

# ¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.  
 Ese apoyo que siempre quiso tener...

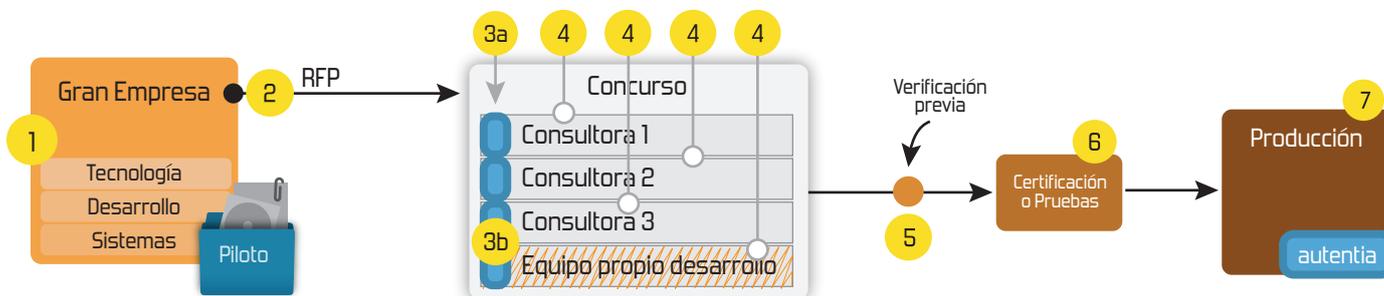
## 1. Desarrollo de componentes y proyectos a medida



## 2. Auditoría de código y recomendaciones de mejora

## 3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



## 4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,  
 HTML5, CSS3, JavaScript-jQuery

Control de autenticación y  
 acceso (Spring Security)  
 UDDI

JPA-Hibernate, MyBatis  
 Motor de búsqueda empresarial (Solr)  
 ETL (Talend)

Gestor portales (Liferay)  
 Gestor de contenidos (Alfresco)  
 Aplicaciones híbridas

Web Services  
 Rest Services  
 Social SSO  
 SSO (Cas)

Dirección de Proyectos Informáticos.  
 Metodologías ágiles  
 Patrones de diseño  
 TDD

Tareas programadas (Quartz)  
 Gestor documental (Alfresco)  
 Inversión de control (Spring)

BPM (jBPM o Bonita)  
 Generación de informes (JasperReport)  
 ESB (Open ESB)



Entra en Adictos a través de 

E-mail

Contraseña

Entrar [Registrarme](#)  
[Olvidé mi contraseña](#)

[Inicio](#) [Quiénes somos](#) [Formación](#) [Comparador de salarios](#) [Nuestros libros](#) [Más](#)

» Estás en: [Inicio](#) [Tutoriales](#) [GitLab: Crear y gestionar nuestro servidor propio de Git](#)



Rubén Aguilera Díaz-Heredero

Consultor tecnológico de desarrollo de proyectos informáticos.

Ingeniero en Informática, especialidad en Ingeniería del Software

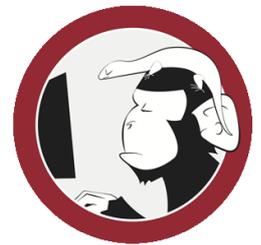
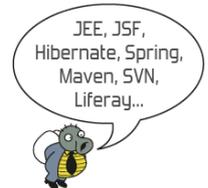
Puedes encontrarme en [Autentia](#): Ofrecemos servicios de soporte a desarrollo, factoría y formación

Somos expertos en Java/J2EE



[Ver todos los tutoriales del autor](#)

## Catálogo de servicios Autentia



Fecha de publicación del tutorial: 2014-07-18

Tutorial visitado 1 veces [Descargar en PDF](#)

# GitLab: Crear y gestionar nuestro servidor propio de Git

## 0. Índice de contenidos.

- 1. Entorno
- 2. Introducción
- 3. Primeros pasos
- 4. Conclusiones

## 1. Entorno

Este tutorial está escrito usando el siguiente entorno:

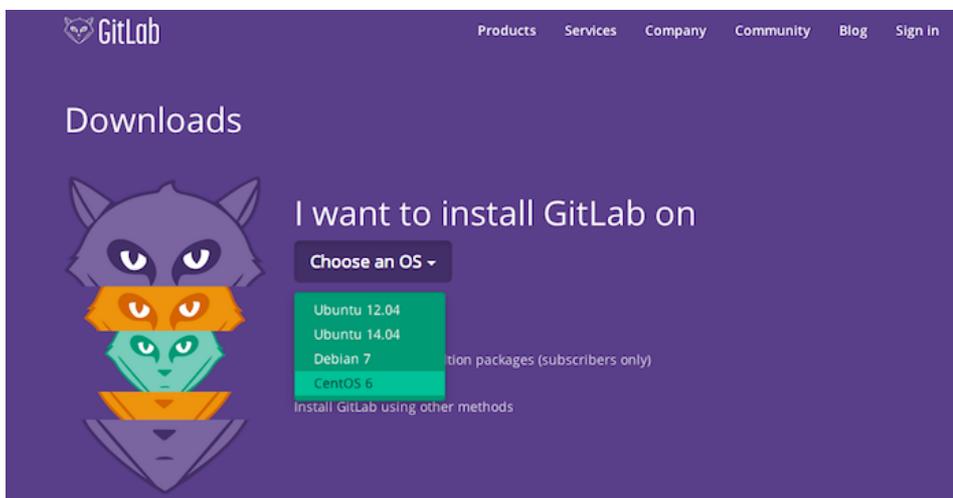
- Hardware: Portátil Mac Book Pro 15" (2,3 Ghz Intel Core i7, 16 GB DDR3)
- Sistema Operativo: Mac OS X Mavericks
- Máquina virtual VMWare con CentOS 6.5

## 2. Introducción

En un reciente [tutorial](#) hablábamos de que como montar un servidor propio de Git en un CentOS 6.5 de forma manual. Hoy os presentamos GitLab, una herramienta que hace la instalación, configuración y gestión de un servidor Git propio mucho más amigable gracias a su interfaz web, muy al estilo de la interfaz de github.com.

## 3. Primeros pasos

Para instalar la herramienta tenemos que acceder a la página de descargas de GitLab y seleccionar el sistema operativo. En nuestro caso CentOS 6.5



Seleccionando el sistema operativo, podremos ver las instrucciones a seguir para la instalación de la herramienta y el username y password por defecto para poder acceder.

## Síguenos a través de:



## Últimas Noticias

» [Comentando el libro Start-up Nation, La historia del milagro económico de Israel, de Dan Senor & Salu Singer](#)

» [Screencasts de programación narrados en Español](#)

» [Sorteo de entradas para APIdays Mediterranea](#)

» [Concurso del Día de la Madre:](#)

» [Aprende gratis ReactiveCocoa](#)

[Histórico de noticias](#)

## Últimos Tutoriales

» [Configurando Notificaciones Push para desarrollos Android con Google Cloud Messaging.](#)

» [Crear servidor propio de Git en CentOS 6.5](#)

» [Primeros pasos con Neo4j](#)

» [Introducción a WSO2 API Manager](#)

» [Introducción a Groovy y Grails con Maven: el patrón CRUD](#)

## Últimos Tutoriales del Autor

- » Crear servidor propio de Git en CentOS 6.5
- » Crear un plugin para Android en PhoneGap
- » Intercomunicación de aplicaciones en IOS
- » Crashlytics en IOS
- » Acceso a la cámara con PhoneGap

Vamos a seguir los pasos que se indican en la página. Entonces conectamos via ssh con la máquina CentOS y lo primero que hacemos es descargar la última versión de la herramienta.

```
01. | view plain print ?
    | wget https://downloads-packages.s3.amazonaws.com/centos-6.5/gitlab-7.0.0_omnibus-1.e16.x86_64.rpm
```

En mi caso, no necesito instalar "openssh-server" ni "postfix" porque ya tengo las últimas versiones así que pasamos directamente al paso de instalación del .rpm

```
01. | view plain print ?
    | sudo rpm -i gitlab-7.0.0_omnibus-1.e16.x86_64.rpm
```

Una vez instalado, vamos a configurarlo para añadir nuestro hostname y que además en el puerto que le indiquemos y no en el 80 que tiene por defecto. Para ello, editamos el fichero "/etc/gitlab/gitlab.rb" ejecutando:

```
01. | view plain print ?
    | sudo -e /etc/gitlab/gitlab.rb
```

Esto hace que se podamos editar el fichero donde deberemos cambiar la "external\_url" que viene por defecto por la url del host y el puerto que queramos, en mi caso, "http://ip\_maquina\_centos:8082". Siempre es mejor que pongáis el nombre de la máquina o la IP para poder acceder desde fuera.

Para que estos cambios surtan efecto tenemos que ejecutar:

```
01. | view plain print ?
    | sudo gitlab-ctl reconfigure
```

El último paso con el comando "lokkit" se utiliza para abrir los puertos dentro de la máquina. Este comando suele dar problemas por lo que recomiendo abrir los puertos necesarios de forma manual con la edición del fichero /etc/sysconfig/iptables añadiendo con los permisos necesarios la siguiente línea:

```
01. | view plain print ?
    | -A INPUT -p tcp -m state --state NEW -m tcp --dport 8082 -j ACCEPT
```

Para que los cambios surtan efecto tendremos que reiniciar el servicio "iptables" ejecutando en el terminal:

```
01. | view plain print ?
    | sudo service iptables restart
```

Si todo ha ido bien desde nuestra máquina cliente abrimos un navegador, nos conectamos a la url: [http://ip\\_maquina\\_centos:8082](http://ip_maquina_centos:8082) y tenemos que ver la página de login de GitLab. Donde nos logamos con las credenciales que nos dan por defecto en las instrucciones de la página de instalación.

# GitLab

GitLab is open source software to collaborate on code.  
Sign in or browse for [public projects](#).

## Sign in

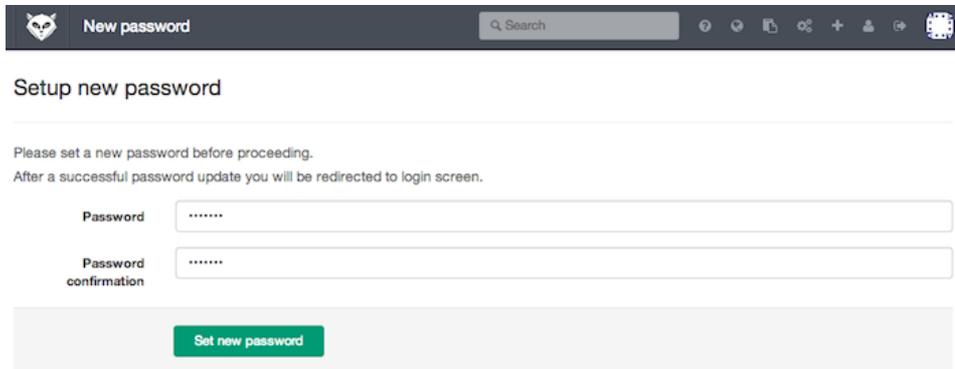
  

Remember me

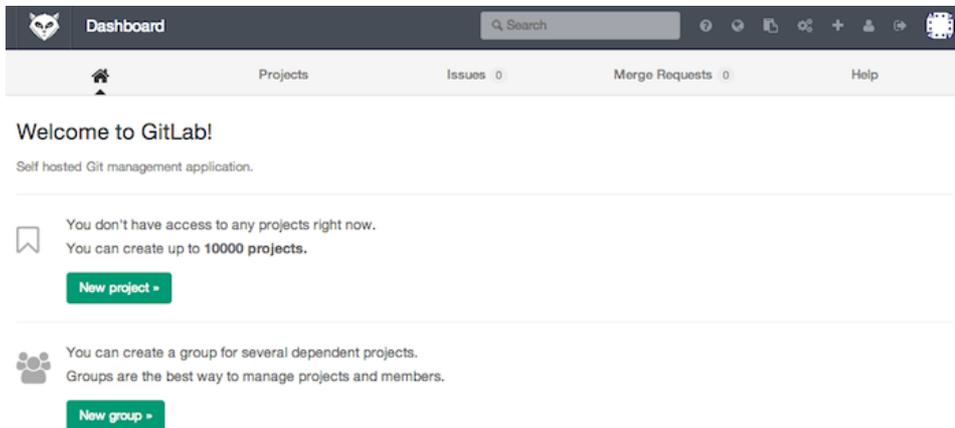
 

Did not receive confirmation email? [Send again](#)

Una vez introducidas estas credenciales, el sistema nos obliga modificar la password por defecto.



Una vez introducida la nueva password el sistema nos redirige nuevamente a la página de login donde tendremos que logarnos con las nuevas credenciales. Accediendo al dashboard principal de la herramienta.



Una vez aquí el siguiente paso sería crear algún usuario. Para ello vamos a la opción de menú "Admin Area" y pulsamos en el botón "New User".

Admin area

Projects Groups **Users** Logs Messages Admin area Background

Active 1

Admins 1

Blocked 0

Without projects 1

Name, email or username

Users (1) **New User**

Administrator (Admin) It's you! admin@local.host Edit

Aquí rellenamos el formulario con la información del usuario y éste recibirá un email con la información para acceder y una password temporal que el sistema le obligará a cambiar la primera vez que haga login en la herramienta.

Admin area

Projects Groups **Users** Logs Messages Hooks Background Jobs

### New user

Account

Name Rubén Aguilera \* required

Username raguilera82 \* required

Email raguilera@autentia.com \* required

Password

Password A temporary password will be generated and sent to user. User will be forced to change it after first sign in

Access

Projects limit 10

Can create group

Admin

Profile

Avatar Choose File No file chosen

Skype

LinkedIn

Twitter @raguilera82

Website http://www.autentia.com

**Create user** Cancel

Ahora es momento de crear nuestro primer proyecto. Si habéis seguido el anterior tutorial, esto lo hacíamos ejecutando comandos en la consola, aquí desde el dashboard principal o desde "Admin Area" lo podemos hacer pulsando en el botón "New Project", donde nos solicita información como el nombre del proyecto, una descripción y de qué tipo va a ser el proyecto: privado, donde solo pueden acceder los usuarios con los permisos explícitos; interno, donde podrá acceder cualquier usuario que esté logado o público, donde cualquiera podrá hacer clone de nuestro proyecto esté o no logado.

Además nos da la posibilidad de personalizar el nombre de nuestro repositorio o de importar alguno ya existente.

Al pulsar en "Create", la herramienta crea el proyecto y nos informa de una serie de instrucciones.

También nos avisa que el usuario no podrá acceder al repositorio hasta que no añada su SSH Key pública para poder conectar con el repositorio sin necesidad de poner la password. Si habéis seguido el anterior tutorial sabréis que esto se puede realizar desde el terminal ejecutando:

```

view plain print ?
01. $> ssh-keygen -t rsa
02. $> cat ~/.ssh/id_rsa.pub

```

Entonces, en la herramienta, podemos pinchar en el enlace que nos muestra o desde la opción de menú "Profile Settings" en la pestaña "SSH Keys", podemos pulsar en el botón "Add SSH Key" donde le damos un nombre a la clave y pegamos nuestra clave ssh pública que es todo el contenido que se muestra en pantalla al hacer el cat.

Profile 🔍 🔔 📄 ⚙️ + 👤 🔄 

[Account](#) [Emails](#) [Password](#) [Notifications](#) **SSH** [Design](#) [Groups](#) [History](#)

## Add an SSH Key

Paste your public key here. Read more about how to generate a key on the [SSH help page](#).

**Title**

**Key**

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQDAQO+UCODY20Rk0oz8QESWzGBkfjYrgwP7pj/cx6SiqJ
ocoBuinf0I3DYepTdhHyir5WcN5yCzTkPPH+hoqlzJxXal0fI6ufGoJQ3RzmvPJks6xk/Us1tKoNF6fd75
1rsTjyFxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxSJV+
u+w8MwkZdnwOwS0HrHAU2zgLgqI3hRnfreMo5wuTJU7MqYhrNxVuiVGsno0Nk7mESznsMeMkY
wwWG6M7real52fKlwa/ygON2sl2qws7XFXqRNkQdLxqJ4rE7SHpEoXYa6adV8GkqEbfP9F
raguilera82@xxxxxx
```

Para probar que todo ha ido bien podemos abrir un terminal e intentar hacer un clone del proyecto.

```
view plain print ?
01. git clone git@ip_maquina_centos:raguilera82/autentia-test.git
```

Si el terminal no solicita que introduzcamos una contraseña significa que la configuración SSH ha ido bien. Ahora podemos interactuar con el repositorio de la forma habitual.

### 4. Conclusiones

En este tutorial hemos dado los primeros pasos con esta fantástica herramienta. Con ella no se echa de menos ninguna funcionalidad que te ofrece Github.com. Podemos configurar los web hooks para integrar con Jenkins de forma gráfica y simple, podemos tener nuestros snippets, nos ofrece la posibilidad de tener una wiki del proyecto, podemos hacer grupos de usuarios, etc... Lo mejor es que la instaléis y os déis una vuelta para ver todas las opciones. Espero con este tutorial que el comienzo sea más suave y que os animéis a utilizar siempre que tengáis la necesidad de tener un repositorio propio de Git.

Cualquier duda o sugerencia en la zona de comentarios.

Saludos.

## A continuación puedes evaluarlo:

[Regístrate para evaluarlo](#)

## Por favor, vota +1 o compártelo si te pareció interesante

Share |  0

¡Anímate y coméntanos lo que pienses sobre este TUTORIAL:

» [Regístrate](#) y accede a esta y otras ventajas «



Esta obra está licenciada bajo [licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5](#)

[Impulsores](#) [Comunidad](#) [¿Ayuda?](#)

-----  
sin clicks      0 personas han traído clicks a esta página  
+   +   +   +   +   +   +   +

powered by [karmacrazy](#)

Copyright 2003-2014 © All Rights Reserved | [Texto legal y condiciones de uso](#) | [Banners](#) | [Powered by Autentia](#) | [Contacto](#)

