

Herramientas libres para el apoyo en el proceso de desarrollo de software

Trac, Subversion, Doxygen

Milton Mazzarri

milmazz@milmazz.com.ve

Encuentro Canaima

7 de noviembre de 2008

Objetivos Generales

- Ofrecer una plataforma para el desarrollo de aplicaciones que brinden soluciones y cubran las necesidades esenciales de la **Administración Pública Nacional (APN)** .

Objetivos Generales

- Ofrecer una plataforma para el desarrollo de aplicaciones que brinden soluciones y cubran las necesidades esenciales de la **Administración Pública Nacional (APN)** .
- Dar cumplimiento al decreto presidencial Nro. 3390 sobre el uso de tecnologías libres en la APN.

Objetivos específicos

- No estar limitado al uso de la APN, puede ser utilizado por cualquier persona.

Objetivos específicos

- No estar limitado al uso de la APN, puede ser utilizado por cualquier persona.
- Brindar un sistema integrado que permita el manejo y seguimiento de proyectos de desarrollo de software.

Objetivos específicos

- No estar limitado al uso de la APN, puede ser utilizado por cualquier persona.
- Brindar un sistema integrado que permita el manejo y seguimiento de proyectos de desarrollo de software.
- Promover las mejores prácticas para el desarrollo de aplicaciones libres.

Objetivos específicos

- No estar limitado al uso de la APN, puede ser utilizado por cualquier persona.
- Brindar un sistema integrado que permita el manejo y seguimiento de proyectos de desarrollo de software.
- Promover las mejores prácticas para el desarrollo de aplicaciones libres.
- Promover la inclusión de **nuevos** desarrollos en la Distribución Canaima.

¿Qué es exactamente?

- Es un sistema wiki, seguimiento y manejo de proyectos mejorado para el desarrollo de proyectos de software.

¿Qué es exactamente?

- Es un sistema wiki, seguimiento y manejo de proyectos mejorado para el desarrollo de proyectos de software.
- Uso de un enfoque minimalista para el manejo de proyectos de desarrollo de software basado en la Web.

¿Qué es exactamente?

- Es un sistema wiki, seguimiento y manejo de proyectos mejorado para el desarrollo de proyectos de software.
- Uso de un enfoque minimalista para el manejo de proyectos de desarrollo de software basado en la Web.
- Tiene como misión ayudar a los desarrolladores a escribir software de excelente calidad, mientras busca no interferir en el proceso y políticas del desarrollo.

Manejo de Proyectos de Desarrollo de Software

- Herramientas comunes para el manejo de proyectos de software:

Manejo de Proyectos de Desarrollo de Software

- Herramientas comunes para el manejo de proyectos de software:
 - Sistemas de Seguimiento.

Manejo de Proyectos de Desarrollo de Software

- Herramientas comunes para el manejo de proyectos de software:
 - Sistemas de Seguimiento.
 - Sistemas de Control de Versiones.

Manejo de Proyectos de Desarrollo de Software

- Herramientas comunes para el manejo de proyectos de software:
 - Sistemas de Seguimiento.
 - Sistemas de Control de Versiones.
 - Sistemas Wiki.

Manejo de Proyectos de Desarrollo de Software

- Herramientas comunes para el manejo de proyectos de software:
 - Sistemas de Seguimiento.
 - Sistemas de Control de Versiones.
 - Sistemas Wiki.
- **Problema:** La información del Proyecto no se concentra en un solo lugar.

Propósito

- Ofrecer una interfaz integrada y consistente para acceder a la información del Proyecto:

Propósito

- Ofrecer una interfaz integrada y consistente para acceder a la información del Proyecto:
 - Sistema de seguimiento de errores integrado.

Propósito

- Ofrecer una interfaz integrada y consistente para acceder a la información del Proyecto:
 - Sistema de seguimiento de errores integrado.
 - Sistema Wiki integrado.

Propósito

- Ofrecer una interfaz integrada y consistente para acceder a la información del Proyecto:
 - Sistema de seguimiento de errores integrado.
 - Sistema Wiki integrado.
 - Integración con Sistemas de Control de Versiones.

Propósito

- Ofrecer una interfaz integrada y consistente para acceder a la información del Proyecto:
 - Sistema de seguimiento de errores integrado.
 - Sistema Wiki integrado.
 - Integración con Sistemas de Control de Versiones.
 - Reportes de tickets.

Propósito

- Ofrecer una interfaz integrada y consistente para acceder a la información del Proyecto:
 - Sistema de seguimiento de errores integrado.
 - Sistema Wiki integrado.
 - Integración con Sistemas de Control de Versiones.
 - Reportes de tickets.
- Ofrecer un sistema totalmente extensible por medio de complementos o *plugins*.

Línea de Tiempo

- Registro de eventos ocurridos a diario.

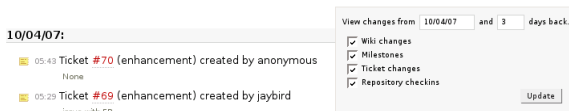


Figura: Línea de tiempo

Línea de Tiempo

- Registro de eventos ocurridos a diario.
- Acceso a los registros desde un solo lugar.

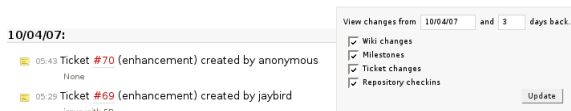


Figura: Línea de tiempo

Línea de Tiempo

- Registro de eventos ocurridos a diario.
- Acceso a los registros desde un solo lugar.
- Ofrecer sindicación **RSS** o *feeds*.



Figura: Línea de tiempo

Sistema Wiki

- Ideal para mantener la base de conocimiento del Proyecto.

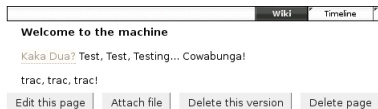


Figura: Sistema Wiki

Sistema Wiki

- Ideal para mantener la base de conocimiento del Proyecto.
- Mantenimiento de la documentación del Proyecto.

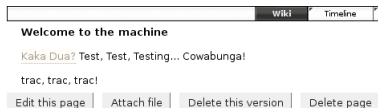


Figura: Sistema Wiki

Sistema Wiki

- Ideal para mantener la base de conocimiento del Proyecto.
- Mantenimiento de la documentación del Proyecto.
- Guías de estilo.

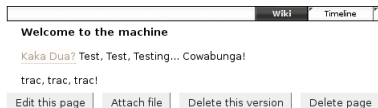


Figura: Sistema Wiki

Sistema Wiki

- Ideal para mantener la base de conocimiento del Proyecto.
- Mantenimiento de la documentación del Proyecto.
- Guías de estilo.
- Recomendaciones a nuevos desarrolladores.

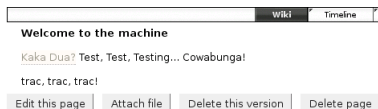


Figura: Sistema Wiki

Consistencia

La misma sintaxis del Wiki es usada en Trac para:

- Páginas Wiki.

Consistencia

La misma sintaxis del Wiki es usada en Trac para:

- Paginas Wiki.
- Tickets (bugs, issues).

Consistencia

La misma sintaxis del Wiki es usada en Trac para:

- Paginas Wiki.
- Tickets (bugs, issues).
- En los mensajes de envío del Sistema de Control de Versiones (en el argot de Subversion nos referimos a esto como commit)

Consistencia

La misma sintaxis del Wiki es usada en Trac para:

- Páginas Wiki.
- Tickets (bugs, issues).
- En los mensajes de envío del Sistema de Control de Versiones (en el argot de Subversion nos referimos a esto como commit)
- En la descripción de los hitos.

Hoja de Ruta

Muestra el porcentaje de avance de la versión actual del proyecto respecto al número de *tickets activos* vs. *tickets cerrados*.

Roadmap

Milestone: Release 1.1a

Due in 3 months (12/25/07)



Closed tickets: 1 Active tickets: 20

☐ Show already completed milestones

Update

Figura: Vista Hoja de Ruta

Integración con Subversion

- Visor del código fuente del proyecto.

Integración con Subversion

- Visor del código fuente del proyecto.

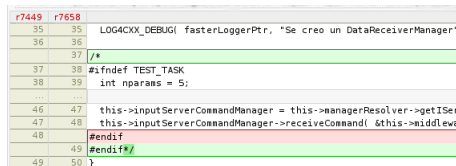
View revision: <input type="text"/>				
Name ▲	Size	Rev	Age	Last Change
 build		1	1 year	TracDemo: Import Trac source
 cgi-bin		1	1 year	TracDemo: Import Trac source
 contrib		1	1 year	TracDemo: Import Trac source
 doc		1	1 year	TracDemo: Import Trac source
 htdocs		1	1 year	TracDemo: Import Trac source

Integración con Subversion

- Visor del código fuente del proyecto.
- Visualización de diferencias en las revisiones, ficheros, etc.

Integración con Subversion

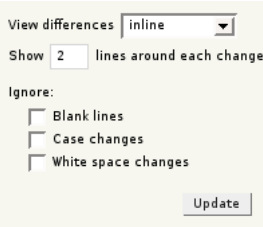
- Visor del código fuente del proyecto.
- Visualización de diferencias en las revisiones, ficheros, etc.



```
r7449  r7658
35      35      LOG4CXX_DEBUG( fasterLoggerPtr, "Se creo un DataReceiverManager"
36      36
37      37      /*
37      38      #ifndef TEST_TASK
38      39      int nparams = 5;
...      ...
46      47      this->inputServerCommandManager = this->managerResolver->getISer
47      48      this->inputServerCommandManager->receiveCommand( &this->middlew
48      48      #endif
49      49      #endif*/
49      50      }
```

Integración con Subversion

- Visor del código fuente del proyecto.
- Visualización de diferencias en las revisiones, ficheros, etc.



View differences inline

Show lines around each change

Ignore:

- ☐ Blank lines
- ☐ Case changes
- ☐ White space changes

Integración con Subversion

- Visor del código fuente del proyecto.
- Visualización de diferencias en las revisiones, ficheros, etc.
- Resaltado de código.

Consultas de *tickets*

Custom Query (65 matches)

Filters

Status ☒ new ☒ assigned ☒ reopened ☐ closed

Add filter

Group results by ☐ descending ☐ Sh

Ticket	Summary	Owner	Priority	
20	None	None	blocker	
24	None	None	blocker	
26	None	None	blocker	
30	None	None	blocker	
40	None	None	blocker	
49	None	None	blocker	
51	None	None	blocker	Release 1.1a

- Summary
- Reporter
- Owner
- Type
- Status
- Priority
- Milestone
- Component
- Version
- Severity
- Resolution
- Keywords
- Cc

Figura: Consultas de tickets personalizadas

Personalización

- Cada organización tiene distintas necesidades.

Personalización

- Cada organización tiene distintas necesidades.
- Capacidad de escribir extensiones en Python para:

Personalización

- Cada organización tiene distintas necesidades.
- Capacidad de escribir extensiones en Python para:
[Macros](#) Definir funciones para usar en el Wiki.

Personalización

- Cada organización tiene distintas necesidades.
- Capacidad de escribir extensiones en Python para:
 - Macros** Definir funciones para usar en el Wiki.
 - Plugins** Extender los componentes actuales o agregar nuevos.

Personalización

- Cada organización tiene distintas necesidades.
- Capacidad de escribir extensiones en Python para:
 - Macros** Definir funciones para usar en el Wiki.
 - Plugins** Extender los componentes actuales o agregar nuevos.
- Cantidad inmensa de **Macros** y **Plugins** disponibles en diversos proyectos de la comunidad del *Software Libre*.

Complementos

- Administración.

Complementos

- Administración.
- Control de SPAM.

Complementos

- Administración.
- Control de SPAM.
- Manejo de cuentas.

Complementos

- Administración.
- Control de SPAM.
- Manejo de cuentas.
- Compatibilidad con Sistemas de Control.

Complementos

- Administración.
- Control de SPAM.
- Manejo de cuentas.
- Compatibilidad con Sistemas de Control.
- Integración con LDAP.

Complementos

- Administración.
- Control de SPAM.
- Manejo de cuentas.
- Compatibilidad con Sistemas de Control.
- Integración con LDAP.
- Integración continua.

Complementos

- Administración.
- Control de SPAM.
- Manejo de cuentas.
- Compatibilidad con Sistemas de Control.
- Integración con LDAP.
- Integración continua.
- ...

Sistema de control de versiones Subversion

Subversion es un sistema de versiones libre y de código abierto. Ofreciendo manejo de directorios y ficheros, además de los cambios hechos en ellos en un intervalo de tiempo. Algunas características que ofrece *subversion* son las siguientes:

- Recuperar revisiones o versiones previas de los datos o código.

Sistema de control de versiones Subversion

Subversion es un sistema de versiones libre y de código abierto. Ofreciendo manejo de directorios y ficheros, además de los cambios hechos en ellos en un intervalo de tiempo. Algunas características que ofrece *subversion* son las siguientes:

- Recuperar revisiones o versiones previas de los datos o código.
- Posibilidad de examinar la historia acerca de cómo ha cambiado el código.

Sistema de control de versiones Subversion

Subversion es un sistema de versiones libre y de código abierto. Ofreciendo manejo de directorios y ficheros, además de los cambios hechos en ellos en un intervalo de tiempo. Algunas características que ofrece *subversion* son las siguientes:

- Recuperar revisiones o versiones previas de los datos o código.
- Posibilidad de examinar la historia acerca de cómo ha cambiado el código.
- Atomicidad en las transacciones de envíos.

Sistema de control de versiones Subversion

Subversion es un sistema de versiones libre y de código abierto. Ofreciendo manejo de directorios y ficheros, además de los cambios hechos en ellos en un intervalo de tiempo. Algunas características que ofrece *subversion* son las siguientes:

- Recuperar revisiones o versiones previas de los datos o código.
- Posibilidad de examinar la historia acerca de cómo ha cambiado el código.
- Atomicidad en las transacciones de envíos.
- Versionado de metadatos.

Sistema de control de versiones Subversion

Subversion es un sistema de versiones libre y de código abierto. Ofreciendo manejo de directorios y ficheros, además de los cambios hechos en ellos en un intervalo de tiempo. Algunas características que ofrece *subversion* son las siguientes:

- Recuperar revisiones o versiones previas de los datos o código.
- Posibilidad de examinar la historia acerca de cómo ha cambiado el código.
- Atomicidad en las transacciones de envíos.
- Versionado de metadatos.
- Manipulación consistente de datos.

Sistema de control de versiones Subversion

Subversion es un sistema de versiones libre y de código abierto. Ofreciendo manejo de directorios y ficheros, además de los cambios hechos en ellos en un intervalo de tiempo. Algunas características que ofrece *subversion* son las siguientes:

- Recuperar revisiones o versiones previas de los datos o código.
- Posibilidad de examinar la historia acerca de cómo ha cambiado el código.
- Atomicidad en las transacciones de envíos.
- Versionado de metadatos.
- Manipulación consistente de datos.
- Ramificación y etiquetados eficientes.

Sistema de control de versiones Subversion

Subversion es un sistema de versiones libre y de código abierto. Ofreciendo manejo de directorios y ficheros, además de los cambios hechos en ellos en un intervalo de tiempo. Algunas características que ofrece *subversion* son las siguientes:

- Recuperar revisiones o versiones previas de los datos o código.
- Posibilidad de examinar la historia acerca de cómo ha cambiado el código.
- Atomicidad en las transacciones de envíos.
- Versionado de metadatos.
- Manipulación consistente de datos.
- Ramificación y etiquetados eficientes.
- Incluye mejoras respecto a su predecesor CVS.

Verdadero historial de versiones

Dado que CVS está limitado al versionado de ficheros, operaciones como copiar y renombrar (las cuales en realidad pueden ocurrir sobre ficheros, pero que realmente son cambios en el contenido del directorio en el que se encuentran) no son soportadas por CVS.

Subversion le permite añadir, borrar, copiar y renombrar ficheros y directorios. Cada fichero nuevo añadido comienza con un historial nuevo, limpio y completamente suyo.

Atomicidad en las transacciones de envío

Una colección cualquiera de modificaciones o bien ingresa por completo al repositorio, o bien no lo hace en absoluto. Permitiendo a los desarrolladores construir y enviar los cambios como fragmentos lógicos e impiden que ocurran problemas cuando sólo una parte de los cambios enviados lo hace con éxito.

Control de versiones de los metadatos

Cada fichero y directorio tiene un conjunto de propiedades (claves y sus valores) asociados a él. Usted puede crear y almacenar cualquier par arbitrario de clave/valor que desee. Las propiedades son versionadas a través del tiempo, al igual que el contenido de los ficheros.

Manipulación consistente de datos

Subversion expresa las diferencias del fichero usando un algoritmo de diferenciación binaria, funciona de la misma manera tanto para ficheros de texto (legibles para humanos) y ficheros binarios (ilegibles para humanos). Ambos tipos de ficheros son almacenados igualmente comprimidos en el repositorio.

Ramificación y etiquetado eficiente

El coste de ramificación y etiquetado no necesita ser proporcional al tamaño del proyecto. *Subversion* crea ramas y etiquetas simplemente copiando el proyecto, usando un mecanismo similar a los enlaces duros en Unix. De este modo las operaciones toman solamente una cantidad de tiempo pequeña y constante.

Doxygen

Doxygen es un sistema de documentación para:

- C++

Doxygen

Doxygen es un sistema de documentación para:

- C++
- C

Doxygen

Doxygen es un sistema de documentación para:

- C++
- C
- Java

Doxygen

Doxygen es un sistema de documentación para:

- C++
- C
- Java
- Objective-C

Doxygen

Doxygen es un sistema de documentación para:

- C++
- C
- Java
- Objective-C
- Python

Doxygen

Doxygen es un sistema de documentación para:

- C++
- C
- Java
- Objective-C
- Python
- ...

¿Cómo puede ayudarle Doxygen?

Doxygen puede ayudarle de varias maneras.

- Puede generar documentación en línea haciendo uso de **HTML** o puede obtener un manual de referencia en **L^AT_EX** a partir de una serie de ficheros de código fuente documentado. También es posible obtener documentos en otros formatos como:
 - RTF (MS-Word)
 - PostScript
 - PDF con enlaces
 - Páginas de Manual Unix

¿Cómo puede ayudarle Doxygen? (Cont.)

- Puede configurar Doxygen para que extraiga la estructura del código fuente sin documentar.
- Usted puede además visualizar la relación entre varios elementos al incluir gráficos de dependencia, diagramas de herencia y colaboración, los cuales son generados automáticamente.
- Usted puede abusar de Doxygen y generar la documentación normal para su proyecto. De hecho, el manual de Doxygen está hecho de esta manera.

Documentando el código

Para los miembros, clases y *namespaces* hay básicamente dos opciones.

- Colocar un bloque de documentación **especial** en frente de la declaración o definición del miembro, clases o namespace.
- **Completar**: Llegamos a la parte que necesita mayor dedicación.

Bloques especiales de documentación

- Un bloque especial de documentación es un comentario al estilo de **C** o **C++** con algunas marcas especiales, de ese modo Doxygen sabe que es un elemento de documentación que necesita incluirse en los resultados que generará.
- Para cada ítem de código hay dos tipos de descripciones, de manera conjunta forman la documentación: Una descripción breve o **brief** y una descripción detallada o **detailed**, ambas son opcionales. Tener una o más descripciones breves o detalladas no está permitido.

Estilo JavaDoc

El estilo **JavaDoc** se caracteriza por agregar un asterisco después de abrir el comentario al estilo **C**, como sigue:

Ejemplo

```
/**  
 * ... texto ...  
 */
```

Estilo Qt

El estilo **Qt** agrega un signo de exclamación después de abrir el comentario estilo **C**, como sigue:

Ejemplo

```
/*!  
* ... texto ...  
*/
```

El asterisco intermedio es opcional

En ambos casos, estilo **JavaDoc** o **Qt**, el ***** intermedio es opcional.

Ejemplo

```
/*!  
... texto ...  
*/  
/**  
... texto ...  
*/
```

Una tercera alternativa

Una tercera alternativa es usar un bloque con al menos dos líneas de comentario en **C++**, donde cada línea comienza con una barra adicional o un signo de exclamación.

Ejemplo

```
///  
/// ... texto ...  
///  
...  
///  
///  
///! ... texto ...  
///!
```

Llamando la atención

Algunas personas prefieren resaltar sus bloques de comentarios en la documentación. Para este propósito usted puede usar lo siguiente:

Ejemplo

```
////////////////////////////////////  
/// ... texto ...  
////////////////////////////////////
```

Haciendo uso del comando \brief

Puede usar el comando `\brief` con uno de los bloques de comentarios mostrados previamente. Este comando termina al finalizar el párrafo, esto quiere decir que la descripción detallada sigue después de la línea en blanco.

Ejemplo

```
/** \brief Descripcion breve.  
 *      Continuación de la descripción.  
 *  
 * Descripción detallada comienza acá. note  
 * que debe dejar una linea en blanco para lograr  
 * tener las dos descripciones.  
 */
```


Documentación de tipos de datos

Ejemplo

```
/**
 * @brief Enumerado de los tipos de datos admitidos
 *
 * Descripción mas detallada de la función de este tipo de dato.
 */

enum DataTypes{ INTEGER, /**<Puede ser un valor de tipo entero */
DOUBLE,    /**<Puede ser un valor de tipo double */
STRING     /**<Puede ser un valor de tipo string */
};
```

Documentación de tipos de datos

Documentación de las enumeraciones miembro de la clase

enum Test3::DataTypes

Enumerado de los tipos de datos admitidos.

Descripción mas detallada de la función de este tipo de dato.

Valores de la enumeración:

INTEGER Puede ser un valor de tipo entero.

DOUBLE Puede ser un valor de tipo double.

STRING Puede ser un valor de tipo string.

Figura: Documentación tipos de datos

Formato de documentación Doxygen

Ejemplo

```
/**
 * @brief Breve descripción de esta función
 *
 * Aquí comienza la descripción detallada de esta función,
 * también se muestra una explicación del valor de retorno de la
 * función si es el caso para esto se utiliza el comando \\@return
 * @return Devuelve un valor entero resultado de la operación
 */
int multFunction ( int c , /**<Primer operando de la operación */
                    int d , /**<Segundo operando de la operación */
                    int e   /**<Tercer operando de la operación */
                  );
```

Formato de documentación Doxygen

```
int Test3::multFunction ( int c,  
                           int d,  
                           int e  
                           )
```

Breve descripción de esta función.

Aquí comienza la descripción detallada de esta función, también

Devuelve:

Devuelve un valor entero resultado de la operación

Parámetros:

- c* Primer operando de la operación
- d* Segundo operando de la operación
- e* Tercer operando de la operación

Figura: Formato de documentación

¡Muchas gracias!

¿Preguntas?

- Milton R. Mazzarri S.
- milmazz@milmazz.com.ve
- <http://www.milmazz.com.ve>