



AGORA@US

Grupo 9: Creación y Administración de Censos



2016/2017

Grupo 2 (tarde)

Id Opera: 9

Pablo Romero Vázquez

Santiago Fraga Martín-Arroyo

Nicolás Lorenz Rosado

Simón Egea Guerrero

Rubén Barrientos Mohedano

Historial de versiones

Versión	Autor	Descripción	Fecha
1.0	Santiago Fraga Martín-Arroyo	Realización del esquema del documento y los apartados de gestión del código deficiente, gestión de incidencias deficientes y mejora de la documentación	30/01/2017
1.1	Pablo Romero Vázquez	Realización del apartado de .gitignore y corrección del documento	30/01/2017
1.2	Santiago Fraga Martín-Arroyo	Corrección de errores y realización del apartado de Uso de Jenkins solo en local	31/01/2017
1.3	Nicolás Lorenz Rosado	Creación del apartado de Integración con otros subsistemas	31/01/2017
1.3	Santiago Fraga Martín-Arroyo	Unión de los distintos apartados, corrección del formato y revisión.	01/02/2017

Índice

Historial de versiones.....	2
Tabla de figuras.....	4
Resumen.....	6
Mejoras realizadas.....	7
Gestión del código deficiente	7
Gestión de incidencias deficiente	8
No se usa el .gitignore	11
Uso de Jenkins solo en local	12
Integración con otros sistemas.....	20
El .war, el despliegue y la construcción se genera de forma manual	25
Mejora de la documentación.....	34

Tabla de figuras

Ilustración 1	Commits de uno de los miembros	7
Ilustración 2	Incidencia en github.....	8
Ilustración 3	Incidencia #23	8
Ilustración 4	Labels de una incidencia	9
Ilustración 5	Archivo gitignore.....	11
Ilustración 6	Ejemplo de git rm.....	11
Ilustración 7	Login de jenkins	12
Ilustración 8	Login de Jenkins.....	12
Ilustración 9	Panel principal de jenkins.....	13
Ilustración 10	Administracion de Jenkins.....	13
Ilustración 11	Instalar plugin Jenkins.....	14
Ilustración 12	Configurar plugin de maven Jenkins	14
Ilustración 13	Configurar jdk en jenkins.....	15
Ilustración 14	Crear tarea en Jenkins.....	15
Ilustración 15	Configurar tarea en jenkins 1	16
Ilustración 16	Configurar disparadores en Jenkins.....	17
Ilustración 17	Configurar origen del codigo en Jenkins	17
Ilustración 18	Configurar pasos previos en Jenkins.....	18
Ilustración 19	Configurar proyecto en Jenkins	18
Ilustración 20	Ver tarea en jenkins	19
Ilustración 21	Demostración conexión 1.....	20
Ilustración 22	Demostración conexión 2.....	21
Ilustración 23	Antiguo método de crear censo	21
Ilustración 24	Modificación de servicio.....	22
Ilustración 25	Modificación controlador	22
Ilustración 26	Adición de línea para crear un censo	22
Ilustración 27	Datos del fichero a usar.....	23
Ilustración 28	Selección del método.....	23
Ilustración 29	Demostración del método.....	23
Ilustración 30	Heroku login	25
Ilustración 31	Heroku create	25
Ilustración 32	Heroku instalación plugin.....	25
Ilustración 33	Heroku deploy	26
Ilustración 34	Heroku panel de control	26
Ilustración 35	Heroku application error	27
Ilustración 36	App Engine	28
Ilustración 37	Elastical creación del entorno	28
Ilustración 38	Elastical entorno creado	29
Ilustración 39	Elastical creación de la base de datos	29
Ilustración 40	Elastical despliegue.....	30
Ilustración 41	Versión de MYSQL Server.....	30
Ilustración 42	Versión de maven	30
Ilustración 43	Versión de git.....	31
Ilustración 44	Creación base de datos	31
Ilustración 45	Despliegue war	31
Ilustración 46	Archivo host.....	31

Ilustración 47Configuración archivo hosts	31
Ilustración 48Vaciado de DNS	32
Ilustración 49ADMCensus	32
Ilustración 50Copy Artifact plugin	32
Ilustración 51Deploy to Websphere container plugin	32
Ilustración 52Jobs	33

Resumen

En la siguiente tabla se muestran todas las mejoras que el profesor de la asignatura nos ha recomendado realizar, a quien se le ha asignado dicha mejora y si se ha realizado correctamente.

Titulo	Descripción	Asignado a	Solucionado
Gestión del código deficiente	Uno de los miembros del grupo solo realiza un commit a lo largo del desarrollo	Pablo Romero Vázquez	SI
Gestión de incidencias deficiente	La gestión de incidencias es demasiado pobre y no se pone en práctica con naturalidad	Grupo	SI
No se usa .gitignore	Cada vez que se hace un commit se suben archivos que no son necesarios como los .class, .classpath o .project	Pablo Romero Vázquez	SI
Uso de Jenkins solo en local	Jenkins solo se usa en local	Grupo	SI
Integración con otros sistemas	El sistema no interacciona con otros subsistemas	Nicolás Lorenz Rosado	SI
El .war se genera de forma manual	La generación del war se realiza de forma manual	Simón Egea Guerrero y Rubén Barrientos Mohedano	SI
El despliegue y construcción se hace manual	Tanto el despliegue como la construcción del sistema se hace de forma manual	Simón Egea Guerrero y Rubén Barrientos Mohedano	SI
Mejora de la documentación	Se debe de mejorar la documentación en su totalidad	Pablo Romero Vázquez y Santiago Fraga Martín-Arroyo	SI

Mejoras realizadas

A continuación, se detallan las mejoras que se han conseguido realizar y como se han llevado a cabo.

Gestión del código deficiente

Uno de los miembros del grupo solo había realizado un “commit” durante todo el desarrollo del subsistema, debido a esto se decidió que realizase el archivo “.gitignore” y algunos test que faltaban por realizar.

Tras la realización de las tareas anteriormente mencionadas dicho miembro ha aumentado sus “commits” hasta llegar a 7, tal y como podemos observar en la siguiente figura.



Ilustración 1 Commits de uno de los miembros

Gestión de incidencias deficiente

Debido a que no teníamos soltura a la hora de gestionar las incidencias generadas en nuestro subsistema se optó por realizar una mejora de esta sección la cual viene detallada a continuación.

Para gestionar las incidencias se deberá exponer dicha incidencia de la forma más clara posible y evitando ambigüedades. Si es necesario se deben de poner los pasos a seguir para reproducir el error en otra máquina distinta.

Comenzamos situándonos en el apartado issues de GitHub:

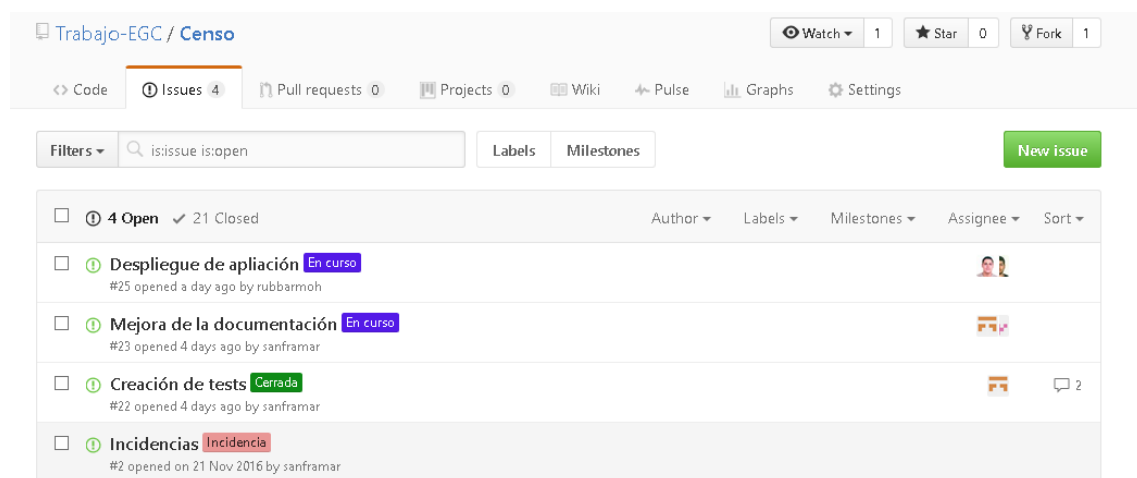


Ilustración 2 Incidencia en github

Como se observa en la imagen, tenemos 25 incidencias, 4 abiertas y 21 cerradas. En este caso vamos a ver la incidencia “Mejora de la documentación”.

Mejora de la documentación #23

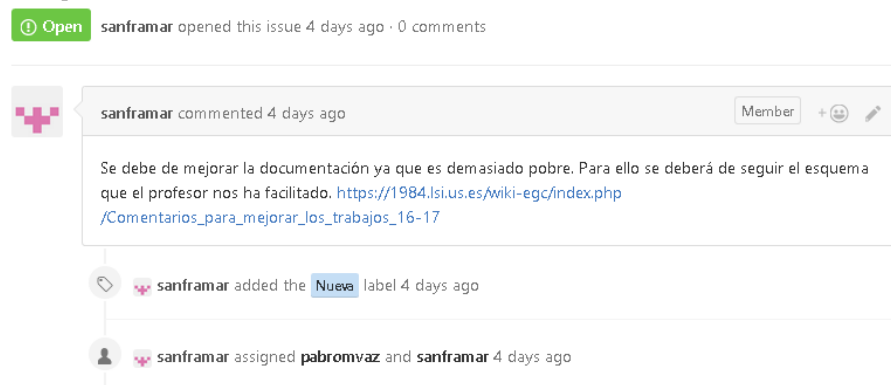


Ilustración 3 Incidencia #23

Tal y como puede observarse, la incidencia sigue el esquema anteriormente mencionado, compuesto de un título, una descripción y una etiqueta.

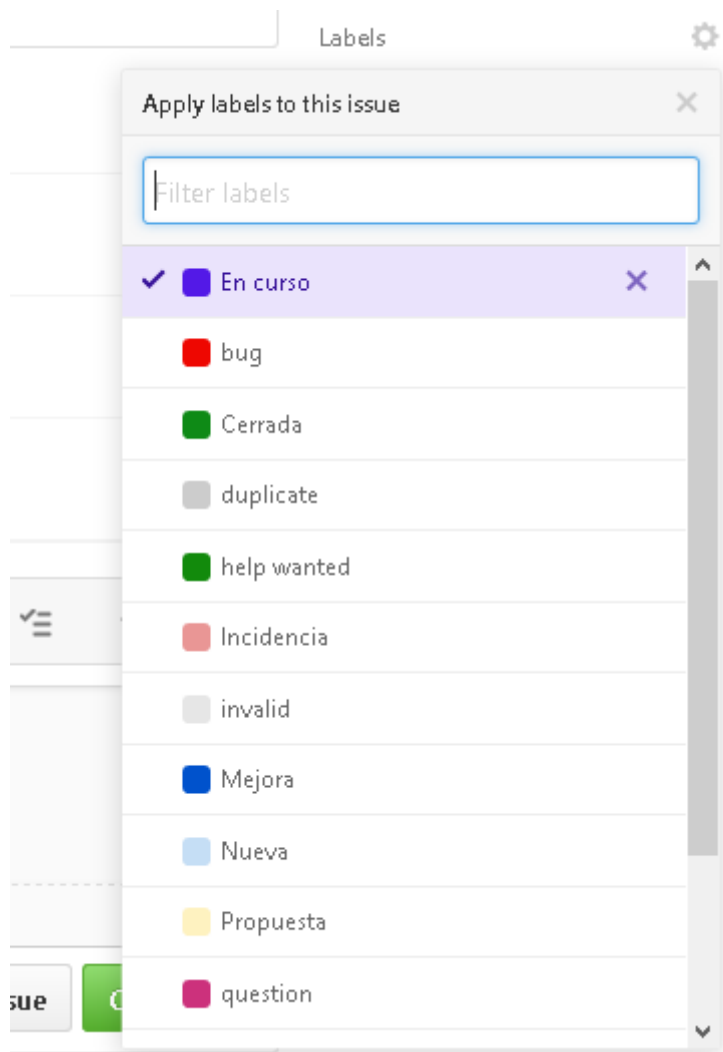


Ilustración 4 Labels de una incidencia

En el apartado “Label”, podremos asignar la etiqueta que creamos oportuna a la incidencia. En caso de no existir la etiqueta que queramos, se puede crear.

En nuestro caso, la evolución de las etiquetas de una incidencia seria la siguiente:

- Nueva: incidencia recién creada y en la que aún no se está trabajando
- En curso: la incidencia ya tiene a un miembro del equipo asignado y está trabajando en ella.
- Revisar: la incidencia ya está implementada en una rama distinta a la rama master para ser probada por un miembro distinto al que trabajó en ella.

- Cerrada: la incidencia funciona correctamente y se ha implementado en la rama master.

Cabe destacar que en la incidencia se puede mencionar a un determinado compañero mediante el comando “@nombreDelCompañero” para que GitHub le notifique. Además, también pueden referenciarse incidencias en los commits mediante los comandos “Fixed”, “Fixes” o “Close” + #numeroDeLaIncidencia. Si por algún casual vuelven a surgir conflictos, la incidencia puede ser reabierta.

No se usa el .gitignore

Con el objetivo de eliminar del repositorio archivos innecesarios y evitar que vuelvan a ser subidos, se ha creado un fichero “.gitignore”.

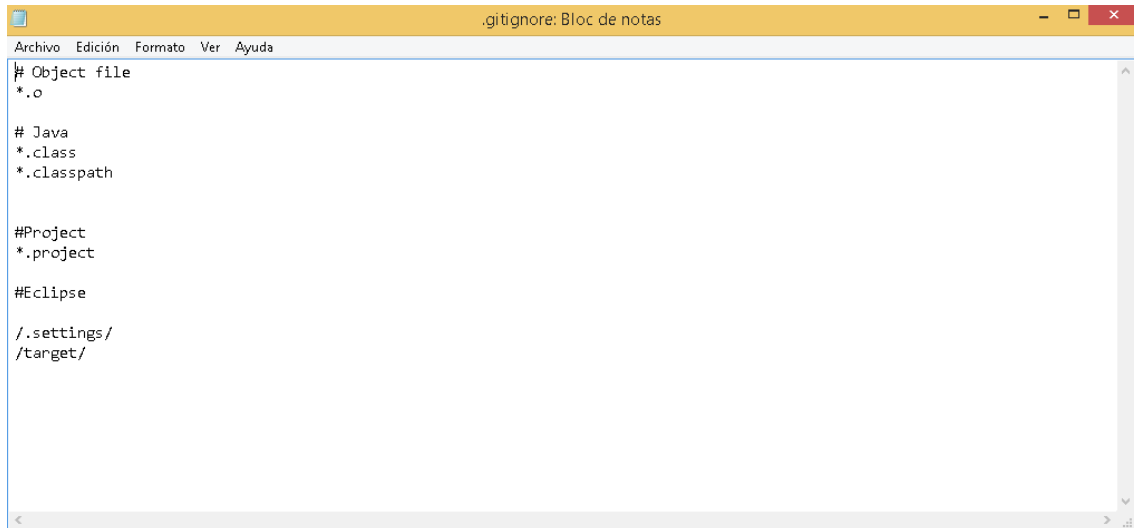


Ilustración 5 Archivo gitignore

En la ilustración podemos observar aquellos archivos que no deseamos que sean subidos. Cabe destacar que en “#Eclipse” hemos indicado que no se suban las carpetas “settings” y “target”, de forma que se ignora todo lo que contenga.

Previamente, para borrar los archivos del repositorio, podemos hacerlo eliminando manualmente los archivos y haciendo un push al repositorio o bien podemos hacerlo con los comandos que nos proporciona git y realizando el push. En nuestro caso, hemos usado los comandos de git. Dichos comandos son “git rm nombreDelArchivo” y “git rm -r nombreDeCarpeta” en caso de que queramos borrar los archivos tanto del repositorio como del sistema y “git rm --cached nombreDelArchivo” si queremos borrar los archivos en el repositorio pero queremos mantenerlos en el sistema. Como los archivos que no se subirán a GitHub son los que genera automáticamente eclipse, se optó por borrarlos también del sistema y una vez creado el archivo “.gitignore”, ya no volverían a ser subidos.

```
c:\Temp\UM\Censo>git rm .project
```

Ilustración 6 Ejemplo de git rm

Uso de Jenkins solo en local

Se ha configurado Jenkins para que automáticamente se descargue el código de nuestro repositorio en Github y ejecute las pruebas para comprobar que todo funciona correctamente, dicha configuración viene detallada a continuación.

La primera vez que entremos en Jenkins nos pedirá un usuario y contraseña, deberemos de poner la cuenta con la que nos registramos a la hora de instalar Jenkins.

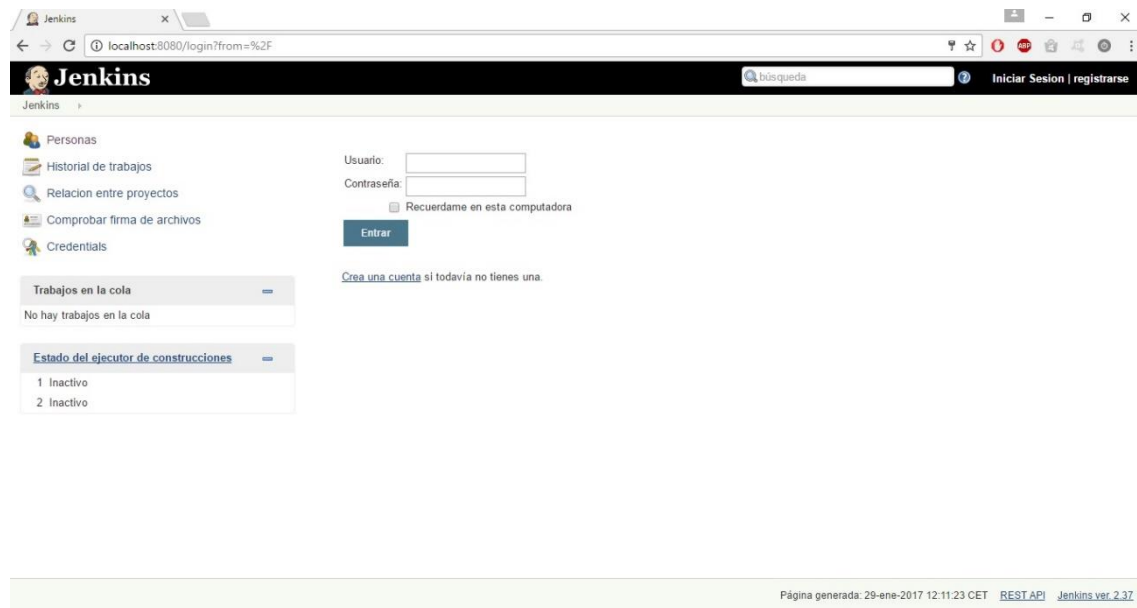
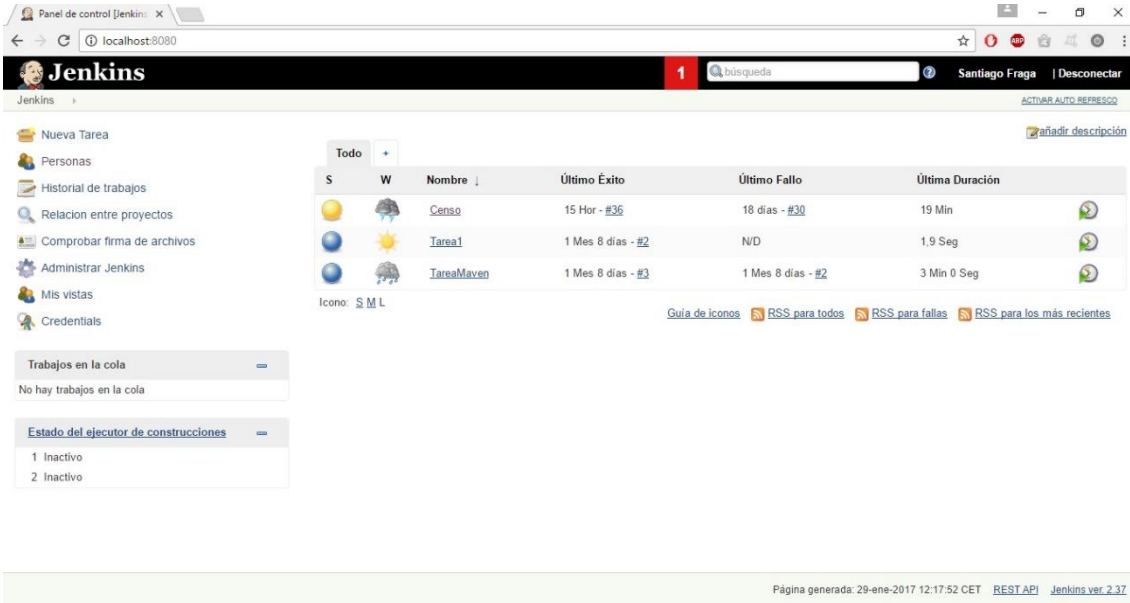


Ilustración 8Login de Jenkins

Una vez logueados en nuestro Jenkins, nos saldrán todos los proyectos que tenemos y una pequeña información sobre si se ejecutaron bien o si por el contrario hubo algún tipo de problema.

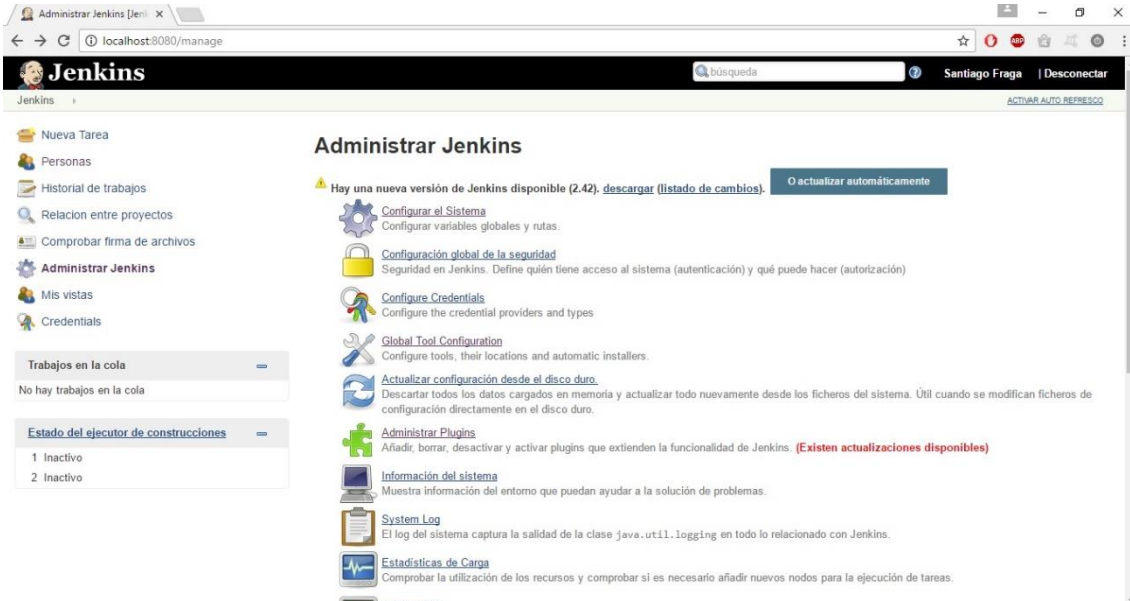


The screenshot shows the Jenkins dashboard. On the left, there is a sidebar with navigation links: Nueva Tarea, Personas, Historial de trabajos, Relacion entre proyectos, Comprobar firma de archivos, Administrar Jenkins, Mis vistas, and Credentials. The main area displays a table of jobs. The table has columns for S (Success), W (Warning), Nombre (Name), Último Exito (Last Success), Último Fallo (Last Failure), and Última Duración (Last Duration). The jobs listed are Censo, Tarea1, and TareaMaven. Below the table, there are links for 'Guía de iconos', 'RSS para todos', 'RSS para fallos', and 'RSS para los más recientes'. On the left sidebar, there are sections for 'Trabajos en la cola' (No hay trabajos en la cola) and 'Estado del ejecutor de construcciones' (1 Inactivo, 2 Inactivo).

S	W	Nombre	Último Exito	Último Fallo	Última Duración
		Censo	15 Hor - #36	18 dias - #30	19 Min
		Tarea1	1 Mes 8 dias - #2	N/D	1.9 Seg
		TareaMaven	1 Mes 8 dias - #3	1 Mes 8 dias - #2	3 Min 0 Seg

Ilustración 9 Panel principal de jenkins

Debido a que nuestro proyecto usa Maven deberemos de introducir el plugin para Maven en nuestro Jenkins. Para ello iremos a “Administrar Jenkins” situado en la parte izquierda de la pantalla.



The screenshot shows the 'Administrar Jenkins' page. The sidebar on the left is the same as in the previous screenshot. The main area is titled 'Administrar Jenkins' and contains a list of configuration options: Configurar el Sistema, Configuración global de la seguridad, Configure Credentials, Global Tool Configuration, Actualizar configuración desde el disco duro, Administrar Plugins, Información del sistema, System Log, and Estadísticas de Carga. A notification at the top indicates that a new version of Jenkins (2.42) is available for download.

Ilustración 10 Administracion de Jenkins

Una vez en la siguiente ventana nos iremos a la sección de “Administrar plugins”.

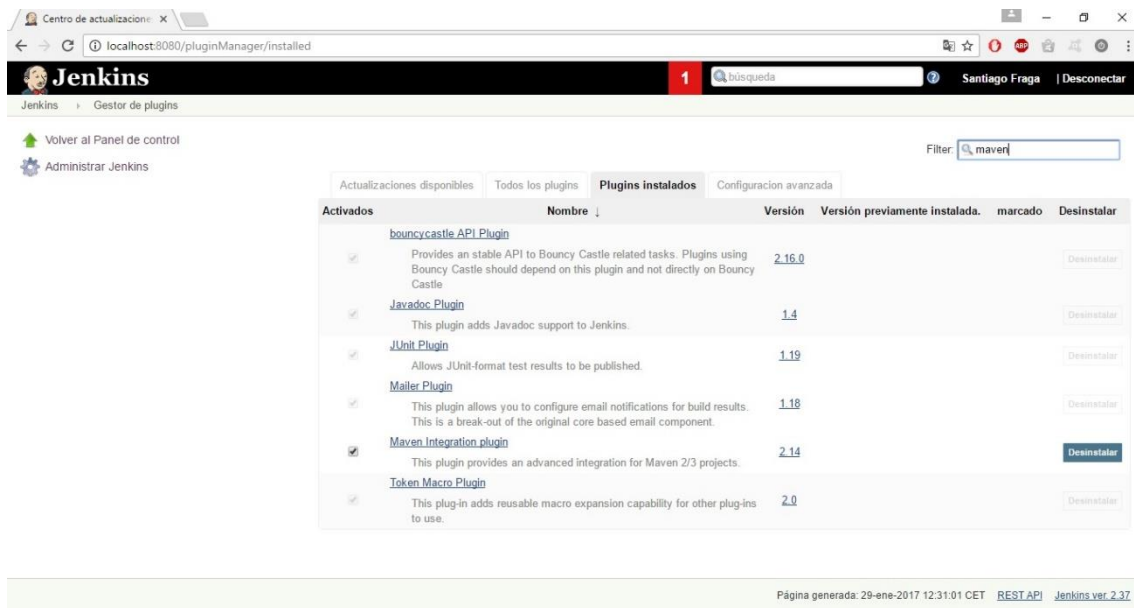


Ilustración 11 Instalar plugin Jenkins

En esta ventana escribiremos en el buscador que nos proporciona Jenkins “maven” y seleccionaremos el plugin llamado “Maven Integration plugin”. Una vez seleccionado Jenkins lo descargará y se reiniciará para hacer efectivos los cambios.

Una vez completado dicho paso volveremos a la ventana de “Administrar Jenkins” pero esta vez accederemos a la opción de “Global Tool Configuration” para configurar mínimamente el plugin de Maven.

Tras acceder a dicha ventana nos iremos a la sección de “Maven” la cuál configuraremos tal y como se muestra en la figura siguiente:

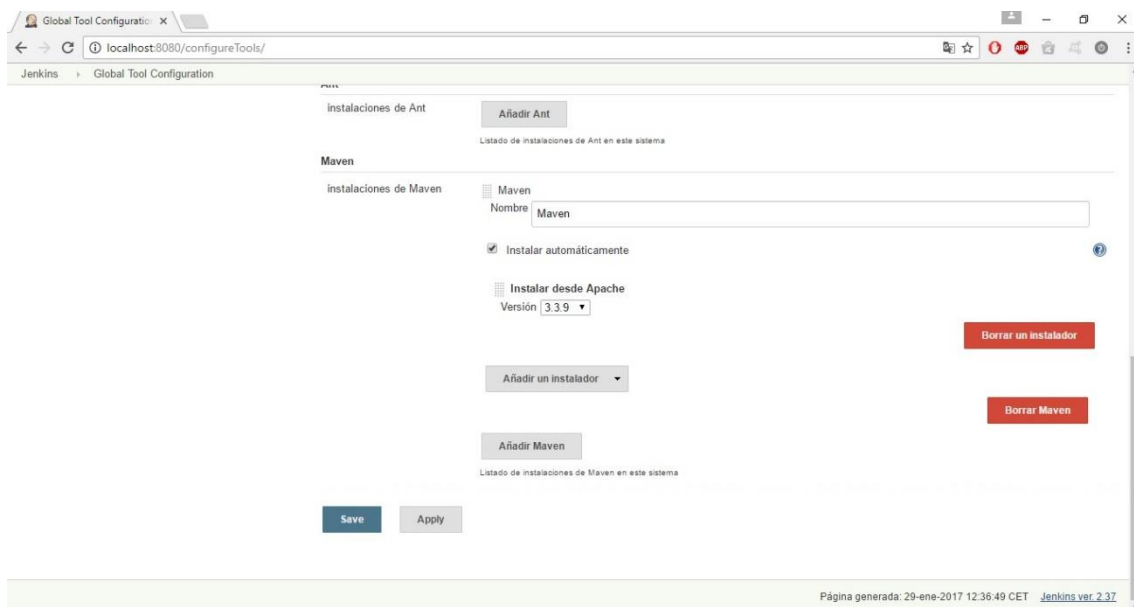


Ilustración 12 Configurar plugin de maven Jenkins

Una vez realizado este paso ya tendremos nuestro maven configurado para Jenkins. Ahora, en la misma ventana debemos de configurar el JDK ya que nuestro proyecto está realizado casi en su totalidad con Java. Para configurar dicho JDK nos iremos a la sección “JDK” y la configuraremos tal y como se muestra en la imagen:

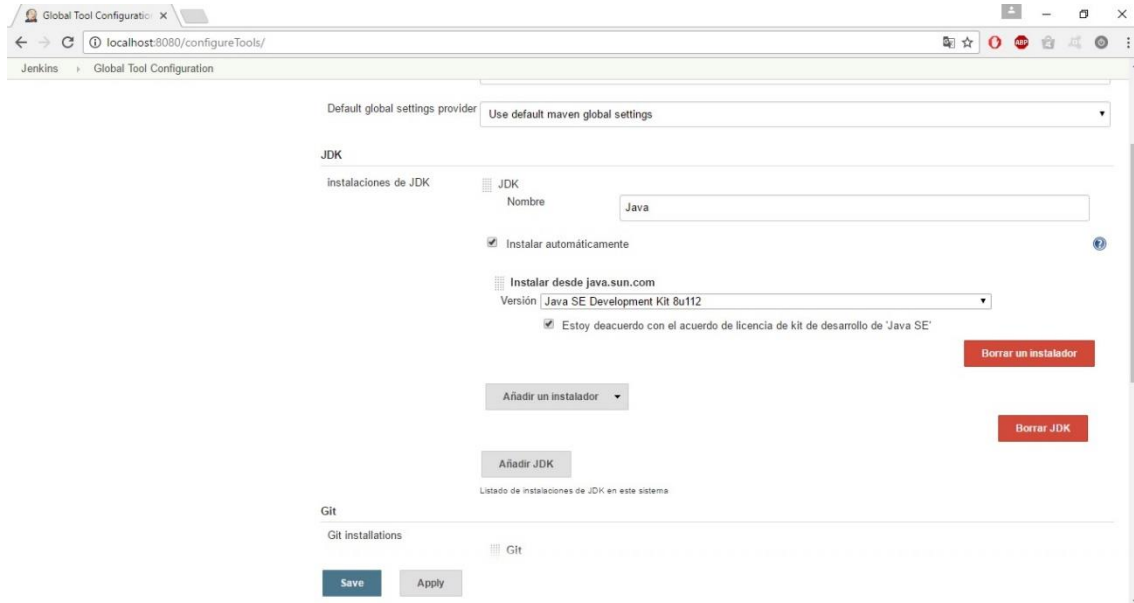


Ilustración 13 Configurar jdk en jenkins

Tras realizar este último paso ya tendremos nuestro Jenkins listo para ejecutar nuestro proyecto.

Ahora regresaremos a la ventana principal y le daremos a “Nueva tarea” situada en la parte izquierda de la pantalla.

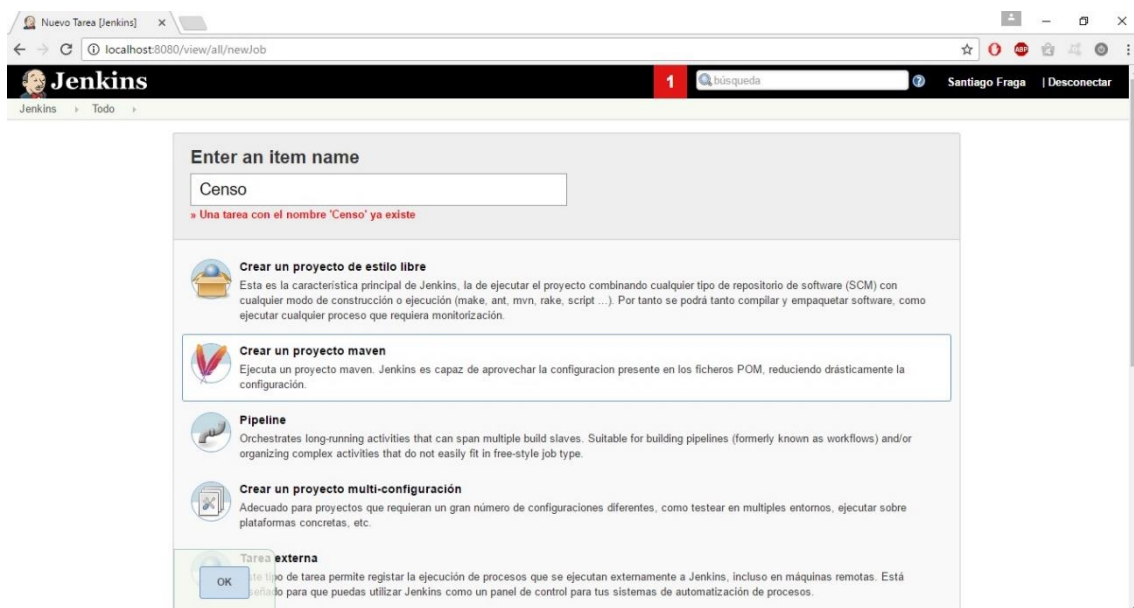


Ilustración 14 Crear tarea en Jenkins

Debido a que en nuestro caso se trata de un proyecto maven seleccionaremos la opción de “Crear un proyecto maven”. Posteriormente, le daremos un nombre y pulsaremos en ok.

Una vez realizada la siguiente acción nos aparecerá la siguiente pantalla para configurar nuestra tarea:

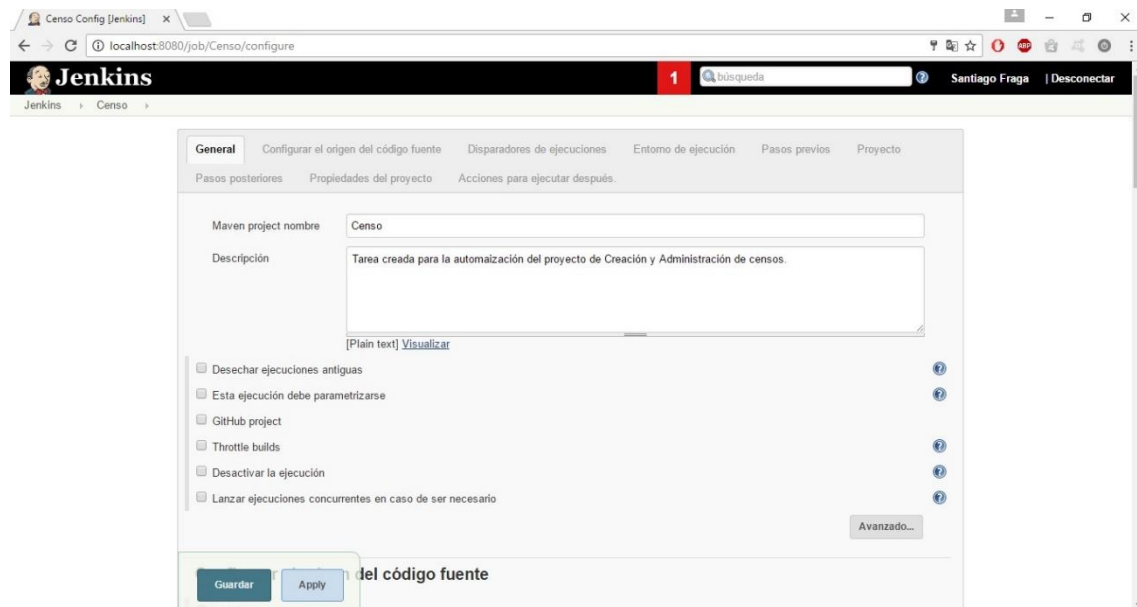


Ilustración 15Configurar tarea en jenkins 1

En la sección “General” solo deberemos de introducir una pequeña descripción de la tarea que hemos creado.

La siguiente sección a configurar es “Configurar el origen del código fuente”, en dicha sección le diremos a Jenkins que nos coja el código de nuestro repositorio de Github y para ser más concretos de nuestra rama master.

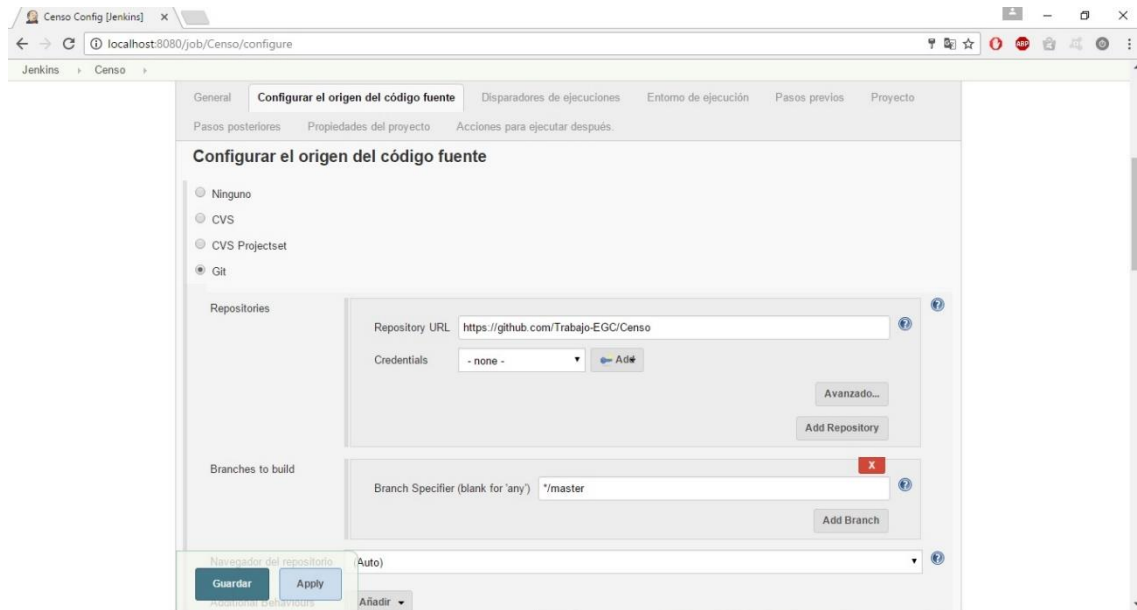


Ilustración 17 Configurar origen del código en Jenkins

El siguiente apartado que debemos de configurar es el de “Disparadores de ejecuciones” en el cuál le diremos a Jenkins que se debe de ejecutar cada vez que se haga un “push” a nuestro repositorio o cada x tiempo.

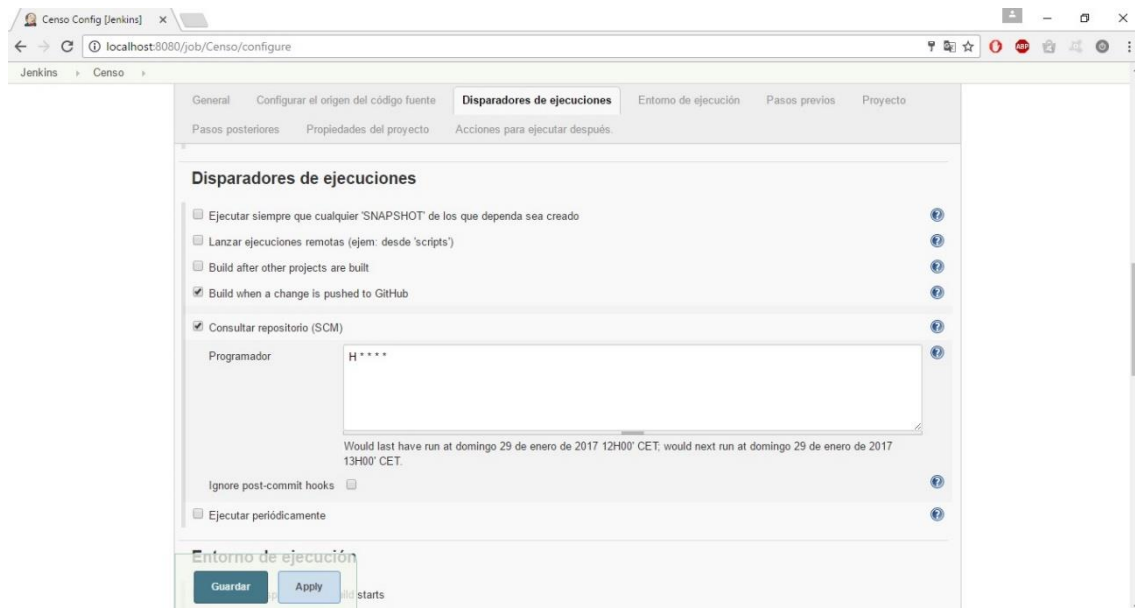
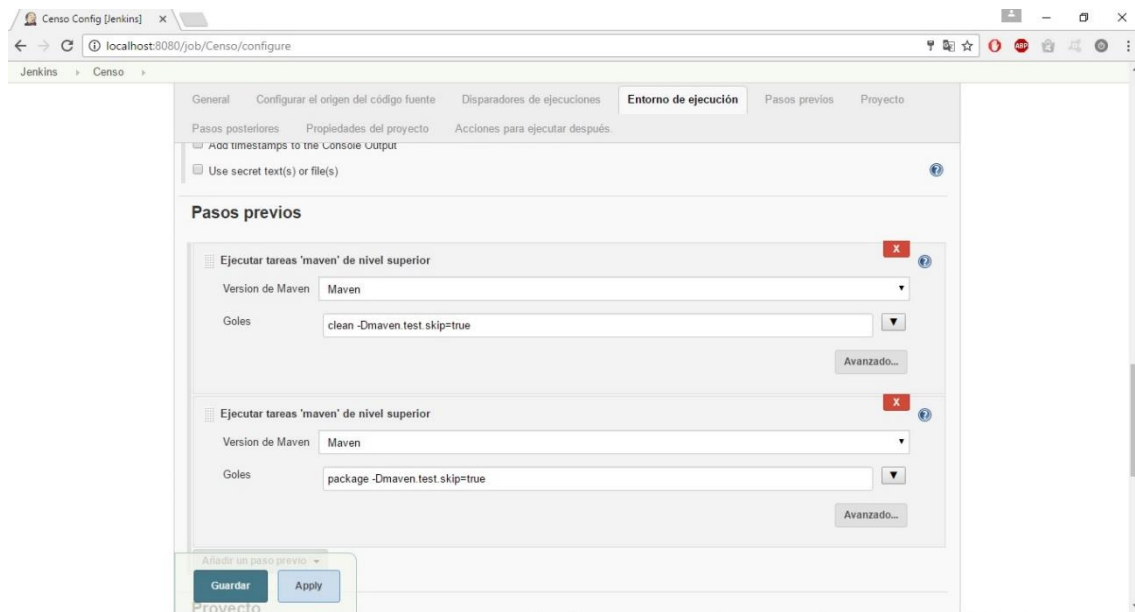


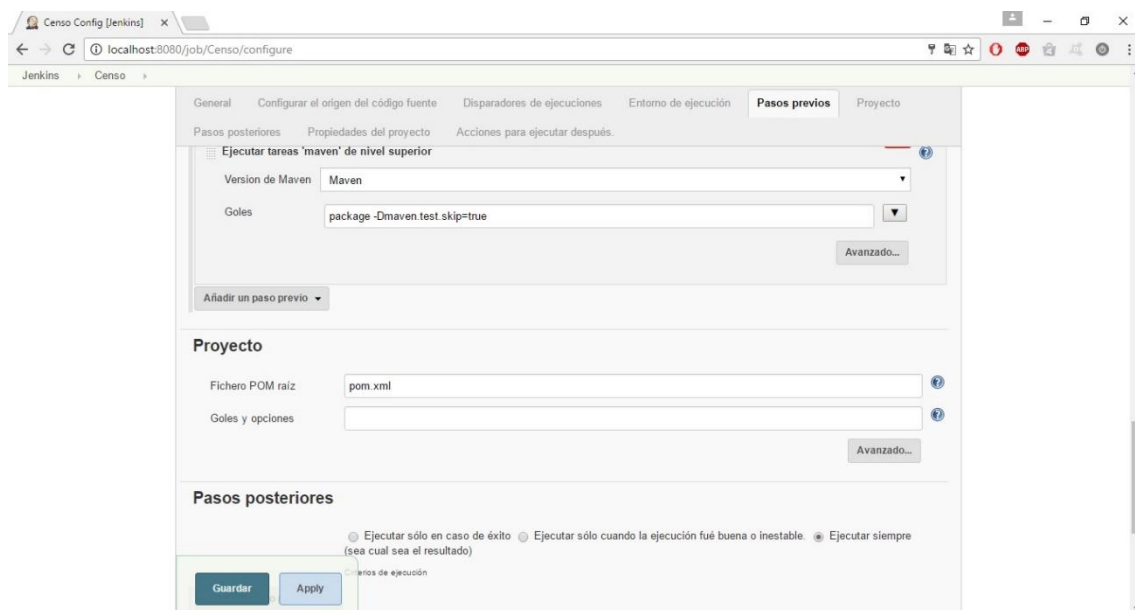
Ilustración 16 Configurar disparadores en Jenkins

Después de configurar esta sección toca irnos a la sección de “Pasos previos” la cuál configuraremos tal y como se nos muestra en la siguiente imagen:



*Ilustración 18*Configurar pasos previos en Jenkins

Tras este paso lo único que nos quedaría por configurar sería la sección de “Proyecto” ya que debemos de decirle donde se encuentra nuestro archivo “pom.xml” que es el que usa Maven para configurar todas las dependencias y demás. Esto lo hacemos de la siguiente forma:



*Ilustración 19*Configurar proyecto en Jenkins

Después de realizar este paso ya tendremos nuestro Jenkins configurado para la ejecución de nuestra aplicación.

Podemos ver con más detalle nuestra tarea si pinchamos en ella desde la ventana principal de Jenkins.

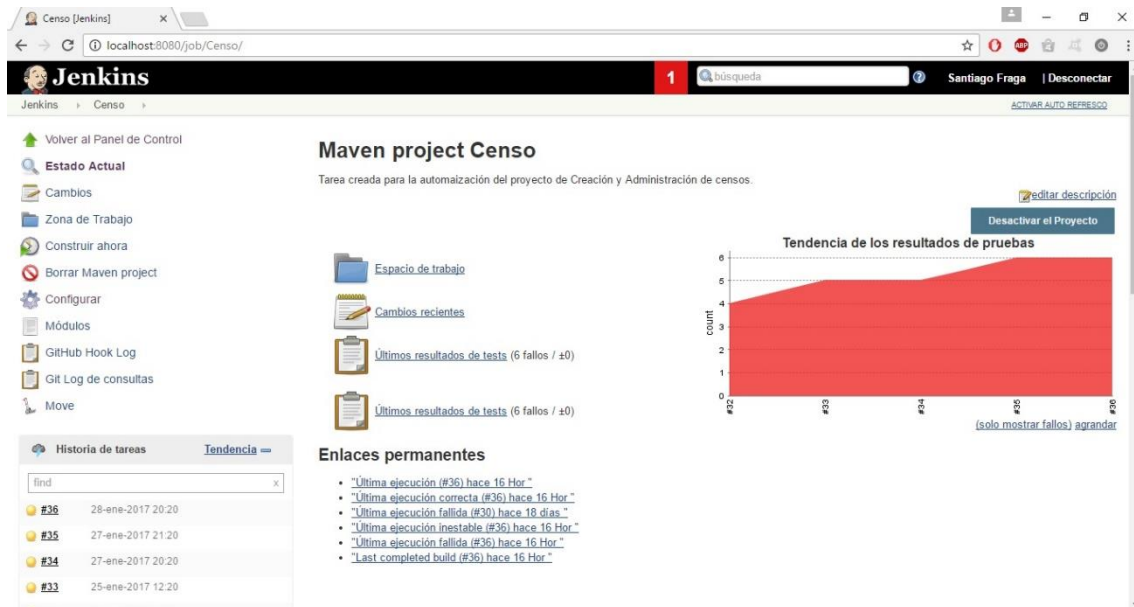


Ilustración 20 Ver tarea en Jenkins

En dicha ventana podemos observar las distintas ejecuciones de nuestra tarea así como si ha dado fallo. Además, podremos forzar una ejecución dándole a “Construir ahora”.

Integración con otros sistemas

Se ha mirado con que otros proyectos se podía realizar la integración y se encontró que podíamos hacerlo con cabina de votación y creación y administración de votaciones.

Detallaré como ha ido la integración con ambas:

Cabina

Lo primero que se hizo fue hacer un clone del proyecto en Git en la carpeta del workspace donde trabajo para así poder investigar sobre que podíamos integrar con nuestro proyecto.

Después de inspeccionar dicho código no encontramos nada para poder realizar la integración así que se pasó directamente a buscar maneras de integrar la otra opción.

Votación

Al igual que con el proyecto de Git, se realizó un clone del proyecto para poder trabajar con él y poder investigar el código.

En este caso, al ejecutar los dos proyectos en paralelo se puede acceder de uno a otro, viendo que están conectados:

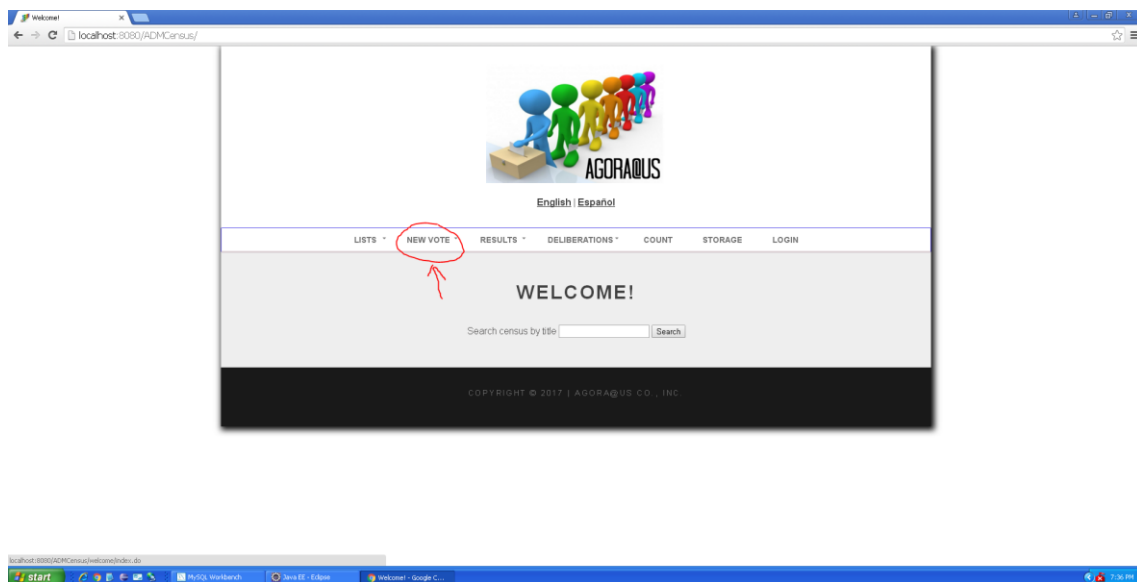


Ilustración 21 Demostración conexión 1

Si le damos a New Vote, nos lleva al sistema de votación, y si desde ahí le damos a volver al menú principal nos llevaría de vuelta a nuestro proyecto:

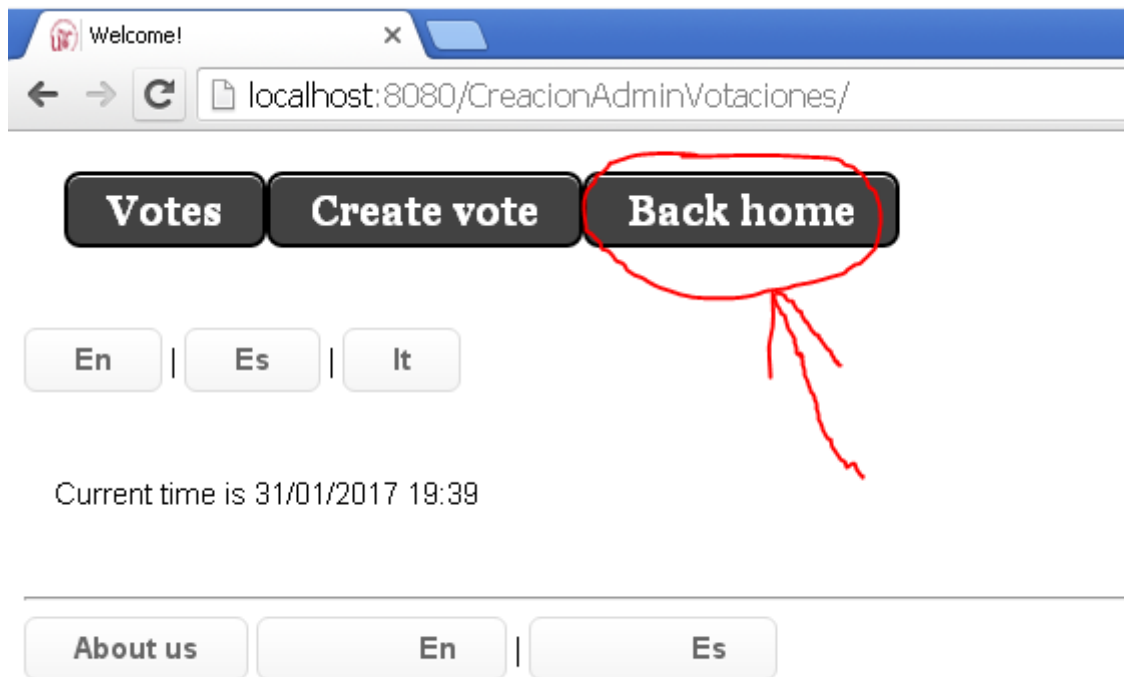


Ilustración 22 Demostración conexión 2

Aunque al igual que el otro proyecto no podemos integrar servicios así que vamos a realizar una simulación como si recibiéramos los datos desde votación y los usaremos en los servicios que hagan uso de estos datos.

Por ejemplo, vemos que para crear un censo necesitamos idVotación, tipoVotación, etc

```
* Crea un censo a partir de una votación
*
* @param idVotacion
*         = Identificador de la votación
* @param username
*         = Nombre de usuario que ha creado la votacion
* @param fechaInicio
*         = Fecha de inicio de la votacion
* @param fechaFin
*         = Fecha de fin de la votacion
* @param tituloVotacion
*         Cadena de texto con el titulo de la votacion
* @param tipoVotacion
*         Cadena de texto con el tipo de la votacion (abierta o cerrada)
* @return census
* @throws ParseException
*/
```

Ilustración 23 Antiguo método de crear censo

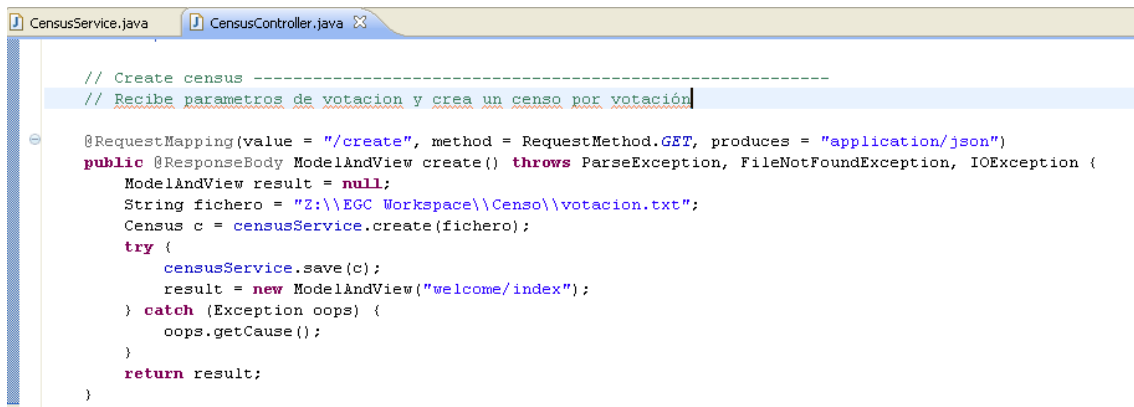
Esos datos los recibiremos desde un fichero, el cual vamos a leer y extraer la información para poder así trabajar con ella.

Se modifican el servicio visto antes, el test para que coincida con el nuevo servicio y por último el controlador.



```
public Census create(String archivo) throws FileNotFoundException, IOException, ParseException {  
  
    FileReader f = new FileReader(archivo);  
    BufferedReader b = new BufferedReader(f);  
  
    String votacion;  
    String username;  
    String fechaInicio;  
    String fechaFin;  
    String tituloVotacion;  
    String tipoVotacion;  
  
    votacion = b.readLine();  
    username = b.readLine();  
    fechaInicio = b.readLine();  
    fechaFin = b.readLine();  
    tituloVotacion = b.readLine();  
    tipoVotacion = b.readLine();  
  
    b.close();  
  
    int idVotacion = Integer.parseInt(votacion);  
  
    Assert.isTrue(!username.equals(""));  
    Assert.isTrue(tipoVotacion.equals("abierta") || tipoVotacion.equals("cerrada"));  
    Census result = new Census();  
    long startDate = Long.parseLong(fechaInicio);  
    long finishDate = Long.parseLong(fechaFin);  
    Date fechaComienzo = new Date(startDate);  
    Date fechaFinal = new Date(finishDate);  
    Assert.isTrue(fechaComienzo.before(fechaFinal));  
}
```

Ilustración 24 Modificación de servicio



```
// Create census -----  
// Recibe parametros de votacion y crea un censo por votación  
  
@RequestMapping(value = "/create", method = RequestMethod.GET, produces = "application/json")  
public @ResponseBody ModelAndView create() throws ParseException, FileNotFoundException, IOException {  
    ModelAndView result = null;  
    String fichero = "Z:\\EGC Workspace\\Censo\\votacion.txt";  
    Census c = censusService.create(fichero);  
    try {  
        censusService.save(c);  
        result = new ModelAndView("welcome/index");  
    } catch (Exception oops) {  
        oops.getCause();  
    }  
    return result;  
}
```

Ilustración 25 Modificación controlador

Una vez tengamos todo esto modificado, modificamos la vista principal para poder hacer uso del servicio:

```
<li><a href="census/create.do"><spring:message code="master.page.census.createCensusFromFile" /></a></li>
```

Ilustración 26 Adición de línea para crear un censo

Ahora solo queda ejecutar el servidor y probar que el servicio, leyendo datos de un fichero funciona:

```

CensusService.java  CensusController.java  header.jsp  votacion.txt
14
user1
1473033600000
1504569600000
Votacion1
abierta

```

Ilustración 27 Datos del fichero a usar

Seleccionamos el método crear censo desde archivo:

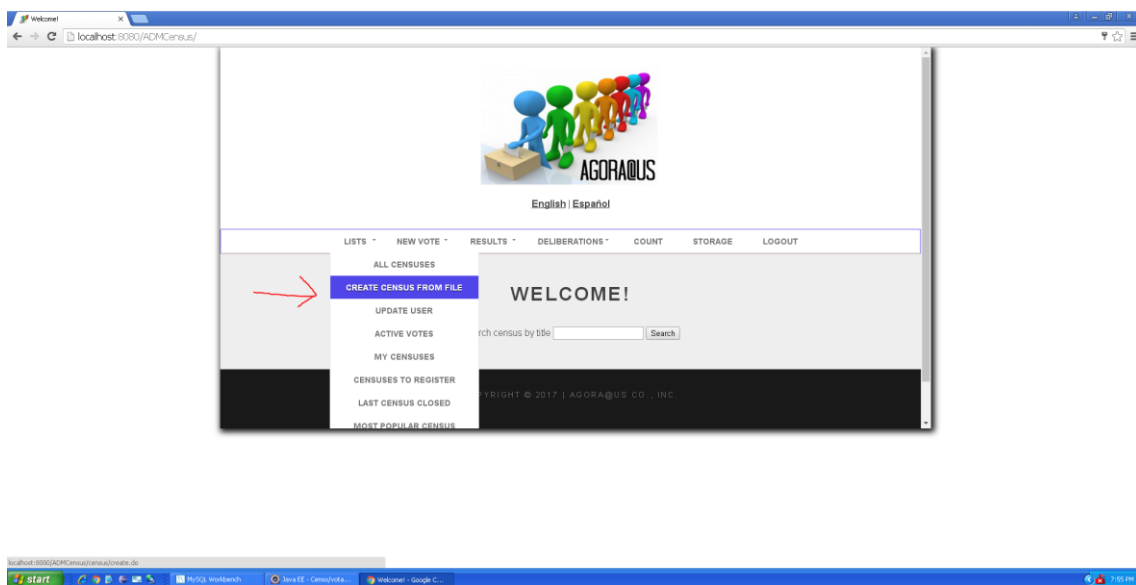


Ilustración 28 Selección del método

Vamos al listado de censos y vemos que tenemos nuestro nuevo censo creado con datos recibidos de un fichero:

LIST ALL CENSUS				
14 items found, displaying 11 to 14 [First/Prev] 1, 2, 3 [Next/Last]				
Name	Title	Type	Start date	End date
admin1	Censo cerrado	cerrado	2016-11-20 00:00:00.0	2016-11-25 00:00:00.0
admin1	Censo abierto	abierto	2016-11-01 00:00:00.0	2016-11-15 00:00:00.0
admin1	Censo abierto	abierto	2016-10-01 00:00:00.0	2016-10-28 00:00:00.0
user1	Votacion1	abierto	2016-09-05 02:00:00.0	2017-09-05 02:00:00.0

Back

Ilustración 29 Demostración del método

Indicar que todo esto es una simulación de integración para poder demostrar cómo funcionaría nuestro proyecto con la integración adecuada si los servicios de otros proyectos funcionaran correctamente.

El .war, el despliegue y la construcción se genera de forma manual

Para realizar el despliegue de nuestro subsistema primero hemos buscado varias páginas que ofrecen servicios de computación en la Nube que soportan distintos lenguajes de programación.

El primero que hemos probado ha sido Heroku. Permite realizar el despliegue en pocos pasos. Primero debemos acceder a <https://www.heroku.com/> donde nos debemos registrar y entonces podemos encontrar una guía de como desplegar nuestra aplicación y también podemos descargar el instalador del cliente. Una vez instalado el cliente nos vamos a acceder introduciendo el comando heroku login en la consola.

```
C:\>heroku login
Enter your Heroku credentials.
Email: ruben_barr@hotmail.com
Password (typing will be hidden):
Logged in as ruben_barr@hotmail.com
```

Ilustración 30 Heroku login

A continuación, debemos crear una aplicación, para ello vamos a ir a la ruta donde se encuentra nuestro proyecto y vamos a introducir el comando heroku create “Nombre app”.

```
C:\>heroku create egccenso
Creating egccenso... done
https://egccenso.herokuapp.com/ | https://git.heroku.com/egccenso.git
```

Ilustración 31 Heroku create

Para poder subir nuestro war debemos instalar un plugin de heroku, para ello vamos a introducir el comando heroku plugins:install heroku-cli-deploy.

```
C:\>heroku plugins:install heroku-cli-deploy
Installing plugin heroku-cli-deploy... done
```

Ilustración 32 Heroku instalación plugin

Una vez instalado el plugin vamos a introducir el comando heroku war:deploy “ruta del war” --app “nombre app”.

```
C:\>heroku war:deploy C:\Censo.war --app egccenso
Uploading Censo.war
-----> Packaging application...
- app: egccenso
- including: Users\Ruru\AppData\Local\Temp\heroku-deploy313658417536402135\heroku/app/webapp-runner.jar
- including: Censo.war
-----> Creating build...
- file: Users\Ruru\AppData\Local\Temp\heroku-deploy313658417536402135\heroku/slug.tgz
- size: 40MB
-----> Uploading build...
- success
-----> Deploying...
remote:
remote: -----> heroku-deploy app detected
remote: -----> Installing OpenJDK 1.8... done
remote: -----> Discovering process types
remote: Procfile declares types -> web
remote:
remote: -----> Compressing...
remote: Done: 88.8M
remote: -----> Launching...
remote: Released v3
remote: https://egccenso.herokuapp.com/ deployed to Heroku
remote:
remote: -----> Done
```

Ilustración 33Heroku deploy

A continuación, podemos acceder a heroku y veremos el panel de control de nuestra aplicación.

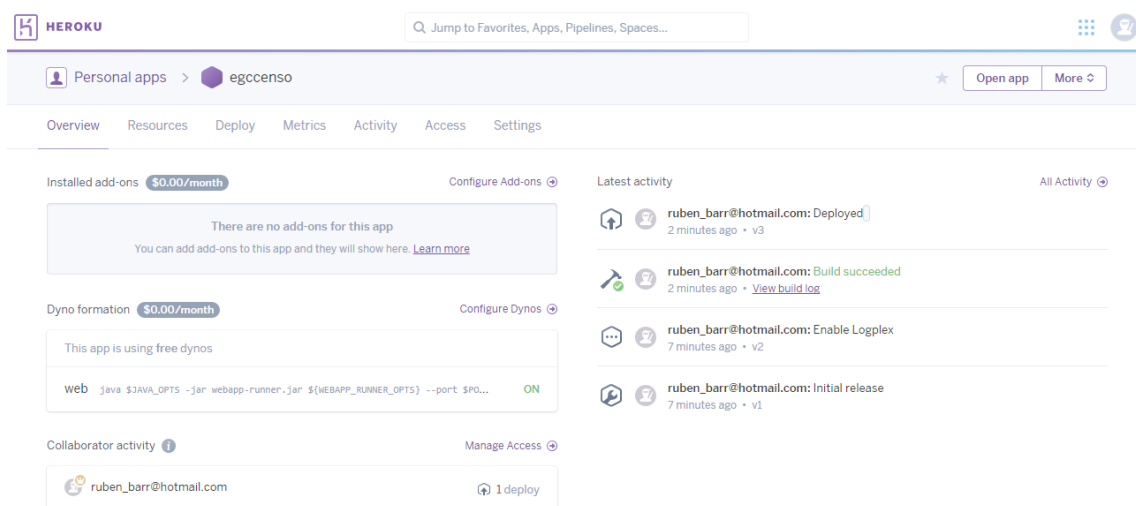
The screenshot shows the Heroku dashboard for an application named 'egccenso'. The top navigation bar includes the Heroku logo, a search bar, and a user profile icon. Below the navigation bar, there's a breadcrumb trail: 'Personal apps > egccenso'. The main content area is divided into several sections. On the left, there's a 'Installed add-ons' section showing '\$0.00/month' and a message that there are no add-ons for this app. Below that is the 'Dynos' section, also showing '\$0.00/month', with a table indicating the app is using free dynos. The 'web' dyno is listed with its command and is currently 'ON'. On the right, there's a 'Latest activity' section showing a list of events: 'Deployed' (2 minutes ago, v3), 'Build succeeded' (2 minutes ago, with a link to view the build log), 'Enable Logplex' (7 minutes ago, v2), and 'Initial release' (7 minutes ago, v1). At the bottom, there's a 'Collaborator activity' section showing a user 'ruben_barr@hotmail.com' with 1 deployment.

Ilustración 34Heroku panel de control

El problema que nos ha presentado heroku es que no soporta mysql.

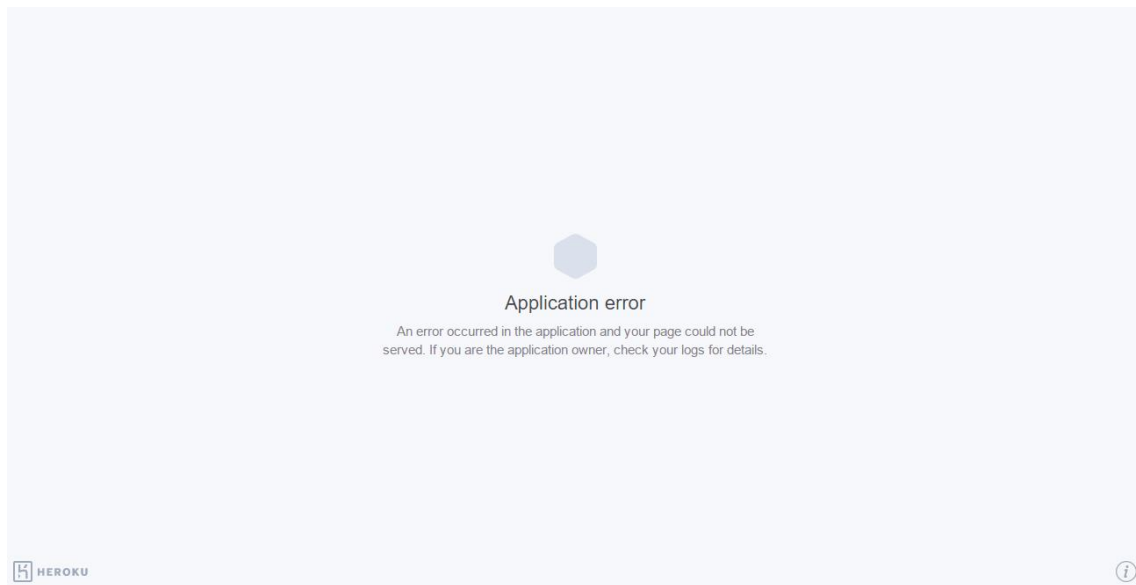



Ilustración 35 Heroku application error

Heroku ofrece varios plugins para migrar de mysql a postgresql pero entonces pide que se introduzca una tarjeta de crédito para poder usarlo de forma gratuita, ningún miembro del grupo ha querido poner su tarjeta de crédito por lo que hemos optado por otras opciones.

La segunda aplicación en la que hemos intentado desplegar ha sido en Google App Engine, el problema es que para su uso también pide que se introduzca una tarjeta de

crédito para así asegurarse que no somos ningún bot. En el caso de no introducirla solo nos permite hacer un tutorial de como se despliega una aplicación de ejemplo.



Forma de pago ⓘ

➡ Añadir tarjeta (crédito/débito) ▼

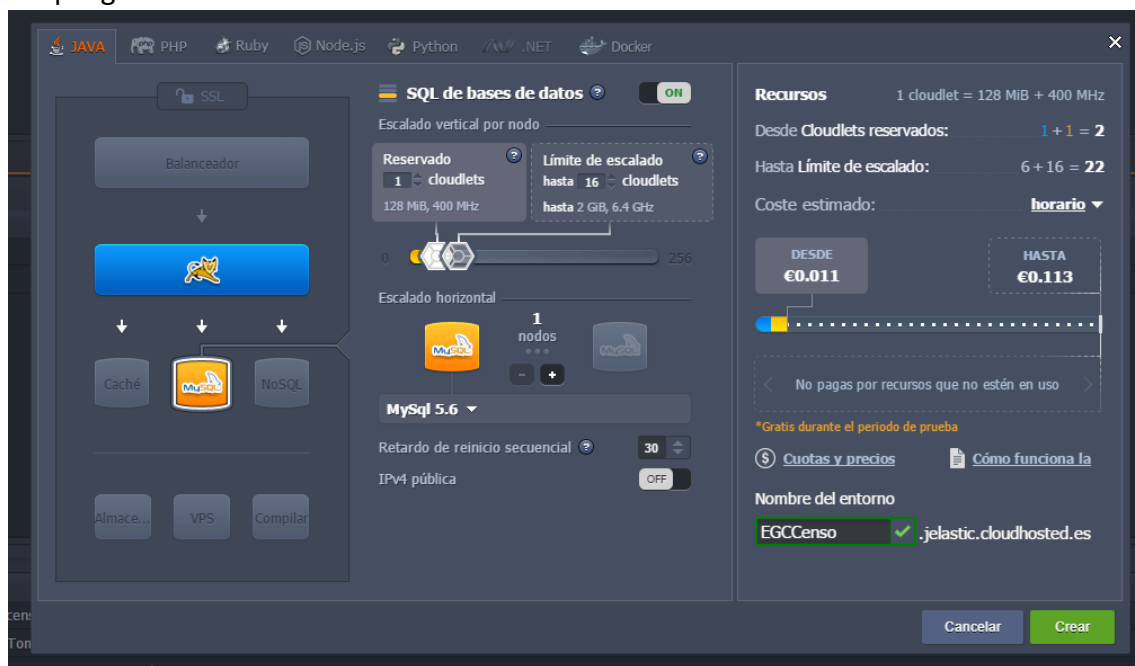
Número de tarjeta	MM	AA	CVC
#	/		
Es obligatorio indicar un número de tarjeta.			
Es obligatorio indicar el mes.			
Es obligatorio indicar el año.			
Es obligatorio indicar el CVC.			

Titular de la tarjeta
Rubén Rbm

☒ La dirección de la tarjeta de crédito o de débito es la misma que figura arriba.

Ilustración 36App Engine

La tercera opción ha sido Jelastical, esta página nos ofrece un periodo de prueba. Esta página nos ofrece varias opciones a la hora de crear el entorno donde se va a realizar el despliegue.



Selec... PHP Ruby Node.js Python .NET Docker

SSL

Balanceador

↓

Caché MySQL NoSQL

Almac... VPS Compilar

SQL de bases de datos ON

Escala... vertical por nodo

Reservado 1 cloudlets 128 MiB, 400 MHz

Límite de escalado hasta 16 cloudlets hasta 2 GiB, 6.4 GHz

Escala... horizontal

MySQL 5.6 1 nodos

Retardo de reinicio secuencial 30

IPv4 pública OFF

Recursos 1 cloudlet = 128 MiB + 400 MHz

Desde Cloudlets reservados: 1 + 1 = 2

Hasta Límite de escalado: 6 + 16 = 22

Coste estimado: horario

DESDE €0.011 HASTA €0.113

No pagas por recursos que no estén en uso

*Gratis durante el periodo de prueba

Cuotas y precios Cómo funciona la

Nombre del entorno EGCCenso .jelastic.cloudhosted.es

Cancelar Crear

Ilustración 37Jelastical creación del entorno

Una vez hemos configurado el entorno debemos esperar un tiempo ya que la creación tarda varios minutos, entonces podremos ver en el panel de control nuestro entorno.

The screenshot shows the Jelastic interface with a table of environments. The environment 'EGCCenso' is in 'En ejecución' (Running) state. Below it, the components 'Tomcat 7.0.73' and 'MySQL 5.6.34' are listed with their respective node IDs and usage percentages.

Nombre	Estado	Desplegado	Uso
EGCCenso egccenso.jelastic.dcloudhosted.es	En ejecución	No desplegado	1% 4/22
Tomcat 7.0.73 ID de nodo: 13717			2% 3/6
MySQL 5.6.34 ID de nodo: 13718			1% 1/16

Ilustración 38Jelastic entorno creado

Podemos gestionar la base de datos mediante phpMyAdmin y así crear nuestra base de datos mediante un script que hemos generado.

Importando en la base de datos "ADMCensus"

Archivo a importar:

El archivo puede ser comprimido (gzip, bzip2, zip) o descomprimido.
Un archivo comprimido tiene que terminar en `[formato].[compresión]`. Por ejemplo: `.sql.zip`

- ☒ Buscar en su ordenador: Lanzador.sql (Máximo: 1,024MB)
- También puede arrastrar un archivo en cualquier página.
- ☐ Seleccionar directorio en el servidor web para subir los archivos `/var/lib/jelastic/phpMyAdmin/upload/`: ¡No hay archivos para subir!

Conjunto de caracteres del archivo:

Importación parcial:

- ☒ Permitir la interrupción de una importación en caso que el script detecte que se ha acercado al límite de tiempo PHP. (Esto podría ser un buen n
- Omitir esta cantidad de consultas (en SQL) o líneas (en otros formatos) desde la primera:

Formato:

Opciones específicas al formato:

- Modalidad SQL compatible:
- ☒ No utilizar AUTO_INCREMENT con el valor 0
- ☐ Leer como conjunto de Bytes

Ilustración 39Jelastic creación de la base de datos

Para desplegar el war en el entorno tenemos que introducir la ruta donde lo tenemos generado y el nombre que queremos que tenga.

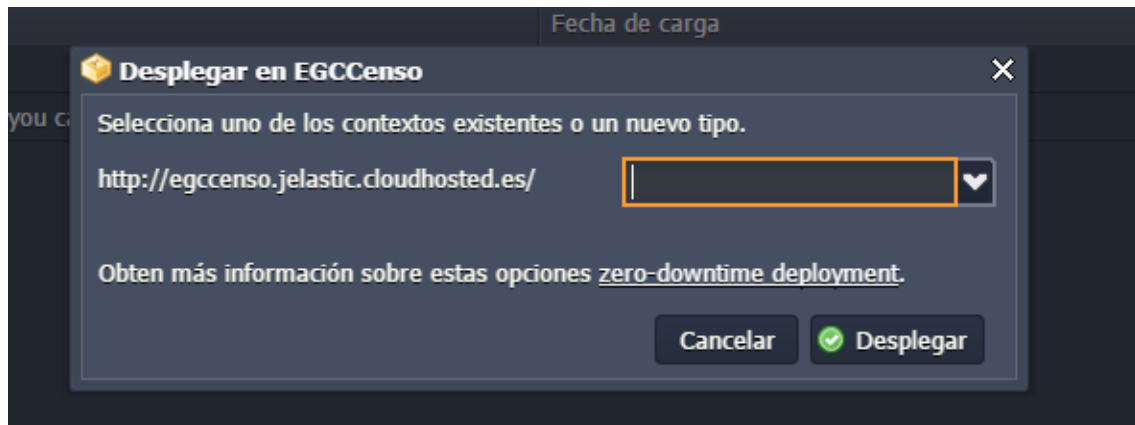


Ilustración 40Jelastical despliegue

A la hora de realizar la conexión entre la base de datos y el servidor de Tomcat nos ha dado problemas, por lo que finalmente hemos optado por configurar una maquina virtual como servidor e instalar todas las herramientas necesarias para poder desplegar de forma automática nuestro subsistema.

Hemos preparado una maquina virtual donde hemos instalado MySql Server, Apache Tomcat, java, maven, git y Jenkins.

```
C:\Program Files\MySQL\MySQL Server 5.5\bin>mysql -uroot -proot
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6
Server version: 5.5.54-log MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Ilustración 41Versión de MYSQL Server

```
C:\Windows\system32>mvn -v
Apache Maven 3.3.9 (bb52d8502b132ec0a5a3f4c09453c07478323dc5; 2015-11-10T17:41:47+01:00)
Maven home: C:\Maven\bin\..
Java version: 1.8.0_121, vendor: Oracle Corporation
Java home: C:\Program Files\Java\jdk1.8.0_121\jre
Default locale: es_ES, platform encoding: Cp1252
OS name: "windows 7", version: "6.1", arch: "amd64", family: "dos"
```

Ilustración 42Versión de maven

```
C:\Windows\System32>git --version
git version 2.11.0.windows.3

C:\Windows\System32>
```

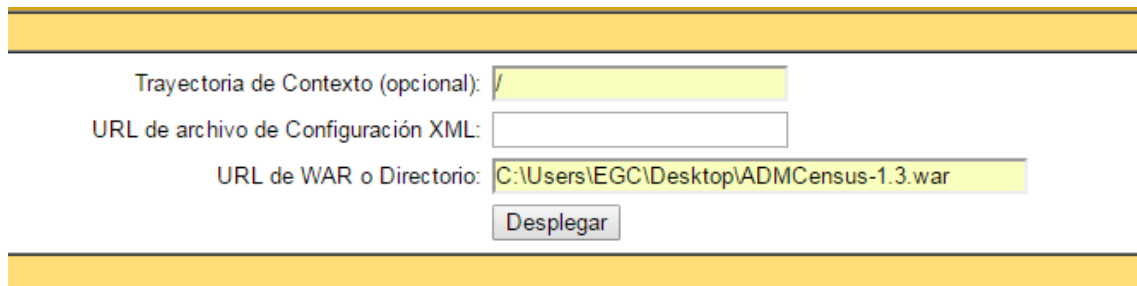
Ilustración 43 Versión de git

Para crear nuestra base de datos vamos a introducir el comando `mysql -uroot -proot < create-ADM Census.sql`, este script contiene todo lo necesario para generar las tablas y usuario de la base de datos con datos de ejemplo para comprobar su funcionalidad.

```
C:\Program Files\MySQL\MySQL Server 5.5\bin>mysql -uroot -proot < create-ADM Census.sql
```

Ilustración 44 Creación base de datos

Para desplegar de forma manual nuestro war debemos acceder a `localhost/manager` introducir el usuario y la pass que son root. A continuación, vamos a introducir la ruta del war como podemos ver en la siguiente imagen.



Trayectoria de Contexto (opcional):

URL de archivo de Configuración XML:

URL de WAR o Directorio:

Ilustración 45 Despliegue war

Para poder acceder desde nuestro pc cliente al servidor que hemos configurado debemos modificar como administrador el archivo `hosts` de nuestra maquina. Para ello vamos a la ruta `C:\Windows\System32\drivers\etc\`

```
C:\Windows\System32\drivers\etc>notepad hosts
```

Ilustración 46 Archivo host

En este archivo debemos introducir la siguiente línea:

```
# localhost name resolution is handled within DNS itself.
#       127.0.0.1       localhost
#       ::1             localhost
192.168.56.103 www.admcensus.com
```

Ilustración 47 Configuración archivo hosts

Una vez hechos estos cambios tenemos que vaciar los DNS para ello vamos a introducir el comando `ipconfig /flushdns`.

```
C:\Windows\System32\drivers\etc>ipconfig /flushdns

Configuración IP de Windows

Se vació correctamente la caché de resolución de DNS.
```

Ilustración 48Vaciado de DNS

Al acceder a www.admcensus.com podremos ver que se carga nuestro subsistema.



Ilustración 49ADMCensus

Para automatizar el despliegue hemos configurado en el servidor Jenkins con los siguientes plugins:

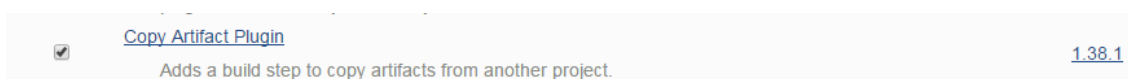


Ilustración 50Copy Artifact plugin

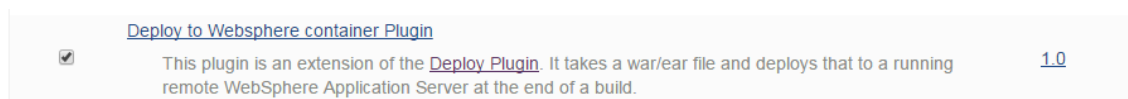


Ilustración 51Deploy to Websphere container plugin

Hemos creado dos tareas. La tarea CensoEGC se va a encargar de revisar el repositorio de forma automática cada hora y en el caso de que exista cambios, realizar los test y generar el war. En caso de que falle mandará un correo a todos los miembros del grupo.

Una vez finalice esta tarea, si no ha tenido ningún problema, se a llama de forma automática a la tarea censusWar, que se va a encargar de copiar el war generado y desplegarlo en tomcat.

Todo		+	
S	W	Nombre ↓	Último Éxito
		CensoEGC	5 Min 6 Seg - #31
		censusWar	3 Min 38 Seg - #44

Icono: [S](#) [M](#) [L](#)

Ilustración 52Jobs

Mejora de la documentación

La documentación se ha vuelto a hacer prácticamente desde cero ya que faltaban demasiadas cosas.

La nueva documentación esta accesible desde la wiki del grupo en la siguiente url

<https://1984.lsi.us.es/wiki-egc/index.php/CreacionAdministracionCensos>