

Herramientas Avanzadas para el Desarrollo de Aplicaciones

Parte I

1. La autoconexión de señales en el marco Glade + Gtk+ se hace:

Sobre un objeto de la clase Builde.

2. Para guardar una copia temporal de tu directorio de trabajo:

Utilizamos la operación Git stash.

3. Git guarda toda la meta-información de un proyecto en:

En un directorio único por proyecto.

4. SQLite:

Ninguna de las anteriores.

(Ni requiere de un proceso servidor, ni requiere de un servidor y de un cliente, y ni en la versión 3 sí requiere un proceso servidor)

5. Trabajando con Git si dos programadores modifican el mismo archivo:

No tiene por qué producirse un conflicto.

6. En Git la clave “SHA-1” está asociada a:

Cada commit.

7. En un sistema de control de versiones centralizado la operación “push”:

No existe.

8. Si en Git quisiéramos deshacer un commit sin perder la parte de la historia del proyecto donde aparece este commit:

Usaremos la orden `git revert`.

9. La biblioteca Gtk+...

Aprovecha la tecnología de señales/manejadores de Glib/Gobject.

10. En Git la operación pull equivale a:

`fetch + merge`.

11. Para modificar un mensaje de un commit:

Utilizamos la opción `--amend` para modificar el último commit.

12. La operación checkout de Git:

La podemos utilizar para cambiar de rama.

13. La signatura de un manejador conectado a una señal:

Está condicionada por la signatura de la señal.

14. En el patrón de la arquitectura MVC:

Un modelo puede tener varias vistas sobre él.

15. En la programación dirigida por eventos, la cola de eventos:

No es necesaria.

16. La operación commit de Git:

Traslada los cambios hechos en la copia de trabajo al repositorio local.

17. A una señal con visibilidad pública:

Le podemos conectar cualquier manejador independientemente de su visibilidad.

18. Los SVC, según la forma de almacenar los contenidos, se clasifican en:

Centralizados y distribuidos.

19. Las librerías dinámicas:

Ninguna de las anteriores.

(Hay que compilarlas, no debe llevar el prefijo lib y la extensión y no hace crecer el ejecutable del final)

20. Git es un sistema de control de versiones:

Distribuido.

21. Imagen:

Gtk+ ejemplo sencillo I

```
// gtkhw.cs
using Gtk;
using System;
class Hello {
    static void Main() {
        Application.Init (); // Iniciacion de labiblioteca Gtk+
```

La inicialización de gtk+ es Application.Init();

22. Imagen:

```
Window window = new Window ("helloworld");
window.Show();

Application.Run (); // Bucle de espera de eventos
}
```

El bucle de espera de eventos se inicia con Application.run();

23. Imagen:

```
// runs when the user deletes the window using
// the "close window" widget in the window frame.
static void delete_event (object obj,
                          DeleteEventArgs args) {
    Application.Quit ();
}
```

Gtk + termina con Application.Quit;

24. Sqlite es para:

Mas. (C, C# y muchas más)

25. Compara MonoDevelop y Visual Studio:

Pueden ejecutar el mismo tipo de proyectos.

26. Para consultar los metadatos de una bb.dd. en sqlite:

Consultamos la tabla sqlite_master.

27. En el interfaz de comandos sqlite para ejecutar en un fichero con comandos sql:

Utilizamos la operación .exec

28. La operacion branch de Git:

Es la única que podemos usar para renombrar una rama.