



**2 Vistos desde otros horizontes**

**Revista de Software Libre ATIX 2009**

## 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.

## Dirección y Coordinación General

Esteban Saavedra López (jesaavedra@opentelematics.org)

## Diseño y Maquetación

Jenny Saavedra López (jennysaavedra@gmail.com)

Esteban Saavedra López (jesaavedra@opentelematics.org)

## Revisiones

Esteban Saavedra López

Jenny Saavedra López

Lucy Ayarde Romero

## Noticias

Ivonne Menacho

Jenny Saavedra López

Marcia Velasquez

## Autores Frecuentes

Arnold Guzmán

Carlos Ramos

Ernesto Rico Smith

Esteban Saavedra López

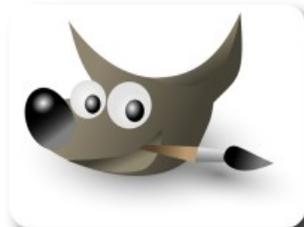
Joseph Sandoval

Lucy Ayarde Romero

Williams Chorolque Choque

## 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**

Cuando uno empieza un nuevo proyecto, generalmente éste es conocido sólo por los miembros del mismo, o por personas del contexto donde se desenvuelve el proyecto, pero al pasar el tiempo, puede ser que el interés que denote éste, haga que día a día, más y más personas se vean inmiscuidas, ya sea como colaboradoras directas o indirectas.

**Vistos desde otros horizontes**, un título que refleja de forma clara, que cuando uno es visto desde otros contextos, se va retroalimentando de los aportes, sugerencias y críticas que podrían generar estos, aspecto que decantará en el fortalecimiento y en el progreso de la actividad o actitud que uno tiene.

Es así que desde su inicio la **Revista Atix** se ha venido nutriendo de comentarios, sugerencias, críticas y actualmente de aportes de quienes comparten nuestro objetivo, revisan y valoran nuestro trabajo.

En éste octavo número ponderamos la presencia de varios autores de fuera de nuestras fronteras, aprovechamos de darles una cordial bienvenida, seguros estamos que sus contribuciones serán de gran valor para nuestros lectores.

## **El ser vistos desde otros horizontes, nos compromete más en nuestro trabajo.**

Bienvenidos a nuestro octavo número

**Esteban Saavedra López**  
Director y Coordinador General

# Contenido

Liberado el 9 de marzo de 2009

- 7 Depuración sobre entornos gráficos
- 15 Desarrollo Ágil con Ruby on Rails (3ra Parte)
- 20 Diseño Avanzado de Aplicaciones Web EXT, Zend Framework y Doctrine
- 32 Grails: Framework para el desarrollo de aplicaciones Web (1ra Parte)
- 42 Gestión de Certificados Digitales con OpenSSL (2da parte)
- 52 CentOS Directory Server en CentOS 5
- 62 Willay news
- 70 Comics
- 71 Conociendo lo nuestro - Turismo y Libertad
- 74 Arte Libre
- 76 Información de contacto
- 77 Números anteriores



# Depuración sobre entornos gráficos

El proceso de depuración es una de la etapas del desarrollo de software en el que se invierte más tiempo. Se tiene la fama de que la programación sobre entornos \*nix es complicada, que todo es pantalla negra con letritas blancas con un monton de términos raros y cosas así por el estilo. Si bien gran parte de esto es cierto, veremos que tan fácil puede resultar el proceso de depuración en una interfaz gráfica.



## Introducción

Antes que nada sean bienvenidos a este minitutorial, el cual marca mi primera participación en este grandioso proyecto llamado ATIX. En consecuencia también quisiera agradecer al equipo de ATIX la oportunidad de pertenecer al mismo.

Como se menciona antes, el proceso de depuración de software consume la mayor parte del tiempo que empleamos en crear software. Por lo anterior es fundamental saber realizar de manera adecuada y precisa este proceso. Para realizar esta tarea existen programas que se encargan de analizar programas, los famosos depuradores o debuggers los cuales nos permiten inspeccionar el comportamiento de nuestros programas en cada paso de ejecución y por ende detectar las fallas que pueda presentar.

Los depuradores no son exclusivos de ninguna plataforma o sistema operativo, existen depuradores para muchos lenguajes de programación y para muchas plataformas y sistemas operativos. Los sistemas \*nix fueron concebidos con el propósito de la

investigación y el desarrollo de de software, por lo tanto, estos sistemas nos brindan un conjunto de herramientas para el desarrollo de aplicaciones, y por supuesto, herramientas para la depuración de las mismas. En este documento trabajaremos sobre el sistema GNU/Linux y las herramientas que este nos brinda.

En números anteriores se abordo ya el tema de la depuración por otros integrantes de la revista, más sin embargo se enfoco sobre la depuración orientada a la consola con el depurador GDB. He aquí el por qué de este documento, en esta ocasión veremos como depurar programas en un entorno gráfico, para lo cual nos apoyaremos del IDE (Integrated Development Environment) Anjuta y de un programita escrito en lenguaje C. Sin mas que decir, pasemos a lo que vinimos.



Gráfico 1. Logo del IDE Anjuta.

## ¿Qué es la depuración?

En términos sencillos podríamos definir a la depuración como una etapa del desarrollo de software en la cual se intenta encontrar y

reparar los errores que puedan presentar los programas. Cabe decir que la depuración se aplica a los programas que ya han pasado el proceso de compilación (para lenguajes compilados) y no antes de eso, ya que aquí el error sería por parte del programador o por factores externos. A pesar de que la depuración es una buena herramienta para solucionar errores en los programas, no se recomienda utilizar demasiado este proceso, es mejor aprender a diseñar buenos algoritmos y solo recurrir a la depuración como último recurso.



**Gráfico 2.** Eliminando bugs

## Lenguaje C

En este documento trabajaremos con un programa escrito en lenguaje C. El motivo de esto es porque en la mayoría de los casos es el primer lenguaje con el que se inicia un usuario recién llegado los sistemas GNU/Linux. Es uno de los lenguajes más difundidos y mas populares en todo el mundo. El núcleo Linux esta escrito en su mayoría en este lenguaje, por lo que me parece prudente utilizar este lenguaje para nuestro ejemplo. Para aquellos que no manejen este lenguaje les recomiendo hacerlo ya que para aquellos que estén interesados realmente en la programación, por ejemplo, el núcleo Linux, este lenguaje sera su pan de cada día (y un poco de assembly claro:-).

## El IDE Anjuta

Anjuta es un Entorno de Desarrollo Integrado para el lenguaje C/C++ sobre sistemas GNU/Linux (también sobre los \*BSD). Ha sido escrito para el entorno de escritorio GNOME, más sin embargo es posible utilizarlo desde cualquier otro entorno de escritorio, tales como KDE, XFCE, entre otros. Es un IDE muy amigable para el usuario ya que provee un conjunto de facilidades para programar en los lenguajes antes mencionados, tales como resaltado de sintaxis, administración de proyectos, asistentes de aplicación y un potente depurador entre otras características.

El programa lo pueden encontrar en la pagina oficial del proyecto en <http://anjuta.sourceforge.net/> al igual que la documentación y toda la información necesaria para poder trabajar con él. En la mayoría de las distribuciones ya viene instalado y configurado por lo que no explicare el proceso de instalación, en caso de no tenerlo es solo cuestión de instalarlo mediante el administrador de paquetes de tu distribución o en caso de no contar con una conexión a internet, siempre puedes descargar los fuentes y compilarlo tú mismo.



**Gráfico 3.** Logo anjuta

## Manos a la obra

Después de un pequeño preámbulo pasamos a nuestro tema en sí. A continuación muestro el código del programa que nos servirá de ejemplo para la depuración:

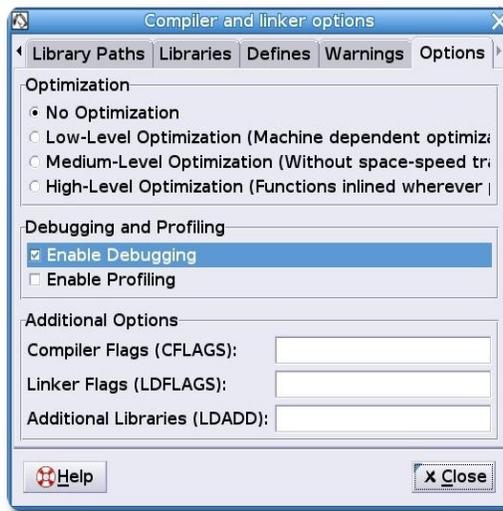
```

1  #include<stdio.h>
2
3  int Factorial(int n);
4  int main()
5  {
6      int a;
7      for(a = 2; a<10; a++)
8          printf("El factorial de %i es: %i\n",a,Factorial(a));
9      return 0;
10 }
11 //*****
12 int Factorial(int n)
13 {
14     int i, val = 1;
15     for(i = 2; i<=n; i++)
16         val *= i;
17     return val;
18 }

```

**Gráfico 4.** Factoriales

Como podrán ver este pequeño código calcula el factorial de  $n$  para una secuencia de valores consecutivos. Lo primero que haremos sera codificar este programa en el editor que provee Anjuta. Para poder utilizar la depuración debemos indicarle al IDE esta opción, la cual habilitaremos desde el menú **settings->compiler and linker options->Options** aqui activamos la casillas de verificación **enable debugging**.



**Gráfico 5.** Habilitación de la depuración.

Una vez hecho esto compilamos nuestro programa desde el IDE con **F9** y posteriormente lo construimos con **F11**. Si todo esta bien el programa compilara de forma correcta y no habrá problemas para continuar. Ahora nos vamos al menú **Debug** y pulsamos la opción **start debugger**, con esto se habilitaran las funciones para el depurador que se encontraban inhabilitadas hasta ese momento.

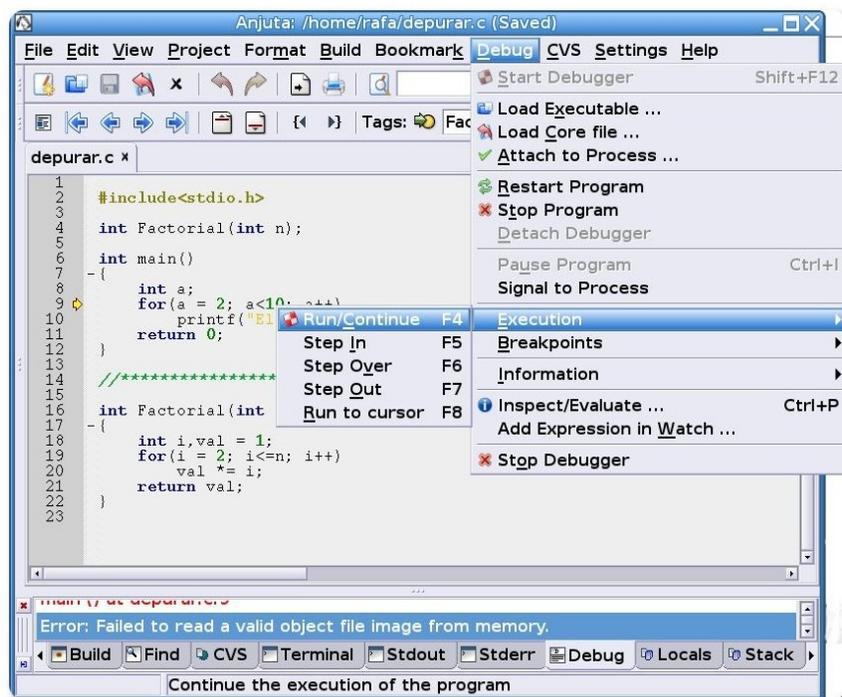


Gráfico 6. Funciones de depuración

Pues bien, ya esta todo listo, ahora empezamos a explicar cada una de las funciones de depuración mas importantes. Antes de empezar cabe decir que este es un ejemplo muy sencillo y he elegido este código ya que permite observar muchos comportamientos de un lenguaje de programación tales como son la variables, ciclos y retorno de valores con lo que creo es más que suficiente para entender bien esto de la depuración.

## Correr programa (run)

El primer paso que realizaremos después de ejecutar el depurador será correr nuestro programa para empezar a trabajar con él. Esto lo haremos desde el menú **Debug->Execution->Run/Continue** o simplemente presionando **F4**.

Cuando iniciemos la ejecución del programa se lanzara una consola (si nuestro programa es de tipo consola) en la cual podremos ir viendo los resultados e interactuar con el programa.

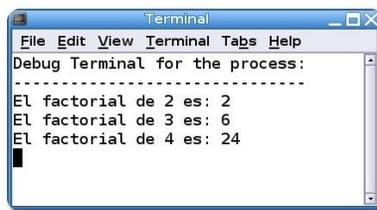


Gráfico 7. Consola modo depuración

## Puntos de ruptura (Breakpoints)

Quando corremos nuestros programas generalmente las instrucciones se ejecutan secuencialmente, es decir, el procesador ejecutara cada una de las líneas de instrucción que contenga nuestro programa una tras otra hasta llegar al final del mismo y dar por terminado el programa. Como a nosotros nos interesa ejecutar el programa paso a paso para poder estudiar su comportamiento, una vez que iniciemos el depurador le debemos indicar en que puntos del

programa queremos que se detenga para checar como van las cosas, por ejemplo, podemos checar el valor que contienen ciertas variables en cierto momento de la ejecución. A estos puntos de parada se les conoce como breakpoints o puntos de ruptura. Para indicar estos puntos al depurador lo haremos de las siguientes maneras:

1. Colocamos el cursor en la línea donde queremos colocar un breakpoint y :
  1. Damos doble clic en el área de numeración de líneas.
  2. O desde el menú **Debug->Breakpoints->Toggle breakpoint**.
2. Desde el menú **Debug->Breakpoint->Set breakpoint**. Aquí nos pide el número de línea donde se colocara el breakpoint y opcionalmente una condición para el mismo.



**Gráfico 8.** Set breakpoint

Una vez puesto el breakpoint notará un círculo de color magenta en el área de numeración de línea. Podemos colocar tantos puntos de ruptura como necesitemos y si quisiéramos eliminar algunos de ellos, solo es cuestión de utilizar las funciones **Disable all breakpoints->clear all break points** del menú **Debug->Breakpoints** o simplemente dar doble clic sobre los círculos de color

```

3
4   int Factorial(int n);
5
6   int main()
7   -{
8       int a;
9       for(a = 2; a<10; a++)
10          printf("El factorial de %i es: %i\n",a,Factorial(a));
11          return 0;
12      }
13

```

**Gráfico 9.** Breakpoint

## Ejecutar hasta el cursor (Run to cursor)

Esta función indica al depurador que ejecute todas las líneas de código desde el punto en el que se encuentra la ejecución hasta el lugar donde esta situado el cursor. Con esta funcionalidad nos evitamos la tarea de colocar breakpoints y después correr o continuar con la ejecución del programa. Para utilizar esta funcionalidad basta con el situarse en la línea el la cual queremos que pare la ejecución y después presionar **F8** o desde **Debug->Execution->Run to cursor**.

## Step in (ejecución paso a paso)

La función **step in** nos permite la ejecución individual de cada una de las sentencias de nuestro programa, con lo que podemos observar detalladamente el orden de ejecución y examinar los valores de las variable cada vez que lo necesitemos. Si la siguiente sentencia a ejecutar es una llamada a una función, el apuntador de ejecución se trasladaran dentro del código de dicha función y ejecutara paso a paso cada sentencia que contenga la función.

```
13
14 //*****
15
16 int Factorial(int n)
17 -{
18     int i, val = 1;
19 |   for(i = 2; i<=n; i++)
20     val *= i;
21     return val;
22 }
23
```

Gráfico 10. Step in

Para utilizar esta función nos vamos a **Debug->Execution->Step in** o simplemente oprimir **F5** cada vez que deseemos ejecutar una sentencia más.

## Step over

Esta opción es muy parecida a la anterior, la única diferencia es que esta función ejecuta las llamadas a funciones como una sola instrucción, es decir no se translada a la función para ejecutar sus sentencias una a una. Nos iremos a **Debug->Execution->Step over** para utilizar esta funcionalidad o presionando **F7**.

## Step out

Realmente step out no ejecuta sentencias individuales, ya que su función es ejecutar todas las sentencias de una función hasta encontrar el punto de retorno. Es decir, si nos encontramos en la primera sentencia de un procedimiento y utilizamos esta opción, el apuntador de ejecución se trasladara hasta el final del procedimiento o punto de retorno. En el menú **Debug->Execution->Step out** se puede encontrar esta opción o simplemente presionando la tecla **F7**.

## Inspección de variables

Como se mencionó anteriormente, a cada paso de ejecución de nuestro programa podremos checar que valores tienen cada una de las variables que utilizamos. Pues bien, si en un momento dado nos detenemos en algún punto de ejecución y queremos inspeccionar valores, nos dirigimos a la ventana de mensajes que se encuentra en la parte baja de la ventana y a nos colocamos sobre la pestaña **Watches**. Si la ventana no aparece la podemos activar desde el menú **View** o con la combinación de teclas **Ctrl + F1**.

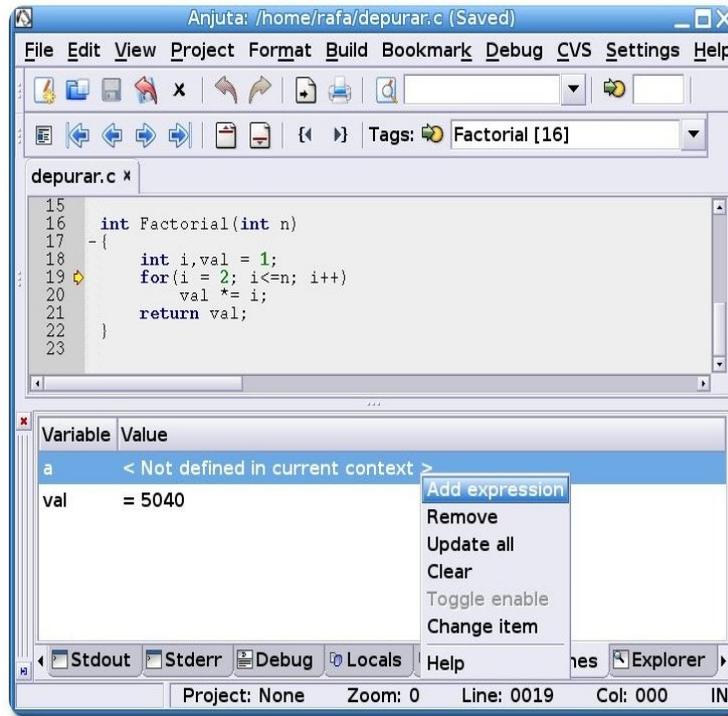


Gráfico 11. Inspección de variables.

Con el botón derecho desplegamos un menú con opciones que nos permiten inspeccionar variables, por ejemplo, añadir expresión, remover, actualizar todo, limpiar todas las inspecciones, entre otras.

### Paso de parámetros

En caso de que nuestro programa necesite recibir parámetros de inicio, se los suministraremos en la caja de parámetros que nos aparece al ejecutar **Build->Set program params**.

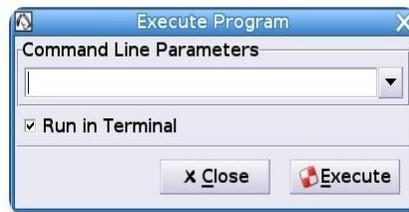


Gráfico 12. Paso de parámetros

### Parar programa

Una vez que terminemos de ejecutar nuestro programa o queramos detener la ejecución del mismo, bastara elegir **Debug->Stop program** o la opción **Restart program** para reiniciarlo.

Hasta aquí llega este mini tutorial. Estas son las operaciones más importantes dentro de la depuración, existen algunas más pero no son tan relevantes como estas, el propio usuario puede deducir su uso y utilidad.

## Conclusión

La depuración es muy importante dentro del desarrollo de software y que puede llegar a ser tan fácil de realizar si utilizamos las herramientas adecuadas pero que más allá de saber utilizar un depurador es mucho más conveniente aprender a diseñar buenos algoritmos para evitar problemas.

## Referencias

- [1] <http://anjuta.sourceforge.net/>
- [2] Manual de Anjuta

## Autor



**Rafael Rendón Pablo**  
Estudiante de Ingeniería en Sistemas Computacionales  
Usuario de software libre  
[smart.dyr@gmail.com](mailto:smart.dyr@gmail.com)



# Desarrollo Ágil con Ruby on Rails (4ta Parte)

Ruby on Rails es un framework de desarrollo web ágil, elaborado por David Heinemeier Hansson, que el 2004 lanzó la versión pública a partir de desarrollar el proyecto Basecamp, Ruby on Rails (RoR) está desarrollado en el lenguaje de programación Ruby, RoR actualmente se encuentra en la versión 2.1.1 en la que contribuyeron más de 1400 desarrolladores del mundo con 1600 parches al framework, por estos datos RoR es uno de los proyectos open source con más movimiento actual.

## Registro y autorización de usuarios en Ruby on Rails.

Actualmente todas las aplicaciones web necesitan incorporar seguridad y control de acceso dentro de sus características, el framework Ruby on Rails dentro de su propia infraestructura no incorpora esta funcionalidad, pero existen plugins que nos permiten incorporarlos, en este caso utilizaremos el conocido plugin de autenticación y autorización: **restful-authenticacion**.

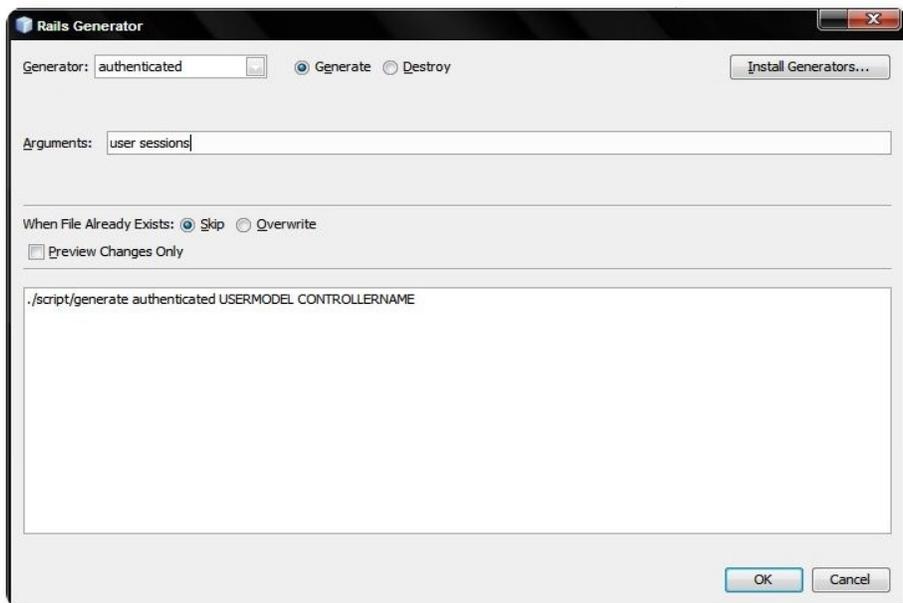
Primeramente se debe de instalar el plugin con el siguiente comando, dentro del proyecto escribimos:

```
./script/plugin install git://github.com/technoweenie/restful-authentication.git
```

o visitar la página del proyecto en <http://github.com/technoweenie/restful-authentication>, para descargar el plugin, copiar dentro de la carpeta **Vendor/plugins** y renombrar la carpeta a **restful\_authentication**, una vez instalado, proceder a generar las clases, controladores y html necesario con el siguiente comando, dentro del folder del proyecto, escribimos:

```
./script/generate authenticated user sessions
```

O también desde Netbeans, click derecho en el proyecto y seleccionamos la opción **Generate...** seleccionamos la opción **authenticated** y en **Arguments** escribimos: **user sessions** como se muestra a continuación



Con esto se generaran los controladores **Users** y **Sessions**, el modelo **User** y una migración que crea a la tabla **users**, la que modificaremos un poco para que al realizar la migración se añada un campo que marque a un usuario si es administrador o es un usuario normal, y cree por defecto un usuario administrador, este es código de la migración en **db/migrate/xxxxxx\_create\_users.rb**.

```
class CreateUsers < ActiveRecord::Migration
  def self.up
    create_table "users", :force => true do |t|
      t.column :login,           :string, :limit => 40
      t.column :name,           :string, :limit => 100
      t.column :email,         :string, :limit => 100
      t.column :encrypted_password, :string, :limit => 40
      t.column :salt,          :string, :limit => 40
      t.column :created_at,     :datetime
      t.column :updated_at,     :datetime
      t.column :remember_token, :string, :limit => 40
      t.column :remember_token_expires_at, :datetime
      t.boolean :admin, :default => false
    end
    add_index :users, :login, :unique => true

    admin = User.new(:name => '', :login => 'admin', :email => "admin@sitio.com",
:password => "admin123", :password_confirmation => 'admin123', :admin => true)
    admin.admin = true
    admin.save!
  end

  def self.down
    drop_table "users"
  end
end
```

Con esto creamos un usuario con **login**: 'admin' y **password**: 'admin123'; tengamos en cuenta que automáticamente valida que nuestra contraseña sea mayor a 6 caracteres, para que sea administrador la asignación lo hacemos explícita, por que el modelo **user** como una medida de seguridad, solo acepta como parámetros públicos los campos: **login**, **email**, **name**, **password** y **password\_confirmation**, y no otros, así no nos podrán inyectar código inseguro o auto-asignarse como administradores.

Bien ahora realizamos la migración con **rake db:migrate** o desde netbeans seleccionando **Migrate Database->To current versión**.

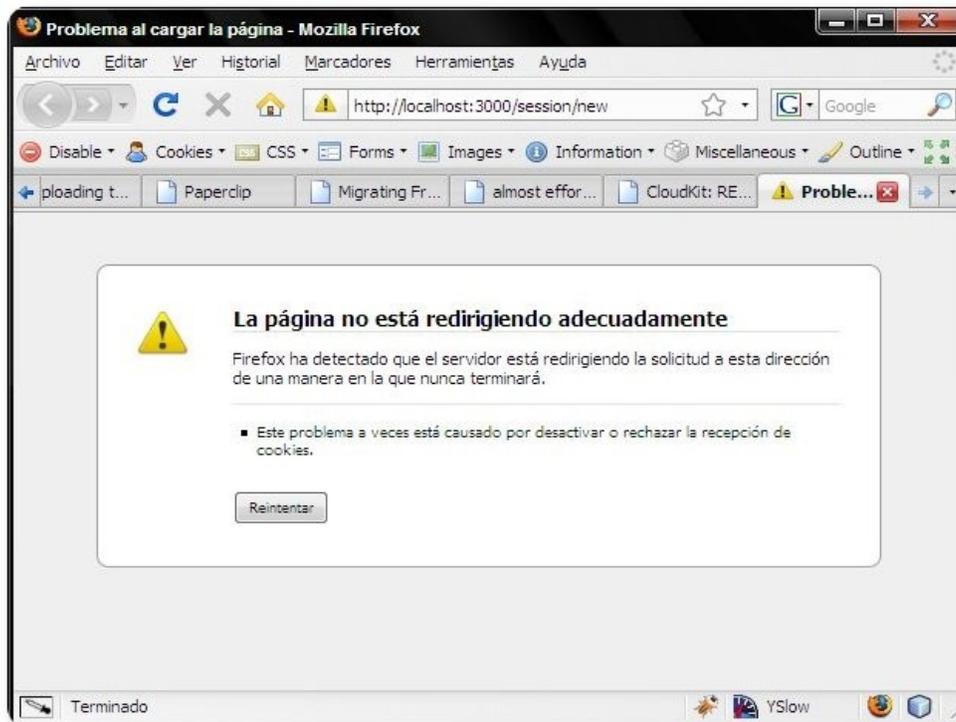
Ahora tenemos que modificar unas cuantas líneas en los controladores, para esto abrimos los archivos **sessions\_controller.rb** y **users\_controller.rb**, en estos borramos la línea seleccionada en el dibujo y esta la copiamos tal cual dentro de **controllers/application.rb**.

```
1 # This controller handles the login/logout function of the site
2 class SessionsController < ApplicationController
3   # Be sure to include AuthenticationSystem in Application Controller
4   include AuthenticatedSystem
5
6   # render new.rhtml
7   def new
8   end
9
10  def create
```

Quedando de la siguiente manera:

```
class ApplicationController < ActionController::Base
  helper :all
  include AuthenticatedSystem
  before_filter :login_required
  protect_from_forgery
  filter_parameter_logging :password
end
```

Con la instrucción **before\_filter :login\_required** le decimos a nuestra aplicación que no permita que ningún usuario anónimo pueda utilizar el sitio, además redireccionará a la página de registro para que pueda identificarse. Ahora iniciamos la aplicación con `./script/server` o Run en Netbean; una vez iniciado veremos el siguiente error dentro de nuestro navegador al ingresar al sitio:



Esto es porque le dijimos que controlara todo el acceso anónimo aun en el momento de registrarse o identificarse y por eso entra en un bucle infinito, arreglamos la anomalía añadiendo dentro de `controllers/sessions_controller.rb` después de definir la clase la línea de código:

```
skip_before_filter :login_required
```

Con esto, se soluciona el error mostrándonos ahora una ventana donde se pide nuestro login y password; para poder manipular el sistema.



Ahora adicionaremos un enlace para que cuando este identificado pueda desconectarse del sistema y si es un usuario anónimo, pueda identificarse. Por último validaremos que solo el administrador pueda crear una revista y no cualquier usuario, además el administrador podrá crear nuevos administradores o usuarios normales mostrando un enlace a esta acción solo al administrador.

#### En views/layouts/application.html.erb

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
  <head>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
    <title>Sistema Atix</title>
    <%= stylesheet_link_tag 'style' %>
  </head>
  <body>
    <div class="header">
      <% if logged_in? && current_user.admin? %>
        <div class="menu">
          <%= link_to("Revistas", revistas_path, :title => "Listado de Revistas") %> |
<%= link_to("Articulos", articulos_path, :title => "Listado de Articulos") %> | <%=
link_to("Autores", autores_path, :title => "Listado de Autores") %>
          <%= " | #{link_to("Usuarios", users_path)}" if current_user.admin? %>
          | <%= link_to("Salir", '/logout') %>
        </div>
      <% else %>
        <div class="menu">
          <%= link_to("Identificarse", '/login') %>
        </div>
      <% end %>
      <br class="clear">
      <h1><%= link_to("Sistema de registros de Atix", root_path, :title => "Sistema de
archivo de Revistas") %></h1>
    </div>
    <div class="body">
      <% flash.each do |key, msg| -%>
        <%= content_tag :div, msg, :class => key, :id => 'flash' -%>
      <% end -%>
      <%= yield %>
    </div>
  </body>
</html>
```

En la vista `views/revistas/index.html.erb` modificamos la última línea:

```
<%= link_to('Nueva revista', new_revista_path) if current_user.admin? %>
```

Y por último modificamos el método `create` dentro del controlador de Revistas, quedando de esta forma:

```
def create
  @revista = Revista.new(params[:revista])

  respond_to do |format|
    if current_user.admin? && @revista.save
      flash[:notice] = 'Revista was successfully created.'
      format.html { redirect_to(@revista) }
      format.xml { render :xml => @revista, :status => :created, :location =>
@revista }
    else
      format.html { render :action => "new" }
      format.xml { render :xml => @revista.errors, :status =>
:unprocessable_entity }
    end
  end
end
```

Bien con esto terminamos esta parte de autenticación de usuarios, utilizando el plugin **Restful-Authentication**; que facilita esta tarea; además adicionamos un rol de administrador a nuestra aplicación, utilizamos este rol para la autorización dentro de los controladores y vistas, para bajar el código fuente visiten <http://github.com/carakan/atix/tree/master>.

## Referencias

- [1] <http://www.rubyonrails.org/>
- [2] <http://www.rubyforge.org/>

## Autor



**Carlos Ramos**

Lic. Informática UMSA  
 Lider de Wiebia, soluciones web 2.0  
 carakan@gmail.com  
 Blog personal <http://ww.carakan.com>  
<http://www.wiebia.com>

# Diseño Avanzado de Aplicaciones Web. EXT – Zend Framework y Doctrine

Describe la propuesta de diseño de aplicaciones utilizando los frameworks Zend, EXT y Doctrine, así como los principales Mecanismos de Diseño y artefactos a generar durante el Análisis y Diseño de la aplicación. Se fundamenta el estilo arquitectónico propuesto y se presentan variantes de diagramas a escoger por el equipo de analistas y arquitectos.



## Introducción

Uno de los elementos bases del proceso de desarrollo de software es diseñar la Arquitectura de Software. Esencialmente sobre ella se sustentan todos los mecanismos de diseño y representaciones de la estructura general de la aplicación a desarrollar. De la cohesión, utilidad y flexibilidad de los componentes de la arquitectura dependerán la calidad final y la utilidad del software. La correcta definición del estilo arquitectónico a utilizar, patrones y mecanismos de diseño es la raíz de lo anteriormente descrito.

## Arquitectura

Una definición de arquitectura del software es:

*“Una arquitectura es el conjunto de decisiones significativas sobre la*

*organización del sistema de software, la selección de los elementos estructurales y sus interfaces, con los que se compone el sistema, junto con su comportamiento tal como se especifica en las colaboraciones entre esos elementos, la composición de esos elementos estructurales y de comportamiento en subsistemas progresivamente más amplios, y el estilo de arquitectura que guía esta organización –estos elementos y sus interfaces, sus colaboraciones, y su composición”.* (Booch, 1999)

Una idea general de la definición anterior es que se necesita lo siguiente:

- ✓ Patrones de Diseño.
- ✓ Principios de Diseño (Patrones y estilos arquitectónicos.
- ✓ Mecanismos de Diseño.
- ✓ Diagramas de Clases de Diseño.
- ✓ Diagramas de Interacción.

Y de esa manera entonces quedaría expuesta la real arquitectura del software, o la base de diseño para ella.

Luego de lo anteriormente descrito y para llegar a la idea que se desea, basados en la máxima de que una buena gestión y organización es más importante que la tecnología, se propone la selección de un patrón arquitectónico de entre los siguientes:

## Capas

El patrón Capas se relaciona con la arquitectura lógica, es decir, describe la organización conceptual de los elementos del diseño en grupos, independiente de su empaquetamiento o despliegue físico. (Larman)

Es uno de los patrones más generalizados y utilizados en el desarrollo de aplicaciones web a nivel global y sencillo de implementar. Se resume en lograr:

Organizar la estructura lógica de gran escala de un sistema en capas separadas de responsabilidades distintas y relacionadas, con una separación clara y cohesiva de intereses como que las capas "más bajas" son servicios generales de bajo nivel, y las capas más altas son más específicas de la aplicación.

La colaboración y el acoplamiento es desde las capas más altas hacia las más bajas; se evita el acoplamiento de las capas más bajas a las más altas.

Muchas ventajas, muchas aplicaciones pero muchos de los frameworks (su uso es una tendencia del desarrollo Web actual) que se usan hoy en día están implementados bajo otros estilos arquitectónicos y no en 3 capas, que no puede ser violado. ¿Será entonces una buena practica desechar la reutilización de código que plantean los frameworks? No es una buena idea, se propone entonces uno de esos estilos.

## MVC

Es un estilo basado en un patrón de diseño que plantea la separación de diferentes clases en dependencia de la función que realizan de modo tal que sea posible manejar dinámicamente la forma en que se procesan solicitudes y se gestiona la manera en que se muestran resultados al usuario final. En otras palabras separa la presentación del dominio de la aplicación. A simple vista ya se tienen ventajas.

Es un principio que utilizan muchos frameworks para basar su funcionamiento, la idea de "Don't call us, we'll call you" (No nos

llame, nosotros lo llamaremos a usted). Esa idea ha hecho que los frameworks que implementan MVC se puedan usar sencillamente implementando interfaces o extendiendo de una clase abstracta que brinda el framework. Muchos de los más usados en PHP son:

Framework	Estilo
Prado	MVC
CakePHP	MVC
Symfony	MVC
Kumbia	MVC
CodeIgniter	MVC
Zend Framework	MVC

Tabla 1 Frameworks sobre MVC

A continuación se muestra una representación del funcionamiento del patrón MVC que es la base del estilo arquitectónico del mismo nombre. Representa una manera muy sencilla de gestionar la presentación de datos a los usuarios finales de la aplicación y si se cuenta además con pequeñas aplicaciones pre-elaboradas como punto de partida es aún más útil. De entre los frameworks mostrados en la tabla anterior se propone utilizar el Zend Framework por un conjunto de características que se mostrarán más adelante.

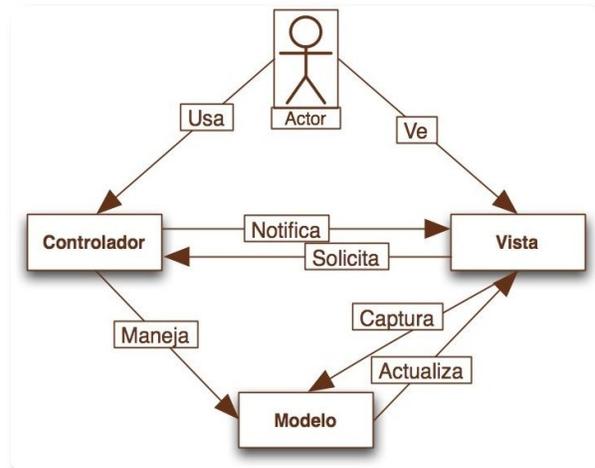


Fig. 1 Estructura de MVC

## Mecanismos de Diseño

Una de las principales preocupaciones de los diseñadores de hoy día es cómo reutilizar modelos ya elaborados, al igual que los desarrolladores desean siempre reutilizar código.

El término de Mecanismos de Diseño fue introducido en (Documentación del Proceso Unificado, 2003) y ha sido utilizado con el objetivo de simplificar los diagramas de clases. Cada diseñador establece sus propios mecanismos de diseño y es el total responsable de sus modelos y su forma siempre y cuando no viole los patrones y estilos seleccionados.

Los mismos reportan beneficios para al menos 3 propósitos:

- ✓ Mantener la homogeneidad en el diseño.
- ✓ Reutilizar soluciones anteriormente probadas.
- ✓ Reutilizar documentación.

A continuación, haciendo uso de este artefacto, se detallan los componentes (clases y frameworks) a utilizar en cada parte de MVC.

## Componentes de la Arquitectura

### Vista

#### Uso de EXTJS

Es un framework para JavaScript muy utilizado en el desarrollo de aplicaciones Web con AJAX. Tiene una librería inmensa que permite configurar las interfaces Web de manera semejante a aplicaciones desktop .

Tiene incluidos la mayoría de los controles de los formularios Web incluyendo Grids para mostrar datos y elementos semejantes a la programación desktop como los formularios, paneles, barras de herramientas, menus y muchos otros. Dentro de su librería de componentes incluye componentes para el manejo de datos, lectura de XML, lectura de datos JSON e implementaciones basadas en AJAX. Presenta el uso de JavaScript con una programación orientada a objetos.

Clase	Descripción
ext-base	Encargada del manejo de las solicitudes y respuestas, trabajo con ajax y manejo de componentes de EXT. Está incluida en el paquete original.
ext-all	Es la encargada de la creación de los componentes visuales de la vista. Está incluida dentro de las clases que trae EXT JS.
Vista	Representa la vista que se muestra al usuario.
js_vista	Fichero js con las funciones Java Script asociadas a la vista. Aquí se establece la referencia a las clases de EXT.

**Tabla 2** Descripción clases EXTJS

En el siguiente diagrama de clases se representa el uso de EXT en la vista de la aplicación resaltando las clases fundamentales utilizadas de entre el resto.

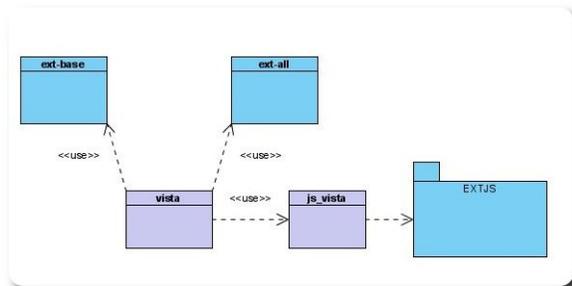


Fig. 2 Diagrama de Clases Genérico para EXTJS

Pero es sabido que en la real representación según el código fuente las clases, ext-base y ext-all están incluidas como clases del paquete original de EXTJS por lo que la propuesta quedaría:

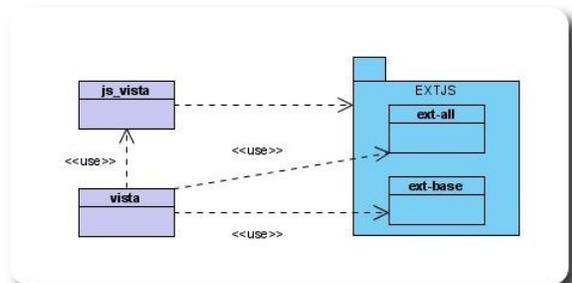


Fig. 3 Diagrama de Clases Genérico para EXTJS

## Controlador

Son las clases que gestionan el manejo de la lógica del negocio. Por lo general incluyen las restricciones y validaciones fundamentales determinadas por las reglas. Para un caso de uso puede representarse una clase controladora o más.

## Uso de Zend Framework

Zend Framework es uno de los más utilizados para PHP y utiliza el estilo MVC como base de su funcionamiento. Es fácilmente integrable a las aplicaciones debido a su composición y a que contiene diferentes clases de gran utilidad, como por ejemplo en la búsqueda dinámica de ficheros a incluir o utilizar.

Cuenta con un importante mecanismo de manejo de controladores y vistas por lo que se propone tenerlo en cuenta para el diseño de estos dos componentes de la arquitectura.

## Elementos Arquitectónicamente significativos de Zend Framework

Los frameworks, como aplicaciones de software que son, deben cumplir con los requerimientos de arquitectura semejante al resto. En este caso se encuentran los patrones de diseño y Zend Framework incluye en su implementación algunos de ellos.

- ✓ **Vista:** Implementa el patrón Decorator en la clase Zend\_View, encargada de asignarle responsabilidades a objetos de manera dinámica y configurarlos con nuevos atributos.
- ✓ **Controlador:** Zend Framework tiene implementado el patrón Front Controller que implica que todas las solicitudes son dirigidas a un único script PHP que se encarga de instanciar al controlador frontal y redirigir las llamadas.

Además tiene una instancia única del controlador frontal disponible mediante el patrón singleton para lograr una vía de entrada única a las solicitudes.

- ✓ **Modelo:** ZF provee una API para el acceso a dato conformada por un conjunto de clases que implementan los patrones Factory, Table Data Gateway y Row Data Gateway.

Por supuesto todos los patrones de diseño anteriormente mencionados implican características y comportamiento específico en los componentes de la arquitectura por lo que deben ser tenidos muy en cuenta. A continuación se modelan las clases fundamentales a utilizar en el controlador.

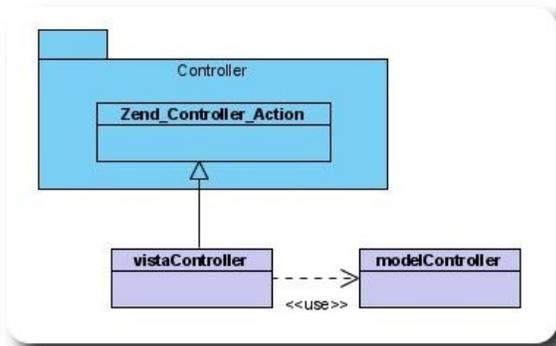


Fig. 4 Diagrama de Clases Genérico para Zend Framework (controladores)

Clase	Descripción
Zend_Controller_Action	De esta clase deben heredar todos los controladores de la aplicación, en ella se incluyen numerosas funcionalidades comunes.
vistaController	Representa el controlador del Caso de Uso en cuestión.
modelController	Es un intermediario entre el controlador y la clase del modelo. No debe heredar de Zend_Controller_Action, incluye las principales funciones para el manejo de los datos.

Tabla 3 Descripción clases Zend Framework

En la representación general e implementación se utilizan además otras dos clases de Zend Framework muy importantes que son Zend\_Controller\_Front y Zend\_Loader que son el controlador frontal y la encargada de búsquedas automáticas respectivamente. Sin ellas el funcionamiento sería imposible.

## Modelo

### Uso de Doctrine PHP

Doctrine es un potente y completo sistema ORM (object relational mapper) para PHP 5.2+ con un DBAL (database abstraction layer) incorporado.

Entre muchas otras cosas tienes la posibilidad de exportar una base de datos existente a sus clases correspondientes y también a la inversa, es decir convertir clases (convenientemente creadas siguiendo las pautas del ORM) a tablas de una base de datos.

Su principal ventaja radica en poder acceder a la base de datos utilizando la programación orientada a objetos (POO) debido a que doctrine utiliza el patrón Active Record para manejar la base de datos, tiene su propio lenguaje de consultas y trabaja de manera rápida y eficiente. Es fácilmente integrado a los principales frameworks de desarrollo utilizados actualmente, por lo que se propone su uso.

Ver Características fundamentales de Doctrine PHP (Man08)

A continuación se representan las clases fundamentales a representar en el modelo. Las mismas son generadas automáticamente.

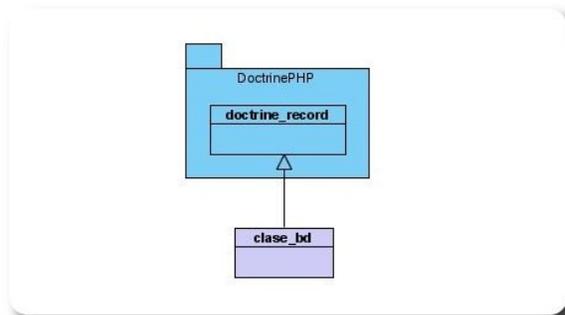


Fig. 5 Diagrama de Clases Genérico para Doctrine PHP

## Modelo – Vista - Controlador

Habiendo finalizado de describir cada componente del estilo, a continuación se propone la vista de Gestión de Modelo de la Arquitectura Base para MVC. Se representan los paquetes fundamentales y el conjunto de clases utilizadas directamente de entre el resto de las contenidas.

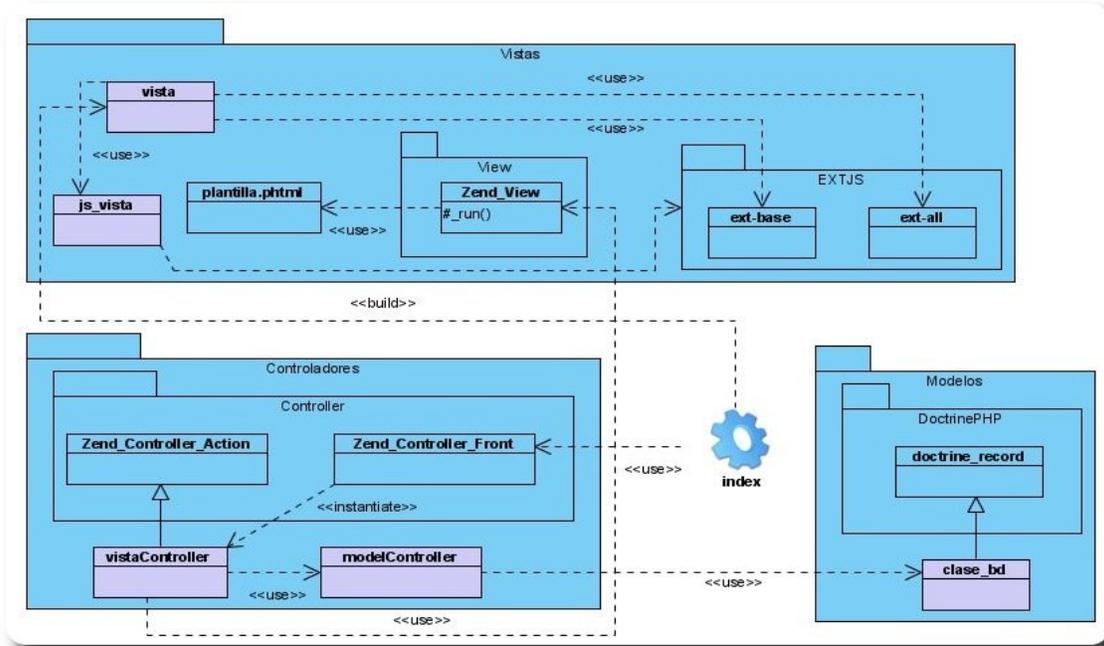


Fig. 6 Vista de Gestión de Modelo Genérico para MVC

Los paquetes View y Controller son los paquetes incluidos por defecto en la carpeta library del Zend Framework.

## Extensiones UML para Web

A finales de los 90, cuando el desarrollo de aplicaciones Web se hizo más importante, Jim Conallen hace uso de las facilidades de extensión brindadas por el UML para, basado en este lenguaje modelar aplicaciones Web. Publica hacia junio del 99 varios artículos incluido “Modelling Web Applications Architectures with UML” donde describe la extensión de UML para el modelado de aplicaciones Web. (UML en acción. Modelando aplicaciones Web, 2006)

Su extensión presenta como elementos más significativos a 3 clases de UML estereotipadas con los siguientes estereotipos “Server Page”, “Client Page”, “Form” empleados para el código servidor, código cliente y formularios respectivamente.

La extensión permite reflejar otros elementos como controles ActiveX, Librerías de Etiquetas (tag library), Ficheros de Scripts, Conjunto de Frames, que prácticamente no se comentan en el presente trabajo, para profundizar en estos, se recomienda consultar (Conallen, 2002).

Las relaciones posibles a establecerse entre los tres elementos claves son:

Hasta Desde	Client Page	Form	Server Page
Client Page	<<Link>> , <<redirect>>	Contiene	<<Link>> , <<redirect>>
Form	Agregado por.	---	<<Submit>>
Server Page	<<Build>>, <<redirect>>		<<Redirect>>

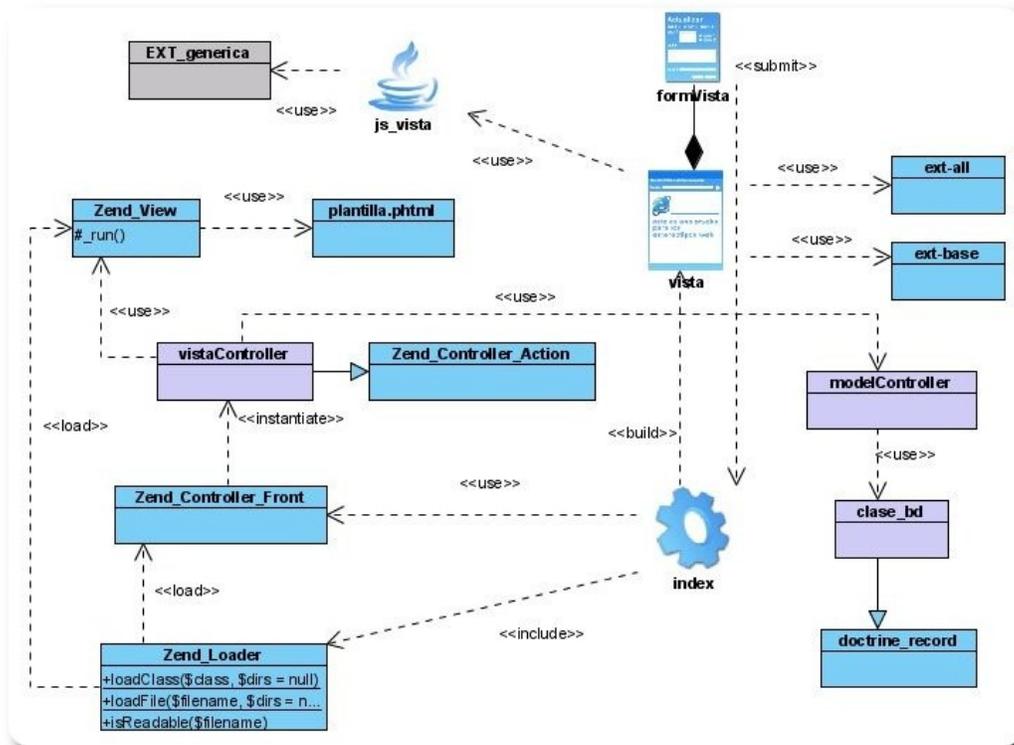
**Tabla 4** Relaciones entre estereotipos Web

Como se está en presencia de una aplicación Web se recomienda entonces hacer uso de dichos estereotipos y sus relaciones para el modelado de la aplicación en general.

### Diagramas de Clases

Una vez definida la base de la arquitectura se presenta a continuación la propuesta para los Diagramas de Clases del Diseño de los Casos de Uso del Sistema de manera genérica. En el diseño se propone no volver a especificar patrones de diseño puesto que la reutilización de EXT, ZF y Doctrine incorporan algunos de los más utilizados para la presentación y el manejo de datos.

Se sugiere el uso de los artefactos (clases, paquetes, relaciones, subsistemas) como medio para simplificar diagramas que puedan ser complejos y lograr un mejor entendimiento.



**Fig. 7** DCD para Caso de Uso Genérico.



A manera de explicación se recuerda que index es el mecanismo que se conecta con el controlador frontal quien redirecciona todas las solicitudes al controlador correspondiente según la solicitud del usuario. A la vez recibe las solicitudes y envíos de los formularios de EXT y deja el trabajo una vez más al controlador frontal quien las enruta a la página destino. La clase index representa el contenedor general que será quien construirá de manera dinámica y mostrará la vista final al usuario. Un mismo caso de uso puede tener hasta n vistas y controladores por lo que se deja en mano de los diseñadores la adaptación del diagrama de clases UML.

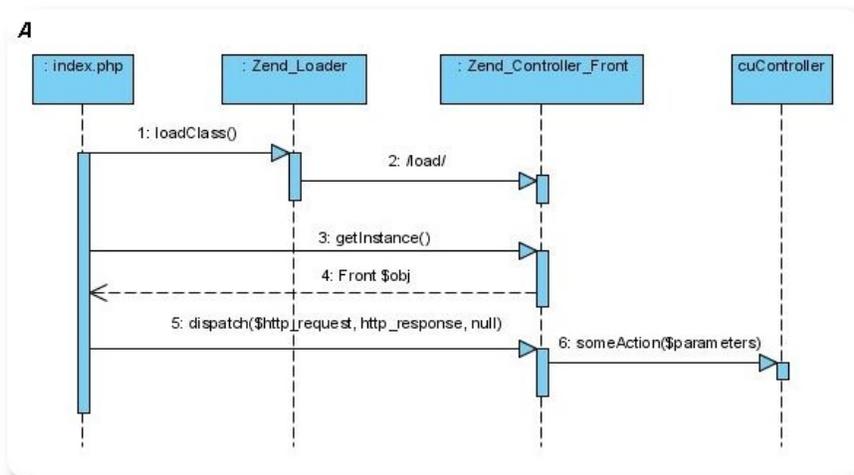
### Diagramas de Interacción

Si bien los diagramas de clases de diseño muestran las relaciones estáticas entre las diferentes clases, los diagramas de Interacción: secuencia y colaboración, muestran la parte dinámica de las relaciones entre las instancias de dichas clases en las realizaciones de los casos de uso. Se representan mediante una secuencia de objetos y sus relaciones, además de los mensajes (llamadas a métodos, creación

destrucción de objetos, referencia) y el tiempo de vida de los objetos (duración de la instancia) y cada uno de los dos a pesar que muestra información semejante tiene sus peculiaridades. Muchos diseñadores seleccionan a su preferencia el diagrama en dependencia de la información que desea mostrar; normalmente podría dejarse a selección pero se recomienda el uso de los diagramas de secuencia para el diseño.

Luego de la explicación anterior y para una mejor comprensión del funcionamiento interno del software así como la interacción entre las clases de la aplicación y el framework se proponen dos diagramas de interacción genéricos (secuencia) para una de las dos operaciones básicas de los casos de uso: mostrar datos y realizar una acción determinada.

Las imágenes a continuación muestran el comportamiento general de los objetos en un caso de uso cualquiera de la aplicación representando los principales métodos y clases de los frameworks utilizados. Se deja en manos del diseñador su propio diseño, lo que se presenta es una propuesta incluyendo mensajes genéricos que se explican más adelante en una pequeña tabla.



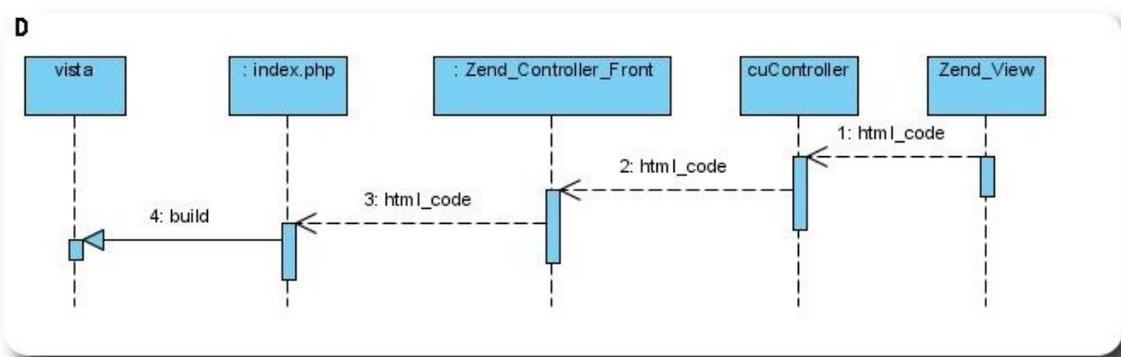
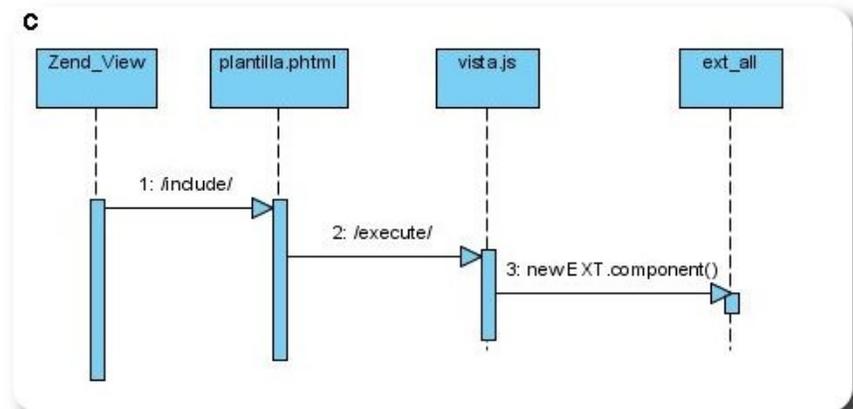
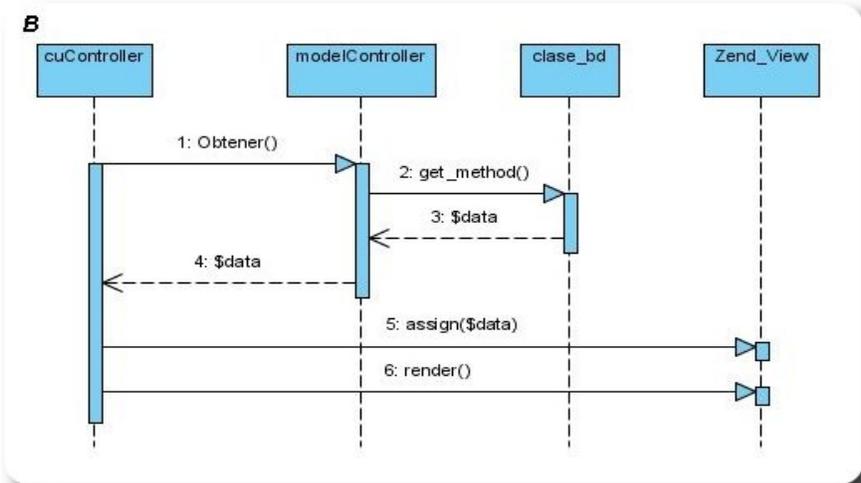


Fig. 9 Diagrama de Secuencia Genérico Mostrar Información

Mensaje	Descripción
someAction	Representa la llamada a uno de los métodos del controlador
get_method	Representa el método de obtención de datos utilizado en el modelo. Por lo general es un método ya establecido por Doctrine.
/include/	Representa el uso de una clase determinada, ya sea su contenido o una funcionalidad incluida.
/execute/	Representa la ejecución del código Java Script por una página cliente.
new EXT.component	Representa la creación de uno de los componentes de EXT, en cada caso podrían crearse varios componentes.
html_code	Representa el retorno del código html de la vista a mostrarse al usuario. Incluye los valores dinámicos así como en contenido de la plantilla.

**Tabla 6** Descripción de mensajes genéricos

## Conclusiones

Con una sólida arquitectura base, un diseño flexible y escalable y el uso de las ventajas y utilidades que brindan los frameworks implementados y disponibles para el desarrollo Web es posible diseñar y posteriormente desarrollar aplicaciones con una considerable rapidez y calidad. El correcto uso de estilos, patrones y mecanismos de diseño en la generación de los artefactos necesarios garantiza la base necesaria para lograr calidad en la propuesta. El uso del Zend Framework, unido a las potencialidades que brinda EXT y un excelente manejo de datos con Doctrine son muestras de esto.

## Referencias

- [1] Manual Online de Doctrine PHP. [En línea] [Citado el: 26 de 04 de 2008.] <http://www.phpdoctrine.org/documentation/manual>.
- [2] Booch, G., Rumbaugh, J, and Jacobson, I. 1999. The Unified Modeling Language User Guide. 1999.
- [3] Conallen, Jim. 2002. Building Web Applications with UML. s.l. : Object Technology Series by Addison Wesley, 2002. Vol. Second Edition.
- [4] Documentación del Proceso Unificado, version 2003.06.00. 2003. Arifact Design Mechanism. Rational Unified Process v 2003.06.00. 2003.
- [5] Larman, Craig. UML y Patrones. 2da Edición. Cap. 30 Epig. 2. Patrón de Arquitectura: Capas.

- [6] SHAW M., Garlan D. 1996. Software Architecture Perspective on an Emerging Discipline. New Jersey : Prentice Hall, 1996.
- [7] UML en acción. Modelando aplicaciones Web. Navarro, José Angel Franco. 2006. Ciudad de la Habana : s.n., 2006.

## Autor



**Yuniel Eliades Proenza Arias**  
Vicedecano de Formación – Facultad 9  
Profesor de Programación  
yproenza@uci.cu



# Grails: Framework para el desarrollo de aplicaciones Web (1ra Parte)

Grails es un framework orientado al desarrollo de aplicaciones web de forma sencilla, rápida y divertida.



## Introducción

Hoy en día, con la aparición de nuevas metodologías y herramientas en el ámbito de la ingeniería de software, el desarrollo de aplicaciones web ha tomado un curso tan distinto a lo que se tenía hace unos años atrás; esto representa que, si antes con el desarrollo y herramientas tradicionales demorábamos semanas y/o meses en prototiparlas, meses o hasta años en desarrollarlas completamente. Hoy en día con la aparición de los frameworks éste tiempo se ha reducido considerablemente, ya que ahora con la ayuda de estas nuevas herramientas como son los frameworks y teniendo como aliadas a las metodologías ágiles de desarrollo, podemos prototipar nuestras aplicaciones en cuestión de minutos u horas, y terminar su desarrollo en una par de semanas.

## Que es un framework

- ✓ Es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado.
- ✓ Se ha convertido en la piedra angular de la moderna ingeniería del software.

- ✓ Es una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación.
- ✓ En otras palabras, es una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta.

## Objetivos de un framework

- ✓ Desarrollo rápido.
- ✓ Desarrollo estructurado.
- ✓ Reutilización de código.
- ✓ Disminución del esfuerzo en el desarrollo.
- ✓ Aprovechar las funcionalidades ya implementadas, no debemos reinventar la rueda.
- ✓ Nos concentramos directamente en la solución del problema.
- ✓ Tener como aliadas a las metodologías de desarrollo ágil (XP, AD).

## Patrón MVC

Desde la aparición de RoR (Ruby on Rails) el padrino de todos los frameworks de desarrollo de aplicaciones web, el patrón MVC ha cobrado una notable importancia, sobre todo por su cualidad de dividir en capas el desarrollo de un proyecto o aplicación web.

- ✓ **Modelo:** Encargado de modelar y contener la lógica del dominio de la

aplicación.

- ✓ **Vista:** Encargada de contener la lógica de presentación.
- ✓ **Controlador:** Encargado de ser el engranaje entre el modelo y la vista; y también contener la lógica del negocio.

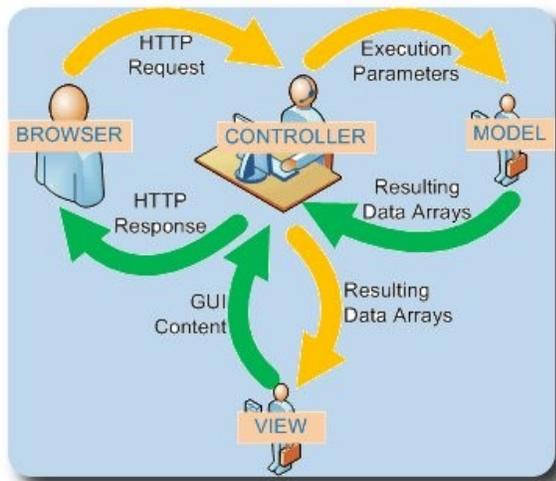


Gráfico 1: Patrón MVC

## Que es Grails

- ✓ Es un framework para el desarrollo de aplicaciones web.
- ✓ Es la respuesta de groovy y java para Ruby on Rails.
- ✓ Permite el desarrollo de aplicaciones web de forma rápida.

## Filosofía de Grails

- ✓ Reutilizar código.
- ✓ Desarrollar aplicaciones web de forma rápida, sencilla y divertida.
- ✓ Utilizar los mejores componentes actualmente disponibles.
- ✓ Combinar todo en un paquete fácil de usar.
- ✓ Grails provee el pegamento, de tal forma que usted no haga el suyo.

## Características de Grails

- ✓ Se basa en el patrón MVC.
- ✓ Tiene todo lo necesario para desarrollar aplicaciones Web.
- ✓ Soporta Ajax.
- ✓ Posee buena documentación.
- ✓ Es multiplataforma.
- ✓ Posee una numerosa comunidad de desarrolladores y usuarios.
- ✓ Inspirado en Ruby on Rails (Padrino de la mayoría de los Frameworks).
- ✓ Está en constante desarrollo.

## Componentes de Grails

- ✓ **Hibernate:** El estándar defecto para ORM en el mundo de Java.
- ✓ **Spring:** El más popular framework para Java.
- ✓ **Quartz:** Un framework que permite realizar tareas programadas.
- ✓ **SiteMesh:** Un framework robusto y estable para renderizar layouts.

## Preparando nuestro entorno de trabajo.

Para empezar a trabajar y desarrollar aplicaciones web en Grails, precisamos cumplir algunos requisitos como ser:

- ✓ Tener instalado el JDK de Java.
- ✓ Tener instalado algún gestor de base de datos (Sqlite, MySQL, PostgreSQL, MS-SQL, Oracle, entre otros).

## Instalación de Grails

- ✓ Obtener la última versión estable de Grails desde su sitio web [1].
- ✓ Descomprimir Grails en un directorio de su preferencia, al cual tenga acceso el usuario.
- ✓ Setear la variable de entorno `GRAILS_HOME`.

- ✓ Adicionar `$GRAILS_HOME/bin` en el `path`.

Para confirmar que nuestra instalación es correcta, introducimos el comando `grails` en la línea de comando.

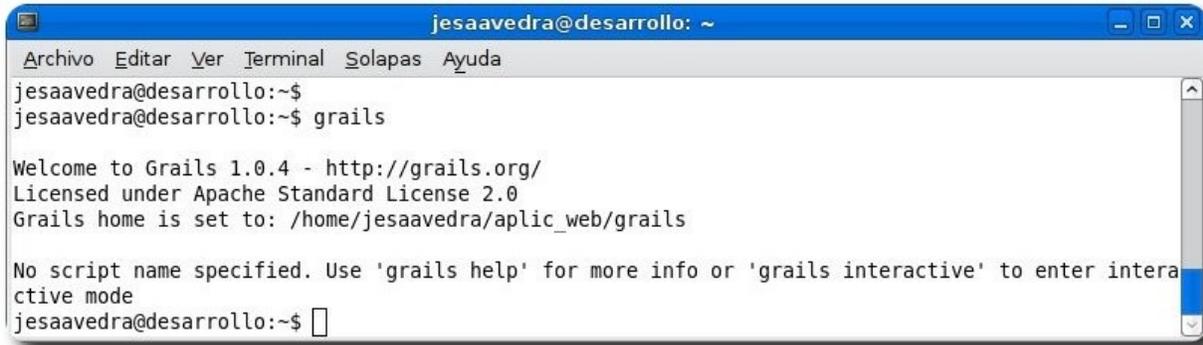


Gráfico 2: Uso del comando `grails`

## Nuestra primera aplicación

Nuestra pequeña aplicación se enmarca en un entorno donde se mantiene el registro de eventos (congresos, jornadas, etc), a los cuales concurren varios expositores de diferentes nacionalidades y donde cada uno de estos tiene una o más conferencias que impartir.

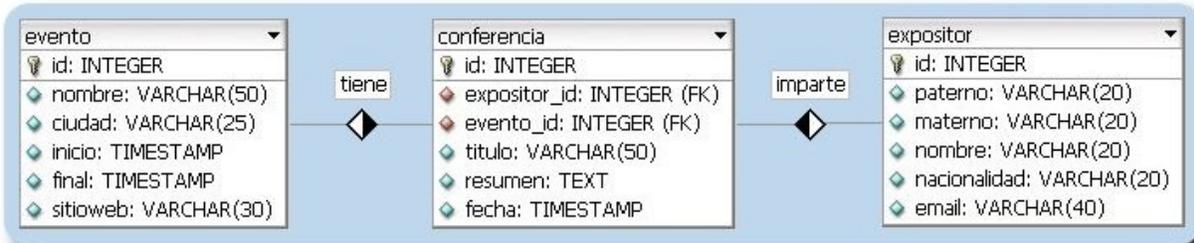


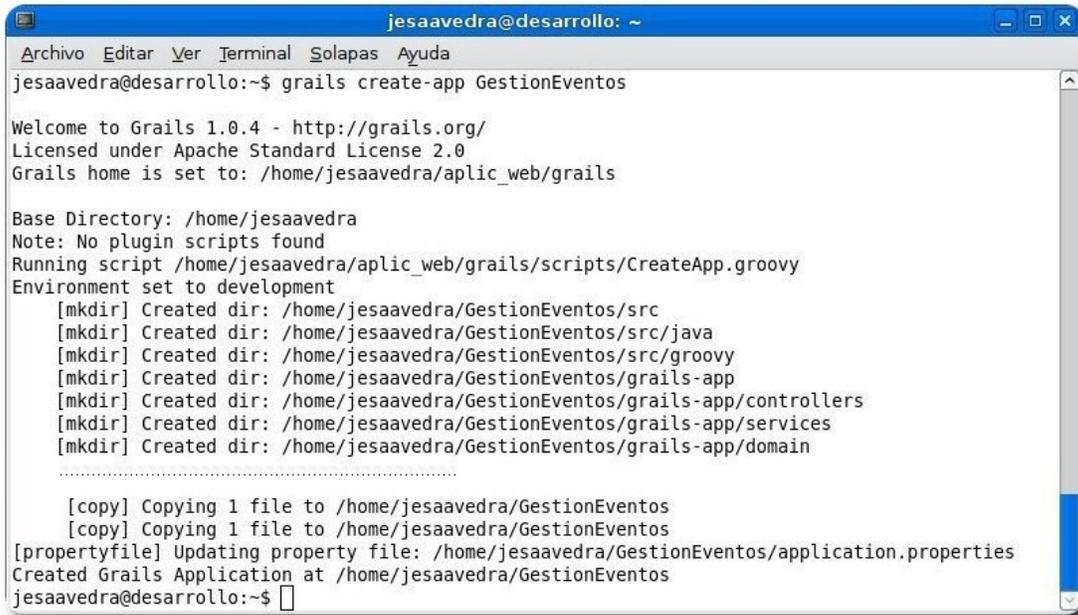
Gráfico 2: Diagrama E/R de nuestra aplicación

Para una mejor comprensión del potencial de Grails en el desarrollo de aplicaciones web, realizaremos una aplicación paso a paso.

## Estructura de un Proyecto

Hoy en día, los frameworks mantienen una forma similar de estructurar sus proyectos, imaginemos como si se tratase de un organigrama donde se consideran las siguientes partes:

1. Un proyecto contiene una o varias aplicaciones
2. Una aplicación tiene uno o varios módulos
3. Un módulo tiene una o varias acciones
4. Una acción puede o no estar asociada a una vista
5. Creación de la aplicación



```

jesaavedra@desarrollo: ~
Archivo Editar Ver Terminal Solapas Ayuda
jesaavedra@desarrollo:~$ grails create-app GestionEventos

Welcome to Grails 1.0.4 - http://grails.org/
Licensed under Apache Standard License 2.0
Grails home is set to: /home/jesaavedra/aplic_web/grails

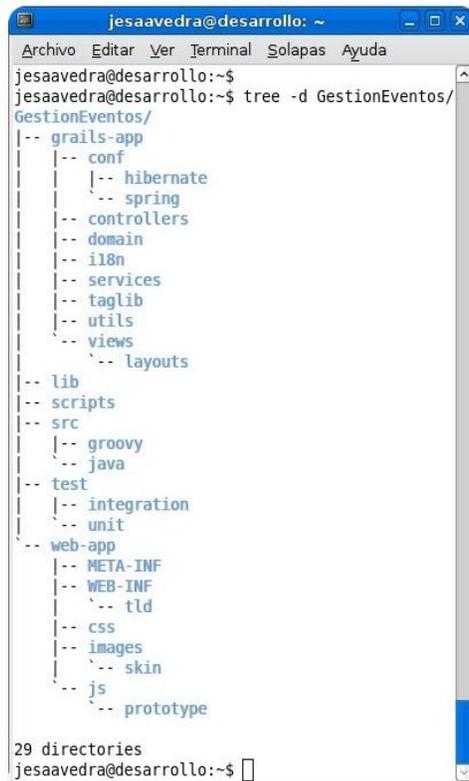
Base Directory: /home/jesaavedra
Note: No plugin scripts found
Running script /home/jesaavedra/aplic_web/grails/scripts/CreateApp.groovy
Environment set to development
[mkdir] Created dir: /home/jesaavedra/GestionEventos/src
[mkdir] Created dir: /home/jesaavedra/GestionEventos/src/java
[mkdir] Created dir: /home/jesaavedra/GestionEventos/src/groovy
[mkdir] Created dir: /home/jesaavedra/GestionEventos/grails-app
[mkdir] Created dir: /home/jesaavedra/GestionEventos/grails-app/controllers
[mkdir] Created dir: /home/jesaavedra/GestionEventos/grails-app/services
[mkdir] Created dir: /home/jesaavedra/GestionEventos/grails-app/domain
.....
[copy] Copying 1 file to /home/jesaavedra/GestionEventos
[copy] Copying 1 file to /home/jesaavedra/GestionEventos
[propertyfile] Updating property file: /home/jesaavedra/GestionEventos/application.properties
Created Grails Application at /home/jesaavedra/GestionEventos
jesaavedra@desarrollo:~$

```

Gráfico 3: Crear una aplicación

## Estructura de directorios

Al igual que la mayoría de los frameworks orientados al desarrollo de aplicaciones web, Grails cuando crea los proyectos y aplicaciones, por defecto crea una estructura de directorios definida, estructura que nos permite mantener un orden y conocer exactamente donde se encuentran todos y cada uno de los componentes de nuestra aplicación.



```

jesaavedra@desarrollo: ~
Archivo Editar Ver Terminal Solapas Ayuda
jesaavedra@desarrollo:~$ tree -d GestionEventos/
GestionEventos/
|-- grails-app
|   |-- conf
|   |   |-- hibernate
|   |   |-- spring
|   |-- controllers
|   |-- domain
|   |-- i18n
|   |-- services
|   |-- tagLib
|   |-- utils
|   |-- views
|   |-- layouts
|-- lib
|-- scripts
|-- src
|   |-- groovy
|   |-- java
|-- test
|   |-- integration
|   |-- unit
|-- web-app
|   |-- META-INF
|   |-- WEB-INF
|   |-- tld
|   |-- css
|   |-- images
|   |-- skin
|   |-- js
|   |-- prototype
29 directories
jesaavedra@desarrollo:~$

```

Gráfico 4: Estructura de la aplicación

## Ejecución de la aplicación

Otra característica similar que poseen la mayoría de los frameworks, va referida a que estos vienen con un pequeño servidor de aplicaciones que nos permite ejecutar y probar nuestros desarrollos, debemos considerar que estos pequeños servidores de aplicación solo deben ser utilizados en etapas de prueba o pequeños desarrollos, ya que para entornos mas grandes o de producción debemos hacer uso de servidores de aplicación mas robustos, orientados y con soporte al lenguaje propio del framework; en el caso de Grails se puede hacer uso de Tomcat, Jboss, Websphere entre otros.

El pequeño servidor de aplicaciones que acompaña a Grails es Jetty, como lo muestra la figura.



```
jesaavedra@desarrollo: ~/GestionEventos
Archivo Editar Ver Terminal Solapas Ayuda
jesaavedra@desarrollo:~/GestionEventos$
jesaavedra@desarrollo:~/GestionEventos$ grails run-app

Welcome to Grails 1.0.4 - http://grails.org/
Licensed under Apache Standard License 2.0
Grails home is set to: /home/jesaavedra/aplic_web/grails

Base Directory: /home/jesaavedra/GestionEventos
Note: No plugin scripts found
Running script /home/jesaavedra/aplic_web/grails/scripts/RunApp.groovy
Environment set to development
[mkdir] Created dir: /home/jesaavedra/.grails/1.0.4/projects/GestionEventos/classes
[groovyc] Compiling 6 source files to /home/jesaavedra/.grails/1.0.4/projects/GestionEventos/classes
.....
2009-02-27 09:56:10.423:/GestionEventos:INFO: Initializing Spring FrameworkServlet 'grails'
2009-02-27 09:56:10.533::INFO: Started SelectChannelConnector@0.0.0.0:8080
Server running. Browse to http://localhost:8080/GestionEventos
```

Gráfico 5: Iniciando la aplicación

## Ejecutando nuestra aplicación



Gráfico 6: La aplicación ejecutándose

## Modelado del dominio

Para describir el dominio de nuestra aplicación, primero debemos crear los dominios correspondientes, de la siguiente forma para todos y cada uno de los dominios de nuestra aplicación:



```

jesaavedra@desarrollo: ~/GestionEventos
jesaavedra@desarrollo:~/GestionEventos$ grails create-domain-class Evento

Welcome to Grails 1.0.4 - http://grails.org/
Licensed under Apache Standard License 2.0
Grails home is set to: /home/jesaavedra/aplic_web/grails

Base Directory: /home/jesaavedra/GestionEventos
Note: No plugin scripts found
Running script /home/jesaavedra/aplic_web/grails/scripts/CreateDomainClass.groovy
Environment set to development
  [copy] Copying 1 file to /home/jesaavedra/GestionEventos/grails-app/domain
Created Domain Class for Evento
  [copy] Copying 1 file to /home/jesaavedra/GestionEventos/test/integration
Created Tests for Evento
jesaavedra@desarrollo:~/GestionEventos$
jesaavedra@desarrollo:~/GestionEventos$

```

**Gráfico 7:** Creando los dominios

Grails cuando crea un dominio, inicialmente crea sólo clases vacías en el directorio **GestionEventos/grails-app/domain/**, las cuales debemos modificar e incluir los siguientes contenidos:

### **/grails-app/domain/Evento.groovy**

```

class Evento {
    String nombre
    String ciudad
    Date inicia
    Date finaliza
    String sitioweb
}

```

### **/grails-app/domain/Expositor.groovy**

```

class Expositor {
    String paterno
    String materno
    String nombre
    String nacionalidad
    String email
}

```

### **/grails-app/domain/Conferencia.groovy**

```

class Conferencia {
    String titulo
    String resumen
    Date fecha
}

```

## Relaciones

Las relaciones de nuestro modelo deben verse reflejados en nuestros dominios. Grails permite mantener relaciones de 1 a 1, de 1 a muchos y de muchos a muchos. Para nuestro ejemplo las relaciones son las siguientes:

- ✓ Un evento tiene muchas conferencias
- ✓ Un expositor puede impartir muchas conferencias

Estas relaciones deben figurar en los dominios y ficheros correspondientes como se muestra a continuación:

### **/grails-app/domain/Evento.groovy**

```
class Evento {
    String nombre
    String ciudad
    Date inicia
    Date finaliza
    String sitioweb
    static hasMany = [conferencias:Conferencia]
}
```

### **/grails-app/domain/Expositor.groovy**

```
class Expositor {
    String paterno
    String materno
    String nombre
    String nacionalidad
    String email
    static hasMany = [conferencias:Conferencia]
}
```

### **/grails-app/domain/Conferencia.groovy**

```
class Conferencia {
    String titulo
    String resumen
    Date fecha
    Evento evento
    Expositor expositor
    static belongsTo = [Evento,Expositor]
}
```

## Scaffolding

Scaffolding es un término introducido por Rails, para representar la generación de los artefactos, que permiten un uso más sencillo y rápido de las interfaces que permiten operaciones básicas, denominados de forma genérica CRUDs (Create, Read, Update y Delete) o ABDs (Altas Bajas y Modificaciones).

Grails permite 2 tipos de scaffolding, dinámicas y estáticas; en esta primera parte del tutorial sólo consideraremos el scaffolding dinámico.

Para cada uno de los dominios, para los que se desee tener un scaffold, se debe crear de forma análoga la creación y adecuación del controlador correspondiente, como se muestra a continuación.

```

jesaavedra@desarrollo: ~/GestionEventos
Archivo Editar Ver Terminal Solapas Ayuda
jesaavedra@desarrollo:~/GestionEventos$
jesaavedra@desarrollo:~/GestionEventos$ grails create-controller Evento

Welcome to Grails 1.0.4 - http://grails.org/
Licensed under Apache Standard License 2.0
Grails home is set to: /home/jesaavedra/aplic_web/grails

Base Directory: /home/jesaavedra/GestionEventos
Note: No plugin scripts found
Running script /home/jesaavedra/aplic_web/grails/scripts/CreateController.groovy
Environment set to development
  [copy] Copying 1 file to /home/jesaavedra/GestionEventos/grails-app/controllers
Created Controller for Evento
  [copy] Copying 1 file to /home/jesaavedra/GestionEventos/test/integration
Created ControllerTests for Evento
jesaavedra@desarrollo:~/GestionEventos$
jesaavedra@desarrollo:~/GestionEventos$ █

```

Gráfico 8: Creando los controladores

Para esto debemos realizar las siguiente operaciones:

- ✓ Crear los controladores para cada uno de los dominios.
- ✓ Modificar los controladores, de tal forma que permitan generar los scaffold de forma dinámica.

Los controladores para nuestra aplicación deberían tener los siguientes contenidos:

**/grails-app/controllers/EventoController.groovy**

```
class EventoController {
    def scaffold = Evento
}
```

**/grails-app/controllers/ExpositorController.groovy**

```
class ExpositorController {
    def scaffold = Expositor
}
```

**/grails-app/controllers/ConferenciaController.groovy**

```
class ConferenciaController {
    def scaffold = Conferencia
}
```

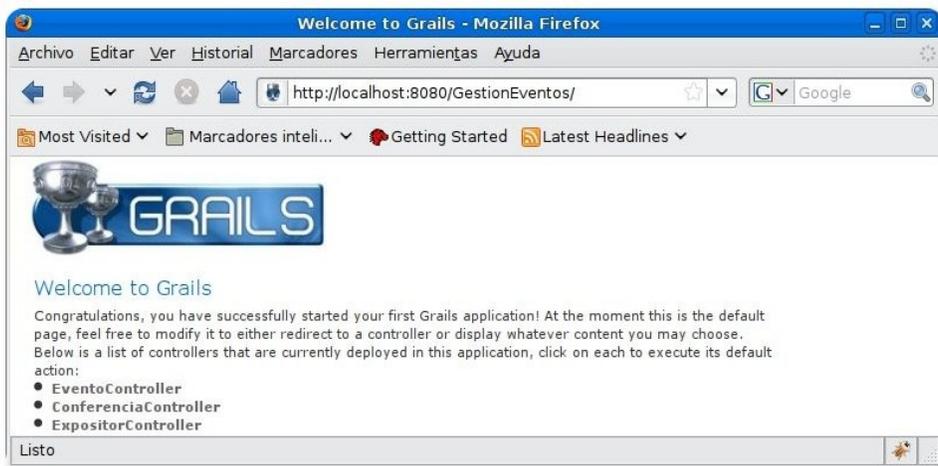


Gráfico 9: Nuestra aplicación y sus controladores

## Ejecución y pruebas

Una vez corriendo nuestra aplicación podemos empezar a probar todas y cada una de las características provistas por los scaffolds generados: Formularios, Ventanas de diálogo, Relaciones, Efectos JavaScript en ciertos campos, etc,...

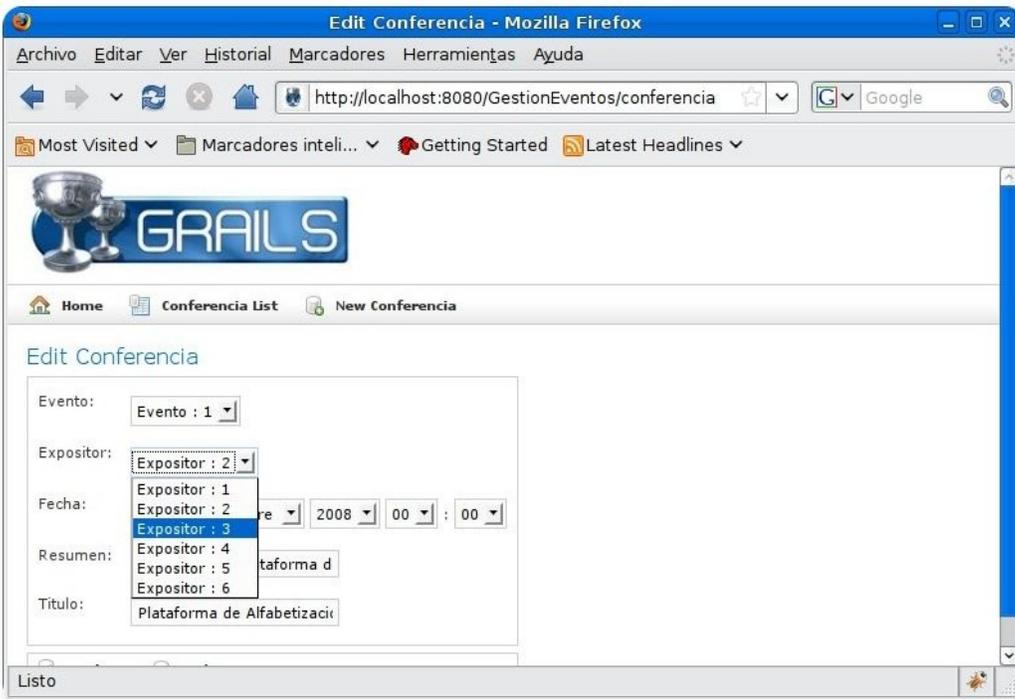
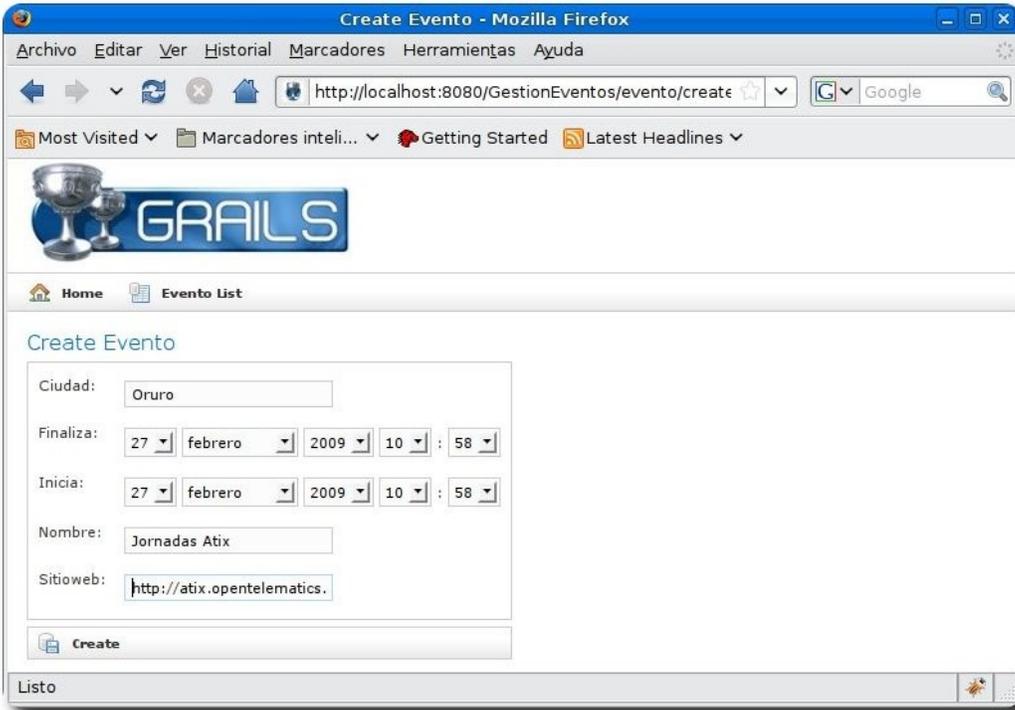


Gráfico 10: Efectos varios en los formularios de creación

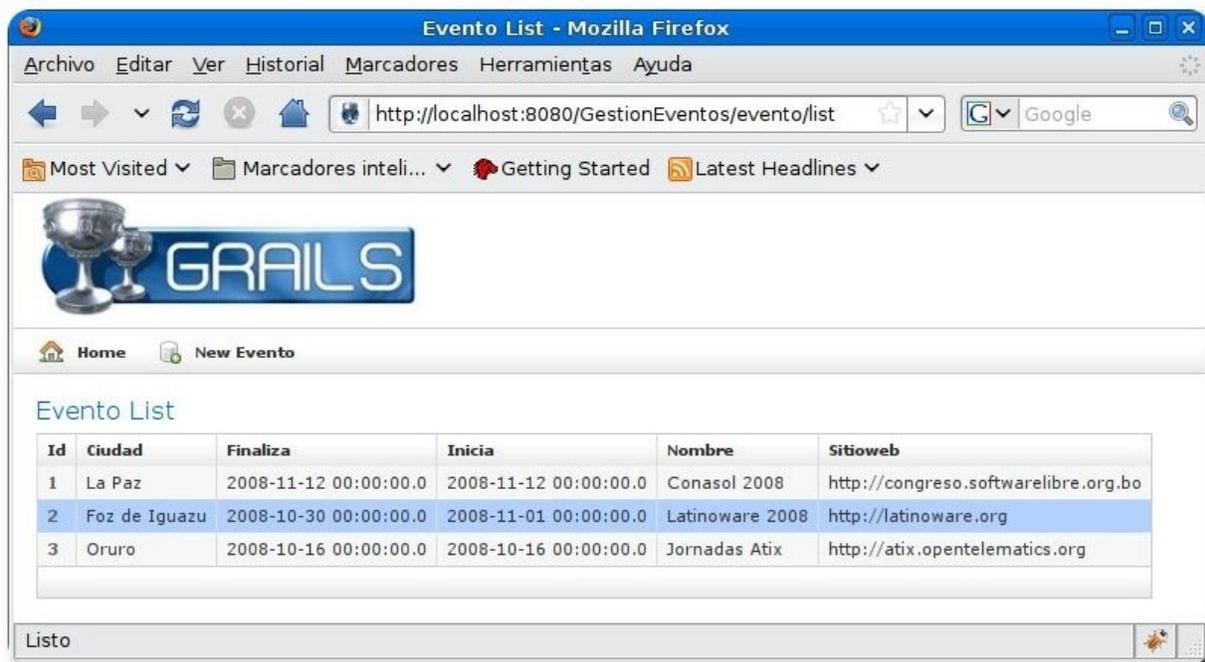


Gráfico 11: Listado de Eventos

## Referencias

- [1] <http://www.grails.org>
- [2] Getting Started with Grails
- [3] The Definitive Guide to Grails

## Autor



### Esteban Saavedra López

Líder de la Comunidad ATIX (Oruro – Bolivia)  
Activista de Software Libre en Bolivia  
[jesaavedra@opentelematics.org](mailto:jesaavedra@opentelematics.org)  
<http://jesaavedra.opentelematics.org>

# Gestión de Certificados Digitales con OpenSSL (2da parte)

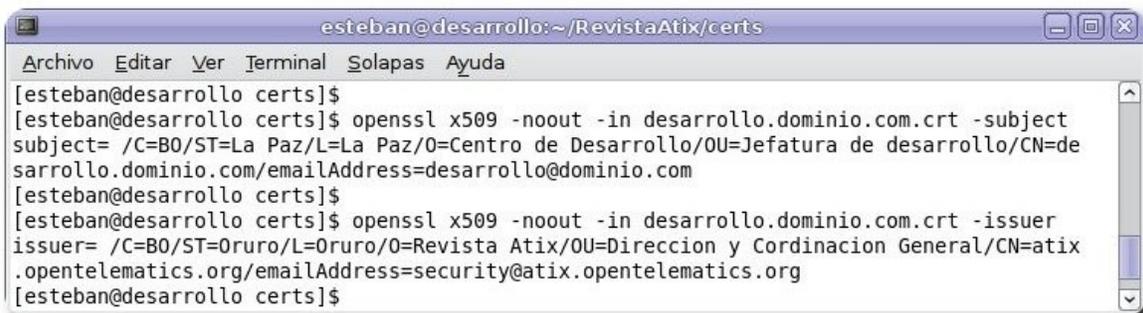
Hoy por hoy el uso de certificados digitales se ha hecho tan frecuente e importante, sobre todo la momento de garantizar la privacidad y seguridad tanto en el intercambio de documentos como en establecer comunicaciones seguras en los distintos servicios que hacen uso del Internet.

## Operaciones sobre certificados digitales

**OpenSSL** provee muchas operaciones criptográficas como encriptación y desencriptación de datos, creación y verificación de resúmenes (digests), cálculo de pares de claves públicas y privadas, manipulación de certificados, etc.

## Detalles de los certificado

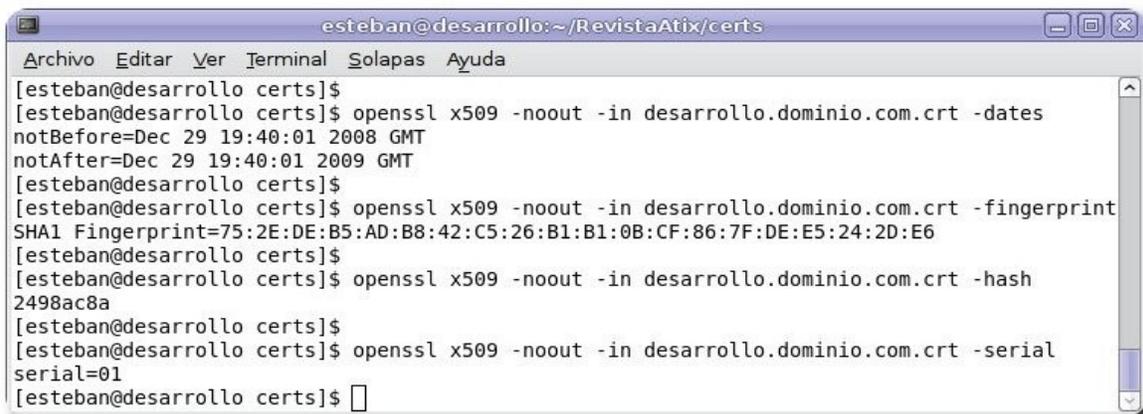
Antes de realizar operaciones sobre los certificados digitales y con estos, primeramente es necesario conocer los detalles de los mismos, son varias las opciones que provee **OpenSSL**, así como se muestran en las figuras siguientes:



```

esteban@desarrollo:~/RevistaAtix/certs
Archivo Editar Ver Terminal Solapas Ayuda
[esteban@desarrollo certs]$
[esteban@desarrollo certs]$ openssl x509 -noout -in desarrollo.dominio.com.crt -subject
subject= /C=B0/ST=La Paz/L=La Paz/O=Centro de Desarrollo/OU=Jefatura de desarrollo/CN=de
sarrollo.dominio.com/emailAddress=desarrollo@dominio.com
[esteban@desarrollo certs]$
[esteban@desarrollo certs]$ openssl x509 -noout -in desarrollo.dominio.com.crt -issuer
issuer= /C=B0/ST=Oruro/L=Oruro/O=Revista Atix/OU=Direccion y Cordinacion General/CN=atix
.opentelematics.org/emailAddress=security@atix.opentelematics.org
[esteban@desarrollo certs]$
  
```

Detalles del dueño del certificado y de la CA que firma el certificado



```

esteban@desarrollo:~/RevistaAtix/certs
Archivo Editar Ver Terminal Solapas Ayuda
[esteban@desarrollo certs]$
[esteban@desarrollo certs]$ openssl x509 -noout -in desarrollo.dominio.com.crt -dates
notBefore=Dec 29 19:40:01 2008 GMT
notAfter=Dec 29 19:40:01 2009 GMT
[esteban@desarrollo certs]$
[esteban@desarrollo certs]$ openssl x509 -noout -in desarrollo.dominio.com.crt -fingerprint
SHA1 Fingerprint=75:2E:DE:B5:AD:B8:42:C5:26:B1:B1:0B:CF:86:7F:DE:E5:24:2D:E6
[esteban@desarrollo certs]$
[esteban@desarrollo certs]$ openssl x509 -noout -in desarrollo.dominio.com.crt -hash
2498ac8a
[esteban@desarrollo certs]$
[esteban@desarrollo certs]$ openssl x509 -noout -in desarrollo.dominio.com.crt -serial
serial=01
[esteban@desarrollo certs]$
  
```

Detalles de vigencia del certificado, huella digital, hash y el serial correspondiente

```

esteban@desarrollo:~/RevistaAtix/certs
Archivo Editar Ver Terminal Solapas Ayuda
[esteban@desarrollo certs]$
[esteban@desarrollo certs]$ openssl x509 -noout -in desarrollo.dominio.com.crt -text
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number: 1 (0x1)
    Signature Algorithm: sha1WithRSAEncryption
    Issuer: C=B0, ST=Oruro, L=Oruro, O=Revista Atix, OU=Direccion y Cordinacion Ge
    neral, CN=atix.opentelematics.org/emailAddress=security@atix.opentelematics.org
    Validity
      Not Before: Dec 29 19:40:01 2008 GMT
      Not After : Dec 29 19:40:01 2009 GMT
    Subject: C=B0, ST=La Paz, L=La Paz, O=Centro de Desarrollo, OU=Jefatura de des
    arrollo, CN=desarrollo.dominio.com/emailAddress=desarrollo@dominio.com
    Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
      RSA Public Key: (1024 bit)
      Modulus (1024 bit):
        00:db:35:20:f4:72:55:89:f9:52:4c:dd:0d:0e:46:
        8a:4d:32:14:12:f7:07:52:48:d3:48:2a:96:e1:05:
        2c:f2:6b:16:76:c1:b5:81:f3:ec:7b:15:30:01:58:
        5a:8d:db:2c:3b:d7:e2:f7:a4:cd:91:fc:1b:ee:f4:
        9f:64:9f:9d:e6:72:d2:34:a7:9f:90:18:a8:30:f3:
        a4:7e:dc:d8:7d:63:b9:fd:19:7b:70:5a:95:9c:93:
        c1:be:ed:a7:14:6e:8f:e0:56:ac:2e:49:36:df:d8:
        bf:a7:0f:92:99:bf:16:fd:7c:0b:bc:a0:d3:a7:c6:
        be:75:7e:d8:42:55:82:89:af
      Exponent: 65537 (0x10001)
    X509v3 extensions:
      X509v3 Basic Constraints:
        CA:FALSE
      Netscape Comment:
        OpenSSL Generated Certificate
      X509v3 Subject Key Identifier:
        04:22:1E:4F:B4:B8:CB:05:46:31:72:3D:8A:82:45:06:6B:D1:E4:54
      X509v3 Authority Key Identifier:
        keyid:AA:C4:3B:0D:C1:D7:BA:83:32:C4:8D:2E:4F:27:5C:9B:F5:C0:DC:61

    Signature Algorithm: sha1WithRSAEncryption
    61:2e:64:d5:36:2d:7a:50:ae:de:25:cd:ef:60:02:f4:52:2c:
    ae:ac:65:b0:d6:a9:b3:b2:30:a4:36:07:ad:b3:2f:32:ce:e9:
    ea:84:5e:fd:0d:b5:97:7c:d8:7b:99:a2:ce:ce:64:a9:34:01:
    5e:91:f3:7d:50:a7:7e:71:91:cb:5d:67:60:46:ac:23:4d:81:
  
```

Detalles generales del certificado de forma conjunta

## Resúmenes de archivos

Los resúmenes (digests) de mensajes (o archivos) son usados para asegurar que un mensaje (o archivo) es válido y no ha sido modificado durante la transmisión.

El resumen es creado aplicando una función de hash sobre el mensaje (o archivo) original. Es extremadamente difícil (sino imposible) encontrar dos mensajes para los cuales se obtenga un mismo valor de resumen.

Los resúmenes son usados comúnmente para proveer integridad a los archivos transmitidos, ellos además poseen propiedades de autenticación y no-repudio para algún tipo de datos a través de firmas digitales.

Un resumen encriptado con la clave privada del autor del archivo es lo equivalente a la firma digital del archivo; otros podrían verificar el resumen del archivo usando la clave pública del

autor.

Para poder obtener resúmenes de archivos, primeramente debemos generar un par de llaves (publica y privada), así como se muestran en las figuras:

```

esteban@desarrollo:~
Archivo Editar Ver Terminal Solapas Ayuda
[esteban@desarrollo ~]$
[esteban@desarrollo ~]$ openssl genrsa -out privada.key 1024
Generating RSA private key, 1024 bit long modulus
.....+++++
.....+++++
e is 65537 (0x10001)
[esteban@desarrollo ~]$
[esteban@desarrollo ~]$ openssl rsa -in privada.key -pubout -out publica.key
writing RSA key
[esteban@desarrollo ~]$
    
```

**OpenSSL**, provee algunas variantes de como hacer uso de los resúmenes, como muestran las figuras siguientes:

```

esteban@desarrollo:~/RevistaAtix/certs
Archivo Editar Ver Terminal Solapas Ayuda
[esteban@desarrollo certs]$
[esteban@desarrollo certs]$ openssl list-message-digest-commands
md2
md4
md5
rmd160
sha
sha1
[esteban@desarrollo certs]$
    
```

Los resúmenes obtenidos por **OpenSSL** son los mismos que se obtienen por utilidades como md5sum y sha1sum, así como se muestra en las figuras siguientes:

```

esteban@desarrollo:~
Archivo Editar Ver Terminal Solapas Ayuda
[esteban@desarrollo ~]$
[esteban@desarrollo ~]$ openssl dgst -md5 presentacion.pdf
MD5(presentacion.pdf)= 177376771ddf12f0ba676ba7cc996c5c
[esteban@desarrollo ~]$
[esteban@desarrollo ~]$ md5sum presentacion.pdf
177376771ddf12f0ba676ba7cc996c5c presentacion.pdf
[esteban@desarrollo ~]$
[esteban@desarrollo ~]$ openssl dgst -sha1 presentacion.pdf
SHA1(presentacion.pdf)= 5133cdccf54c6980d0868c65e375f9ba24cdca93
[esteban@desarrollo ~]$
[esteban@desarrollo ~]$ sha1sum presentacion.pdf
5133cdccf54c6980d0868c65e375f9ba24cdca93 presentacion.pdf
[esteban@desarrollo ~]$
    
```

### Firma digital

Una de las grandes utilidades de los resúmenes es la de poder simular una firma normal, básicamente consiste en obtener el resumen de un archivo y cifrar este resumen para protegerlo y de esta forma poder garantizar que cierto archivo fue enviado conjunto a su resumen cifrado por una persona en particular, aspecto que es posibilita realizar la verificación haciendo uso del archivo resumen, la llave pública y el archivo origen, así como se muestra en la figura siguiente:



```

esteban@desarrollo:~
Archivo Editar Ver Terminal Solapas Ayuda
[esteban@desarrollo ~]$
[esteban@desarrollo ~]$ openssl enc -d -aes-256-cbc -a -in presentacion.enc -out presentacion.pdf
enter aes-256-cbc decryption password:
[esteban@desarrollo ~]$
    
```

Descifrado simétrico

## Cifrado asimétrico

Las técnicas de cifrado asimétrico se basan en el uso de dos claves, una pública y otra privada, de modo que lo que una de ellas cifra, sólo puede descifrarlo la otra, y viceversa. Se inventaron para evitar el problema del intercambio de claves de los sistemas de cifrado simétricos. Tanto el emisor como el receptor poseen dos claves: una privada (conocida sólo por él) y una pública (conocida por cualquiera), de manera que no es necesario que el emisor y el receptor intercambien claves secretas.

```

esteban@desarrollo:~
Archivo Editar Ver Terminal Solapas Ayuda
[esteban@desarrollo ~]$
[esteban@desarrollo ~]$ openssl rsautl -encrypt -pubin -inkey publica.key -in mensaje.txt -out mensaje.enc
[esteban@desarrollo ~]$
    
```

Cifrado simétricos

```

esteban@desarrollo:~
Archivo Editar Ver Terminal Solapas Ayuda
[esteban@desarrollo ~]$
[esteban@desarrollo ~]$ openssl rsautl -decrypt -inkey privada.key -in mensaje.enc -out mensaje.txt
[esteban@desarrollo ~]$
    
```

Descifrado asimétrico

Un aspecto importante a considerar en el cifrado asimétrico, va referido al tamaño del archivo origen (`mensaje.txt`) no debe ser mayor a 128 bytes o 1024 bits.

A diferencia del cifrado simétrico, el comando `rsautl` no admite textos largos, tan sólo de unos. Esto es debido a que normalmente no se cifran grandes documentos mediante algoritmos asimétricos. Como ya se ha comentado los documentos se cifran con algoritmos simétricos sólo las claves secretas se cifran con algoritmos asimétricos.

En caso de desear encriptar asimétricamente archivos de mayor tamaño, se puede hacer uso de la opción `smime`, tal como se muestra en las figuras siguientes:

```

esteban@desarrollo:~
Archivo Editar Ver Terminal Solapas Ayuda
[esteban@desarrollo ~]$
[esteban@desarrollo ~]$ openssl smime -encrypt -in texto.txt -out texto.enc esteban.crt
[esteban@desarrollo ~]$
    
```

Cifrado asimétrico

```

esteban@desarrollo:~
Archivo Editar Ver Terminal Solapas Ayuda
[esteban@desarrollo ~]$
[esteban@desarrollo ~]$ openssl smime -decrypt -in texto.enc -out texto.txt -inkey esteban.key -recip esteban.crt
Enter pass phrase for esteban.key:
[esteban@desarrollo ~]$

```

Descifrado asimétrico

## Confidencialidad: Cifrar

- ✓ **cifrar un mensaje para enviarlo a un receptor:** si el emisor desea transmitir un mensaje al receptor, de manera que sólo el receptor pueda conocer su contenido generará el mensaje cifrado utilizando un algoritmo de cifrado asimétrico y la clave pública del receptor. En este caso sólo se garantiza la confidencialidad del mensaje, ya que sólo el receptor, utilizando el algoritmo inverso y su clave privada podrá reproducir el mensaje original, pero no la identidad del emisor ni la integridad del mensaje. Este método se denomina cifrado simple.
- ✓ **cifrar un mensaje para guardarlo con seguridad:** si el mensaje lo ciframos con nuestra clave pública, sólo nosotros podremos descifrarlo. Esto se utiliza para proteger archivos que no queremos que nadie pueda leer.

## Autenticación: Firmar

- ✓ Si el emisor desea transmitir un mensaje al receptor, de manera que éste tenga garantizado que la única persona que pudo haberlo generado es el emisor, generará el mensaje cifrado utilizando un algoritmo de cifrado asimétrico y su clave privada. El receptor, utilizando el algoritmo inverso y la clave pública del emisor podrá reproducir el mensaje original. En este caso sólo se asegura la autenticación, no la confidencialidad (ya que la clave pública del emisor es conocida por cualquiera, por lo que cualquiera podría haber obtenido el mensaje) ni la integridad. Este método se denomina firma digital.

## Confidencialidad + Autenticación: Cifrar + Firmar

Cuando el emisor desea transmitir un mensaje al receptor asegurando la confidencialidad y la autenticación, realizará los dos pasos anteriores consecutivamente:

- ✓ primero cifra el mensaje con la clave pública del receptor.
- ✓ a continuación lo firma con su clave privada.

El receptor, utilizando el algoritmo inverso, su clave privada y la clave pública del emisor podrá reproducir el mensaje original. Este método se denomina doble cifrado.

Los algoritmos de cifrado asimétrico más utilizados son RSA (libre) y DSA (libre).

## Cifrado híbrido

La complejidad computacional de los algoritmos de cifrado asimétrico hace que sea muy costoso el uso del doble cifrado, por lo que generalmente se utiliza el cifrado híbrido, que combina cifrado asimétrico, cifrado simétrico y funciones hash. Herramientas como GPG,

OpenSSH y **OpenSSL** utilizan cifrado híbrido: criptografía asimétrica para intercambiar la clave de cifrado simétrico y criptografía simétrica para la transmisión de la información.

**OpenSSL** permite cifrado simétrico/asimétrico de datos a través de varios algoritmos y sus variantes.

## Utilidades de OpenSSL

**OpenSSL**, permite generar claves, con la particularidad que esta cada vez es distinta, así como muestra la figura.

```

esteban@desarrollo:~
Archivo Editar Ver Terminal Solapas Ayuda
[esteban@desarrollo ~]$
[esteban@desarrollo ~]$ openssl passwd miclave
0DxYe8TllFv2c
[esteban@desarrollo ~]$
[esteban@desarrollo ~]$ openssl passwd miclave
.6j2ofufae4No
[esteban@desarrollo ~]$
    
```

Para repetir una clave hacemos uso de la opción **-salt** seguido de los 2 caracteres iniciales del hash generado, así como muestra la figura.

```

esteban@desarrollo:~
Archivo Editar Ver Terminal Solapas Ayuda
[esteban@desarrollo ~]$
[esteban@desarrollo ~]$ openssl passwd -salt at miclave
atNDjFsqsD70s
[esteban@desarrollo ~]$
[esteban@desarrollo ~]$ openssl passwd -salt at miclave
atNDjFsqsD70s
[esteban@desarrollo ~]$
    
```

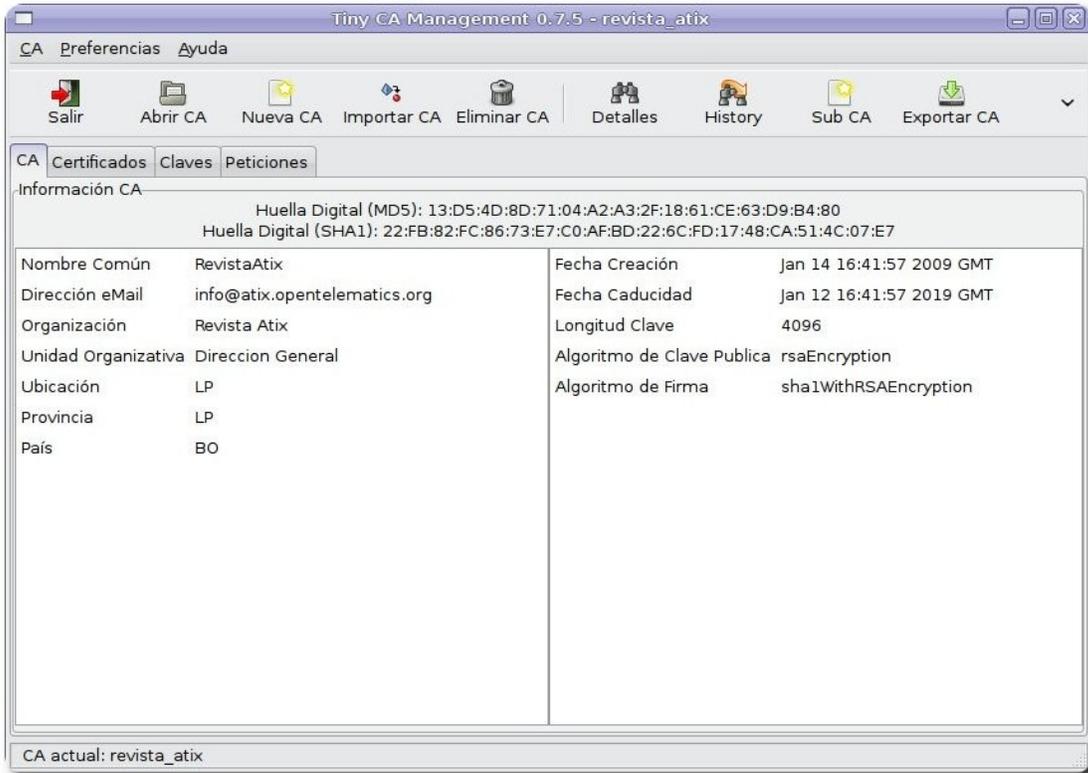
Para generar hashes al estilo **shadow** en sistemas GNU/Linux, podemos hacerlo con la opción **-1** y para repetir la clave hacemos uso de la opción **-salt** seguido de los 8 caracteres situados entre el segundo y tercer \$, como se muestra en la figura.

```

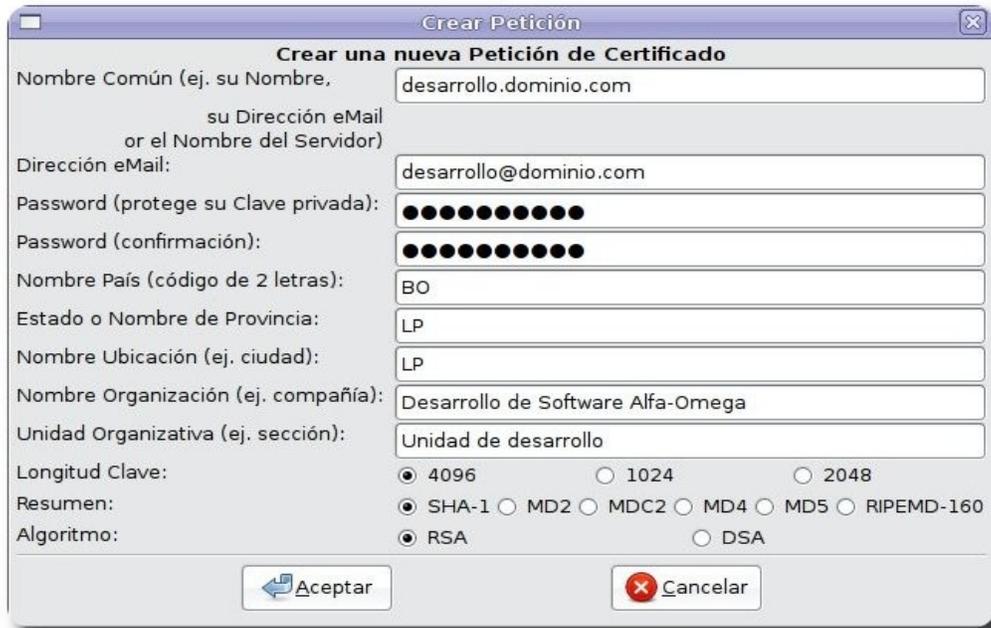
esteban@desarrollo:~
Archivo Editar Ver Terminal Solapas Ayuda
[esteban@desarrollo ~]$
[esteban@desarrollo ~]$ openssl passwd -1 miclave
$1$uc0Yw0aR$GudhMXXnDUeBtbGSnGbwa/
[esteban@desarrollo ~]$
[esteban@desarrollo ~]$ openssl passwd -1 -salt uc0Yw0aR miclave
$1$uc0Yw0aR$GudhMXXnDUeBtbGSnGbwa/
[esteban@desarrollo ~]$
    
```

## Frontends

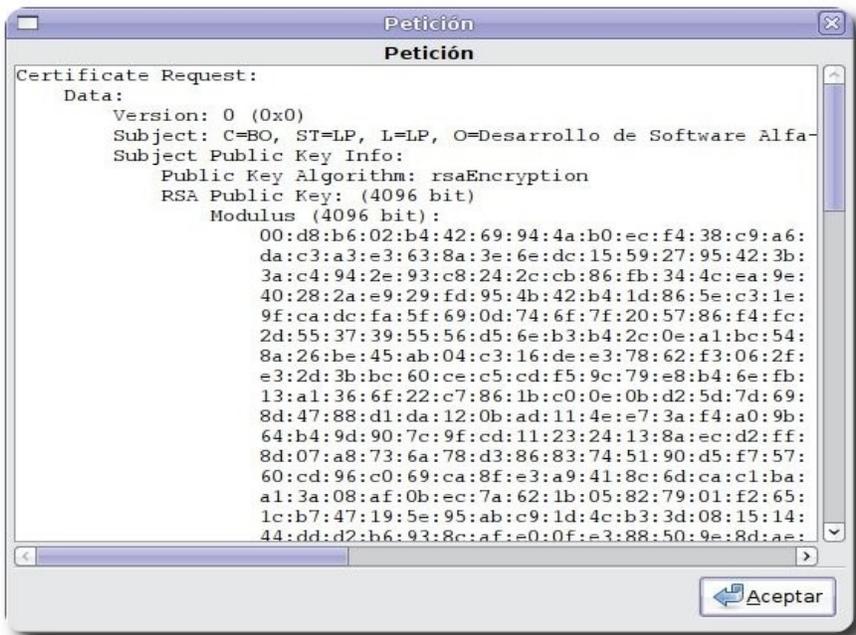
Actualmente existen varios frontends, para la gestión de certificados digitales, desde las aplicaciones de escritorio hasta aplicaciones accesibles vía browser (OpenCA); de entre todos uno de los frontends mas difundidos es el TinyCA, que se caracteriza por su sencillez y facilidad al momento de hacer uso del mismo. A continuación presentamos algunas imágenes que muestran algunas de sus capacidades:



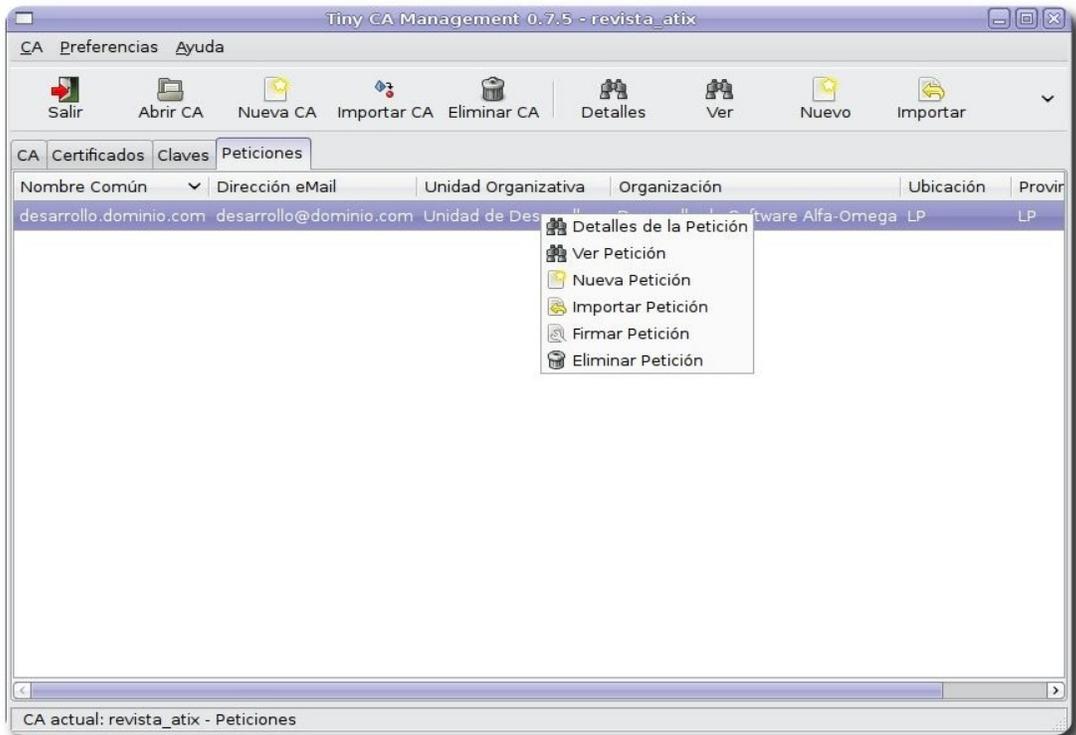
Detalles de la Entidad Certificadora



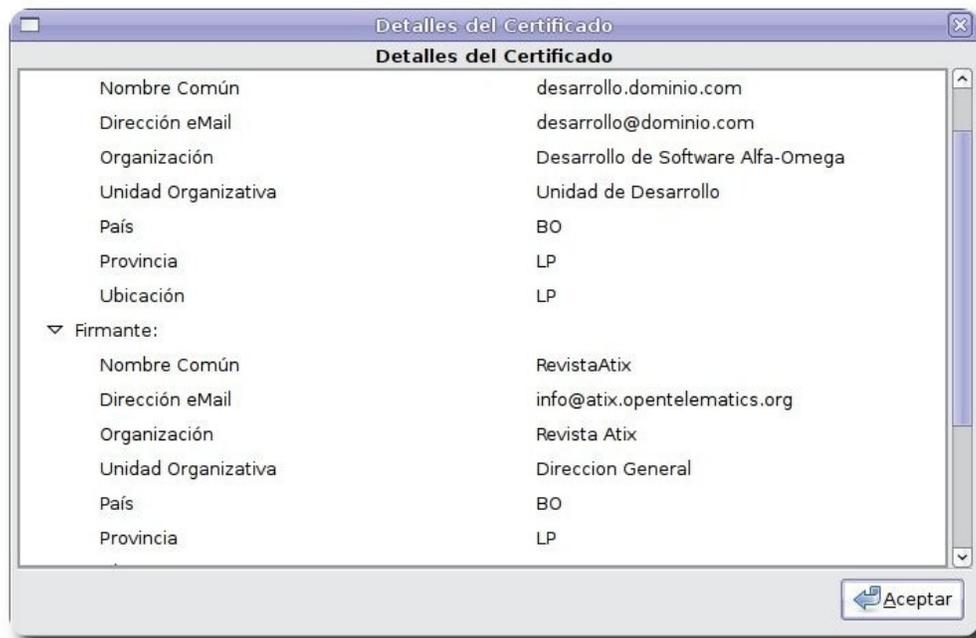
Creación de un solicitud de certificado



Vista de la petición de certificado



Opciones permitidas para una solicitud de certificado



Detalles del certificado

## Referencias

- [1] <http://www.openssl.org/>
- [2] <http://es.wikipedia.com>

## Autores



**Esteban Saavedra López**  
 Líder de la Comunidad ATIX (Oruro – Bolivia)  
 Activista de Software Libre en Bolivia  
[jesaavedra@opentelematics.org](mailto:jesaavedra@opentelematics.org)  
<http://jesaavedra.opentelematics.org>

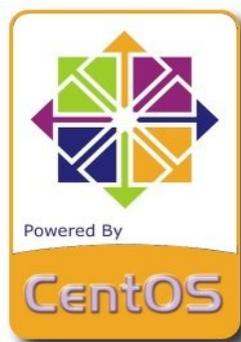


**Joseph Sandoval Falomici**  
 Profesor universitario  
 Entusiasta de Software Libre  
[josephsandoval@gmail.com](mailto:josephsandoval@gmail.com)

# CentOS

## Directory Server (CentOS 5)

Un servicio de directorio (SD) es una aplicación o un conjunto de aplicaciones que almacena y organiza la información sobre los usuarios de una red de ordenadores, sobre recursos de red, y permite a los administradores gestionar el acceso de usuarios a los recursos sobre dicha red. Además, los servicios de directorio actúan como una capa de abstracción entre los usuarios y los recursos compartidos.



### Introducción

**CentOS Directory Server** es un servidor de directorios basado en RedHat Directory Server, a la manera de un fork, tal como la distro RedHat Enterprise Linux y Fedora.

Al parecer la mayor parte de las componentes del producto de software están bajo GPL, pero otras tienen copyright de RedHat.

### Características

- ✓ Soporta LDAPv3.
- ✓ Documentación extensa, ya que utiliza la de RedHat Directory Server, la cual puede ser accedida desde internet sin necesidad de tener una licencia de RedHat.
- ✓ Soporta autenticación segura con SSLv3, TLSv1 y SASL.
- ✓ Implementa características avanzadas de replicación, es decir, Multimaster Replication o Replicación Multi Maestro.

- ✓ Puede sincronizar usuarios y grupos con Active Directory (vamos, a mi no me agrada mucho andar sincronizando cosas con productos de la Microsoft, pero es una característica y hay que mencionarla).
- ✓ Posee una interfaz gráfica de administración muy completa en donde centraliza todo lo relacionado al manejo de los servicios, usuarios, grupos, logs y es posible realizar respaldos, entre otras cosas.

### Aplicaciones del Servicio de Directorio

Las aplicaciones o utilidades de un servicio de directorio son variadas, y a continuación se muestra una lista de algunas de ellas:

- ✓ Principalmente se utiliza para propósitos de autenticación, por lo cual cualquier aplicación que lo necesite, y posea soporte para LDAP, podrá autenticarse y obtener la información que necesita del servidor de directorios, tal como dirección de correo electrónico, teléfono, dirección, etc.
- ✓ Existen diferentes servicios que soportan LDAP, por ejemplo el proxy Squid, servidor smtp Postfix, y se utiliza para implementar "ldap naming".
- ✓ El cliente de correo Thunderbird (al igual que otros, como Evolution)

posee soporte para LDAP, por lo cual puedes conectarte a un servidor de este tipo y obtener tu libreta de direcciones de correo.

- ✓ En sistemas Linux/Unix es posible autenticar usuarios del sistema operativo contra un servidor LDAP, vía PAM y libuser, en vez de la autenticación por defecto (`/etc/passwd`, `/etc/shadow`, `/etc/group`). Esto provee de autenticación centralizada, lo que facilita enormemente la administración de las cuentas para sistemas de terminales livianas, o de acceso compartido.

## Pasos previos

Los pasos previos a la instalación son detallados a continuación:

## Comprobando el hostname

Para que no haya problemas con la instalación, es necesario que el hostname del nodo en el cual estamos instalando CDS corresponda al **FQDN** del mismo (**fully qualified domain name**, el nombre completo):

```
[morenisco@dirserver1 ~]$ hostname
dirserver1.cds1.cl
```

## Versión del sistema operativo

Esta documentación fue realizada para CentOS 5.2:

```
[morenisco@dirserver1 ~]$ cat /etc/issue
CentOS release 5.2 (Final)
Kernel \r on an \m
```

## Obteniendo el archivo del repositorio de testing de CentOS

```
[root@dirserver1 ~]# cd /etc/yum.repos.d/
[root@dirserver1 yum.repos.d]#
[root@dirserver1 yum.repos.d]# ls -l
total 16
-rw-r--r-- 1 root root 2049 Jun 19 2008 CentOS-Base.repo
-rw-r--r-- 1 root root 626 Jun 19 2008 CentOS-Media.repo
[root@dirserver1 yum.repos.d]#
[root@dirserver1 yum.repos.d]# wget -v http://dev.centos.org/centos/5/CentOS-Testing.repo
--09:21:16-- http://dev.centos.org/centos/5/CentOS-Testing.repo
Resolving dev.centos.org... 204.15.73.242
Connecting to dev.centos.org|204.15.73.242|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 710 [text/plain]
Saving to: `CentOS-Testing.repo'

100%[=====>] 710          --.-K/s   in 0s
09:21:17 (60.0 MB/s) - `CentOS-Testing.repo' saved [710/710]
[root@dirserver1 yum.repos.d]#
```

## Excluyendo una versión específica de java del repositorio de testing

En el archivo que describe el repositorio es necesario agregar la línea `exclude=java-1.7.0-icedtea`, con el fin de excluir este paquete para ser instalado.

```
[root@dirserver1 ~]# cd /etc/yum.repos.d/
[root@dirserver1 yum.repos.d]# ls -l
total 24
-rw-r--r-- 1 root root 2049 Jun 19 2008 CentOS-Base.repo
-rw-r--r-- 1 root root 626 Jun 19 2008 CentOS-Media.repo
-rw-r--r-- 1 root root 710 Apr 17 2007 CentOS-Testing.repo
[root@dirserver1 yum.repos.d]#
[root@dirserver1 yum.repos.d]# vim CentOS-Testing.repo
[c5-testing]
name=CentOS-5 Testing
baseurl=http://dev.centos.org/centos/$releasever/testing/$basearch/
enabled=0
gpgcheck=1
gpgkey=http://dev.centos.org/centos/RPM-GPG-KEY-CentOS-testing
# Added by hand
exclude=java-1.7.0-icedtea
# CentOS-Testing:
# !!!! CAUTION !!!!
# This repository is a proving grounds for packages on their way to CentOSPlus and
CentOS Extras.
# They may or may not replace core CentOS packages, and are not guaranteed to function
properly.
# These packages build and install, but are waiting for feedback from testers as to
# functionality and stability. Packages in this repository will come and go during the
# development period, so it should not be left enabled or used on production systems
without due
# consideration.
```

## Instalando algunas dependencias necesarias.

```
[root@dirserver1 ~]# yum install xorg-x11-xauth bitstream-vera-fonts dejavu-lgc-fonts
urw-fonts
Setting up Install Process
[...salida suprimida...]
Installed: urw-fonts.noarch 0:2.3-6.1.1
Complete!
[root@dirserver1 ~]#
```

## Instalación de CentOS Directory Server

Para instalar el meta paquete que contiene la distribución de **CentOS Directory Server**, es necesario habilitar el repositorio de testing, para que el meta paquete sea obtenido desde ahí.

El comando para realizar ambas operaciones junto a su salida es el siguiente:

```
[root@dirserver1 ~]# yum --enablerepo=c5-testing install centos-ds
c5-testing          100% |=====| 1.9 kB    00:00
primary.sqlite.bz2 100% |=====| 389 kB    00:01
Excluding Packages from CentOS-5 Testing
Finished
Setting up Install Process

Installed: centos-ds.i386 0:8.0.0-1.4.e15.centos.4

[...salida suprimida...]

Complete!
[root@dirserver1 ~]#
```

## Verificando la versión de java en uso

```
[morenisco@dirserver1 ~]$ rpm -qa | grep java
java-1.6.0-openjdk-1.6.0.0-0.16.b09.el5.centos
tzdata-java-2007k-2.el5.centos
java-1.4.2-gcj-compat-1.4.2.0-40jpp.115

[morenisco@dirserver1 ~]$ java -version
java version "1.6.0"
OpenJDK Runtime Environment (build 1.6.0-b09)
OpenJDK Client VM (build 1.6.0-b09, mixed mode)
[morenisco@dirserver1 ~]$
```

## Configurando el servicio de directorio

En esta sección aparece con **negrita** las líneas en las cuales hay que responder, o simplemente asentir presionando enter.

Ejecutar el utilitario de configuración:

```
[root@dirserver1 ~]# /usr/sbin/setup-ds-admin.pl
=====
This program will set up the CentOS Directory and Administration Servers.

It is recommended that you have "root" privilege to set up the software.
Tips for using this program:
  - Press "Enter" to choose the default and go to the next screen
  - Type "Control-B" then "Enter" to go back to the previous screen
  - Type "Control-C" to cancel the setup program

Would you like to continue with set up? [yes]: yes
=====
BY SETTING UP AND USING THIS SOFTWARE YOU ARE CONSENTING TO BE BOUND BY
AND ARE BECOMING A PARTY TO THE AGREEMENT FOUND IN THE
LICENSE.TXT FILE. IF YOU DO NOT AGREE TO ALL OF THE TERMS
OF THIS AGREEMENT, PLEASE DO NOT SET UP OR USE THIS SOFTWARE.

Do you agree to the license terms? [no]: yes
=====
Your system has been scanned for potential problems, missing patches,
etc. The following output is a report of the items found that need to
be addressed before running this software in a production
environment.

CentOS Directory Server system tuning analysis version 10-AUGUST-2007.
NOTICE : System is i686-unknown-linux2.6.18-92.el5 (1 processor).

WARNING: 503MB of physical memory is available on the system. 1024MB is recommended for
best performance on large production system.

NOTICE : The net.ipv4.tcp_keepalive_time is set to 7200000 milliseconds
(120 minutes). This may cause temporary server congestion from lost
client connections.

WARNING: There are only 1024 file descriptors (hard limit) available, which
limit the number of simultaneous connections.

WARNING: There are only 1024 file descriptors (soft limit) available, which
limit the number of simultaneous connections.
```

**Would you like to continue? [no]: yes**

=====  
Choose a setup type:

1. Express  
Allows you to quickly set up the servers using the most common options and pre-defined defaults. Useful for quick evaluation of the products.
2. Typical  
Allows you to specify common defaults and options.
3. Custom  
Allows you to specify more advanced options. This is recommended for experienced server administrators only.

To accept the default shown in brackets, press the Enter key.

**Choose a setup type [2]:**

=====  
Enter the fully qualified domain name of the computer on which you're setting up server software. Using the form <hostname>.<domainname>  
Example: eros.example.com.  
To accept the default shown in brackets, press the Enter key.

**Computer name [dirserver1.cdsl.cl]:**

=====  
The servers must run as a specific user in a specific group. It is strongly recommended that this user should have no privileges on the computer (i.e. a non-root user). The setup procedure will give this user/group some permissions in specific paths/files to perform server-specific operations.

If you have not yet created a user and group for the servers, create this user and group using your native operating system utilities.

**System User [nobody]:**

**System Group [nobody]:**

=====  
Server information is stored in the configuration directory server. This information is used by the console and administration server to configure and manage your servers. If you have already set up a configuration directory server, you should register any servers you set up or create with the configuration server. To do so, the following information about the configuration server is required: the fully qualified host name of the form <hostname>.<domainname> (e.g. hostname.example.com), the port number (default 389), the suffix, the DN and password of a user having permission to write the configuration information, usually the configuration directory administrator, and if you are using security (TLS/SSL). If you are using TLS/SSL, specify the TLS/SSL (LDAPS) port number (default 636) instead of the regular LDAP port number, and provide the CA certificate (in PEM/ASCII format).

If you do not yet have a configuration directory server, enter 'No' to be prompted to set up one.

Do you want to register this software with an existing

**configuration directory server? [no]:**

=====  
 Please enter the administrator ID for the configuration directory server. This is the ID typically used to log in to the console. You will also be prompted for the password.

Configuration directory server

**administrator ID [admin]:**

Password:

Password (confirm):

=====  
 The information stored in the configuration directory server can be separated into different Administration Domains. If you are managing multiple software releases at the same time, or managing information about multiple domains, you may use the Administration Domain to keep them separate.

If you are not using administrative domains, press Enter to select the default. Otherwise, enter some descriptive, unique name for the administration domain, such as the name of the organization responsible for managing the domain.

**Administration Domain [cdsl.cl]:**

=====  
 The standard directory server network port number is 389. However, if you are not logged as the superuser, or port 389 is in use, the default value will be a random unused port number greater than 1024. If you want to use port 389, make sure that you are logged in as the superuser, that port 389 is not in use.

**Directory server network port [389]:**

=====  
 Each instance of a directory server requires a unique identifier. This identifier is used to name the various instance specific files and directories in the file system, as well as for other uses as a server instance identifier.

**Directory server identifier [dirserver1]:**

=====  
 The suffix is the root of your directory tree. The suffix must be a valid DN. It is recommended that you use the dc=domaincomponent suffix convention. For example, if your domain is example.com, you should use dc=example,dc=com for your suffix. Setup will create this initial suffix for you, but you may have more than one suffix. Use the directory server utilities to create additional suffixes.

**Suffix [dc=cdsl, dc=cl]:**

=====  
 Certain directory server operations require an administrative user. This user is referred to as the Directory Manager and typically has a bind Distinguished Name (DN) of cn=Directory Manager. You will also be prompted for the password for this user. The password must be at least 8 characters long, and contain no spaces.

**Directory Manager DN [cn=Directory Manager]:**

Password:  
Password (confirm):

```
=====
The Administration Server is separate from any of your web or application
servers since it listens to a different port and access to it is
restricted.
```

```
Pick a port number between 1024 and 65535 to run your Administration
Server on. You should NOT use a port number which you plan to
run a web or application server on, rather, select a number which you
will remember and which will not be used for anything else.
```

**Administration port [9830]:**

```
=====
The interactive phase is complete. The script will now set up your
servers. Enter No or go Back if you want to change something.
```

```
Are you ready to set up your servers? [yes]:
Creating directory server . . .
Your new DS instance 'dirserver1' was successfully created.
Creating the configuration directory server . . .
Beginning Admin Server creation . . .
Creating Admin Server files and directories . . .
Updating adm.conf . . .
Updating admpw . . .
Registering admin server with the configuration directory server . . .
Updating adm.conf with information from configuration directory server . . .
Updating the configuration for the httpd engine . . .
Starting admin server . . .
The admin server was successfully started.
Admin server was successfully created, configured, and started.
Exiting . . .
Log file is '/tmp/setupmKPwo4.log'
```

```
[root@dirserver1 ~]#
```

## Algunas comprobaciones post instalación/configuración

Verificando si se inició el servicio de directorio

```
[root@dirserver1 ~]# /etc/init.d/dirsrv status
dirsrv dirserver1 (pid 4879) is running...
```

==> está corriendo.

## Verificando si se “levantó el puerto” 389

```
[root@dirserver1 ~]# netstat -tln | grep 389
tcp      0          0      :::389      :::*        LISTEN
```

==> está escuchando.

## Veamos qué programa está utilizando el puerto 389

```
[root@dirserver1 ~]# lsof -i:389
COMMAND  PID  USER  FD  TYPE  DEVICE  SIZE  NODE  NAME
ns-slapd 6394 nobody  7u  IPv6  18701          TCP *:ldap (LISTEN)
[root@dirserver1 ~]#
```

Ok, el binario ns-slapd es quien está implementando el servicio y ocupando el puerto 389, solo para saber...

## Comprobando si se inició el servicio de administración

Esto es importantísimo, ya que a través de este servicio podremos logearnos a la aplicación de administración cuando sea necesario, desde donde se tiene control del árbol del directorio. Se puede realizar operaciones como detención, inicio y reinicio del servicio de directorio (naturalmente también se puede hacer desde una línea de comandos), configurar Multi Master Replication, etc.

```
[root@dirserver1 ~]# /etc/init.d/dirsrv-admin status
dirsrv-admin (pid 6482) is running...

==> Ok, está corriendo.
```

## Lanzando la consola de administración

Lanzamos la consola de administración con el comando que se muestra en la imagen:

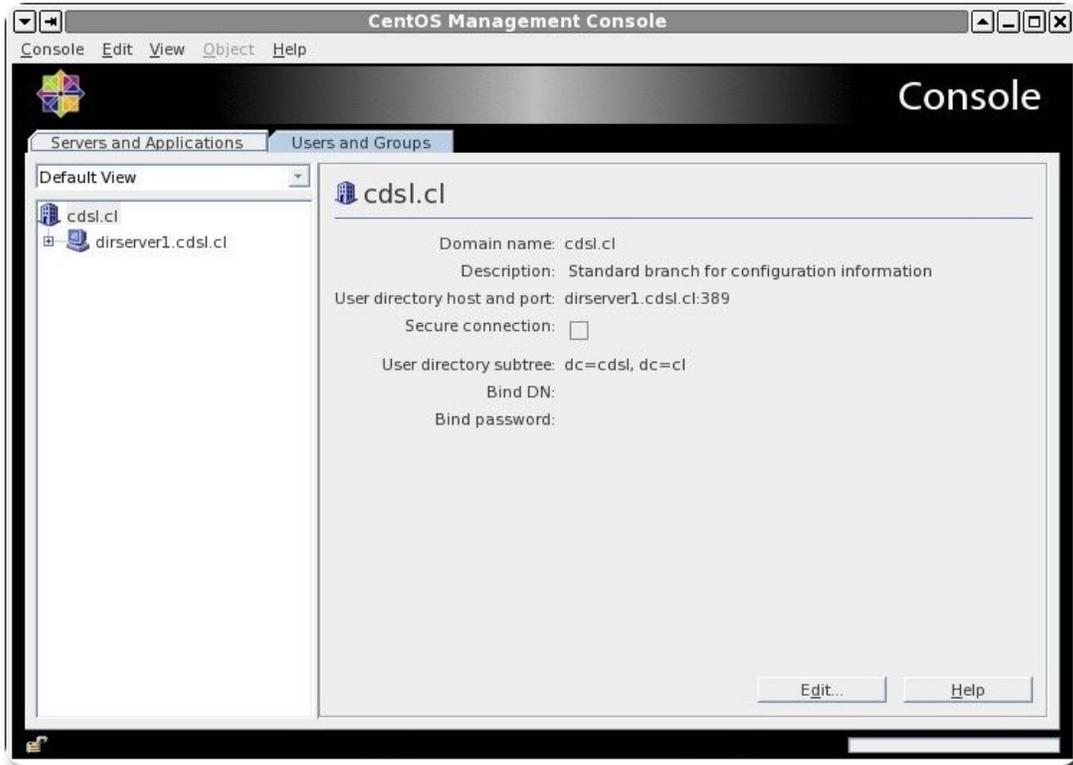


**Nota:** el puerto 9830 se seteó durante el proceso de configuración. Y si especificaron otro, entonces deberían reemplazarlo en el comando.

Luego se presenta la ventana de login, en donde se debe ingresar el login (por defecto es admin, a menos que hayan especificado otro durante el proceso de configuración), como se ve en la siguiente imagen:



Finalmente, tenemos acceso a la consola de administración:



## Autenticando a un usuario y realizando una búsqueda

Esta prueba tiene por objetivo probar en forma simple que el servicio de directorio está funcionando, por medio de realizar dos operaciones básicas (bind, search)

El bind significa acceder con algún tipo de credenciales al sistema ldap, ya sea un acceso anónimo, que no es en si una cuenta anónima, o especificando un usuario y password (muchas veces se permite el bind y search anónimo o con un usuario sin password; los cual por cierto no es lo más seguro).

En esta parte creé un usuario, el cual tiene por DN (distinguished name, o nombre distintivo por decirlo de alguna forma) lo siguiente :

```
"uid=lvivero,ou=people,dc=cdsl,dc=cl"
```

Entonces, utilizando una de las herramientas que provee **openldap-clients.i386**, **ldapsearch**, realicé una búsqueda simple, especificando el DN y password del usuario, de la siguiente forma:

```
[morenisco@dirserver1 ~]$ ldapsearch -x -D "uid=lvivero,ou=people,dc=cdsl,dc=cl" -w
L34rn1n6 -b "ou=people,dc=cdsl,dc=cl" "objectclass=*"
# extended LDIF
# LDAPv3
# base <ou=people,dc=cdsl,dc=cl> with scope subtree
# filter: objectclass=*
# requesting: ALL
```

```
# People, cds1.cl
dn: ou=People, dc=cdsl, dc=cl
objectClass: top
objectClass: organizationalunit
ou: People

# LVivero, People, cds1.cl
dn: uid=LVivero,ou=People, dc=cdsl, dc=cl
mail: morenisco@noc-root.net
uid: LVivero
givenName: Luis
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetorgperson
sn: Vivero
cn: Luis Vivero

# search result
search: 2
result: 0 Success

# numResponses: 3
# numEntries: 2
```

Como se observa, se obtuvo la salida fruto de la búsqueda, permitida por un bind correcto.

Si uno ingresa mal el password, o si la autenticación no funciona, entonces obtendríamos lo siguiente:

```
[morenisco@dirserver1 ~]$ ldapsearch -x -D "uid=lvivero,ou=people,dc=cdsl,dc=cl" -w
L34rn1n7 -b "ou=people,dc=cdsl,dc=cl" "objectclass=*"
ldap_bind: Invalid credentials (49)
```

## Referencias

[1] Documento completo en: <http://morenisco.belvil.eu/blog/?p=67>

## Autor



**Morenisco**  
morenisco@cdsl.cl



A large, semi-transparent grey circle containing the word "Willay" in white, bold, sans-serif font.

**Willay**

**news**

Información Actual

# Willay news

## Nova Baire la Distro Cubana

Cuba ha lanzado a mediados del mes de Febrero su propia Distro denominada “**Nova Baire**” basado en Gentoo, con una interfaz amigable similar a Windows y Macintosh.



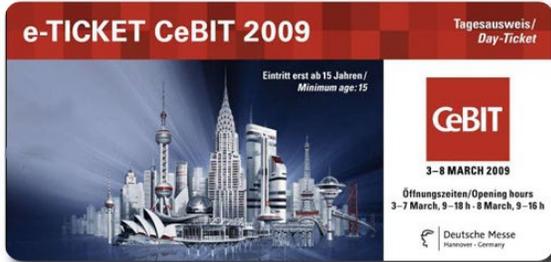
Hace tres años ya había rumores sobre distribuciones cubanas pero se habían quedado en el camino, hasta que en el 2008 se comento que los estudiantes y docentes de la Universidad de Ciencias Informáticas de la Habana estaban realizando la nueva Distro Nova. Según el Ministro de Comunicaciones de Cuba, Ramiro Valdés quién está a la cabeza de una comisión encargada para impulsar el uso de Software Libre dijo que es “muy importante tener un mayor control en todos los procesos informáticos”; y los desarrolladores partidarios del software libre celebran esta decisión ya que es más cercano a la ideología cubana. Además que Cuba tiene un bloqueo comercial por parte de Estados Unidos, por ésto la mayoría de los cubanos utilizan el Sistema Operativo Windows de manera ilegal, esto hace que no tengan las actualizaciones apropiadas y legales.



Nova fue presentado en la Feria de Informática en la Habana, lo interesante de esta Distro es que tiene la interfaz sencilla que permite el uso de aplicaciones modernas, no presenta muchos requerimientos en cuanto a hardware y es por ello que se puede trabajar en máquinas obsoletas. Si deseas más información sobre Nova Baire ve a la página oficial <http://www.nova.uci.cu> o puedes ver el vídeo promocional dirigiéndote a la página <http://www.cubaleo.net/2009/02/nova-linux-made-in-cuba-oficial/>.

## CEBIT 2009

Luego del **Mobile World Congress** que no tuvo muchas novedades este año, el CEBIT 2009 se viene con todo pese a la crisis económica mundial.



La mayor feria mundial de Tecnologías de información y telecomunicación CEBIT (**Centro para la Tecnología de la Información y de la Oficina – Centrum der Büro-und Informationstechnik**) tendrá lugar en Hannover Alemania desde el 3 al 8 marzo del presente año, en esta oportunidad CEBIT está asociado con California cuyo Gobernador Arnold Schwarzenegger indicó que estaría presente en la inauguración de la feria y personajes importantes en cuanto a este mundo de la tecnología se refiere.

CEBIT ha demostrado en anteriores ferias bastante éxito, mucha innovación, negociaciones, participación de empresas importantes, afluencia de personas profesionales en el área de Informática y apasionados por la tecnología y mucho más; este año la Feria espera seguir teniendo el éxito de siempre e incluso superarlo en todo aspecto.



Uno de los principales temas a tratarse en esta versión son la **"Green IT"** o verde IT, cuyo reto de las empresas de hoy es hacer que sus productos sean más eficientes en el consumo de energía; y de esta manera apoyar los esfuerzos para frenar el Calentamiento Global que es un tema latente a nivel mundial. Se considera que las Tecnologías de Información y Telecomunicación son una herramienta importante para reducir la contaminación ambiental; según Greenpeace en CEBIT 2008 sólo 3 productos vencieron la prueba ambiental, lo rescatable es que varias empresas eliminaron algunos productos tóxicos y se espera que este año se supere éste número y se tenga más éxito en la prueba ambiental. Otro de los puntos a tratar es la **"Webciety"** o Internet en la Sociedad que es un tema fundamental en la Feria, debido a que hay miles y millones de personas que utilizan este recurso a nivel personal, social y laboral; todo esto mucho más lo verás en el CEBIT 2009.

### Autor



#### Ivonne Karina Menacho Mollo

Titulada de la carrera de Ingeniería de Sistemas e Informática (F.N.I.)

Conductora programa radial **"Doble Clic"**

ivonnekarina2003@hotmail.com

# Willay news

## Un Poderoso Sistema de Sistemas - FCS

**Future Combat Systems FCS (Sistemas de Combate del Futuro)** una pieza poderosa de la modernización del ejército de los EEUU. Este proyecto tiene una gran cobertura, que va desde robots móviles hasta plataformas de comunicación, búsquedas tridimensionales, sistemas futuristas de guerra terrestres o aéreos.



Solo imaginemos: comunicar toda una cadena de comandos, compartir datos en tiempo real, reaccionar a los cambios en diferentes condiciones con precisión y rapidez en cada una de las unidades, vehículos aéreos y terrestres, tripulados y no tripulados. Sin duda alguna, deberá estar bajo un sistema que no tenga fallos, bastante estable y confiable.

Es justo aquí donde entra LynxOS-178 (de la empresa **LinuxWorks**) sistema operativo embebido que ofrece compatibilidad con Linux y POSIX (Interfaz para Sistemas Operativos migrables basados en UNIX)

Las normas Abiertas del Sistema Operativo son necesarias para satisfacer las necesidades del Ejército en cuanto a la información en tiempo real y la seguridad. LinuxWorks permite a los desarrolladores mover aplicaciones de FCS entre aquellas que han sido desarrolladas en entornos Linux y aquellas dirigidas a plataformas **RTOS (The real-time Operating Systems)** embebido, sin costoso o lento proceso de conexión.



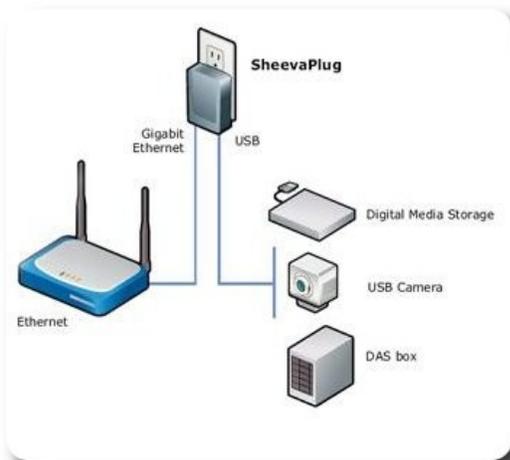
Actualmente este proyecto supera las 63 millones líneas de código y ha tenido un alcance mayor de lo que puede manejar el Ejército.

Con un calendario de desarrollo que llega hasta el año 2030, esperamos que este proyecto que pareció inicialmente un cuento de ciencia ficción, se consolide y en un futuro cercano podremos ver un campo de batalla intermediado por la tecnología.

## Computadora del tamaño de un adaptador de corriente?



**SheevaPlug** es el nombre que le ha dado Marvell a su nueva creación, ¡una computadora del tamaño de un adaptador de corriente! sí!!!, yo tampoco lo creí, con un microprocesador de 1.2 GHz, 512 MB de RAM, 512 MB de almacenamiento FLASH, una tarjeta Ethernet Gigabit y conectividad USB 2.0 que permite a los usuarios ejecutar aplicaciones que requieran un disco duro u otro periférico (monitor, impresora) transformándolo en un pequeñísimo servidor, cualquiera podrá tener un servidor compacto con tan solo conectarlo a un enchufe de la pared.



Tiene un consumo de energía menor a 5 vatios, pero no vayamos a pensar que tiene un bajo rendimiento pues el procesador que contiene es de la serie Marvell Kikwood que permite a las interfaces operar a velocidad de cable, de esta manera los usuarios no ven una disminución en el rendimiento utilizando un enchufe en vez de una computadora convencional. Lo que se extrañara un poco será una salida de vídeo más convencional y una conexión inalámbrica.



Este pequeño dispositivo que por cierto es compatible con las distribuciones GNU/Linux a partir del Kernel 2.6, tiene un costo de \$99, aunque rumores dicen que la empresa quiere rebajar su precio a \$49, grandioso no lo creen? Uno después de esto se preguntara que hacemos con computadoras tan grandes, que consumen 10 veces más energía que esta propuesta.

### Autor



#### **Marcia Estrella Velasquez Aguilar**

Egresada de la carrera de Ingeniería de Sistemas (F.N.I.)

mevaguerradelaluz@gmail.com

# Willay news

## Amistad con la naturaleza



Al parecer el 2009 es el año de la ecología y es que todas las empresas están lanzando productos más y más amigables con el medio ambiente. Samsung ha lanzado la serie de F2EG, discos duros SATA que vienen en tamaños de 500GB, 1TB y 1.5TB. Son llamados "eco-friendly" porque efectivamente están siendo fabricados con materiales que los hacen fáciles de reciclar, pero además, porque también estos nuevos discos duros ayudan a reducir un 43% de energía respecto a sus competidores directos.



Motorola no quiso quedarse atrás, es así que se está dedicando a promocionar su primer teléfono móvil que se realizó utilizando las botellas de agua de plástico recicladas. El teléfono tiene mayor duración de batería con hasta nueve horas de tiempo de conversación y hasta 18 días de tiempo de espera. Además, el teléfono GSM y embalaje incluidos en la caja de materiales se imprimen en el 100% reciclado

post-consumo de papel. Otras características incluyen 2 GB de memoria extraíble opcional, navegador WAP, mini USB, y de 128 x 128 65K CSTN 1,6 pulgadas en la pantalla.



El proyecto "Green Heart Phone" de Sony Ericsson, también trata de encaminarse en el tema de hacer uso de materiales reciclables. Actualmente la mayoría de los teléfonos celulares no son biodegradables, ya que contienen pequeñas cantidades de sustancias potencialmente nocivas en las baterías que, si no se eliminan adecuadamente, podrían dañar el medio ambiente.



"Green Heart Phone" investigará nuevas tecnologías para incluir en sus carcasas material bio-plástico, teclados de plástico reciclado, con cargadores de 3.5mW de poder, manuales de usuario basados en HTML, aplicaciones educativas "Ecomate" y el medio ambiente presente en todos los envases.

## Evadir sistemas biométricos

Desde que se descubrieron los sistemas biométricos, fueron muchas las aplicaciones que empezaron a hacer uso de esta tecnología, principalmente para sistemas donde la autenticación es un tema crítico.



Japón y EE.UU están usando desde hace algún tiempo lectores de huellas digitales biométricos, que son usados para prevenir el acceso de terroristas e inmigrantes ilegales reconocidos a dichos países. Sin embargo, el pasado mes salió a la luz que varios ciudadanos coreanos lograron pasar este filtro de huellas digitales usando una protección en sus dedos.



Ahora los medios de comunicación en Japón y Corea, han develado que esta "protección" para saltar estos controles cuesta tan sólo \$70 dólares, al parecer muchos criminales coreanos están promoviendo mucho este tipo de protecciones de silicona.

Parece increíble ver que hace unos años atrás hablamos de los sistemas biométricos como los sistemas más seguros, pero actualmente tan sólo toma 1 día crear una protección a base de silicona, y evadir estos controles.

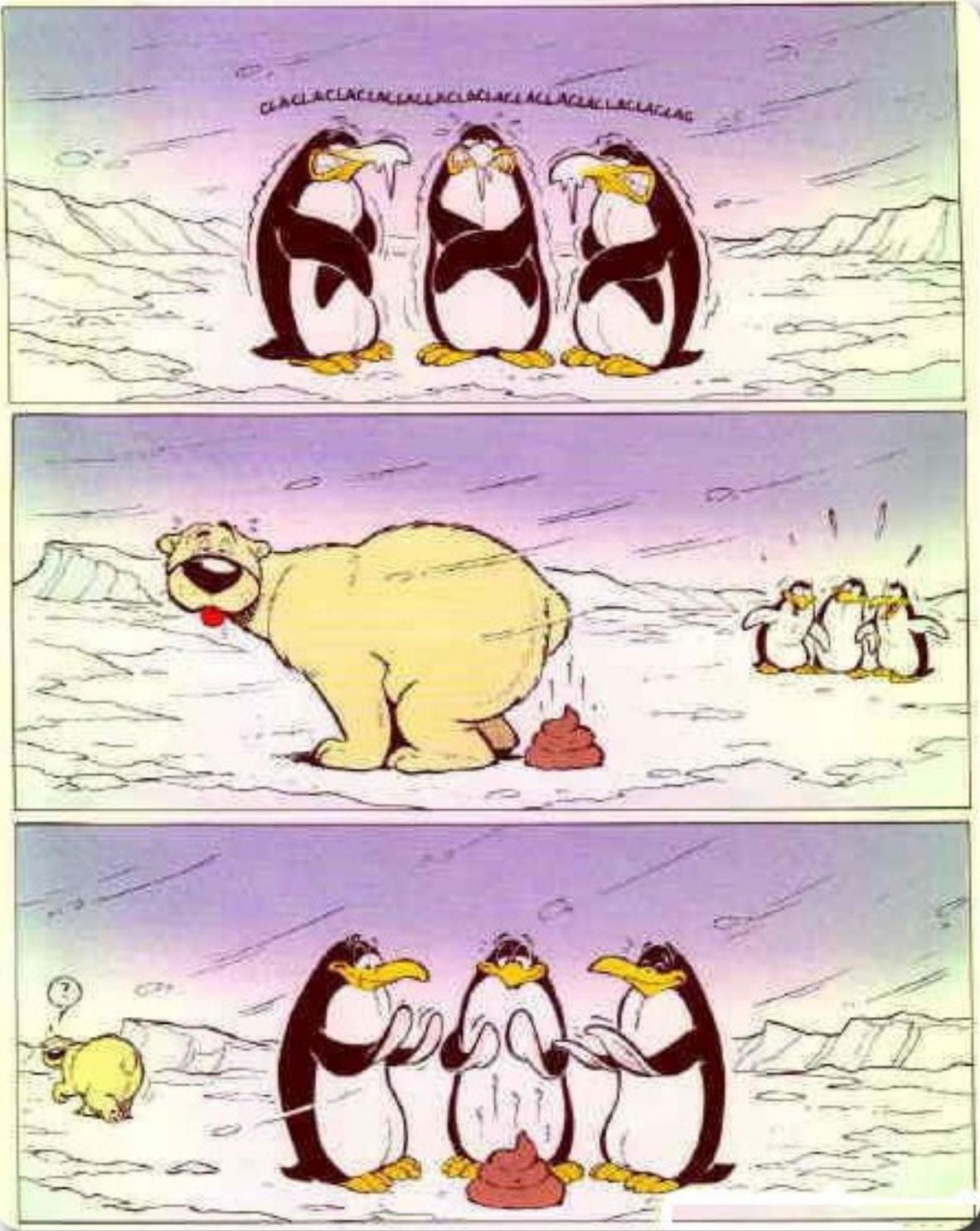
## Autor



**Jenny Saavedra López**  
 Diseño y Edición Revista Atix  
[jennysaavedra@gmail.com](mailto:jennysaavedra@gmail.com)

# QUEREMOS LA REVISTA ATIX....







# Conociendo lo Nuestro

**Oruro**



Carnaval de Oruro, Obra Maestra del Patrimonio Oral e Intangible de la Humanidad



Santuario de la Virgen del Socavón



Vista nocturna del Santuario del Socavón



Imágenes del fastuoso Carnaval de Oruro



Morenada



Tobas



Figura de la Morenada



Figura de Los Incas



Negritos



Tinkus



Caporales



Chinas de la Diablada



Pujllay

**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**

# ATIX



Reviewista

## Contacto

Para solicitar cualquier información, puedes contactar a:

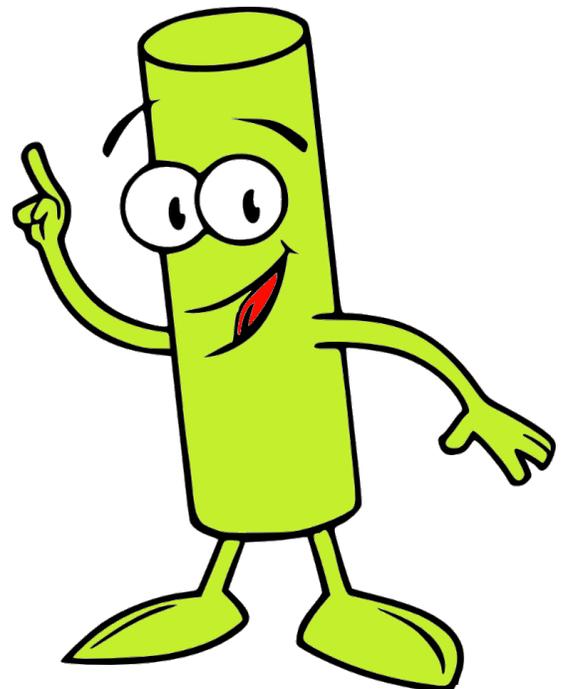
- ✓ Esteban Saavedra López (jesaavedra@opentelematics.org)
- ✓ Williams Chorolque Choque (williamsis@gmail.com)

## 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.

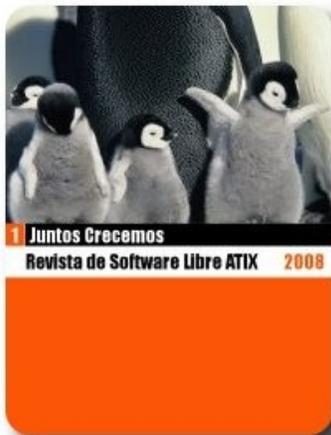
Usa siempre  
Software Libre



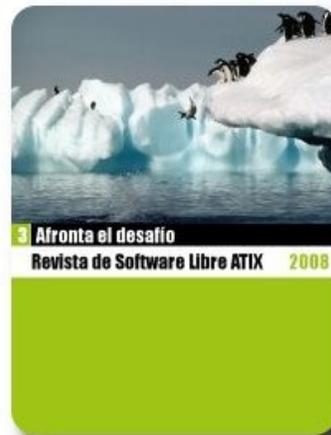
Empezamos a registrar nuestra historia:



28 de abril del 2008



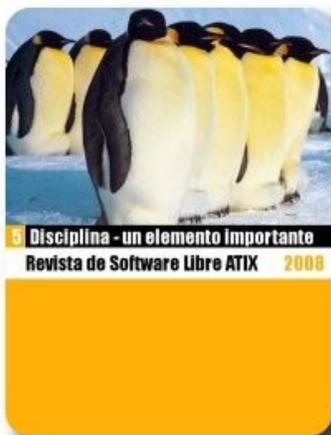
20 de junio del 2008



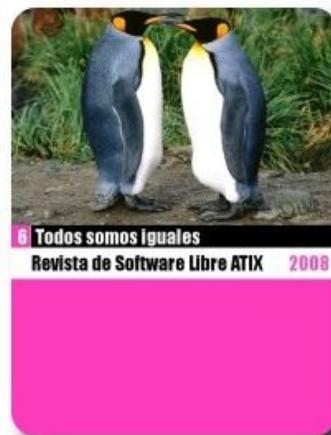
6 de agosto del 2008



17 de septiembre del 2008



10 de noviembre del 2008



17 de diciembre del 2008



26 de enero del 2009



**Marcamos Huella**



<http://atix.opentelematics.org>