

# Desarrollando Aplicaciones Gnome con Glade

## Un Tutorial

Eddy Ahmed

<http://eddy.writelinux.com>

*Traducción al español: Waldo Ramírez Montaña.*

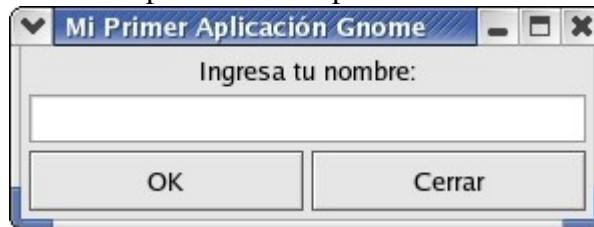
*Corregido por: José Daniel Muñoz Frías*

## Introducción

Este tutorial intenta proporcionar una guía paso a paso para comenzar a escribir aplicaciones Gnome usando Glade. Para ello necesitará:

- [Librerías Gnome](#)
- [Glade](#)
- Conocimientos básicos de C

Al finalizar el tutorial habrá escrito una aplicación del tipo "Hello World!" con la siguiente apariencia:



## ¿Por qué usar Glade?

Glade permite al programador crear rápida y eficientemente un interfaz gráfico de usuario de forma visual. Esto le permite concentrarse en la implementación real del programa en lugar de en los múltiples detalles de la interfaz del usuario.

## ¿Cómo?

1) Arranque glade desde el menú gnome o si lo prefiere desde un terminal tecleando `glade-2`. Se le mostrarán las tres ventanas mostradas en la Ilustración 1.

La primera ventana controla tu proyecto. La paleta te permite añadir *widgets* (pequeños artefactos como botones, cuadros de diálogo, etc.) a tu aplicación, mientras que la ventana de propiedades permitirá la modificación de propiedades del widget elegido y la definición de señales (más adelante veremos más sobre ellas).

2) Lo primero que necesitaremos es un lugar para colocar los widgets. Por tanto en primer

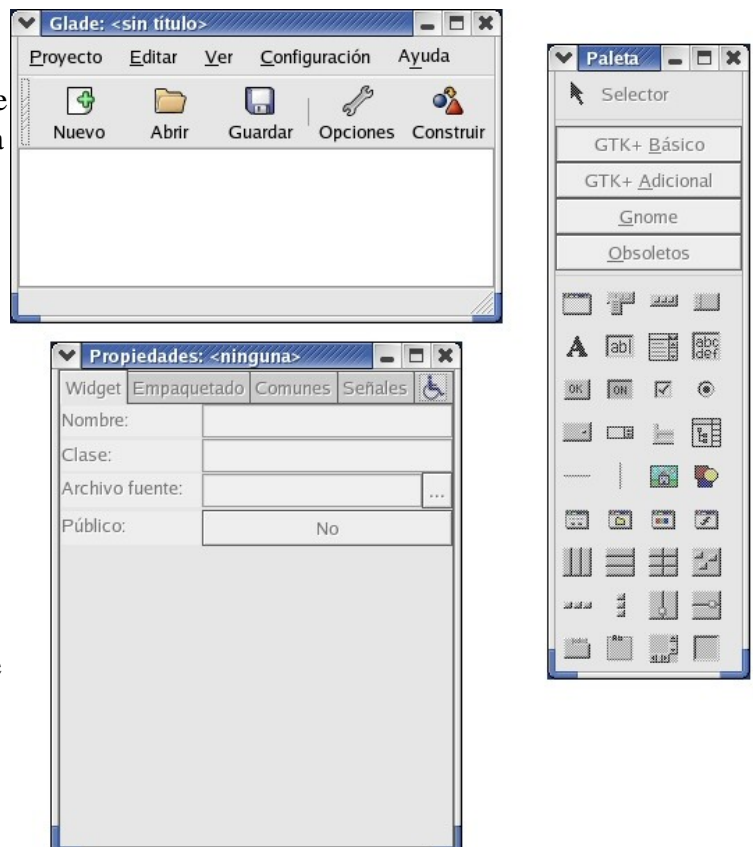


Ilustración 1: Ventanas de Glade

lugar crearemos una ventana haciendo clic en el icono de ventana ubicado en la paleta.



Icono de Ventana, en la paleta.

Con esto aparecerá una nueva ventana lista para ser modificada. Mire las propiedades para la ventana, en particular que el nombre de la ventana es “*window1*”.

Modifiquemos ahora el título de esta ventana, el cual se mostrará en su barra de título, por algo un poco más descriptivo. Para ello, dentro de la ventana de propiedades, basta con situarse en la pestaña Widget y en el apartado Título cambiar “*window1*” por algo parecido a “*Mi Primera Aplicación Gnome*”.

3) Muy bien, ahora la ventana está lista. A continuación vamos a añadir algunos botones y etiquetas... ¡pero espera! No podemos añadir widgets aún... En las aplicaciones Gnome los Widgets están *empaquetados*. Al principio empaquetar widgets puede parecer una carga adicional, pero realmente es un método ingenioso para permitir muchas cosas útiles, siendo la más inmediata el cambio de tamaño automático. Cuando un usuario cambia el tamaño de una aplicación, usualmente nos gusta que los widgets en la ventana también incrementen el tamaño para aprovechar el aumento de espacio, o lo contrario, que se encojan con el objetivo de poder caber en la ventana de acuerdo a su nuevo tamaño. El empaquetado permite realizarlo automáticamente, librando al desarrollador de la escritura del código específico para los cambios de tamaño. El empaquetado se logra creando *cajas* o *tablas*. Estos widgets son invisibles, significando que no pueden ser vistos en el tiempo de ejecución, aunque tienen un efecto directo en la aplicación.

Si observas nuevamente la imagen de la aplicación que crearemos (al principio del documento), puedes ver que es posible dividir la ventana en tres filas: la etiqueta superior, el campo de texto de entrada en la mitad y los dos botones inferiores. Por tanto lo que haremos es crear una *Caja Vertical* con tres filas. Haz clic sobre el icono de Caja Vertical en la paleta y después haz clic en cualquier lugar de la ventana que acabamos de crear.



Icono de Caja Vertical, en la paleta.

Especifica que quieres que sean creadas tres filas y haz clic en Aceptar. La ventana estará ahora dividida en tres filas, lista para que puedan ser añadidos más widgets.

En este momento sería bueno abrir la ventana *Árbol de Widget* eligiendo *Mostrar árbol de widgets* del menú *Ver*.



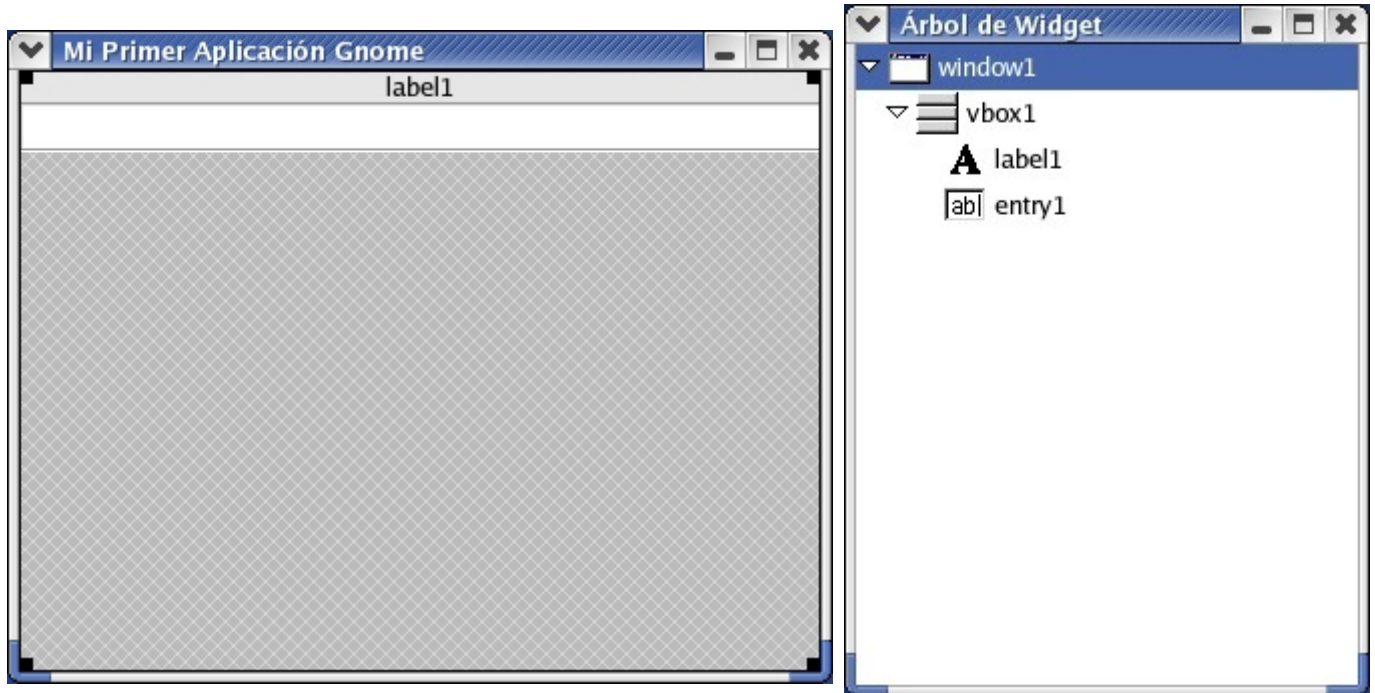
Esta ventana te permite fácilmente obtener la visión de la jerarquía de tus widgets y también su selección.

Ahora podemos añadir con facilidad nuestros widgets etiqueta y entrada de texto en la primera y segunda fila respectivamente. Para ello basta con hacer un clic en el icono de *Etiqueta* de la paleta y luego hacer un clic en la primer fila de nuestra ventana. De manera similar, añade un widget *Entrada de texto* en la segunda fila.



Iconos de Etiqueta y Entrada de texto, respectivamente, en la paleta.

Tu ventana de aplicación y el árbol de widget correspondiente deberían de verse como sigue:



Antes de que podamos añadir los botones en la tercera fila debemos crear una caja horizontal dentro de la tercera fila de la caja vertical. La nueva caja horizontal tendrá dos columnas, una para cada botón. Hagamos esto haciendo clic en el icono de caja horizontal de la paleta y haciendo clic en la tercera fila de nuestra ventana.



Icono de Caja Horizontal, en la paleta.

Especifica dos columnas y haz clic en OK. Ahora la tercera columna se dividirá horizontalmente en dos secciones iguales.

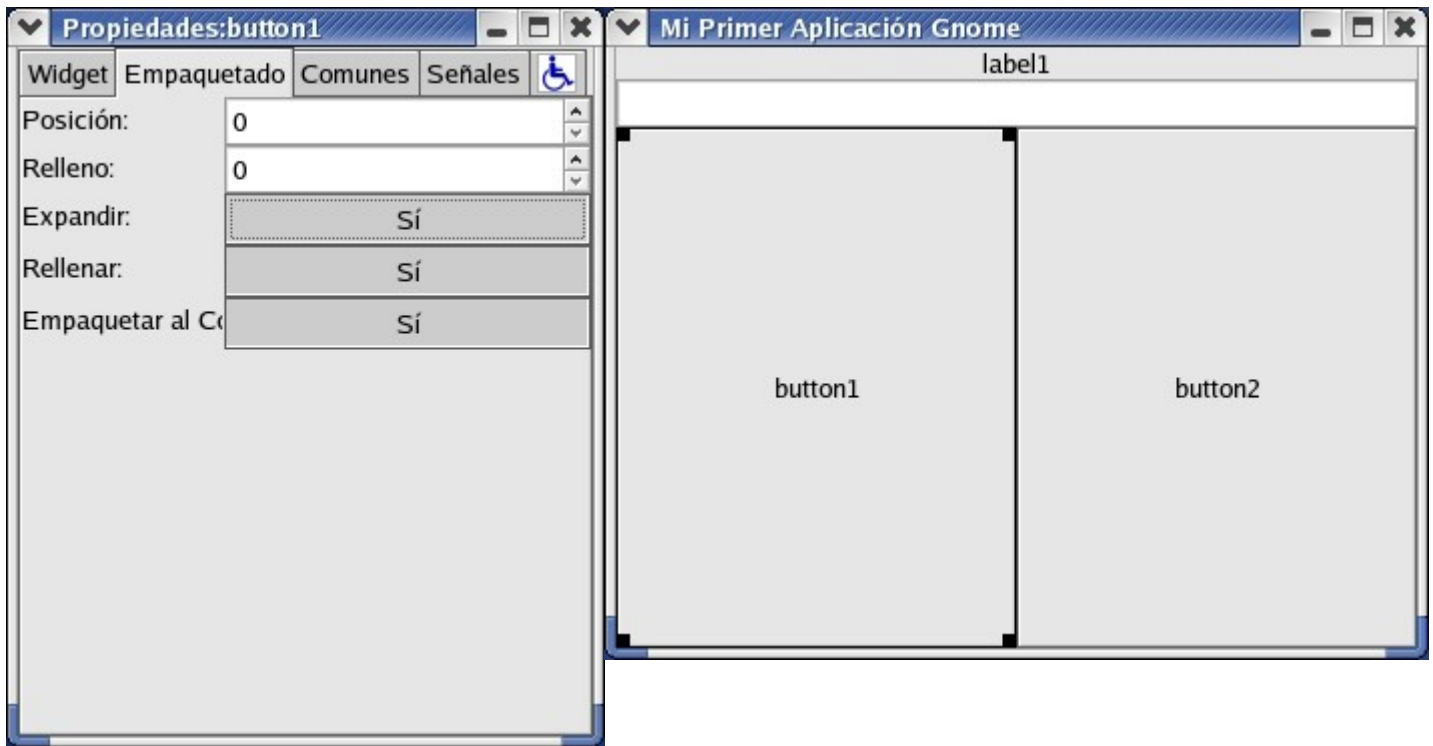
Tal vez este sea un buen momento para guardar nuestro proyecto. Por defecto, Glade salvará tu proyecto dentro de un directorio de *Proyectos* en tu directorio home. Dale un nombre simple a tu proyecto como por ejemplo *hola*.

Sigamos con los botones. Cada sección que hemos creado en la tercera fila tendrá un botón. Por tanto vamos a añadir un botón a cada sección haciendo clic en el icono de *Botón* en la paleta y haciendo clic en la primera sección. Repite la acción, pero ahora para la segunda sección.



Icono de Botón, en la paleta.

Listo, entonces nuestros botones están ahí... pero no se ven muy bien. Esto es porque debemos empaquetarlos adecuadamente. Hazlo dándole clic al primer botón que creaste y en la ventana de *Propiedades* selecciona la pestaña de *Empaquetado*. Dale permiso a las propiedades *Expandir* y *Rellenar* para el botón elegido asignándoles *Si* a ambas. Deberás notar que el primer botón rellena un poco. Realiza lo mismo para el segundo botón y podrás lograr que la aplicación se vea un poco mejor:



Podemos hacer que las cosas se vean aún mejor. Dentro de la ventana de propiedades cambia lo siguiente en las pestañas “Widget” de cada uno de los siguientes widgets (Recuerda que puedes elegir fácilmente a los widgets en la ventana *Árbol de Widgets*):

- Cambia la propiedad *Homogéneo* del widget *hbox1* a *Si*.
- Cambia el *Ancho del Borde* del widget *vbox1* a *4*.
- Cambia el *Espaciado* de los widgets *hbox1* y *vbox1* a *4*.
- Cambia la propiedad *Etiqueta* del widget *button1* a *OK*.
- Cambia la propiedad *Etiqueta* del widget *button2* a *Cerrar*.

Lo grandioso del desarrollo con Glade es que puedes ver inmediatamente como se verá tu aplicación... Es posible jugar alterando las propiedades hasta que la aplicación se vea justo como quieras.

Falta colocar una última propiedad antes de iniciar la escritura de código. Cambia la propiedad de *Etiqueta* de *label1* a "Ingresa tu nombre:".

Sin escribir una sola línea de código, ¡hemos realmente creado una aplicación completamente funcional! Dile a Glade que construya código fuente haciendo clic en *Construir* de la barra de herramientas o seleccionando *Escribir código fuente* del menú Archivo.

Si ahora observas el directorio de tu Proyecto (recuerda que lo guardamos en `/home/[tu nombre de usuario]/Proyectos/hola`) verás todos los archivos que Glade ha creado.

El código fuente real reside en el subdirectorio "src". Algunos archivos como README, ChangeLog y otros, probablemente los modifiques tú mismo cuando desarrolles una aplicación real. Por ahora, sin embargo, podemos dejarlos vacíos. Para construir los Makefiles es necesario abrir un terminal de comandos, situarse dentro del directorio del proyecto y ejecutar **./autogen.sh**. Se visualizarán una serie de mensajes mientras se comprueba la configuración de tu ordenador y se crean los Makefiles adecuados. Cuando finaliza autogen.sh podemos construir nuestra aplicación simplemente ejecutando el comando **make**.

Si todo va bien deberás tener un archivo ejecutable hola en el directorio src. Ejecútalo tecleando **./hola** ¡Voilà! ¡Tu primera aplicación Gnome! Intenta cambiar de tamaño su ventana para observar la magia del

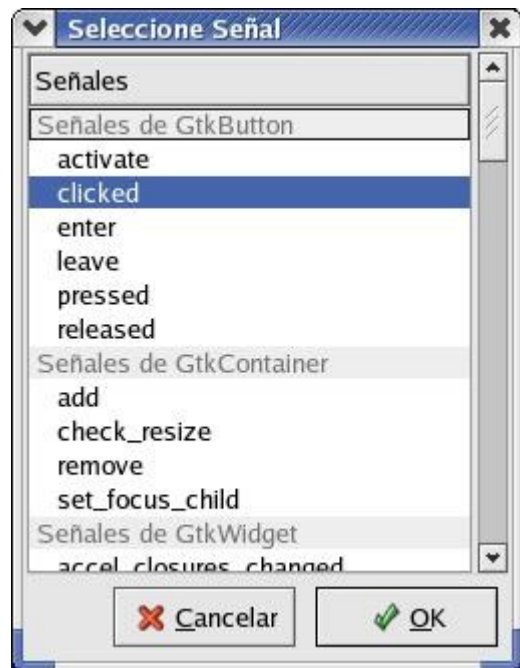
empaquetado trabajando.

Ahora bien, toda esta diversión es buena y bonita... pero nuestra aplicación es aún algo vacío y en realidad no hace nada útil. ¡Entonces vamos a empezar a escribir el código!

Como este es sólo un tutorial de introducción, no haremos nada muy lujoso. Lo que nos gustaría lograr es que cuando el usuario teclee su nombre en la entrada de texto y después haga clic en OK, la etiqueta cambie por un saludo al usuario, consistiendo en un Hola seguido por el nombre que tecleó. También el botón Cerrar deberá finalizar la aplicación.

Para lograrlo debes entender el concepto de señales y respuesta a señal (en inglés, *callbacks*). Una señal sucede en cualquier momento en el que ocurra un evento, tal como el apretar un botón del ratón. Por ello, primero debemos crear un manejador de señal (en inglés, *signal handler*) para un evento del cual queremos que suceda algo. Una vez que se ha creado el manejador de señal hacemos que corresponda a una función de respuesta a señal (en inglés, *callback function*). Esta función es lo que realmente se ejecuta cada vez que sucede el evento. Comencemos por tanto con la creación de dos señales, una para cada botón. Empezaremos con el botón Cerrar, ya que es el más fácil de implementar.

Selecciona el botón Cerrar (button2) y en la ventana de *Propiedades* selecciona la pestaña de *Señales*. Un botón puede generar muchas señales. Sin embargo, en este caso sólo estamos interesados en cuando se hace clic sobre el botón. Para ello, haz clic sobre los puntos suspensivos que están a continuación del campo de *Señal* y de la lista que se muestra selecciona *clicked* y dale a OK. El campo del manejador corresponde al nombre de la función que modificarás para este evento en particular (se puede dejar el valor por defecto). Finalmente añade esta nueva señal haciendo clic en *Añadir*.



Añadamos algo de código trivial a nuestro recién creado manejador, justo para ver a nuestro evento en acción. Todas las respuestas se definen por defecto en el archivo **callbacks.c**, dentro del directorio **src** de tu proyecto. Antes de iniciar el código asegúrate de construir tu proyecto para actualizar nuestros archivos con el nuevo manejador que hemos creado. Cambia a tu terminal e ingresa al directorio **src**. Edita el archivo **callbacks.c** con tu editor de texto favorito. Notificarás que nuestra nueva función del manejador está lista para ser traída a vida:

```
void on_button2_clicked (GtkButton *button, gpointer user_data)
{
}
```

Añadamos algo de código trivial sólo para ver a nuestro evento en acción. La función `g_print` se usa para imprimir en la salida estándar. Escribamos pues algo de código para averiguar que el botón2 recibió un clic. Modifica nuestra función de respuesta como sigue:

```
void on_button2_clicked (GtkButton *button, gpointer user_data)
{
    g_print("El botón2 recibió un clic\n");
}
```

Guarda el archivo y teclea **make**, estando situado en el directorio **src**. Corre el ejecutable **./hola** y dale clic al botón Cerrar. ¡Bien! ¡"El botón recibió un clic" se muestra en la ventana del terminal!

Ahora que sabemos que nuestra función *callback* está siendo llamada, vamos a asignarle algo más útil. Puedes usar la función `gtk_main_quit` para finalizar la aplicación. Entonces todo lo que debes hacer es sustituir la línea de `g_print` por la siguiente:

```
gtk_main_quit();
```

¡Ya tienes un botón Cerrar completamente funcional!

Recordemos qué es lo que queremos que haga el botón OK. Después de que un usuario teclee su nombre en la entrada de texto, cambiaremos la etiqueta para que diga "Hola Usuario", donde Usuario sea el contenido del campo de entrada de texto. Esto envuelve dos conceptos básicos: recuperación de datos provenientes de un widget y establecimiento de propiedades de un widget.

Las funciones específicas de widgets estan bien documentadas en un sin fin de lugares. De entre otras cosas estos recursos documentan funciones específicas para nuestras necesidades inmediatas: etiquetas y entradas de texto.

Lo primero que necesitaremos es una manera de obtener el texto que el usuario ha tecleado. Esto se hace con la función `gtk_entry_get_text()`. En segundo lugar, necesitamos una manera de establecer el texto en nuestra etiqueta. Esto se hace con la función `gtk_label_set_text()`. Ambas funciones requieren en sus argumentos punteros al widget apropiado. En Glade esto se logra con la función `lookup_widget()`.

Por tanto, ahora que tenemos las herramientas vamos a prepararlas. Primero debemos crear nuestro manejador de señal para el botón OK. Selecciona el botón OK (`button1`). En la ventana de Propiedades selecciona la pestaña de Señales y añade una señal *clicked* con el nombre de Manejador por defecto. Construye el código fuente y ahora deberías ver una función `on_button1_clicked` vacía en tu archivo `callbacks.c`.

Ahora lo que debemos hacer es obtener punteros para los widgets que estaremos usando: `label1` y `entry1`. `lookup_widget` devuelve un puntero de tipo `GtkWidget`. Por tanto, vamos a crear dos punteros llamados `label` y `entry` correspondiendo a nuestros widgets. El código deberá verse algo así:

```
GtkWidget * label = lookup_widget(GTK_WIDGET(button), "label1");
GtkWidget * entry = lookup_widget(GTK_WIDGET(button), "entry1");
```

El primer parámetro en `lookup_widget` es un puntero a los datos del widget que llama, en este caso `button` (fíjate que es el primer parámetro de nuestra función de respuesta (*callback*) `on_button1_clicked`). El segundo parámetro es el nombre del widget del cual queremos el puntero. Bien, ahora que tenemos los widgets ¡hagamos algo con ellos!

Antes de que coloquemos el texto de la etiqueta necesitaremos crear una cadena con lo que queramos decir. Recuerda que primero quedemos decir "Hola " entonces algo de código Crápido y sucio que se encargará de eso es:

```
gchar output[50]="Hola ";
```

Todo lo que hicimos fue crear un vector de caracteres llamado `output` de tipo `gchar` almacenándole "Hola" y un espacio extra para inicializar la primera parte de nuestro mensaje. Lo siguiente que tendremos que hacer

es concatenarle lo que haya en el widget de entrada. Para obtener este texto, llamaremos a la función `gtk_entry_get_text` la cual simplemente devolverá cualquier texto que se encuentre en el widget de entrada. Una vez obtenido dicho texto basta con concatenarlo a nuestra cadena `output` mediante la función `strcat`:

```
strcat(output, gtk_entry_get_text(GTK_ENTRY(entry)));
```

Recuerda que queremos pasar un puntero a `gtk_entry_get_text`. También debemos hacer un cast de tipo al puntero, para no obtener warnings durante un `make`.

Ahora lo que hacemos es colocar nuestra cadena en el widget de etiqueta usando la función `gtk_label_set_text` y ¡listo!

```
gtk_label_set_text(GTK_LABEL(label), output);
```

Entonces nuestra función completa es:

```
void on_button1_clicked(GtkButton *button, gpointer user_data)
{
    GtkWidget * label = lookup_widget(GTK_WIDGET(button), "label1");
    GtkWidget * entry = lookup_widget(GTK_WIDGET(button), "entry1");
    gchar output[50]="Hello ";
    strcat(output,gtk_entry_get_text(GTK_ENTRY(entry)));
    gtk_label_set_text(GTK_LABEL(label), output);
}
```

¡Estamos listos! Ejecuta un `make` en el directorio `src` e inicia tu aplicación. Teclea tu nombre en la entrada de texto y haz clic a OK. ¡Hola Usuario! ¿A que no está tan mal?



Ahora con optimismo, estarás en camino al desarrollo de tus propias aplicaciones Gnome usando Glade. Incluso con la simplicidad de esta aplicación, el conocimiento que has ganado con la ejecución de este tutorial debería ser suficiente para desarrollar acciones más complicadas y también aplicar ideas similares a otros widgets. Comprueba la infinidad de otros recursos y no temas a experimentar. ¡Espero te hayas divertido!

Por favor contáctame con cualquier pregunta, comentario, cosas que debería cambiar para hacer al tutorial más claro, ideas, historias de éxito, quejas, o cualquier otra cosa.

*El tutorial original es en inglés y propiedad de Eddy Ahmed. La traducción al español fue hecha con autorización de Eddy, respetando los derechos de autor.*

*Tutorial original, Eddy Ahmed: eddy@writelinux.com*

*Traducción y adaptación al español, Waldo Ramírez: waldo.ramirez@correo.unam.mx*

*Dirección para el documento original en inglés:*

<http://eddy.writelinux.com>

*Content in this document translated from the original with permission from the author.*

*Copyright © 2003 Eddy Ahmed*