) y pon el valor de la propiedad Text vacío para que el formulario siempre aparezca en blanco.



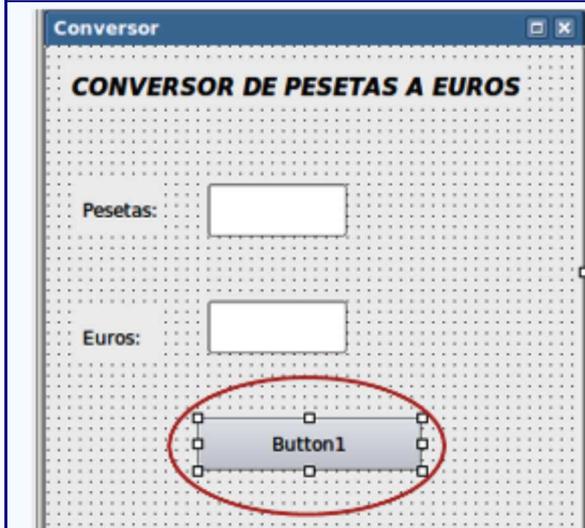
? Autoevaluación

Tercera fase: asignar a un botón la acción de calcular el valor de salida en función del valor de entrada.

1. Insertar un botón (Button) y escribir el texto "Convertir".
2. En la ventana de código, FMain.class, escribir el código de la acción: dividir el número introducido en la casilla de pesetas por 166.386 para calcular los euros equivalentes.



```
PUBLIC SUB Button1_Click()  
    TextBox2.text = TextBox1.text / 166.386  
END
```



+ Para saber más

Es común que los botones respondan a teclas para realizar su función, bien para agilizar el proceso, bien para sustituir al ratón en caso de problemas con este dispositivo.

Esta acción se denomina "atajo de teclado" y se realiza escribiendo el símbolo ampersand (&) delante de la letra que activará el botón.

En nuestro ejemplo lo más lógico sería asignar esta función a la letra C del botón Convertir. Para ello, en la propiedad Text tendrás que escribir *&Convertir*.





Ejercicio resuelto

Ahora sólo queda probar el programa. Para ejecutarlo pulsa F5 o selecciona el menú Depuración. ¿Se parece a este tu formulario?

Conversor

CONVERSION DE PESETAS A EUROS

Pesetas: pts

Euros: €

Convertir

2.3. Eventos y procedimientos



La programación orientada a objetos, a diferencia de la programación estructurada, otorga al usuario el control completo sobre el programa. Es el usuario, mediante la interactividad con el ratón o el teclado, quién determinará que parte del código se va a ejecutar.

Esto es posible gracias a los eventos que permiten a los objetos realizar acciones, hacer clic sobre un botón, cerrar una ventana, incluso hacer doble clic sobre un texto, son acciones que tienen asignado un código.



Importante

Un **procedimiento de evento** es el evento o eventos que desencadena un objeto, junto con el código que contienen.

Su sintaxis es:

```
PUBLIC SUB Control_Evento
```



Autoevaluación

¿Qué control y qué evento aparecen en estos fragmentos de código?

Código 1	Código 2
<pre>PUBLIC SUB Form_Open() END</pre>	<pre>PUBLIC SUB Button1_Click() TextBox2.text = TextBox1.text / 166. END</pre>

	Control	Evento
Código 1	<input type="text"/>	<input type="text"/>
Código 2	<input type="text"/>	<input type="text"/>

Comprobar

Cada control puede desencadenar una serie de eventos. En el navegador de ayuda se puede consultar toda la información relativa a cada control: propiedades, eventos y métodos. Para la clase genérica control se obtiene el siguiente resultado:

Propiedades	Métodos	Eventos
Action BackColor	Delete Drag Grab	DbClick Drag
Background Cursor	Hide Lower Move	DragMove Drop
Design Drop Enabled	MoveScaled Raise	Enter GotFocus
Expand Font ForeColor	Refresh Reparent	KeyPress KeyRelease
Foreground H Handle	Resize	Leave LostFocus
Height Id Ignore Left	ResizeScaled	Menu MouseDown
Mouse Name Next	SetFocus Show	MouseDown
Parent Previous ScreenX		MouseMove
ScreenY Tag ToolTip		MouseUp
Top Visible W Width		MouseWheel
Window X Y		



Curiosidad

Mi pequeño diccionario de Gambas

La mayoría de las especificaciones de Gambas están en inglés, pero su significado prácticamente puede deducirse. Te damos algunas pistas y te proponemos que completes tú el diccionario.

- ▶ **DbClick:** se activa al hacer doble clic con el ratón
- ▶ **KeyPress:** se activa al pulsar una tecla
- ▶ **Drag:** arrastrar (el ratón, un texto, ...)
- ▶ **Drop:** soltar (el elemento arrastrado)



Para saber más

Te proponemos un tutorial de David Asorey en el que encontrarás un ejemplo realizado con la versión 1 de Gambas. Tratándose de las TIC puede considerarse un poco antiguo, pero te servirá de ayuda para comprender los eventos.

[Tutorial_Gambas](#)

Usar un evento en un programa implica una serie de acciones:

- ▶ **EVENT.-** se usa para declarar el evento.
- ▶ **RAISE.-** lanza un evento.
- ▶ **STOP EVENT.-** cancela un evento.



Autoevaluación

a) Un evento es una acción diferente a hacer clic con el ratón.

Verdadero Falso

b) Cualquier control puede soportar cualquier evento.

Verdadero Falso

2.4. Control de errores



Gambas incluye un control de errores que permite al programador anticipar posibles acciones que provocarían un final inesperado o inadecuado del programa.



Importante

Un **control de error** se usa para dirigir el comportamiento del programa cuando se presente una situación anómala.

Las instrucciones utilizadas en Gambas son las siguientes:

- ▶ CATCH.- detecta un error en una función.
- ▶ DEBUG.- imprime un mensaje de depuración.
- ▶ TRY.- intenta ejecutar una instrucción sin disparar un error.
- ▶ ERROR.- devuelve el valor TRUE si se ha producido un error.
- ▶ FINALLY.- ejecuta una instrucción, aún habiendo ocurrido un error.



Autoevaluación

a) Un error provoca el final del programa.

Verdadero Falso



Curiosidad

Los mensajes que Gambas comunica están numerados según un código. En esta dirección puedes encontrar los mensajes ordenados alfabéticamente y por código.

[wiki encyclopedia_Gambas](#)

Primero traduce la página a castellano, después haz clic en el enlace *Mensajes de error*.

División por cero (#26)

Se trató de realizar una división de un número por cero.

Img 17. Mensaje de error de Gambas



Para saber más

Es lógico que varios términos trabajen juntos. En este ejemplo puedes observar el trabajo de los errores TRY-ERROR-CATCH-FINALLY.

```
' Mostrar un archivo en la consola

SUB PrintFile(fileName AS STRING)

    DIM hFile AS File
    DIM sLig AS STRING

    OPEN fileName FOR READ AS #hFile

    WHILE NOT EOF(hFile)
        LINE INPUT #hFile, sLig
        PRINT sLig
    WEND

    FINALLY ' Siempre se ejecuta, incluso si hay error
        CLOSE #hFile

    CATCH ' Se ejecuta sólo si hay error

        PRINT "Imposible mostrar el archivo "; fileName

END
```

Img 18. Manejo de errores en Gambas.

Artículo traducido de Wikibooks. Licencia CC.

3. Archivo ejecutable



El último paso de la programación es la creación del archivo ejecutable. Un paso previo a la creación del ejecutable es "limpiar" el código. Después pasamos a la compilación, con la que obtendremos el fichero ejecutable.



Importante

El **depurador** es una herramienta del IDE que permite detectar y corregir errores de sintaxis en el código del programa.



Img 19. Controles de ejecución

Gambas incorpora la posibilidad de ejecutar el código paso a paso para realizar un seguimiento más exhaustivo. Esta forma de ejecutar el programa en cualquier momento se puede

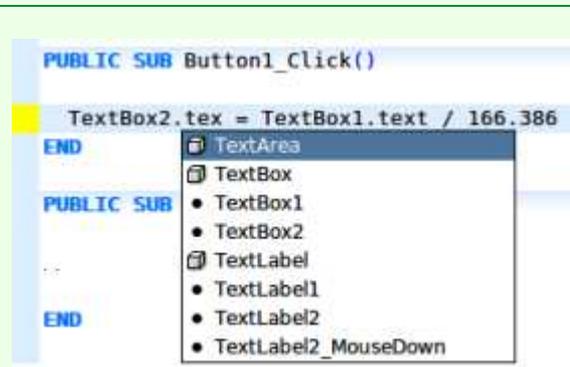
realizar pulsando F5.



Curiosidad

Durante la redacción del código, Gambas ofrece ayuda mediante el llamado "autofilling" o autorrelleno. Cuando escribimos una letra se abre una ventana con las posibles opciones. Esto evita errores, aparentemente pequeños, que pueden generar un código imposible de depurar y compilar.

Img 20. Autorrelleno





Pero el objeto de toda aplicación es ejecutarse independientemente del programa o del lenguaje con el que haya sido creada. Esta operación se denomina **crear el archivo ejecutable** y se lanza desde el menú *Proyecto* → *Compilar* o en el icono de la figura.

Gambas crea un archivo con el nombre del proyecto y una carpeta que contiene todos los formularios, implementación y demás componentes necesarios. Sin embargo, al igual

que ocurre con las aplicaciones de Java, es necesario disponer de un intérprete capaz de ejecutar el código generado (puesto que no es código máquina).



Importante

Gambas Runtime es un componente de Gambas (gbx) cuya misión es servir de intérprete, permitiendo ejecutar las aplicaciones generadas por el programa.

Este componente se instala sin necesidad de instalar el entorno completo. Es el componente gbx.



Autoevaluación

1) Compilar un programa es:

- Comprobar el código.
- Crear el archivo ejecutable.

2) El depurador de código permite:

- Ejecutar el programa paso a paso.
- Eliminar sentencias repetidas.

3) Para ver el resultado de un programa es necesario crear el archivo ejecutable.

- Se puede ver con el comando Vista Previa.
- Se puede ver pulsando F5.

4) Para ejecutar una aplicación es necesario usar el programa Gambas.

- Verdadero.
- Falso.