



ATIX

R e v i s t a D i g i t a l

Libertad es proteger lo nuestro

2011



Reconocimiento-Compartir bajo la misma licencia

Usted es libre de:



copiar, distribuir y comunicar públicamente la obra



hacer obras derivadas

Bajo las condiciones siguientes:



Reconocimiento. Debe reconocer los créditos de la obra de la manera especificada por el autor o el licenciador (pero no de una manera que sugiera que tiene su apoyo o apoyan el uso que hace de su obra).



Compartir bajo la misma licencia. Si altera o transforma esta obra, o genera una obra derivada, sólo puede distribuir la obra generada bajo una licencia idéntica a ésta.

- Al reutilizar o distribuir la obra, tiene que dejar bien claro los términos de la licencia de esta obra.
- Alguna de estas condiciones puede no aplicarse si se obtiene el permiso del titular de los derechos de autor
- Nada en esta licencia menoscaba o restringe los derechos morales del autor.

Colaboradores

Dirección y Coordinación General

Esteban Saavedra López (esteban.saavedra@atixlibre.org)

Diseño y Maquetación

Jenny Saavedra López (jenny.saavedra@atixlibre.org)

Esteban Saavedra López (esteban.saavedra@atixlibre.org)

Revisores

Esteban Saavedra López

Jenny Saavedra López

Noticias

Jenny Saavedra López

Lucy Ayarde Romero

Autores Frecuentes

Esteban Saavedra López

Joseph Sandoval

Mario Carrión

Martín Márquez

Renzo Martínez

Rafael Rendón

Herramientas

La edición de esta revista fue realizada de forma íntegra haciendo uso de Software Libre





**Palabra quechua,
con un sentimiento profundo
y con gran significado filosófico**

El que lo sabe

El que lo intenta

El que lo puede

El que lo logra

Editorial

Un nuevo año, un nuevo desafío; lastimosamente por una serie de inconvenientes nuestra revista no se publicó con la frecuencia que se tenía acostumbrado, pero esperamos que es nuevo año sea el que marque el retorno de nuestra continuidad en las publicaciones de los distintos números.

Libertad es proteger lo nuestro, un título con el que deseamos expresar cuan importante es el cuidar lo nuestro, lo que nos rodea, nuestra familia, nuestro hogar, nuestro mundo, ahora más que nunca cuando atravesamos una serie de cambios en el accionar de nuestro mundo, es cuando mayor empeño debemos poner en colaborar a que este siga brindándonos su calor, su belleza y sobre todo la satisfacción de vivir.

Los profesionales técnicos debemos trabajar en hacer de este mundo cada día un mundo mejor, dejando de lado el egoísmo y sobre todo trabajando en compartir nuestro conocimiento para el bien de todos y no solo de algunos pocos.

Con este primer número del 2011, queremos renovar nuestro compromiso de seguir brindando un instrumento que permita la interacción a todos los amantes de la libertad y la tecnología, seguros estamos que pondremos nuestro mayor esfuerzo en lograrlo, pero así mismo solicitar vuestra ayuda en la difusión de este noble emprendimiento y así mismo invitar a ser parte del mismo.

Libertad es proteger lo nuestro

Bienvenidos a nuestro décimo octavo número

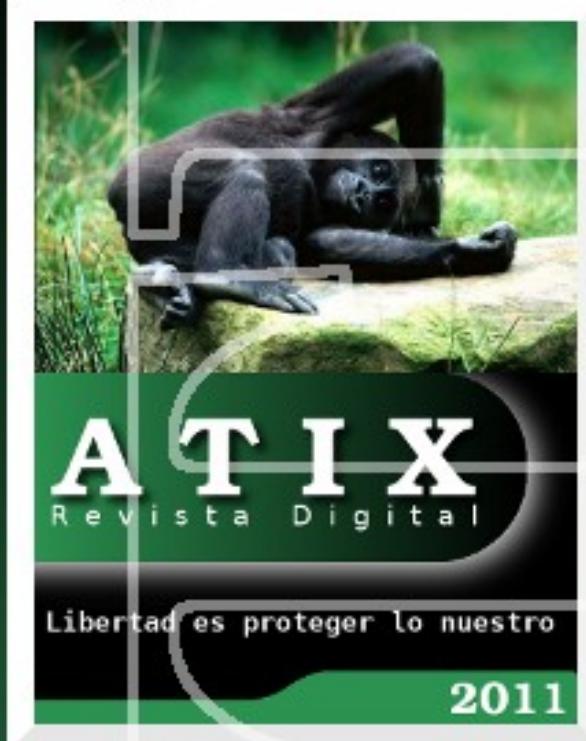


Esteban Saavedra López
Presidente Fundación AtixLibre

Contenido

Liberado el 28 de enero del 2011

- 7 Una alternativa libre para representar la información en la Web: Las ontologías
- 10 Generación de Certificados CSR con el Complemento Keymanager de Firefox
- 16 Entendiendo expresiones lambda en C# con Mono
- 22 Servidor de Mensajería Instantánea - Jabberd2
- 31 RIA Rich Internet Application
- 36 Introducción básica a Apache Ant: Alternativa a make para Java.
- 43 Willay news
- 49 Comics
- 50 Arte Libre
- 52 Información de contacto
- 53 Números anteriores
- 54 Fundación AtixLibre



Una alternativa libre para representar la información en la Web: Las ontologías

En la actualidad, las ontologías han ganado espacio en muchos de los campos de investigación de la informática como una alternativa eficaz de representación de la información y base para el razonamiento sobre el conocimiento almacenado. Por las facilidades que brindan, la sencillez de las metodologías para su desarrollo, las características de los lenguajes para su expresión y las herramientas para su elaboración, constituyen una alternativa al alcance de todos. El objetivo fundamental de este trabajo es presentar de forma resumida y rápida, las ventajas de las ontologías como una alternativa libre para la representación y el subsecuente razonamiento sobre el conocimiento.

Introducción

Desde mediados de los 90 y durante la primera década de este siglo XXI hemos sido testigos del auge de una nueva forma de representación del conocimiento: las ontologías. Aunque se tenga un concepto informático formal de una ontología como “una especificación explícita de una conceptualización” (Gruber, 1993), y que desde su surgimiento la idea inicial era lograr una homogenización entre los diferentes conceptos que manejaban personas y aplicaciones inteligentes en la Web (fundamentalmente), las ontologías han ganado espacio en muchos de los campos de investigación de la informática en la actualidad como una alternativa eficaz de representación de la información y base para el razonamiento sobre el conocimiento almacenado.

Desarrollo

Como base fundamental de la Web Semántica, las ontologías pretenden convertirse en una fuente de conocimiento accesible no solo a personas, sino también a aplicaciones inteligentes y motores de búsqueda en internet, de forma tal que la información disponible pueda ser enriquecida de manera semántica, lo que mejoraría en

gran medida el proceso de búsqueda en la red de redes.

Como formas de representación del conocimiento presentan una estructura sencilla (en su forma más simple) consistente en una taxonomía de clases que representan los conceptos asociados a un dominio específico. Se definen además, atributos y relaciones entre las clases, así como reglas o axiomas que rigen el comportamiento. En muchos casos, las ontologías cuentan con individuos o instancias de las clases especificadas para enriquecer el conocimiento almacenado. Existen clasificaciones asociadas a la estructura y el uso de las ontologías y además en dependencia de su formalidad. En un análisis un poco más minucioso, las ontologías se pueden clasificar usando disímiles criterios, como los explicados en (Ramos y Núñez, 2007).

Para la representación de ontologías se utilizan RDF (Resource Description Framework) y OWL (Ontology Web Language). RDF es un modelo estándar para la representación e intercambio de datos en la Web, basado en triples de la forma sujeto – predicado – objeto; para una forma sencilla de representación de la información y en URIs (Uniform Resource Identifier) para asociar información a cada elemento de la tripla. Un ejemplo pudiera ser: María disfruta

la comida italiana, en RDF se representaría de la forma: sujeto (María) - predicado (disfruta) - objeto (la comida italiana). Por supuesto utilizando los URIs se puede enriquecer esto con mucha más información. RDF permite una serialización a XML (W3C, 2004), debido a que es un modelo y no como tal un formato, lo que permite que la información almacenada pueda ser interpretada fácilmente como un XML por los sistemas.

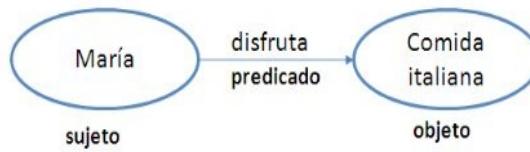


Figura. 1 Ejemplo de grafo RDF

Por otro lado, OWL (W3C, 2007) es un lenguaje pensado para ontologías Web (como sus siglas en inglés indican) y tiene una capacidad de representación semántica superior a RDF. Su creación se centra fundamentalmente en la idea de crear facilidades para la representación de información contenida en la WWW, de manera que sea mejor interpretada por las máquinas incorporando tareas de razonamiento sobre los datos.

Ambas representaciones, tanto RDF como OWL, están basadas en la capacidad de representación de XML, por lo que constituyen una alternativa viable y libre para el almacenamiento de datos en forma de ontologías. Para lograr este propósito, existen algunas herramientas libres que permiten el diseño y desarrollo de ontologías.

Entre esas herramientas se encuentra Protégé (actualmente en su versión 4.1 beta),

desarrollada en el Centro de Investigaciones de Informática Biomédica, en Stanford, California. Es un editor para ontologías, de código abierto, gratis que puede ser descargado online desde su sitio oficial, así como un conjunto de plug-ins para mejorar sus capacidades. Protégé permite el diseño de ontologías basadas en RDF, OWL y OWL 2, consultas SPARQL sobre los datos, razonamiento sobre datos utilizando numerosos razonadores para ontologías y brindando la posibilidad de integrar otros nuevos. Permite además integrar e importar ontologías existentes, integración de plug-ins para minería de texto, importar datos desde bases de datos y otros.

Conclusiones

Desde su propia concepción, las ontologías han perseguido el propósito de la disponibilidad de la información, la interoperabilidad entre aplicaciones y el acceso abierto y libre a recursos compartidos. Sus aplicaciones se extienden desde el área de la medicina, el comercio, los Sistemas de Información Geográfica, la Inteligencia Artificial, la aeronáutica y muchas otras áreas claves que necesitan alta disponibilidad de información y capacidad predictiva y de razonamiento sobre datos almacenados. Por las características de los lenguajes para su expresión y las herramientas para su desarrollo, constituyen una alternativa al alcance de todos. Entonces, cabe hacer la reflexión de si estamos realmente utilizando a cabalidad las potencialidades de esta nueva e interesante forma de representación y razonamiento sobre el conocimiento e información almacenada.

Referencias

- [1] Gruber., T. R., "A Translation Approach to Portable Ontologies." *Knowledge Acquisition*. 1993, pp. 199-220.
- [2] Ramos, Esmeralda and Núñez, Haydemar., ONTOLOGIAS: componentes, metodologías, lenguajes, herramientas y aplicaciones. Laboratorio de Inteligencia Artificial (LIA), Centro de Ingeniería de Software y Sistemas (ISYS). Caracas, Venezuela : s.n., 2007. Reporte técnico. ISSN 1316-6239.

- [3] RDF Primer. [Online] W3C, 2004. <http://www.w3.org/TR/2004/REC-rdf-primer-20040210/#rdfxml>].
- [4] Lenguaje de Ontologías Web (OWL). Vista General. [Online] W3C, 2007. <http://www.w3.org/2007/09/OWL-Overview-es.html>.

Autor



Yuniel Eliades Proenza Arias

Universidad de las Ciencias Informáticas. Ciudad de la Habana, Cuba.

yproenza@uci.cu



Generación de Certificados CSR con el Complemento Keymanager de Firefox

Para la generación de certificados CSR (Certificate Signing Request) utilizaremos el complemento KeyManager el cual nos brinda una interfaz grafica amigable, permitiendo acelerar el proceso de generacion de solicitudes de certificados.

Introducción

En el anterior numero de esta revista, en el articulo “**Instalación de Certificados Digitales en Apache2**”, vimos el procedimiento de como instalar un certificado digital en un servidor Apache2, para lo cual utilizamos la herramienta OpenSSL para generar el certificado CSR (Certificate Signing Request), en este articulo veremos la herramienta KeyManager la cual reemplaza a la herramienta OpenSSL.

El complemento KeyManager del navegador web FireFox es una herramienta para la generación de llaves, inscripción de certificados (certificate enrollment), firmas CRL, delegación de identidad y autoridad (identity and authority delegation).

Pasos para instalar el complemento

1. Abrir el navegador FireFox.
2. En el menú elegir la opción Herramientas, dentro la cual se debe seleccionar la opción Complementos como se muestra a continuación:

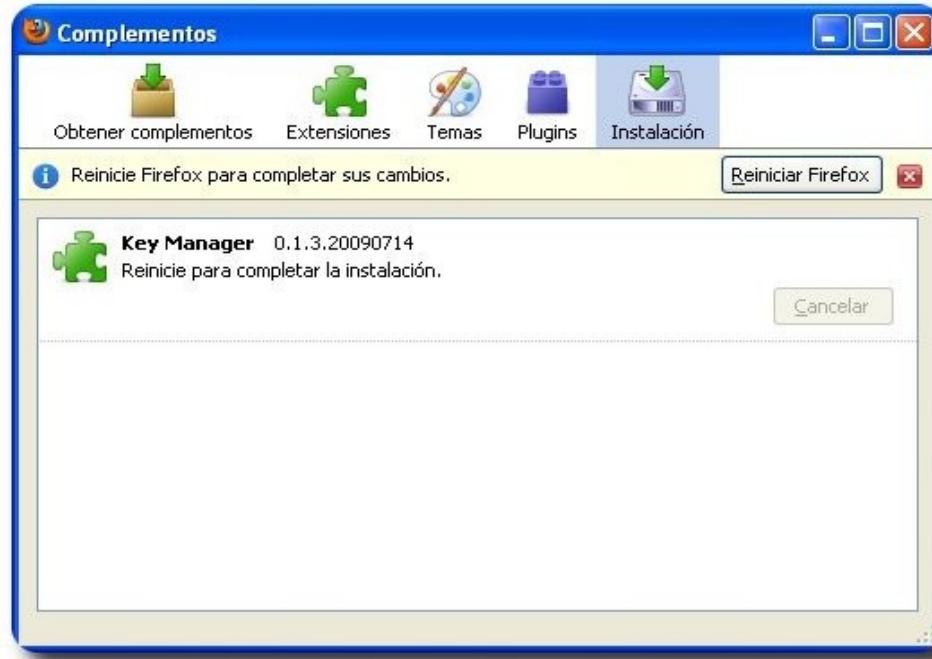


3. En la ventana de Complementos que se abrirá se debe buscar el complemento KeyManager e instalarlo:
- 4.

General

La auténtica libertad es sentirse en paz con uno mismo



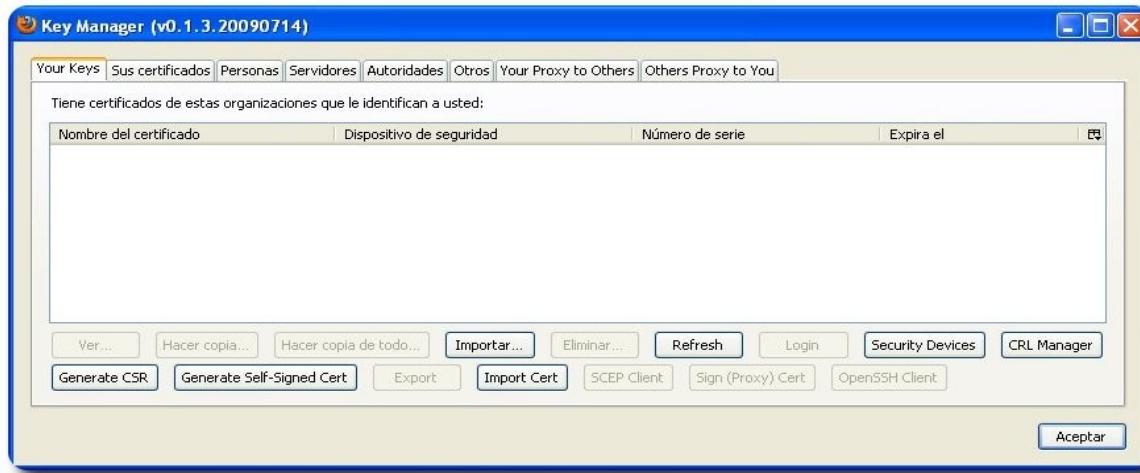


5. Cerramos la ventana de Complementos y tendremos en el navegador FireFox una barra de herramienta KeyManager como se muestra a continuación:



Procedimiento para generar un certificado CSR

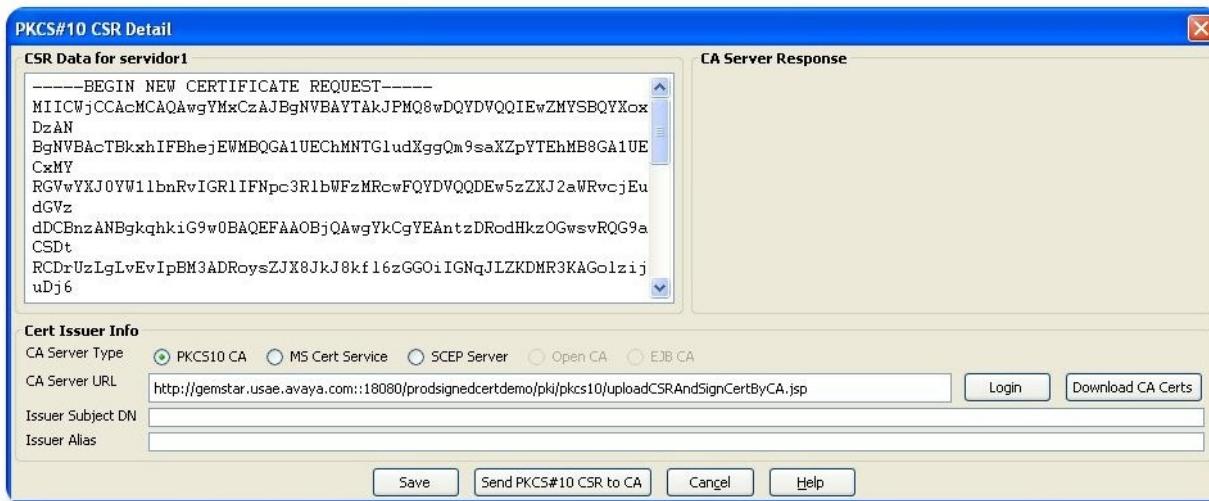
1. En la Barra de Herramientas seleccionamos KeyManager y se despliega la siguiente ventana:



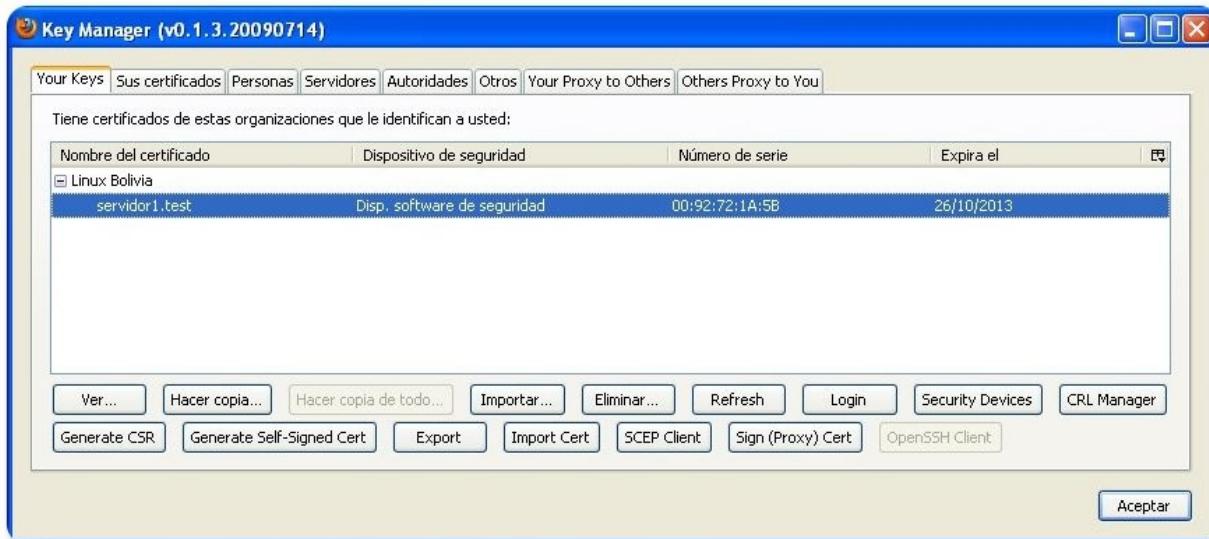
2. Elegimos el botón de Generate CSR y se abrirá la siguiente ventana, en la que se deben introducir los datos para el Certificado Digital:

A screenshot of the "Certificate Signing Request Form" dialog box. It has tabs for "Cert Basic Profile" and "Advanced". The "Default Certificate Profile" tab is selected, showing fields for "Common Name (cn)" (servidor1.test), "Organization Unit (ou)" (Departamento de Sistemas), "Organization (o)" (Linux Bolivia), "Locality (l)" (La Paz), "State (st)" (La Paz), "Country (c)" (BO), "Subject DN Suffix", "Subject (dn)" (cn=servidor1.test,ou=Departamento de Sistemas,o=Linux Bolivia,l=La Paz,st=La Paz,c=BO), and "Alias" (servidor1). Below this is the "Subject Public Key Info" section with "Key Type" set to RSA, "Sig Algorithm" to SHA1, and "Key Size" to 1024 (High Grade). The "Subject Alternative Name" section contains an "E-mail" field with the value renzomp@lycos.com. At the bottom are buttons for "Generate PKCS#10", "Generate CRMF", "Generate SPKAC", "Help", and "Cancel".

3. Una vez llenados los datos elegimos el botón Generate PKCS#10 y se generara el Certificado CSR:



4. Debemos grabar el Certificado CSR eligiendo el botón Save, después de grabado se vera en la consola principal el certificado CSR:



5. Este Certificado CSR debe ser enviado a una entidad emisora de certificados digitales CA (Certificate Authority). Existen varios CAs gratuitos como por ejemplo www.cacert.org ó www.openca.org. Debemos suscribirnos en estos sitios y cargar el CSR.
6. Una vez que el CA ha procesado la solicitud nos informará por correo electrónico y nos proporcionara el certificado digital, para nuestro ejemplo el archivo **servidor.crt**, el CA por lo general tarda de uno a dos días en procesar la solicitud.
7. Una vez que el certificado digital es enviado por la entidad emisora CA, continuaremos con la carga del certificado digital en el servidor Web instalado (ver artículo “**Instalación de Certificados Digitales en Apache2**” en el anterior número de esta revista).

Conclusiones

Con esta herramienta ahorramos tiempo en la generación de certificados CSR de una forma simple y utilizando una interfaz gráfica amigable.

Esta herramienta tiene muchas mas funcionalidades para facilitar la gestión de certificados digitales.

Referencias

- [1] <https://addons.mozilla.org/es-ES/firefox/addon/4471/>
- [2] <http://pubs.research.avayalabs.com/pdfs/ALR-2006-044.pdf>

Autor

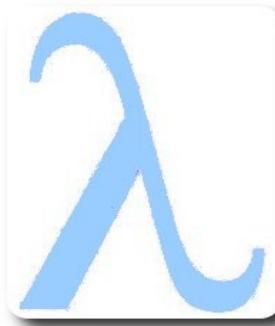


Renzo Martinez Pardo
Ingeniero Sistemas Electrónicos
renzomp@lycos.com



Entendiendo expresiones lambda en C# con Mono

Una expresión lambda es una función o una subrutina sin nombre que se puede utilizar donde haya un delegado válido. Las expresiones lambda pueden ser funciones o subrutinas y tener una sola línea o varias líneas. Puede pasar valores del ámbito actual a una expresión lambda.



Introducción

La programación imperativa, es uno de los paradigmas de computación mas ampliamente utilizados por la mayoría de lenguajes de programación de alto nivel; debido al gran soporte académico y comercial y a que los programas son relativamente independientes del tipo de computadora donde se ejecutan, porque los lenguajes de programación deben abstraer las operaciones del modelo de maquina para la cual se diseñaron.

La programación imperativa se basa en el modelo de la máquina de Von Newman, de la cual la mayoría de las computadoras personales y estaciones de trabajo tienen elementos comunes.

Aunque menos utilizado existe otro paradigma que a diferencia del imperativo se basa en las matemáticas (aplicación de funciones) con el cual igualmente podemos expresar operaciones computacionales de forma mas compacta y abstracta, este paradigma se conoce como programación funcional.

Uno de los muchos elementos del paradigma funcional que .NET incluye desde

la versión 3.0 son las expresiones lambda (lambda expression).

Programación Funcional

Los conceptos básicos de la programación funcional datan de 1956 a 1958 con el trabajo de Jonh McCarthy en el desarrollo y diseño de LISP (List Processor), este lenguaje esta basado en el cálculo lambda que sentó las bases de los lenguajes funcionales, características como:

- ✓ **Recursion:** se utiliza para realizar operaciones repetitivas, no utiliza la iteración.
- ✓ **Funciones de primer orden:** las funciones tienen el mismo nivel que cualquier otro elemento del lenguaje, pueden aplicarse a valores, evaluarse, regresar un valor y ser parámetros de otras funciones.
- ✓ **No requiere asignación:** el computó se realiza aplicando funciones a los argumentos.
- ✓ **Garbage collector:** Se reclaman los objetos que no están siendo utilizado por el programa.
- ✓ **Tipado dinámico (Dynamic typing):** La comprobación del tipo se realiza en tiempo de ejecución, los valores tienen tipos pero las variables no.

El paradigma funcional se basa en el concepto matemático de función, que la mayoría de los lenguajes de programación imperativos y funcionales comparten y cuya definición es

- (1) Una función f es una regla que asigna a

cada elemento x de un conjunto A exactamente un elemento llamado $f(x)$ de conjunto B

Donde la programación funcional marca su diferencia con la imperativa, es que para la programación funcional cada programa es equivalente a esta definición donde x es el argumento o dominio de f mientras que y es el rango de f o la salida sea los programas son cajas negras donde solo importa el que se está computando y no el como se está computando que es el caso de la programación imperativa.

En resumen cuando se programa de forma funcional se piensa más en expresiones y su significado que en una secuencia de operaciones en memoria.

Tipos Delegate y métodos anónimos

Desde sus primeras versiones .NET introdujo

En versiones anteriores a C# 2.0 (1.1,1.0) los delegate se utilizaban como en el siguiente listado:

Listado 1.1 Uso de métodos como parámetros en C# 1.1 using System;

```
namespace Samples
{
class Program
{
    //definimos al objeto que guardara las referencias a los métodos
    delegate double GetTemp(double d);
    static void Main(string[] args)
    {
        int x = 44;
        Console.WriteLine("{0} Fahrenheit = {1:0.00} Celsius", x, ApplyF(x,
Temp.GetCelsius));
        Console.WriteLine("{0} Fahrenheit = {1:0.00} Kelvin", x, ApplyF(x,
Temp.GetKelvin));
        Console.Read();
    }
    //el método que aplicará el método que es su segundo argumento
    static double ApplyF(double d,GetTemp f){
        return f(d);
    }
}
//la implementación de cada método
class Temp{
    public static double GetCelsius(double fahrenheit)
    {
        return (fahrenheit - 32) * (5 / 9D);
    }
}
```

el objeto delegate (delegado) que es un tipo particular de objeto (un delegate deriva de la clase base System.Delegate), que puede encapsular la referencia a un método estático o de una instancia como si fuera un mecanismo de callback(devolución de llamada) similar a los apuntadores de función de C y C++ pero con la importante diferencia de que proporciona un tipado seguro (type-safety) para evitar errores en tiempo de ejecución y que puedan detectarse en tiempo de compilación si la función no coincide con la firma del método al que hace referencia.

Esto posibilita en un contexto de programación orientada a objetos que los métodos pueden recibir como argumentos otros métodos además de tipos primitivos y de referencia.

Veamos un ejemplo para ilustrar estos conceptos con C#.

```
public static double GetKelvin(double fahrenheit)
{
    return fahrenheit + 460;
}
```

Aquí observamos que los métodos que implementan la funcionalidad deben declararse de una manera completamente procedural e imperativa.

```
public static double GetCelsius(double fahrenheit)
{
    return (fahrenheit - 32) * (5 / 9D);
}

public static double GetKelvin(double fahrenheit)
{
    return fahrenheit + 460;
}
```

C# 2.0 al incorporar los métodos anónimos, se acerca más a la programación funcional al asociar un bloque de código a un delegate sin necesidad de tener toda su implementación en un método sino dentro de la misma declaración del objeto, como mostramos en el siguiente listado que es el listado anterior pero usando métodos anónimos.

Listado 1.2 Métodos como parámetros utilizando métodos anónimos.

```
using System;

namespace Samples
{
class Program
{
delegate double GetTemp(double d);
static void Main(string[] args)
{
    int x = 44;
    Console.WriteLine("{0} Fahrenheit = {1:0.00} Celsius", x, ApplyF(x, delegate(double
fahrenheit) {
        return (fahrenheit - 32) * (5 / 9D);
}));
    Console.WriteLine("{0} Fahrenheit = {1:0.00} Kelvin", x, ApplyF(x, delegate(double
fahrenheit) {
        return fahrenheit + 460;
}));
    Console.Read();
}

static double ApplyF(double d, GetTemp f)
{
    return f(d);
}
}
}
```

Aquí observamos la diferencia con respecto al código anterior del listado 1.1.

```
Console.WriteLine("{0} Fahrenheit = {1:0.00} Celsius", x, ApplyF(x, delegate(double fahrenheit) {
    return (fahrenheit - 32) * (5 / 9D);
}));
Console.WriteLine("{0} Fahrenheit = {1:0.00} Kelvin", x, ApplyF(x, delegate(double fahrenheit) {
    return fahrenheit + 460;
}));
```

Expresiones Lambda (Lambda Expressions)

Las expresiones lambda provienen del cálculo lambda (lambda calculus) desarrollado por Alonzo Church en los años 1930's como una notación para representar todas las funciones computables equivalentes a una máquina de Turing, todos los lenguajes funcionales pueden ser vistos como una variante sintáctica del cálculo lambda.

Las expresiones Lambda son útiles para sintetizar funciones con pocos parámetros que regresan algún valor, esta expresión consiste básicamente en una regla de sustitución que expresa tal cual una función o sea un mapeo de los elementos del conjunto dominio a los elementos de un codominio por ejemplo en la siguiente expresión:

```
cuadrado : integer → integer donde cuadrado(n) = n2
```

se reduce a una notación que produce una función anónima donde los únicos símbolos son la letra lambda (λ) y el punto (.)

```
 $\lambda n . n^2$ 
```

En la mayoría de los lenguajes funcionales las funciones anónimas son valores representados por la palabra reservada lambda, como el caso de LISP.

Lambda (n)(**n)

Aunque C# no utiliza los símbolos de la notación matemática lambda, el operador lambda es => que significa "tiende a" o "va hacia a", la estructura de una expresión lambda en C# es:

```
(Argumentos de entrada) => (salida al procesarlos)
```

En caso de únicamente una variable la sintaxis es:

```
(x) => (x*x)
```

En caso de múltiples argumentos la sintaxis es:

```
(x, y, z) => ()
```

En este punto es importante saber que es el tipo delegate que dicta el tipo de los parámetros de entrada y de salida.

Listado 1.3 Métodos como parámetros utilizando expresiones lambda

```
using System;

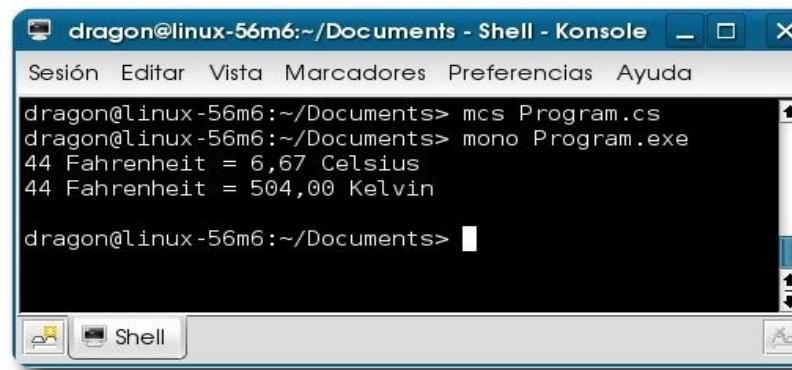
namespace Lambdas
{
    class Program
    {
        delegate double GetTemp(double d);
        static void Main(string[] args)
        {
            int x = 44;
            Console.WriteLine("{0} Fahrenheit = {1:0.00} Celsius", x, ApplyF(x,
(fahrenheit) => ((fahrenheit - 32) * (5 / 9D))));
            Console.WriteLine("{0} Fahrenheit = {1:0.00} Kelvin", x, ApplyF(x, (fahrenheit)
=> (fahrenheit + 460)));
            Console.Read();
        }

        static double ApplyF(double d, GetTemp f)
        {
            return f(d);
        }
    }
}
```

Podemos observar que de los métodos anónimos a las expresiones Lambda, nos queda una sintaxis más legible y compacta.

```
(fahrenheit) => ((fahrenheit - 32) * (5 / 9D))
(fahrenheit) => (fahrenheit + 460))
```

El resultado de la ejecución del programa es el mismo con cada uno de los listados.



Conclusión

Para los nuevos retos en el desarrollo de software, es importante que los lenguajes de programación incorporen características de un paradigma de programación diferente a ellos para extender sus capacidades y así poder expresar algoritmos de una manera compacta y más concisa esto da como resultado un código más legible.

Los ejemplos pueden ser descargados de <http://xomalli.blogspot.com/>

Este documento está protegido bajo la licencia de documentación libre Free Documentacion License del Proyecto GNU, para consulta ver el sitio <http://www.gnu.org/licenses/fdl.txt>, toda persona que lo desee está autorizada a usar, copiar y modificar este documento según los puntos establecidos en la «Licencia FDL»

Autor



I.S. Martín Márquez

xomalli@gmail.com



Servidor de Mensajería Instantánea – Jabberd2

Jabber es un protocolo extensible, abierto y estándar basado en XML para el intercambio en tiempo real de mensajes y presencia entre dos puntos en Internet.



Introducción

La comunicación es uno de los pilares básicos de cualquier empresa, más aún con el avance de las Tecnologías de Información y Comunicación (TIC), nuestros clientes y usuarios exigen tener la información de manera inmediata. Una de las soluciones a este requerimiento es un servidor de mensajería.

Este servidor debe cumplir con los estándares, condiciones y adecuación a nuestros requerimientos, surgiendo Jabberd que conforma una federación de mensajería instantánea en la que todo el mundo puede hablar entre sí sin restricción alguna y lo mas importante LIBRE.

Este pequeño artículo pretende introducirnos en la instalación, configuración y administración de Jabberd2 en una LAN, el cual puede ser implementado de forma personalizada, cumpliendo los requerimientos de nuestra empresa.

¿Que es Jabberd?

Jabber es un protocolo abierto y libre para mensajería instantánea, que está basado en

XML y con núcleo XMPP, para el intercambio en tiempo real de mensajes y presencia entre dos puntos en una red, utilizando el puerto 5222 preferentemente.

¿Para que nos sirve?

Como su nombre lo dice nos permite enviar mensajes instantáneos a los usuarios registrados en la LAN, permitiendo enviar mensajes cuando el destinatario no está conectado, así como poder tenerlo abierto en varios programas (clientes jabberd) a la vez, permitiendo asignar prioridades. Este sistema permite, por ejemplo, asignar prioridad uno a la aplicación que tienes en la oficina y dos en la casa, lo que permitira que los mensajes lleguen a la oficina mientras estés conectado, caso contrario se enviarán a la aplicación de tu casa.

El servidor Jabberd2 soporta SSL, ipv6, transferencias de archivos (comentar que en algunas pruebas se pasó incluso imágenes ISO completas) que puedes parar y reanudar en cualquier momento.

Soporta salas de chat, interoperabilidad entre distintos servidores, usuarios ilimitados y muchas más particularidades. Empresas y servicios de internet ya lo usan, como iChat, tan solo deberemos de elegir un servidor y un cliente y ya nos podemos conectar. Esto permite poder elegir libremente una plataforma y con ella un programa los que cada vez son más numerosos.

Características

Jabberd2 presenta las siguientes características:

- ✓ **Protocolo abierto:** Como todo software libre, se puede programar un servidor o un cliente a medida o ver el código y modificarlo, entre otras cosas.
- ✓ **Descentralizado:** Se puede crear un servidor para Jabber, y se puede interoperar o unirse al resto de la red Jabber.
- ✓ **Extensible:** Se puede ampliar con mejoras sobre el protocolo original. Las extensiones comunes son manejadas por la XMPP Standards Fundation.
- ✓ **Seguro:** Cualquier servidor Jabber está aislado del exterior. El servidor de referencia permite SSL para comunicaciones cliente-servidor y algunos clientes aceptan GPG como cifrado de las comunicaciones usando cifrado asimétrico.
- ✓ **Multiredes:** Un transporte o pasarela permite comunicarse con otros protocolos usados por clientes como Windows Live Messenger, ICQ, AOL o Yahoo.
- ✓ **Salas de conversación:** Conocido como Multi-User Chat. Es una de las extensiones que han sido añadidas a la mensajería Jabber, la cual le permite la creación de grupos de debate como en las redes IRC, con la posibilidad de poseer usuarios con distintos privilegios (moderadores, participantes e invitados), iniciar conversaciones privadas y transferir archivos.

Que precisamos

Necesitamos instalar las librerías de **libgc.so**, **libgsasl.so**, **libntlm**, **libudns.so** y **libmysqlclient.so**; así también **mysql-server** para el almacenamiento y **openldap** para la **autenticacion**. Jabberd2 es un servicio que puede ser instalado en diferentes distribuciones Linux, mediante sus respectivos gestores de paquetes:

- ✓ Debian y Ubuntu con **apt-get**
- ✓ RHEL, CentOS y Fedora con **yum**.

Configuración Inicial

Necesitamos identificar a nuestro servidor, con un IP y puerta de enlace válido dentro de nuestra red y un nombre en formato FQDN (Fully Qualified Domain Name), que es un nombre que incluye el nombre de la computadora y el nombre de dominio asociado a ese equipo, en nuestro caso será **jabberd2.atix.net**.



```
root@jabberd2:~#
[root@jabberd2 ~]# ifconfig 192.168.1.1 netmask 255.255.255.0 broadcast 192.168.1.255 &&
> route add default gw 192.168.1.1 &&
> hostname jabberd2.atix.net
```

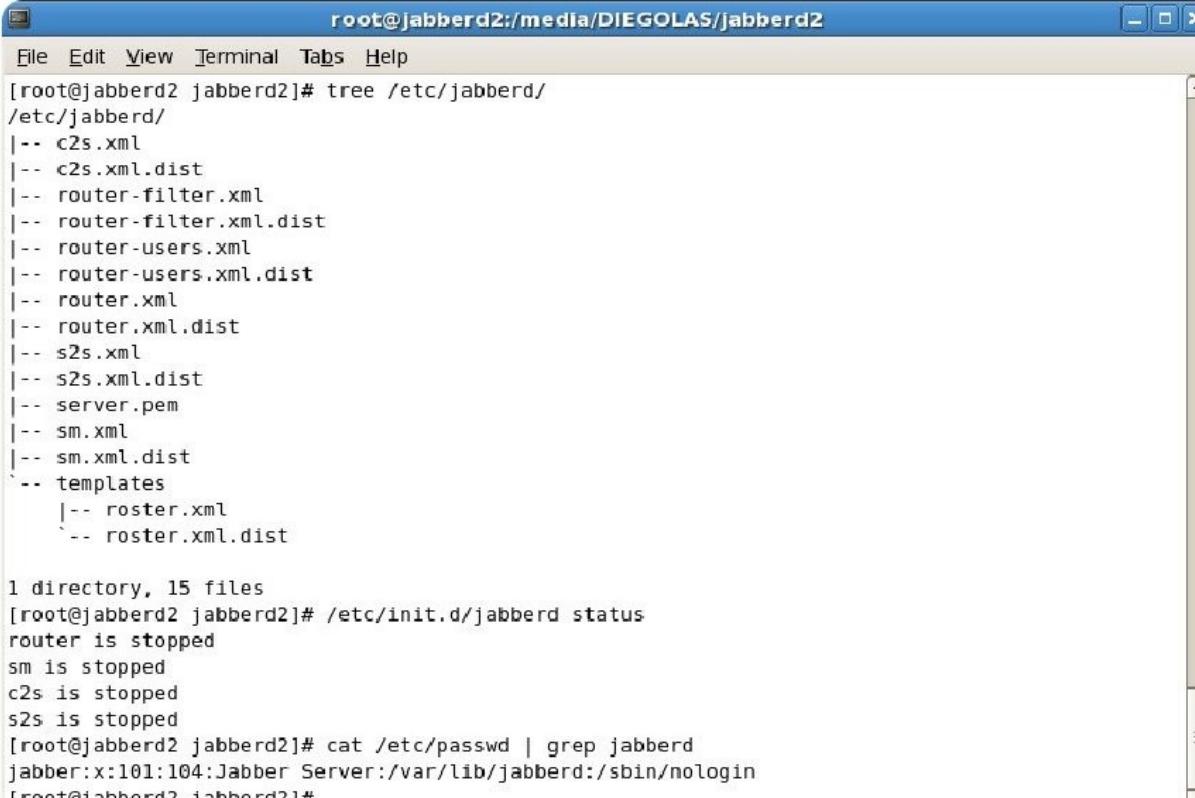
Instalación

Ahora nos queda instalar el paquete **jabberd2**, si es que tenemos acceso a internet lo podemos hacer con **apt-get install jabberd2** en distribuciones basadas en Debian, y **yum install jabberd2** en distribuciones basados en Red Hat, en mi caso no tengo acceso a internet y solo tengo el paquete **.rpm** y lo instalamos.



```
root@jabberd2:/media/DIEGOLAS/jabberd2
File Edit View Terminal Tabs Help
[root@jabberd2 jabberd2]# rpm -ivh jabberd-2.2.8-2.el5.i386.rpm
warning: jabberd-2.2.8-2.el5.i386.rpm: Header V3 DSA signature: NOKEY, Key ID 217521f6
Preparing... ################################ [100%]
1:jabberd ################################ [100%]
[root@jabberd2 jabberd2]#
```

Verificamos si se crearon los archivos, el servicio y el usuario correspondiente



```
root@jabberd2:/media/DIEGOLAS/jabberd2
File Edit View Terminal Tabs Help
[root@jabberd2 jabberd2]# tree /etc/jabberd/
/etc/jabberd/
|-- c2s.xml
|-- c2s.xml.dist
|-- router-filter.xml
|-- router-filter.xml.dist
|-- router-users.xml
|-- router-users.xml.dist
|-- router.xml
|-- router.xml.dist
|-- s2s.xml
|-- s2s.xml.dist
|-- server.pem
|-- sm.xml
|-- sm.xml.dist
`-- templates
    |-- roster.xml
    `-- roster.xml.dist

1 directory, 15 files
[root@jabberd2 jabberd2]# /etc/init.d/jabberd status
router is stopped
sm is stopped
c2s is stopped
s2s is stopped
[root@jabberd2 jabberd2]# cat /etc/passwd | grep jabberd
jabber:x:101:104:Jabber Server:/var/lib/jabberd:/sbin/nologin
[root@jabberd2 jabberd2]#
```

Almacenamiento a MYSQL

Jabberd2 nos presenta varias alternativas de base de datos como mysql, oracle, postgresql y otros, para almacenar a los contactos, grupos, datos personales y otros, en nuestro ejemplo lo vamos a tener en mysql. Para lo cual necesitamos crear una base de datos, crear tablas y crear un usuario con los privilegios correspondientes.

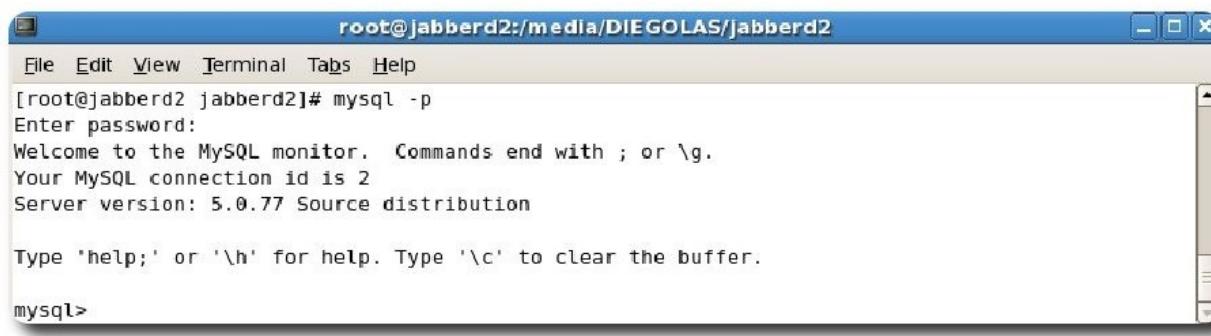
En la instalación de Jabberd2 incorpora scripts de creación de base de datos y tablas:



```
root@jabberd2:/media/DIEGOLAS/jabberd2
File Edit View Terminal Tabs Help
[root@jabberd2 jabberd2]# tree /usr/share/jabberd/
/usr/share/jabberd/
|-- db-setup.mysql
|-- db-setuppgsql
|-- migrate.pl
`-- pipe-auth.pl

0 directories, 4 files
[root@jabberd2 jabberd2]#
```

Nos conectamos a mysql con un usuario con privilegios de poder crear base de datos, en mi caso root:

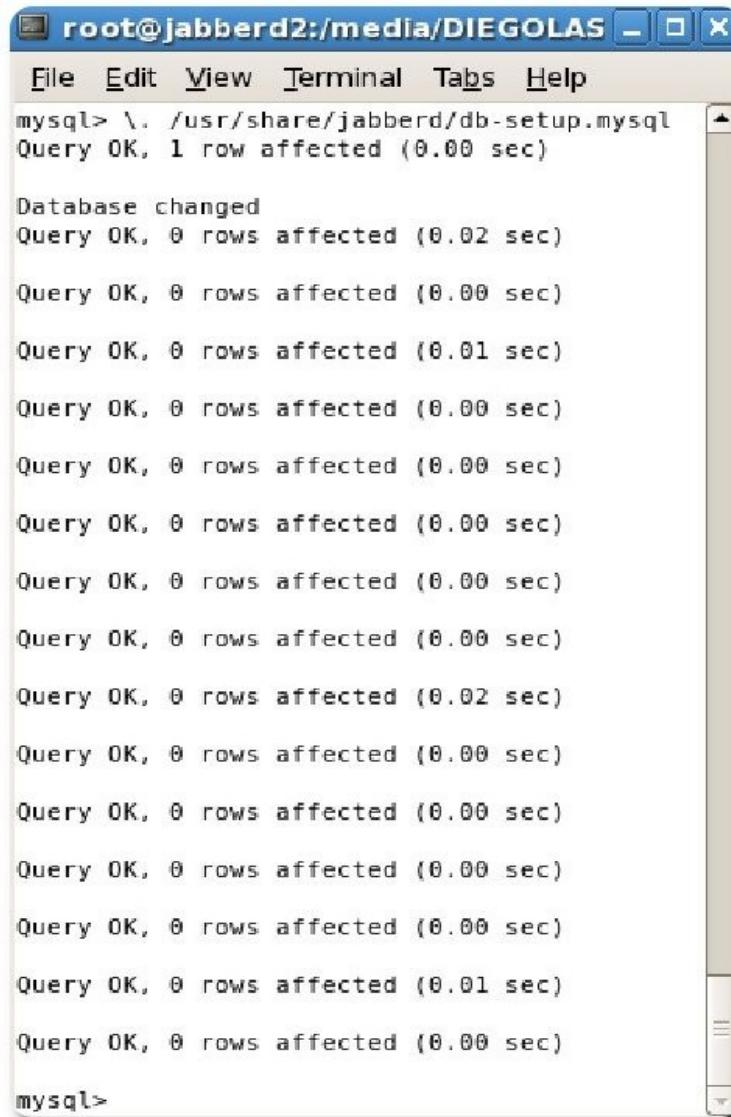


```
root@jabberd2:/media/DIEGOLAS/jabberd2
File Edit View Terminal Tabs Help
[root@jabberd2 jabberd2]# mysql -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.0.77 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

Aquí, hacemos correr el script que corresponde a mysql como sigue:



```
root@jabberd2:/media/DIEGOLAS -
File Edit View Terminal Tabs Help
mysql> \. /usr/share/jabberd/db-setup.mysql
Query OK, 1 row affected (0.00 sec)

Database changed
Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected (0.00 sec)

mysql>
```

Damos privilegios al usuario jabberd2 para la base de datos jabberd2.

```
root@jabberd2:/media/DIEGOLAS/jabberd2
File Edit View Terminal Tabs Help
mysql> GRANT select,insert,delete,update ON jabberd2.* to jabberd2@localhost IDENTIFIED by 'secret';
Query OK, 0 rows affected (0.00 sec)

mysql>
```

Verificamos la creación de la base de datos

```
root@jabberd2:/media/DIEGOLAS/jabberd2
File Edit View Terminal Tabs Help
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| jabberd2 |
| mysql |
| test |
+-----+
4 rows in set (0.00 sec)

mysql>
```

Ahora estamos listos para comenzar la primera configuración del servidor de mensajería instantánea Jabberd2.

Configuración con MYSQL

Se usarán un par de archivos de configuración XML para configurar Jabberd2, de manera que la autenticación y almacenamiento de usuarios sea con mysql.

1. Edite `/etc/jabberd/sm.xml` cambiando `<id>localhost.localdomain</id>` con el nombre de anfitrión u otra id de su servidor, en nuestro ejemplo será `<id>jabberd2.atix.net</id>`. Ahora en la sección `<storage>` cambie el `<driver>db</driver>` a `<driver>mysql</driver>` con esto le indicamos a Jabberd2 que el almacenamiento sera a la base de datos mysql; ahora verificamos que en la sección `<storage>` este habilitado `<auto-create>/>` para que usuarios nuevos ser registrados sin la necesidad de estar en el almacenamiento mysql, esta puede ser estar o no en la política de creación de usuarios, todo depende de la empresa. Como configuramos el almacenamiento a mysql le indicaremos en al sección `<mysql>` el usuario y contraseña `<user>jabberd2</user><pass>secret</pass>`, con esto terminamos con el archivo `sm.xml`.
2. Edite `/etc/jabberd/c2s.xml` cambiando `<id>localhost.localdomain</id>` con el nombre de anfitrión u otra id de su servidor, habilitando el registro de los usuarios, en nuestro ejemplo será `<id register-enable='true'>jabberd2.atix.net</id>`. Ahora en la sección `<authreg>` cambie el `<module>db</module>` a `<module>mysql</module>` con esto le indicamos a Jabberd2 que la autenticación será a la base de datos mysql. Como configuramos el almacenamiento a mysql le indicaremos en al sección `<mysql>` el usuario y contraseña

<user>jabberd2</user><pass>secret</pass>, con esto terminamos con el archivo c2s.xml.

Con estos pasos tenemos nuestra primera configuración de Jabberd2.

Inicio de Servicio y configuracion de un cliente jabberd

Seguimos los siguientes pasos:

1. Iniciamos nuestro servidor, asegurando que el servicio mysql esta arriba.



```
root@jabberd2:/media/DIEGOLAS/jabberd2
File Edit View Terminal Tabs Help
[root@jabberd2 jabberd2]# /etc/init.d/mysql status
mysql (pid 4030) is running...
[root@jabberd2 jabberd2]# /etc/init.d/jabberd start
Initializing jabberd processes ...
Starting router: [ OK ]
Starting sm: [ OK ]
Starting c2s: [ OK ]
Starting s2s: [ OK ]
[root@jabberd2 jabberd2]#
```

2. Instalamos un cliente jabbers para Linux como PSI, si es que tenemos acceso a internet lo podemos hacer con **apt-get install psi** en distribuciones basadas en Debian, y **yum install psi** en distribuciones basadas en Red Hat, en mi caso no tengo acceso a internet y solo tengo el paquete .rpm y lo instalamos.



```
root@jabberd2:/media/DIEGOLAS/jabberd2
File Edit View Terminal Tabs Help
[root@jabberd2 jabberd2]# rpm -ivh psi-0.10-4.el5.rf.i386.rpm
warning: psi-0.10-4.el5.rf.i386.rpm: Header V3 DSA signature: NOKEY, key ID 6b8d79e6
Preparing... ################################ [100%]
1:psi ################################ [100%]
[root@jabberd2 jabberd2]#
```

3. Iniciamos nuestro psi en entorno gráfico desde **aplicaciones-> Internet->Psi**



4. Indicamos el nombre del perfil con el cual vamos a trabajar, habilitamos el registro de usuario y le damos adicionar (**add**).
5. Se llenan los datos de la cuenta a ser registrada con su respectiva clave y le damos registrar (**register**).



6. Se confirma el registro satisfactorio y le damos aceptar (**ok**).



7. Una vez registrado, debemos guardar los datos de la cuenta en el perfil de psi y le damos guardar (**save**).



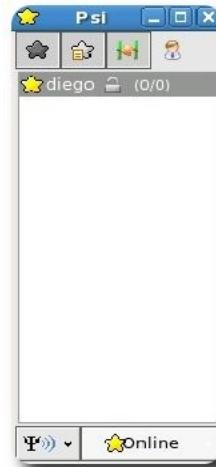
8. Así configuramos psi para un usuario nuevo, por defecto estamos fuera de línea (**offline**), nos queda ponerlo en linea (**online**), para utilizar el servicio e ingresar los datos del usuario.



9. Una vez introducidos nuestros datos le damos publicar (**publish**).



10. Se confirma el almacenamiento de los datos de usuario y le damos aceptar (**ok**).



Como vemos ya estamos en línea usando el servidor recién instalado. Hasta el momento solo hemos invertido aproximadamente 20 minutos en la configuración del servidor de mensajería instantánea Jabberd2 que se encuentra ya en producción.

Ahora tenemos Jabberd2 funcionando con mysql.

En el siguiente número de la revista veremos mas sobre la configuración e incorporaremos la autentificación con LDAP y otros trucos más, hasta la siguiente.

Bibliografía

- [1] <http://www.jabberes.org/>
- [2] <http://codex.xiaoka.com/wiki/jabberd2:start>

Autor



Diego Parraga Menchaca
parragdiego@gmail.com



RIA

Rich Internet Application

Las aplicaciones RIA son un nuevo tipo de aplicaciones con más ventajas que las aplicaciones Web tradicionales. Estas surgen como una combinación de las ventajas que ofrecen las aplicaciones Web y las aplicaciones tradicionales (aplicaciones de escritorio).

Adicionalmente se encuentran las capacidades multimedia donde no hace falta ningún reproductor del Sistema Operativo del usuario.



Introducción

Los valiosos beneficios que las aplicaciones Ricas de Internet (RIA) representan para las empresas ya han sido ampliamente documentados y son claramente reconocidos; desde un bajo costo total de propiedad, hasta la disponibilidad total de la aplicación, mayor seguridad y una experiencia del usuario mejorada.

Contextos de implementación de RIA

Inicialmente las RIA estaban enfocadas a dos posibles entornos:

- ✓ **Usar una Plataforma como servicio (PasS):** Con ejemplos como Force.com, Google, Microsoft, Amazon y LongJump. Si esta fuera la opción elegida, entonces deberá considerar que si bien todas las partes del sistema son provistas por

usted, la plataforma está todavía en algún lugar en la “Nube”.

- ✓ **Usar una plataforma de tipo cliente:** Para empresas con cierta resistencia a ubicar sus activos en la Nube, la única forma de adquirir capacidades para construir RIA es hacerlo por su propia cuenta empleando alguna de la variedad de herramientas de tipo Plataforma de Cliente RIA, como Adobe Flex/Air, Java FX o por medio de algún framework Javascript.

Que obtener con RIA

- ✓ **Menor costo total de propiedad:** Con RIA no hay necesidad de instalar software del lado del Cliente ni ocuparse de mantener a cada nuevo usuario. La aplicación se configura automática y transparentemente en el lado del Cliente y toda la aplicación y las actualizaciones de la misma y así como de la plataforma se llevan a cabo en el servidor .
- ✓ **Estar listo para SaaS :** Las aplicaciones SaaS representarán el 25% de todo negocio de software. RIA es un componente esencial en el desarrollo de una propuesta exitosa de Software como Servicio (SaaS).
- ✓ **Mejor experiencia del usuario y productividad :** A diferencia de las aplicaciones de Internet típicas, las RIA no requieren una comunicación Cliente-Servidor para cada dato procesado y cada actividad. La

- ✓ **aplicación corre más rápido y más agradablemente,** los empleados hacen su trabajo más rápido y los clientes reciben un mejor servicio.
- ✓ **Mas disponibilidad de la aplicación:** Las RIA pueden ejecutarse remotamente desde cualquier parte y pueden accederse en cualquier momento vía un amplio abanico de dispositivos móviles portátiles. Los usuarios pueden acceder a los datos necesarios para trabajar desde su propia casa o desde fuera de la empresa.
- ✓ **Mejor seguridad corporativa y protección de derechos de propiedad intelectual :** La arquitectura multicapa de RIA le permite ocultar los elementos sensibles de su aplicación y evitar el robo corporativo, el abuso del activo, clientes y el malware.

Características básicas

- ✓ Facilidad en la conectividad
- ✓ Distribución de la aplicación entre cliente y servidor
- ✓ Acceso local o remoto
- ✓ Inteligencia en la interfaz

Características clave

- ✓ **Accesibilidad:** AJAX nativo en los navegadores web y es el único RIA framework que puede ser encontrado por los diferentes motores de búsqueda.
- ✓ **Comunicaciones avanzadas:** con servidores que soporten nuevas tecnologías se puede mejorar la experiencia del usuario al utilizar protocolos de red optimizados y entradas y salidas asíncronas. Se requiere de una conexión de banda ancha confiable
- ✓ **Complejidad:** las soluciones avanzadas puede ser más difíciles de diseñar, desarrollar, implementar y

depurar que las aplicaciones web tradicionales

- ✓ **Consistencia:** la interfaz de usuario y las experiencias pueden ser controladas por el sistema operativo, el monitoreo del rendimiento y diagnóstico de errores puede tornarse difícil.
- ✓ **Instalación y mantenimiento:** no se precisa instalación y su mantenimiento es automático
- ✓ **Offline:** puede ser soportada reteniendo el estado en la máquina cliente
- ✓ **Seguridad:** se mejora la seguridad por medio de actualizaciones automáticas y "sandbox". Pero las extensiones por si mismas son sujetas a vulnerabilidades
- ✓ **Rendimiento:** puede ser mejorado dependiendo de la aplicación y de las características de la red. Aplicaciones que pueden procesar localmente en el cliente evitando viajes hacia el servidor lo que incrementa su rendimiento. Dándole más trabajo al cliente también se puede incrementar el rendimiento del servidor. Por el contrario los recursos necesarios hacen difícil que las aplicaciones puedan ejecutarse en dispositivos pequeños, móviles o embebidos
- ✓ **Riqueza:** añadiendo características que no son nativas en los navegadores web como captura de vídeo

Beneficios

- ✓ No necesitan instalación (solo es necesario mantener actualizado el navegador web).
- ✓ Las actualizaciones hacia nuevas versiones son automáticas.
- ✓ Se pueden utilizar desde cualquier ordenador con una conexión a Internet sin depender del sistema operativo que este utilice.
- ✓ Generalmente es menos probable la

infección por virus, que utilizando por ejemplo programas ejecutables.

- ✓ Mayor capacidad de respuesta, ya que el usuario interactúa directamente con el servidor, sin necesidad de recargar la página.
- ✓ Ofrecen aplicaciones interactivas que no se pueden obtener utilizando solo HTML, incluyendo arrastrar y pegar, cálculos en el lado del cliente sin la necesidad de enviar la información al servidor.
- ✓ Evita la problemática del uso de diferentes navegadores al abstraerse de ellos a través de un framework.

Que puede hacer RIA

- ✓ Emular el comportamiento de aplicaciones de escritorio
- ✓ Variedad de efectos
- ✓ Manejo de datos del lado del cliente
- ✓ Lógica del negocio del lado del cliente
- ✓ Redefinición de la metáfora "Pagina"
- ✓ Emulación del comportamiento de aplicaciones de escritorio
 - ✓ Arrastrar y soltar
 - ✓ Menús y barra de herramientas
 - ✓ Menús contextuales
 - ✓ Ventanas y wizards
 - ✓ Paneles
 - ✓ Arboles
 - ✓ Validación de formularios
 - ✓ Acciones del teclado
 - ✓ Controles No HTML
 - ✓ Acordeones
 - ✓ Combobox
 - ✓ Spinner box
 - ✓ Sliders
 - ✓

Arquitectura

Generalmente se tiene una aplicación cliente stateful y una capa de servicios separada. RIA se apoya más sobre un desarrollo cliente-servidor en vez de un desarrollo web tradicional, en donde el estado se mantiene en el servidor en sesiones. El cliente sabe acerca de si mismo y el tipo de datos que esta solicitando y únicamente solicita los datos que necesita sin ninguna otra información.

- ✓ **Cliente:** Se maneja la interacción entre el usuario y la interfaz del usuario, el usuario invoca comandos, actualiza vistas y carga datos. Aquí se mantiene el estado de la aplicación, se manejan todas las peticiones de datos hacia el servidor y se controla como se presentan los datos.
- ✓ **Servicios:** Aquí se manejan y se procesan todas las peticiones de la aplicación cliente y delega las acciones en el servidor, estas pueden ser, guardar datos en la base de datos, actualizar los archivos del sistema, retornar datos al servidor, o algún tipo de proceso analítico. Determina y le da formato a los datos que son retornados al cliente.

Tecnologías

A continuación enumeramos algunas de las tecnologías utilizadas por las RIA:

- ✓ DHTML
- ✓ HTML
- ✓ Javascript
- ✓ CCS
- ✓ AJAX
- ✓ XML
- ✓ Flash, Extensiones de los browsers
- ✓ Java
- ✓ ActiveX
- ✓ XUL
- ✓ CURL

Frameworks JScript

Dojo



Rico



Prototype



YUI



Qooxdoo



Backbase



GWT



ZK



Ext Js



JavaFx



Script.aculo.us



OpenLaszlo



Algunas aplicaciones en la web



Autor



Esteban Saavedra López

Presidente de la Fundación AtixLibre (Bolivia)

Activista de Software Libre en Bolivia

jesaavedra@opentelematics.org

esteban.saavedra@atixlibre.org

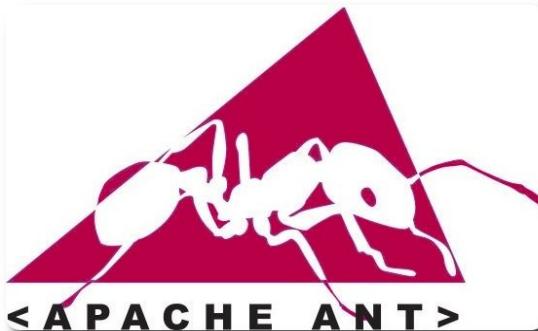
<http://jesaavedra.opentelematics.org>

<http://www.atixlibre.org>



Introducción básica a Apache Ant: Alternativa a make para Java.

Apache Ant es una herramienta open source y cross platform desarrollada en Java para facilitar el proceso de compilación de aplicaciones Java. El trabajo que realiza Ant es muy parecido al que realiza la herramienta GNU make con C/C++ para facilitar la compilación de proyectos con la ventaja de ser más fácil de usar.



Introducción

Erase una vez un joven programador que se aventuró en el mundo Java. En sus principios y como lo manda el reglamento de todo buen programador tecleo su primer **Hello World!**. Desde una consola tecleaba **javac MiClase.java** y después con un solo **java MiClase** su aplicación posaba frente a él sin mayor problema. Pasa el tiempo y aquel joven programador que escribía solo pequeños programas se aventura cada vez más en la construcción de aplicaciones que se vuelven complejas cada vez más y más. Este programador del que les hablo se da cuenta de que sus aplicaciones se hacen cada vez más grandes y eso es bueno, pero al mismo tiempo ve que amontonar todas sus clases en un mismo directorio convierten su entorno de trabajo en un verdadero desorden; archivos **.java**, **.class**, bibliotecas externas y demás componentes que usa en sus aplicaciones.

En un intento por poner orden a su trabajo el programador del que les hablo primeramente aprende a organizar su código mediante el uso de paquetes, en sus respectivos

directorios, un directorio para las bibliotecas externas, otra para los archivos **bytecode(.class)**, etc. Y efectivamente, ya hay algo de orden. Sin embargo, este orden tiene un costo y ese es que el proceso de compilación se vuelve muy complicado, porque tenemos que pasar muchos argumentos al compilador que indiquen la ubicación de los recursos. Y por si fuera poco este no es el problema, imaginen tener que teclear a mano toda una gran linea de comandos para compilar nuestro proyecto cada vez que deseemos probar un cambio en nuestra aplicación, por más pequeño que éste sea.

He aquí el problema. Y para eso esta Ant, para ayudarnos en esta tarea.

¿Qué es apache Ant?

Ant es una herramienta open source que permite compilar, ensamblar, probar y ejecutar aplicaciones Java. Ant además es una herramienta cross platform(plataforma cruzada), lo cual significa que trabaja de forma idéntica en diferentes plataformas. Si bien inicialmente fue desarrollada para Java ésta herramienta también puede extenderse a otros lenguajes. Ant fue desarrollada inicialmente(ya no más) por James Duncan Davidson cuando trabajaba en Sun Microsystems en el 2001 y posteriormente donada a la Apache Software Foundation la cual mantiene el desarrollo activo hasta el día de hoy. La versión más reciente es la 1.8.1, sin embargo, en este documento se empleará la versión 1.8.0.

Instalación de Ant

Antes que nada, es necesario instalar los paquetes necesarios para poder emplear Ant. En mi caso estoy utilizando debian y únicamente es necesario una línea en la consola:

```
# apt-get install ant
```

Este proceso idéntico en ubuntu y derivados de debian así que no hay gran complicación. Ahora, para distribuciones basadas en paquetes RPM tales como openSuse, o fedora con una línea como esta sera suficiente:

```
# yum install ant
```

Por supuesto, siempre podemos descargar el paquete desde su pagina Web en <http://ant.apache.org>. En este sitio tenemos la opción de descargar el código fuente o bien el paquete binario, además de la documentación y otros recursos de apoyo. La instalación desde código fuente es muy simple y no requiere mayor instrucción, desde el paquete binario aun más.

```
package ant.example;

import java.awt.BorderLayout;
import java.awt.FlowLayout;
import javax.swing.JOptionPane;
import javax.swing.JFrame;
import javax.swing.JLabel;

public class HelloAnt extends JFrame{
    private JLabel lblPresentacion;
    public HelloAnt()
    {
        super( ".:Presentacion:" );
        setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
        setSize(600, 100);
        setLayout( new FlowLayout() );
        inicializaComponentes();
        setVisible(true);
    }
    public void inicializaComponentes()
    {
        lblPresentacion = new JLabel();
        String nombre = JOptionPane.showInputDialog(
                "Su nombre aqui por favor:"
        );
    }
}
```

Desarrollo de una aplicación en Java

Para ver como trabaja Ant y cuales son las ventajas de trabajar con ella a continuación se elabora un pequeño programa que primeramente se construye siguiendo la metodología tradicional y posteriormente salen a relucir las virtudes de Ant en este mismo proceso. Cabe mencionar que el programa no pretende ser de utilidad, solo como ejemplo.

Para nuestro proyecto crearemos el directorio Presentacion, src, ant, example, build, classes según la la estructura que se muestra en la siguiente imagen.

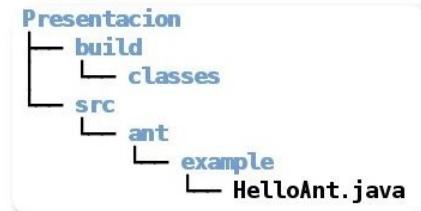


Gráfico 1. Estructura de directorios de nuestro proyecto.

Pasamos el siguiente código a algún editor de texto(kate, gedit, vim, geany, etc.) lo guardamos en el directorio src.

```

        String mensaje = "<html><h3> Hola " +
                        "<font color=\"blue\"><i>" +
                        nombre + "! </i></font>" +
                        "bienvenido a tu primer proyecto con Ant. " +
                        "En ATIX, por supuesto!." +
                        "</h3></html>";
        lblPresentacion.setText( mensaje );
        add(lblPresentacion);
    }

    public static void main(String[] argumentos)
    {
        HelloAnt helloAnt = new HelloAnt();
    }
}

```

Código 1. Aplicación de ejemplo.

Compilación y ejecución

Ya tenemos nuestro proyecto y ahora vamos a compilarlo. Desde una consola nos ubicamos en el directorio raíz del proyecto y ejecutamos el siguiente comando:

```
$ javac -sourcepath src/ -d build/classes src/ant/example/HelloAnt.java
```

Sin todo sale bien ahora seremos capaces de ejecutar la aplicación de la siguiente forma:

```
$ java -cp build/classes/ ant.example.HelloAnt
```

Algo parecido a esto nos debe aparecer.

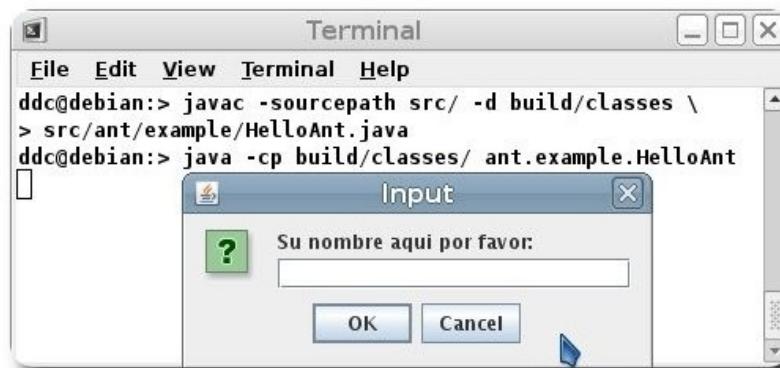


Gráfico 2. Comparación y ejecución de la aplicación.

Empaquetar aplicación

Hasta ahora todo bien, nuestra aplicación compila y se ejecuta correctamente, sin embargo cuando se escriben aplicaciones es muy probable que los usuarios finales, a quienes va dirigido el software, no le interese, o aun peor, le desagrade tener que ejecutar la aplicación tal cual lo hemos hecho nosotros en la línea de comandos, el usuario final desea obtener su software, dar un clic y tener su aplicación trabajando, sin importarle las clases y demás componentes que la integran. Ante esta situación vamos a tratar de dar solución a este problema empaquetando nuestro programa, lo cual nos permite tratar la aplicación como un único objeto autocontenido.

Sin muchos detalles creamos el directorio jar dentro build y un archivo llamado Manifest y dentro de él tecleamos lo siguiente:

```
Main-Class: ant.example.HelloAnt
```

Ahora ejecutamos:

```
$ jar cfm build/jar/HelloAnt.jar Manifest -C build/classes/ .
```

```
$ java -jar build/jar/HelloAnt.jar
```

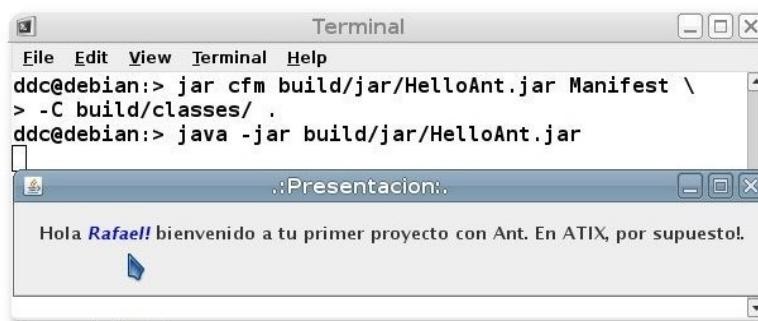


Gráfico 3. Ejecutando la aplicación desde un .jar.

Hasta ahora ya hemos echo todo lo necesario para construir nuestra aplicación. Si bien esto no es mucho ya que estamos tratando con un proyecto muy muy pequeño la verdad es que los pasos para realizar el mismo proceso con aplicaciones grandes son un poco mas engorrosos y repetitivos.

Ant en acción

Ahora veamos como realizar el mismo proceso que hicimos anteriormente pero ahora con Ant. La forma de trabajar de Ant consiste en analizar un archivo XML y realizar las tareas que ahí se especifican. El archivo esta compuesto de objetivos(targets) y tareas(tasks).

Targets

Un target es un contenedor de tareas que colabora para alcanzar un estado deseado durante el proceso de construcción(build).

```
<target name="targetName">
    <!-- Aqui las tareas -->
    .
    .
</target>
```

Tasks

Una tarea es una pieza de código que puede ser ejecutada. Una tarea puede tener multiples atributos o argumentos. La estructura común de una tarea es la siguiente:

```
<nombre atributo1="valor1" atributo2="valor2" ... />
```

Bien. Ya que vimos sencillamente como se estructura el archivo veamos como quedaría para nuestro proyecto.

Por cierto, el nombre del archivo puede ser el que ustedes gusten pero por defecto Ant intenta abrir un archivo con nombre build.xml por lo cual no necesitan indicarlo si su archivo se llama exactamente así.

```
<project>
    <target name="clean">
        <delete dir="build"/>
    </target>

    <target name="compile">
        <mkdir dir="build/classes"/>
        <javac srcdir="src" destdir="build/classes"/>
    </target>

    <target name="jar">
        <mkdir dir="build/jar"/>
        <jar destfile="build/jar/HelloAnt.jar" basedir="build/classes">
            <manifest>
                <attribute name="Main-Class" value="ant.example.HelloAnt"/>
            </manifest>
        </jar>
    </target>

    <target name="run">
        <java jar="build/jar/HelloAnt.jar" fork="true"/>
    </target>
</project>
```

Código 2. Archivo build.xml para nuestro proyecto.

Ahora cada vez que necesiten probar los cambios de su proyecto bastara con 3 líneas:

```
$ ant build
$ ant jar
$ ant run
```

Tres comandos que bien pueden reducirse a uno solo que además de reducir el número de comandos también reduce problemas en archivo build.xml, tales como tener que recordar el nombre de los directorios y archivos durante la escritura del mismo.

```
<project name="HelloAnt" basedir=". " default="main">

    <property name="src.dir"      value="src"/>
    <property name="build.dir"   value="build"/>
    <property name="classes.dir" value="${build.dir}/classes"/>
    <property name="jar.dir"     value="${build.dir}/jar"/>

    <property name="main-class"  value="ant.example.HelloAnt"/>

    <target name="clean">
        <delete dir="${build.dir}"/>
    </target>

    <target name="compile">
        <mkdir dir="${classes.dir}"/>
        <javac srcdir="${src.dir}" destdir="${classes.dir}"/>
    </target>

    <target name="jar" depends="compile">
        <mkdir dir="${jar.dir}"/>
        <jar destfile="${jar.dir}/${ant.project.name}.jar"
            basedir="${classes.dir}">
            <manifest>
                <attribute name="Main-Class" value="${main-class}"/>
            </manifest>
        </jar>
    </target>

    <target name="run" depends="jar">
        <java jar="${jar.dir}/${ant.project.name}.jar" fork="true"/>
    </target>

    <target name="clean-build" depends="clean,jar"/>

    <target name="main" depends="clean,run"/>
</project>
```

Con este nuevo código tenemos un poco mas de legibilidad ya que las referencias a todos los elementos las definimos en un principio y después se accede a ellas mediante el identificador que se específico y no con el valor exacto en cada ocasión.

Desde la consola ejecutar ant sin argumentos y ella hará todo. Y bueno, eso es todo, espero les sirva. Hasta la próxima.

Conclusión

Los proyectos de software son complicados no solo por las complejidad del problema que abordan sino también por el esfuerzo que cuesta mantenerlos y construirlos. Apache Ant es la herramienta open source que nos ayuda con la segunda cuestión. Apache Ant es una herramienta muy extensa por lo que aquí solo hemos incluido una mera introducción a esta misma. Para profundizar en el tema una buena fuente es la documentación que incluye Ant.

Referencias

- [1] <http://ant.apache.org/manual/tutorial-HelloWorldWithAnt.html>
- [2] Documentación de Apache Ant 1.8.0.

Autor



Rafael Rendón Pablo

Estudiante de Ingeniería en Sistemas Computacionales

Usuario de software libre

smart.dyr@gmail.com

www.infoscience.260mb.com



willay news

Información Actual

Willay news

Tecnología que permite hacer llover



Con los últimos cambios climáticos, los científicos se han dedicado a realizar inventos y hacer uso de la tecnología para producir algunos cambios en el clima, un ejemplo de ello se produjo en el verano pasado (invierno para el hemisferio sur) que fue realmente lluvioso en Oriente Medio. De hecho, fue demasiado lluvioso e inusual para un desierto.

Ahora una compañía llamada **Meteo Systems** ha declarado que ellos fueron los causantes de la lluvia debido al uso de un sistema secreto de control del tiempo que se basa en una tecnología desarrollada por Nikola Tesla en 1890.

La compañía usó unas torres con forma de paraguas que miden 9 metros que disparan iones negativos a la atmósfera cuando la humedad supera el 30%. Los iones atraen partículas de polvo, las cuales atraen moléculas de agua, y luego ¡puf! Ya tenemos nubes grises y lluvia.

El método de “fabricación” de lluvia fue usado en cinco sitios cercanos a Abu Dhabi y **Meteo Systems** se lleva el crédito por 52 aguaceros.

Diseño automotriz futurista



Peter Vardai y Tamas Fodor han sido los encargados de la creación del automóvil conceptual que vemos en la imagen, el fantástico **Blackbird** Mercedes, el cual sin dudas tiene fuerte inspiración en la tecnología vista en **Tron**, aunque obviamente no cumpliría las mismas funciones.

Vardai ha dicho que la forma está basada en los aerodinámicos de un avión y también que sus colores son negro y plateado, aunque en las imágenes solamente podemos ver el negro y las luces de color.

Sin dudas se trata de uno de los autos más sobresalientes, y como muchos de ustedes, esperamos que salga de la fase conceptual pronto y entre en producción.

Tinta para camuflar tanques de guerra



BAE Systems es una firma británica que se especializa en darle algunas de las más geniales y locas ideas al ejército. Basándose en diseños conceptuales que se cree pueden ser convertidos en realidad en un corto período de tiempo, la nueva idea de camuflar tanques podría llevar a la tinta electrónica a un nuevo nivel.

Lo necesario son unos cuantos sensores y un tanque cubierto de la citada tinta. Los sensores tomarían lecturas de los alrededores, incluyendo terreno, colores y otra información que sería enviada a la tinta, la cual la proyectaría sobre el tanque, dejándolo camuflado.

Desafortunadamente para **BAE Systems**, la tecnología necesaria para convertir a esto en una realidad es simplemente demasiado costosa, y sobretodo para usarla en el campo de batalla.

Baby Keyboard



Baby Keyboard es un software para bebés que evitara que éstos borren tus archivos mientras estén usando el teclado de la computadora. Tu bebé podrá usar el teclado libremente sin causar ningún tipo de problema.

Cuando el pequeño le da a una tecla o clickea el mouse aparecerá en pantalla una imagen o una palabra con su respectivo sonido. El programa también puede pasar las imágenes automáticamente, solo hace falta esperar. Obviamente otra cosa que se puede hacer es ajustar el tiempo de uso del teclado por parte de tu bebé.

Willay news

Control Paterno



Sin duda alguna muchos padres están preocupados por tener un control sobre la actividad de sus hijos cuando estos están en Internet, pues bien ahora mismo tenemos varios programas que permiten hacer esto, uno de los últimos es **Facebook Spy Monitor**, que es un software de monitoreo y control paterno. Se dedica a registrar lo que hacen los usuarios de **Facebook** en el PC que está instalado, incluyendo perfiles, chat, mensajes privados, contactos y eventos e incluso de vez en cuando hace una captura de pantalla. También registra las búsquedas online que el usuario realiza sobre **Facebook**, clientes instantáneos de **Facebook** y sitios web asociados.

Podrás ver exactamente lo que el usuario ha hecho, como si estuvieras viendo un vídeo grabado por una cámara de seguridad.

Salesforce compra Dimdim



Siguiendo al efecto **Facebook** en la manera en la que la gente colabora y se relaciona, el servicio de computación en nube **Salesforce.com** ha anunciado la adquisición de la firma de comunicaciones en tiempo real **Dimdim**.

La compra se ha llevado a cabo por una cantidad de 31 millones de dólares.

Con ésta adquisición, **Salesforce.com** espera darle a la plataforma de comunicación de la compañía "la misma colaboración integrada y experiencia que ha hecho de **Facebook** la página más popular de internet".

Chatter, la plataforma de **Salesforce.com**, se presentó en junio de 2010 y lleva acumulados 60,000 usuarios desde ese momento.

La compra y la mejora que hay planificada para Chatter forma parte de lo que **Salesforce.com** cree el paso al servicio en nube 2, el siguiente avance de la computación en nube, que será más social, móvil y en tiempo real, y que pretende sorprender a propios y extraños.

Fundada en 2007 por DD Ganguly y Prakash Khot, Dimdim ofrece una plataforma web que permite a los usuarios organizar o visitar reuniones online, demostraciones y seminarios web desde su navegador de internet

Google Summer of Code (GSoC)



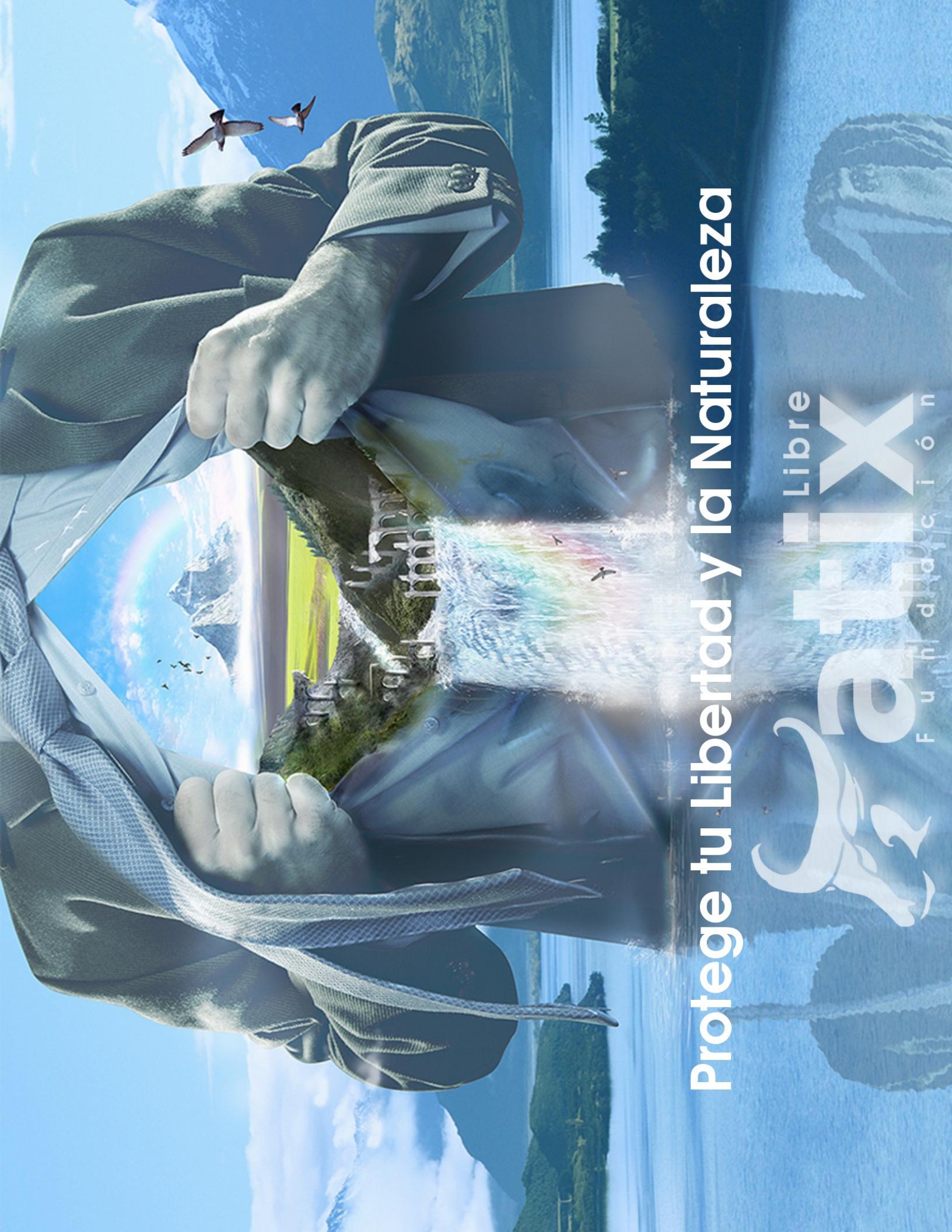
Autor



Jenny Saavedra López
Diseño y Edición Revista Atix
jenny.saavedra@atixlibre.org

A través de Google Summer of Code, los solicitantes aceptan a estudiantes emparejados con un mentor o mentores de los proyectos participantes, ganando así la exposición a los escenarios del mundo real de desarrollo de software y la oportunidad para el empleo en áreas relacionadas con sus actividades académicas. A su vez, los proyectos participantes son capaces de identificar con mayor facilidad y traer a los nuevos desarrolladores. Lo mejor de todo, más código fuente se ha creado y puesto en libertad para el uso y beneficio de todos.

Será el 7º año consecutivo que se organiza. No hace falta que repita qué es el SoC. Los pasados 6 años han sumado en total 4,500 estudiantes de más de 85 países distintos, o sea un éxito total y un gran esfuerzo por parte de Google para fomentar el Software Libre. Este año Google aumentará en un 25% el número de estudiantes y organizaciones aceptadas.



Protege tu libertad y la Naturaleza

Libre
natur
Fundaación



Comics

Curiosidades



Si tenemos una moneda y la intercambiamos,
cada uno quedara con una moneda.



Si tenemos una idea y la intercambiamos,
cada uno quedara con dos ideas.



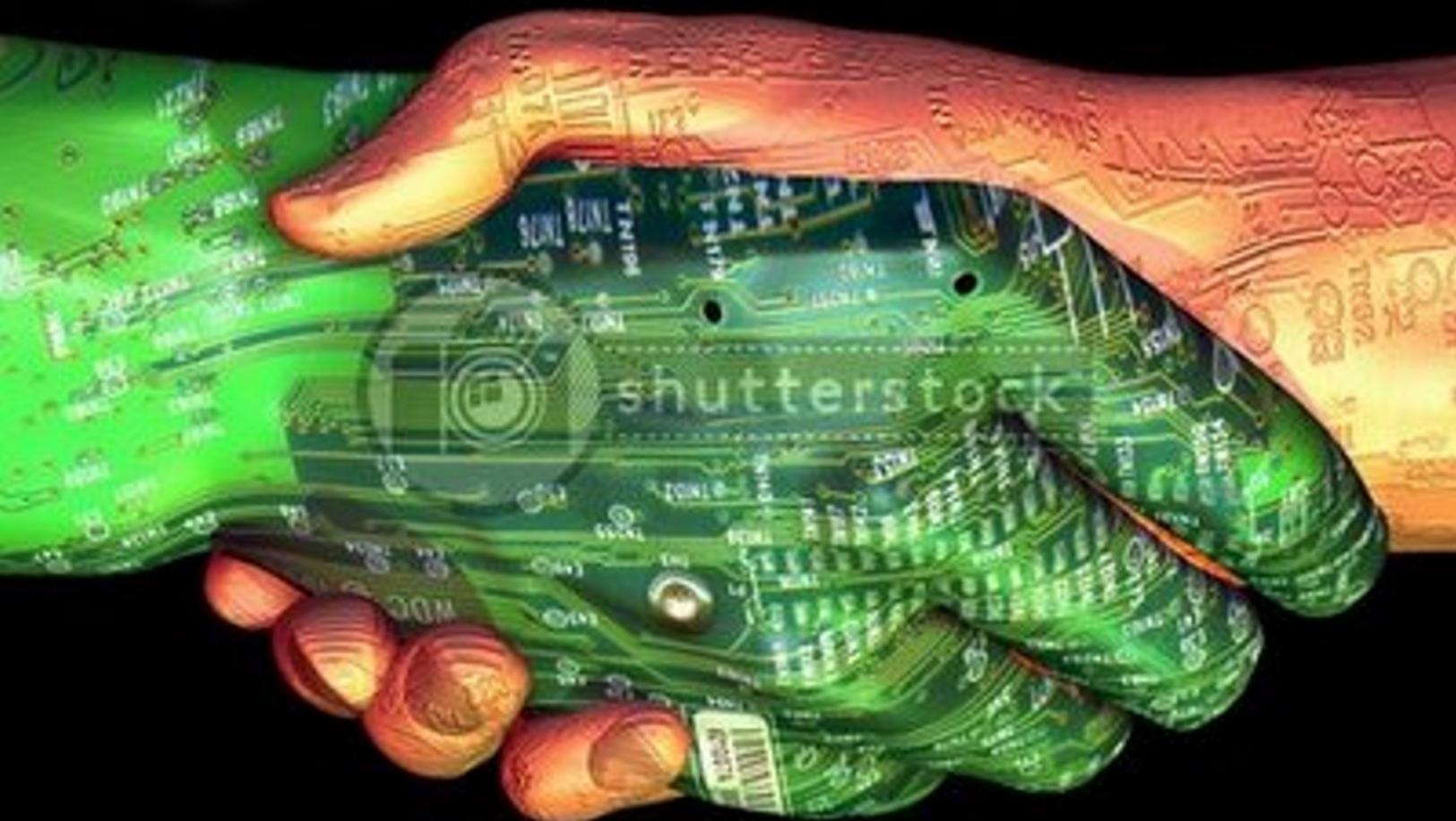
Libres para pensar, libres para decidir, libres para crear

Arte Libre

Te ofrecemos este espacio para mostrar tu Creatividad



Envíanos tus diseños y creaciones para publicarlos



De la mano con la Tecnología

Consultas

Contacto

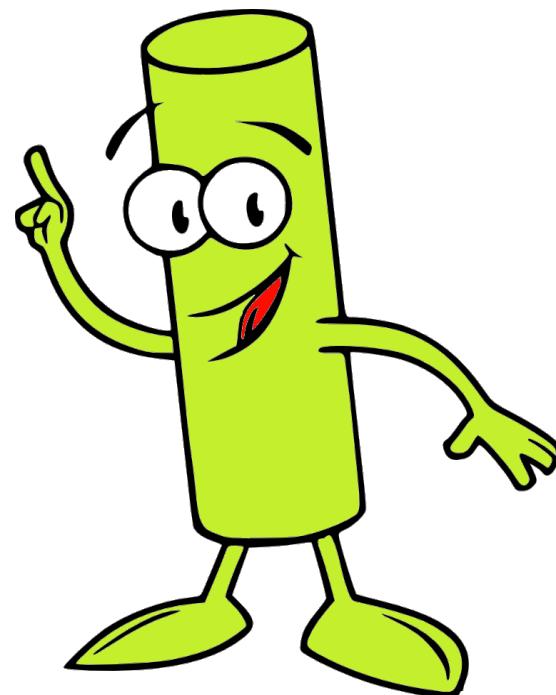
Para solicitar cualquier información, puedes contactar a:

- ✓ Esteban Saavedra López (esteban.saavedra@atixlibre.org)
- ✓ Jenny Saavedra (jenny.saavedra@atixlibre.org)

Publicación

Te invitamos a ser parte de la **Revista ATIX**. La forma de participar puede ser enviándonos:

- ✓ Artículos referidos a áreas como:
 - ✓ Instalación y personalización de Aplicaciones
 - ✓ Scripting
 - ✓ Diseño gráfico
 - ✓ Programación y desarrollo de aplicaciones
 - ✓ Administración de servidores
 - ✓ Seguridad
 - ✓ y cualquier tema enmarcado dentro del uso de Software Libre
- ✓ Trucos y recetas.
- ✓ Noticias.
- ✓ Comics.
- ✓ Links de interés.



Bitacora

Empezamos a registrar nuestra historia



Nuestros números anteriores

Hacia un Futuro Innovador



<http://www.atixlibre.org>

Por un Mundo Ético, Libre y Justo



Por un Mundo Ético, Libre y Justo