

INSTITUTO TECNOLÓGICO SUPERIOR ZACATECAS OCCIDENTE



MATERIA: VALIDACIÓN Y VERIFICACIÓN DE SOFTWARE

CARRERA: INGENIERIA EN SISTEMAS COMPUTACIONALES

TEMA: CONFIGURAR SERVIDOR DE INTEGRACION
CONTINÚA

DOCENTE:

I.S.C. ERICKA JAZMÍN ROBLES GÓMEZ

ELABORADO POR:

FABIAN ARMANDO HERRERA AVALOS

OSIEL BARRIENTOS RAMIREZ

OSIEL CHAVEZ FLORES

RUBEN GOMEZ BARRIENTOS

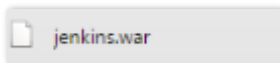
FECHA: 29 DE ENERO DEL 2016

Jenkins, como servidor de integración continua, permite planificar y realizar multitud de tareas, simplificando los procesos involucrados en el ciclo de vida de un proyecto. Algunas de sus características más importantes son:

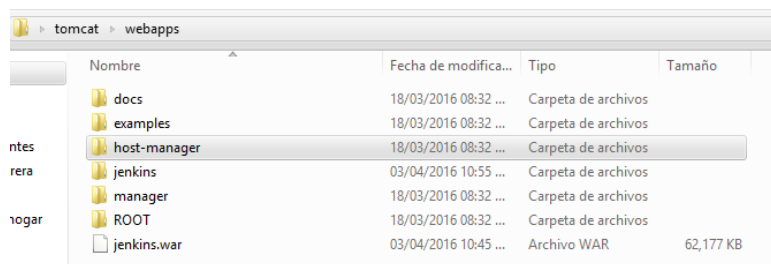
- Comprobación cada cierto periodo de tiempo si se ha realizado algún commit en el repositorio de control de versiones (GIT), y en caso de ser así, compilar el código y ejecutar las pruebas para testearlo.
- Notificación de errores que se hayan detectado tras la ejecución de pruebas, por ejemplo vía mail, twitter, chat, etc.
- Generación y publicación de binarios.
- Ejecución de métricas de calidad y visualización los resultados.
- Generación de documentación asociada a un proyecto.

Instalación de Jenkins

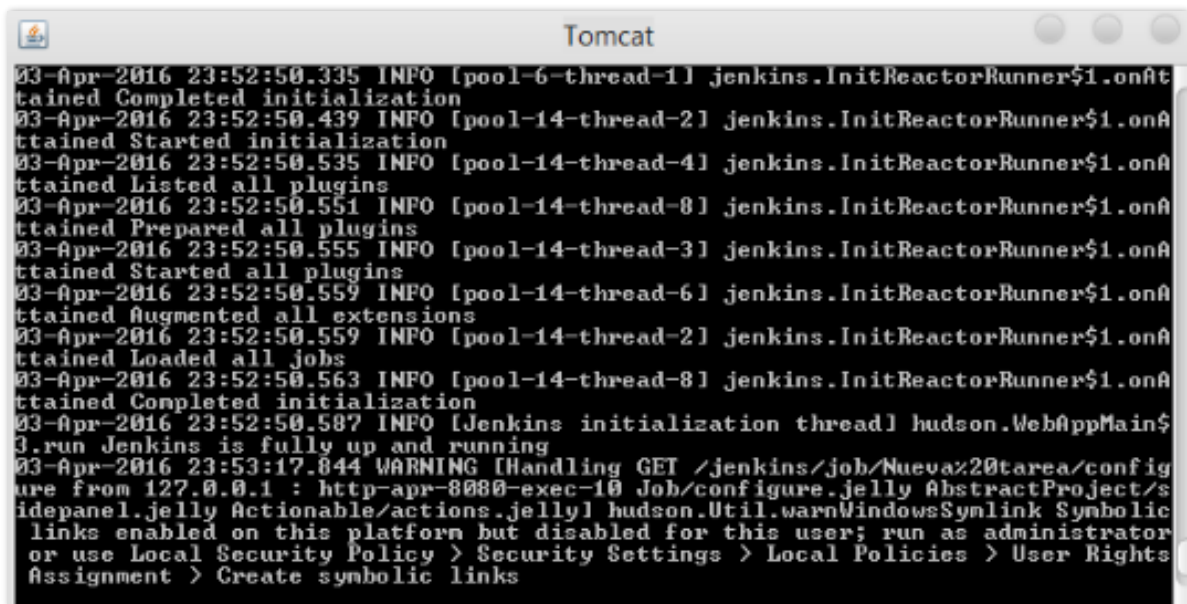
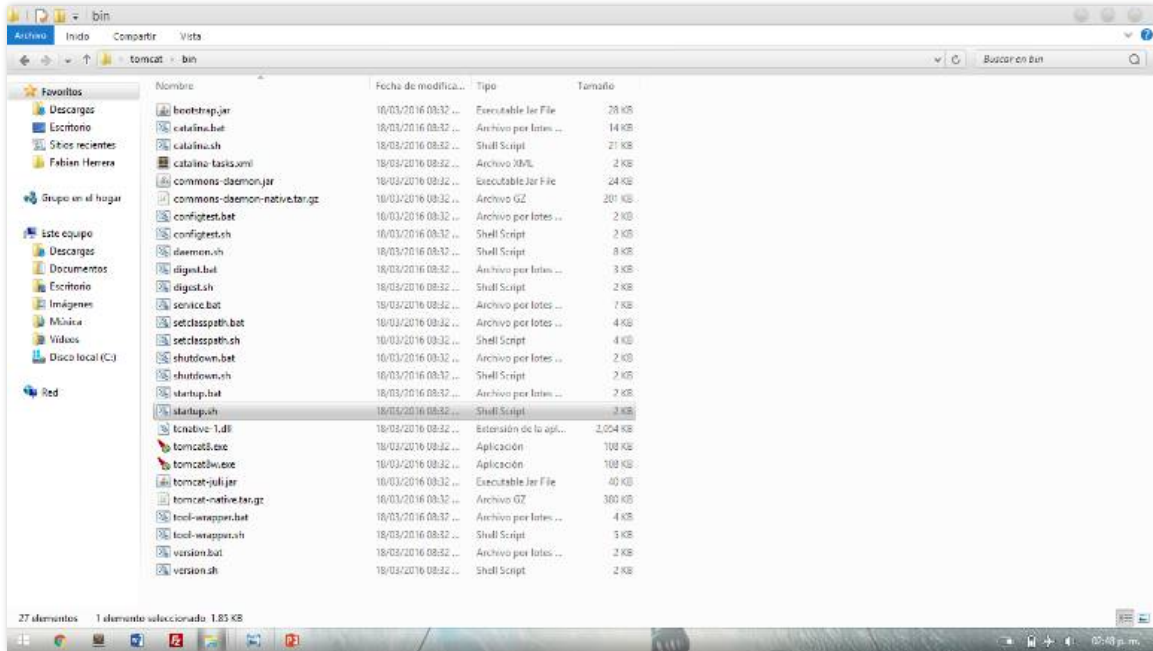
Descarga la herramienta de la página oficial. Observa que puedes descargar Jenkins como paquete nativo para tu sistema operativo o en formato WAR. Para nuestro ejemplo, te recomiendo la segunda opción.

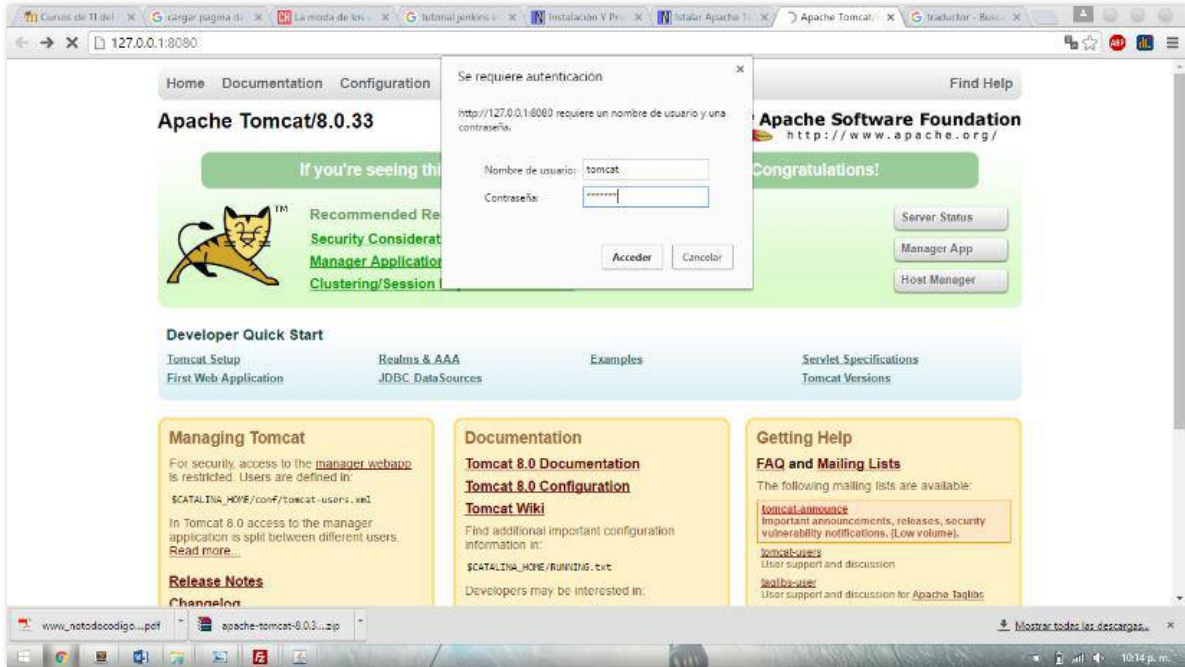


Ubica el WAR descargado (**war**) en la carpeta donde quieras instalar Jenkins.



Por ultimo levantamos el servidor con nuestro servidor Tomcat 8.





¡Bienvenido a Jenkins!

Por favor, **crea una nueva tarea** para empezar.

Daremos clic en crear una nueva tarea

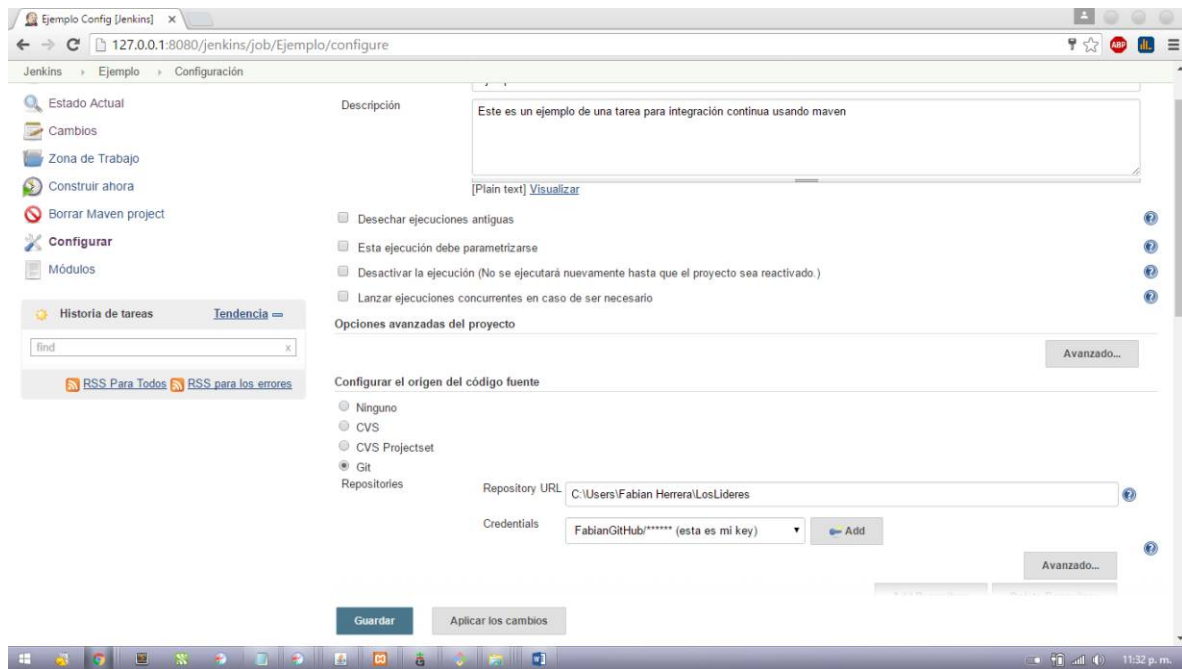
Y seleccionaremos la primera opción

Nombre de la Tarea

- ☒ **Crear un proyecto de estilo libre**
Esta es la característica principal de Jenkins, la de ejecutar el proyecto combinando cualquier tipo de repositorio de software (SCM) con cualquier modo de construcción o ejecución (make, ant, mvn, rake, script ...). Por tanto se podrá tanto compilar y empaquetar software, como ejecutar cualquier proceso que requiera monitorización.
- ☐ **Crear un proyecto maven**
Ejecuta un proyecto maven. Jenkins es capaz de aprovechar la configuración presente en los ficheros POM, reduciendo drásticamente la configuración.
- ☐ **Crear un proyecto multi-configuración**
Adecuado para proyectos que requieran un gran número de configuraciones diferentes, como testear en múltiples entornos, ejecutar sobre plataformas concretas, etc.
- ☐ **External Job**
Este tipo de tareas te permite registrar la ejecución de un proceso externo a Jenkins, incluso en una máquina remota. Está diseñado para usar Jenkins como un panel de control de tu sistema de automatización. Para más información consulta esta [página](#).

Y damos ok.

La siguiente pantalla nos mostrara lo siguiente.



Una vez asignemos un Nombre a nuestro proyecto, podremos agregar una descripción del mismo.

Al igual que seleccionamos el repositorio al que nuestro proyecto hará sus referencias para el control de versiones en mi caso es Git, pero cabe mencionar que también funciona github con este pluggin.

Cabe mencionar que mi configuración es de forma nativa, como quien dice tengo una copia del repositorio en mi máquina y mediante git y unos pequeños comandos, todos los archivos los vínculo con mi servidor en la nube, permitiéndome tener un respaldo y que mis colaboradores, puedan trabajar en el proyecto.

En fin sigamos con el documento, como mencionaba gracias a maven ya preinstalado, podremos tener nuestro servidor para pruebas de integración, ya que Jenkins cuenta con infinidad de pluggins para lograr programar en diversos lenguajes.

Bueno terminada la configuración del repositorio, solo pasaremos a la parte buena, en donde solo queda guardar nuestra tarea.

Una vez guardada, deberemos vincular con nuestro servidor para poder hacer lo de integración, si todo sale bien. Podremos tener acceso a los archivos del servidor.

Jenkins

Jenkins ▾ Ejemplo ▸

📌 Volver al Panel de Control

🔍 Estado Actual

📝 Cambios

📁 Zona de Trabajo

🗑️ Limpiar el espacio de trabajo

🕒 Construir ahora

🚫 Borrar Proyecto

⚙️ Configurar

Workspace of Ejemplo on principal

📁 .git

📄 Buena.jpg 152.48 KB [vista](#)

📄 index.php 191 B [vista](#)

📄 Thumbs.db 10.00 KB [vista](#)

📦 [\(Descargar archivos en formato zip\)](#)

🔆 Historia de tareas [Tendencia](#)

find x

🌐 #1 18-abr-2016 23:51

📡 [RSS Para Todos](#) 📡 [RSS para los errores](#)

Como se puede ver el proyecto se ha cargado de forma correcta.

Ahora haremos algunos cambios en el mismo para ver que pasa si volvemos a contruir nuestro proyecto.

```
MINGW64:/c/Users/Fabian Herrera/LosLideres

Fabian Herrera@hp MINGW64 ~/LosLideres (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 2 commits.
(use "git push" to publish your local commits)
nothing to commit, working directory clean

Fabian Herrera@hp MINGW64 ~/LosLideres (master)
$ git pull
Already up-to-date.

Fabian Herrera@hp MINGW64 ~/LosLideres (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 2 commits.
(use "git push" to publish your local commits)
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        diagrama.jpg

nothing added to commit but untracked files present (use "git add" to track)

Fabian Herrera@hp MINGW64 ~/LosLideres (master)
$ git add diagrama.jpg

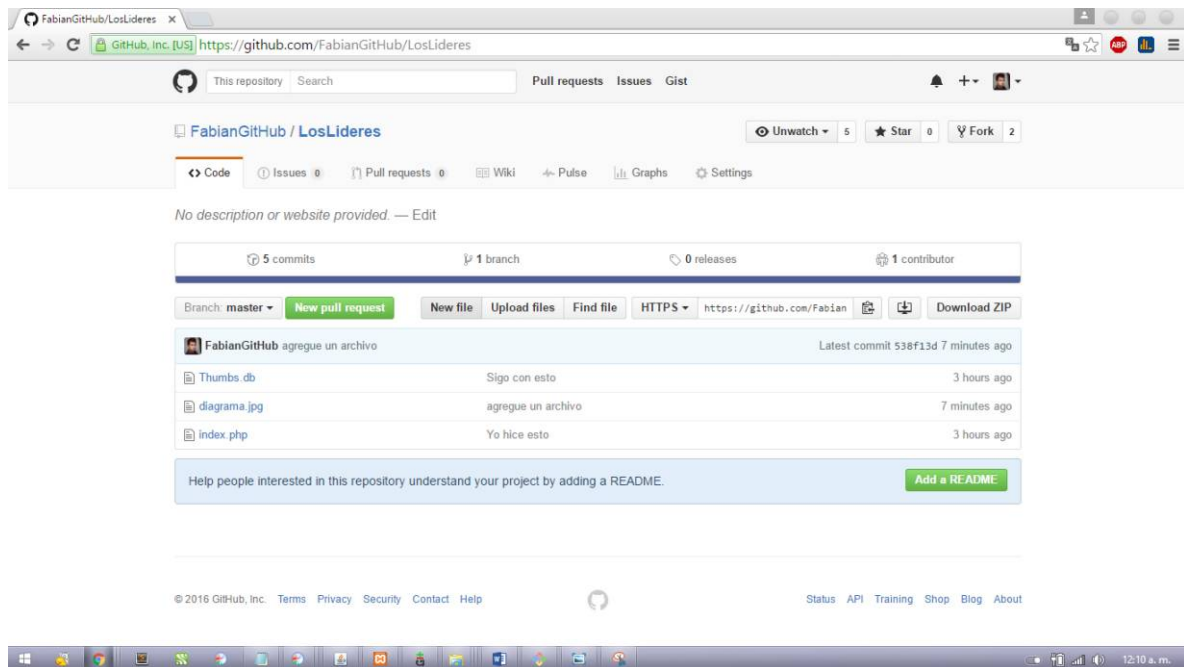
Fabian Herrera@hp MINGW64 ~/LosLideres (master)
$ git commit -m "agregue un archivo"
[master 538f13d] agregue un archivo
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 diagrama.jpg

Fabian Herrera@hp MINGW64 ~/LosLideres (master)
$ git pull
Already up-to-date.

Fabian Herrera@hp MINGW64 ~/LosLideres (master)
$ |
```

```
Fabian Herrera@hp MINGW64 ~/LosLideres (master)
$ git push
Counting objects: 8, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (8/8), done.
Writing objects: 100% (8/8), 290.41 KiB | 0 bytes/s, done.
Total 8 (delta 1), reused 0 (delta 0)
To https://github.com/FabianGitHub/LosLideres.git
1269b87..538f13d master -> master
```

Ahora veamos nuestro repositorio



Como podemos ver, los cambios han sido efectuados.

Ahora actualizamos el proyecto en Jenkins.



Workspace of Ejemplo on principal

Historia de tareas

Build	Time
#3	19-abr-2016 0:14
#2	18-abr-2016 23:58
#1	18-abr-2016 23:51

Página generada: 19-abr-2016 0:15:20 Jenkins ver. 1.642.4

Como podemos ver, se ha actualizado nuestro repositorio, gracias a esto podremos tener la integración continua de una manera moderada y gracias a nuestro repositorio, podremos laborar de forma remota con nuestros colaboradores.