



UNIVERSIDAD TÉCNICA DE AMBATO

FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES E INFORMÁTICOS

Tema:

“PLATAFORMA COMO SERVICIO (PAAS) PARA LA CREACIÓN, DESARROLLO Y DESPLIEGUE DE APLICACIONES WEB EN LA FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL”.

Trabajo de Graduación. Modalidad: Proyecto de Investigación, presentado previo la obtención del título de Ingeniero en Ingeniero en Sistemas, Computacionales e Informáticos.

SUBLINEA DE INVESTIGACION: Sistemas operativos

AUTOR: José Bernabe Vera Marin

TUTOR: Ing. Franklin Oswaldo Mayorga Mayorga, Mg.

Ambato - Ecuador
(Marzo-2018)

APROBACIÓN DEL TUTOR

En mi calidad de Tutor del Trabajo de Investigación sobre el tema: **“PLATAFORMA COMO SERVICIO (PAAS) PARA LA CREACIÓN, DESARROLLO Y DESPLIEGUE DE APLICACIONES WEB EN LA FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL”** , del señor Vera Marin José Bernabe N.N1803632767, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, considero que el informe investigativo reúne los requisitos suficientes para que continúe con los trámites y consiguiente aprobación de conformidad con el numeral 7.2 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.

Ambato Marzo, 2018

EL TUTOR

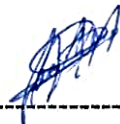


Ing. Franklin Oswaldo Mayorga Mayorga, Mg.

AUTORÍA

El presente Proyecto de Investigación titulado: **“PLATAFORMA COMO SERVICIO (PAAS) PARA LA CREACIÓN, DESARROLLO Y DESPLIEGUE DE APLICACIONES WEB EN LA FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL”**, es absolutamente original, auténtico y personal, en tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato Marzo, 2018



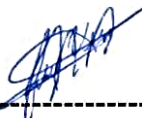
Vera Marin José Bernabe
CC: 1803632767

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación, con fines de difusión pública, además autorizo su reproducción dentro de las regulaciones de la Universidad.

Ambato marzo, 2018



Vera Marin José Bernabe
CC: 1803632767

APROBACIÓN DE LA COMISIÓN CALIFICADORA

La Comisión Calificadora del presente trabajo conformada por los señores docentes Ing. Guevara David, Ing. Nogales Rubén, revisó y aprobó el Informe Final del Proyecto de Investigación titulado **“PLATAFORMA COMO SERVICIO (PAAS) PARA LA CREACIÓN, DESARROLLO Y DESPLIEGUE DE APLICACIONES WEB EN LA FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL”**, presentado por el señor Vera Marin José Bernabe de acuerdo al numeral 9.1 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.



Ing. Mg. Elsa Pilar Urrutia Urrutia
PRESIDENTA DEL TRIBUNAL



Ing. Guevara Aulestia David Omar

DOCENTE CALIFICADOR



Ing. Nogales Portero Ruben Eduardo

DOCENTE CALIFICADOR

INDICE

APROBACION DEL TUTOR.....	II
AUTORIA.....	III
DERECHOS DE AUTOR.....	IV
APROBACION COMISIÓN CALIFICADORA.....	V
INTRODUCCIÓN.....	XI
ABSTRAC.....	XII
CAPITULO I.....	1
1.1 Tema.....	1
1.2 Planteamiento del problema.....	1
1.3 Delimitación.....	2
1.4 Justificación.....	2
1.5 Objetivos.....	3
1.5.1 Objetivo general.....	3
1.5.2 Objetivos específicos.....	3
CAPITULO II.....	4
2.1 Antecedentes investigativos.....	4
2.2 Fundamentación teórica.....	5
2.2.1 Cloud Computing o “la nube”.....	5
2.2.2 Características generales de la computación en nubes.....	6
2.2.3 Tipos De Nubes.....	7
2.2.4. Computación en la nube.....	7
CAPITULO III.....	11
3.1. Modalidad de la investigación.....	11
3.2 Población y muestra.....	11
3.3 Recolección de información.....	11
3.4 Procesamiento y análisis de datos.....	12
3.5 Desarrollo del proyecto.....	12
CAPITULO IV.....	13
DESARROLLO DE LA PROPUESTA.....	13
4.1 Análisis de las plataformas como servicio (PaaS) que se ajusten a la infraestructura disponible.....	13

4.1.1 Análisis de infraestructura disponible	13
Windows azure	14
Openshift origin	15
4.1.2 Análisis de lenguaje de programación, frameworks, IDEs y SGBDs utilizados.	15
4.1.4 Contratación de requerimientos PaaS con infraestructura disponible.	19
4.1.5 Selección de PaaS adecuada a la infraestructura disponible	20
4.2 Procesos de instalación y configuración de PaaS.	21
4.2.1 Instalación de PaaS elegida.	21
4.2.2 Instalación de frameworks, lenguajes de programación y SGBDs	41
4.3 Implementar PaaS y pruebas de funcionamiento.	54
4.3.1 Pruebas de creación y despliegue de proyectos de desarrollo.....	54
 CAPITULO V	 73
5.1 Conclusiones.....	80
5.2 Recomendaciones.....	80
 Bibliografía	 75

Índice de tablas

Cuadro N.-1 Infraestructura FISEI.....	21
Cuadro N.-2 lenguajes de programación, frameworks , IDEs, SGBDs, FISEI.....	16
Cuadro N.- 3 Análisis Lenguaje de programación, Frameworks, IDEs, SGBDs.....	17
Cuadro N.-4 Análisis Lenguaje de programación, Frameworks, IDEs, SGBDs.....	18
Cuadro N.-5 requisitos Windows Azure	19
Cuadro N.-6 Requisitos Openshift Origin.....	19
Cuadro N.-7 IaaS- PaaS Openshift Origin	20
Cuadro N.-8 IaaS- PaaS Windows Azure	20

Índice de imágenes

Imagen N.1 versiones de openshift origin.....	21
Imagen N.-2 Configuración de repositorio.....	29
Imagen N.-3 Verificación contenido DNS.....	31
Imagen N.- 4 Configuración del named.conf.....	33
Imagen N.- 5 Configuración de la interfaz de ActiveMQ.....	38
Imagen N.- 6 Interfaz gráfica de ActiveMQ.....	39
Imagen N.- 7 Verificación de funcionamiento con Json.....	
Imagen N.- 8 Verificación con la consola web.....	48
Imagen N.- 9 Verificación con la consola web ventana de openshift.....	48
Imagen N.- 11 Lista de cartuchos disponibles en openshift origin.....	50
Imagen N.- 12 lista de cartuchos escogida para la instalación.....	53
ImagenN.- 13 Configuración de límite de almacenamiento en disco.....	54
Imagen.- 14 Configuración de límite de almacenamiento memoria RAM.....	58
Imagen N.- 15 Habilitación de las cuotas de disco.....	58
Imagen N.- 16 Configuración del dominio para el node1.....	
Imagen N.- 17 Ingreso al browser para pruebas.....	60
Imagen N.-18 Añadir excepción.....	61
Imagen N.-19 Inicio de sesión.....	62
Imagen N.-20 verificación de usuario.....	62
Imagen N.-21 Ventana principal de Openshift Origin.....	63
Imagen N.-22 lista de herramientas de desarrollo.....	63
Imagen N.-23 proyecto de wordPress.....	64
Imagen N.-24 Creación de plantilla WordPress.....	64
Imagen N.-25 Creación de proyecto WordPress.....	65
Imagen N.-26 proyecto WordPress.....	66
Imagen N.-27 proyecto WordPress paso 2.....	66
Imagen N.-28 selección de idioma para instalar wordpress.....	67
Imagen N.-29 creacion de usuario y contraseña WordPress.....	67
Imagen N.-30 Instalación correcta de WordPress.....	68
Imagen N.-31 Ingreso a WordPress.....	69
Imagen N.-32 ventana principal de WordPress.....	69
Imagen N.-33 interfaz del proyecto wordPress.....	70
Imagen N.-34 Visualización del proyecto.....	70
Imagen N.-35 Instalación de Git.....	71
Imagen N.-36 Creación de la llave de acceso.....	72

Imagen N.-37 Creación de la llave de acceso paso 2	72
Imagen N.-38 Creación de la llave de acceso paso 3	73
Imagen N.-39 vincular llave entre git y openshift.....	73
Imagen N.-40 visualización de llaves en openshift.....	74
Imagen N.-41 Creación de aplicación PHP	74
Imagen N.-42 Creación de aplicación PHP paso 2	75
Imagen N.-43 Creación de aplicación PHP paso 3	75
Imagen N.-44 Clonación de repositorios en git	76
Imagen N.-45 Clonación de repositorios en git paso 2	76
Imagen N.-46 Prueba en el browser de la clonación.....	77
Imagen N.-47 edición del fichero index.php.....	77
Imagen N.-48 ingreso de comentario para le versión	78
Imagen N.-49 pruebas comando git push.....	78
Imagen N.-50 vista de aplicación modificada.....	79

RESUMEN

En el presente proyecto se revisan los conceptos asociados con el Cloud Computing y las herramientas para su utilización. En la actualidad Cloud Computing es una de las tendencias predominantes en la cual los Departamentos de TIC se apoyan para la gestión de recursos que permiten a las empresas migrar servicios al Cloud Computing.

El enfoque principal de esta propuesta de investigación se centra en el análisis de las plataformas como Servicio (PaaS), su implementación y su despliegue en entornos de producción. De esta manera proporcionar información relevante que permita a la Facultad de Ingeniería en Sistemas Electrónica e Industrial incursionar en un modelo de negocios basados en la tecnología Cloud Computing, mediante un diseño que se convertirá en una guía hacia la innovación.

Con la instalación de OpenShift Origin en el proyecto, se proporcionara los beneficios una nube pública, con un entorno mucho más controlado, ya que se configurara a las necesidades de los estudiantes.

ABSTRACT

In this project we review the concepts associated with Cloud Computing and the tools for its use. Currently, Cloud Computing is one of the predominant trends in which the TIC Departments support the management of resources that allow companies to migrate services to the cloud.

The main focus of this research proposal is focused on the analysis of platforms as a service (PaaS), its implementation and its deployment in production environments. Providing relevant information that allows the faculty of engineering in electronic and industrial systems to venture into a business model based on Cloud Computing technology through a design that will become a guide to innovation.

With the OpenShift Origin installation in the project, it provides the same benefits as a public cloud, but in a much more controlled environment, since it will be configured to the needs of the students.

CAPITULO I

EL PROBLEMA

1.1 Tema

“Plataforma como servicio (PaaS) para la creación, desarrollo y despliegue de aplicaciones web en la Facultad de Ingeniería en Sistemas, Electrónica e Industrial”.

1.2 Planteamiento del problema

El cloud computing o “computo en la nube” es una evolución natural de la adopción generalizada de la virtualización, se introdujo por primera vez en el 2006 por Amazon con el término **Elastic Compute Cloud (EC2)** como un servicio comercial que permite a pequeñas y medianas empresas alquilar los servidores donde poder hacer funcionar sus propias aplicaciones [1].

Actualmente, es posible acceder a Internet a través de una serie de dispositivos, básicamente desde cualquier lugar del mundo, de forma segura y rápida. Así mismo la tecnología evoluciona constantemente y con ella los avances en el mundo de la comunicación. Cloud Computing, facilita las tareas de almacenar, gestionar, compartir y poner a disposición datos, software, aplicaciones y servicios a través de Internet. [2]

Este concepto es cada vez más conocido y utilizado, ya que se trata de un modelo de computación que permite a las organizaciones ahorrar inversiones en infraestructura, sistemas e incluso personal.

En Ecuador el Cloud Computing empieza a ser conocido por el sector empresarial y gubernamental, es así que muchas de las empresas públicas y privadas están migrando a esta nueva tecnología y como en los demás países se busca la innovación y el avance

tecnológico.

En este contexto el personal administrativo de los laboratorios de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial pierden demasiado tiempo en la puesta a punto de sus plataformas de desarrollo, la configuración del software, buscando que se cumplan los requerimientos mínimos que se necesitan para la instalación del software, sobretodo la necesidad de comprar un dispositivo específico, un sistema operativo especial o un programa para ese dispositivo en particular.

Actualmente la facultad no cuenta con una plataforma hardware ofrecidos como servicios a través de una conexión a internet o a una red que les permita a los estudiantes desarrollar sus aplicaciones sin la necesidad de configurarla.

1.3 Delimitación

Área Académica: Hardware y Redes

Línea de Investigación: Sistemas Administradores de Recursos

Sub Línea de Investigación: Sistemas operativos

Delimitación Espacial: La presente investigación se realizará en la Facultad de Ingeniería en Sistemas, Electrónica e Industrial

Delimitación Temporal: El desarrollo de esta propuesta tomará 6 meses a partir de la fecha de aprobación por parte del Consejo Directivo

1.4 Justificación

Es claro que estamos migrando de la era de la PC (toda la información almacenada y disponible en un solo lugar) a la era del Cloud Computing, información que se mueve y a la que se accede desde cualquier lugar y dispositivo.

La Facultad de Ingeniería en Sistemas Electrónica e Industrial, cuenta con una infraestructura adecuada para instalar Cloud Computing en los servidores que posee la facultad, y al tener PaaS instalado y configurado se evitara la contratación de personal capacitado.

Uno de los aspectos importantes que se debe destacar en el desarrollo del presente

proyecto es el beneficio que va a brindar a la institución y a los estudiantes al implementar PaaS, esta brindara las facilidades para apoyar el ciclo de vida completo del desarrollo de las aplicaciones, incluyendo el diseño, depuración, pruebas y despliegue de aplicaciones web.

El desarrollo del presente trabajo aportara con el conocimiento de las nuevas tecnologías de cloud computing en PaaS. Pone a disposición del usuario las herramientas necesarias para desplazamiento de servicios personalizado, permitiéndole el ahorro de recursos.

El desarrollo de este trabajo es factible porque se cuenta con los suficientes conocimientos en cuanto al problema y su solución, una ventaja es que se desarrollará el proyecto bajo herramientas informáticas libres, por lo que se presenta como una alternativa de fácil acceso.

En cuanto a los recursos bibliográficos existen gran cantidad de documentación que brinda ayuda tanto en instalación configuración y administración además de existir foros especializados y grupos activos que brindan soporte.

1.5 Objetivos

1.5.1 Objetivo general

Implementar una Plataforma como servicio (PaaS) para la creación, desarrollo y despliegue de aplicaciones web en la Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

1.5.2 Objetivos específicos

- Analizar las plataformas como servicio que se ajusten a la infraestructura disponibles.
- Realizar los procesos de instalación con sus correspondientes frameworks y soportes de lenguaje.
- Implementar (PaaS) y pruebas de funcionamiento.

CAPITULO II

MARCO TEORICO

2.1 Antecedentes investigativos

Se debe manifestar que en la Facultad de Ingeniería en Sistemas Electrónica e Industrial, no se han realizado implementación de openshift u otra plataforma como servicio no obstante si se aprecian documentación sobre trabajos realizados acerca de su utilización. Entre estos trabajos en [3], podemos citar el alojamiento de una plataforma Moodle, La investigación tiene como propósito diseñar un Aula Virtual alojada en una plataforma como servicio (PaaS) en este caso sobre openshift online, implementada en el Instituto Superior Tecnológico Baños, por lo tanto esta información sirve de base para la realización del proyecto.

De igual manera en [4], se aprecia un trabajo sobre el alojamiento de un servicio de tele operación bilateral aplicado para brazos robóticos, en la cual se utiliza se configura una plataforma como servicio PaaS openshift Origin para la realización de pruebas en tiempo real.

A nivel nacional, se revisó repositorios de las instituciones de nivel superior dando como resultado trabajos relacionados al análisis y comparativa de PaaS. Pero ningún trabajo relacionado a la implementación de ninguna plataforma como servicio, sin embargo en [5], se puede apreciar la propuesta de incorporación de la tecnología de desarrollo en la nube PaaS en la cual se ha realizado un estudio experimental entre 5 plataformas dedicadas a este tipo de servicio, llegando a la selección de la mejor herramienta en base a las necesidades presentadas en la etapa de desarrollo de software por la empresa en IntSoft Solutions

En este mismo campo en [6], se propone la realización de un prototipo de plataforma como servicio PaaS basado en software libre, este se enfoca en analizar las ventajas y desventajas que tienen los PaaS para la instalación y configuración y al final elegir el PaaS adecuado para realizar el prototipo, por lo tanto este sería una gran ayuda para la realización del proyecto.

Una vez analizado a nivel nacional y local los trabajos realizados acerca de plataformas como servicio (PaaS), se aprecia un consumo de los servicios de PaaS online para que los proyectos alojados aprovechen todas las características de un PaaS, pero no se apreció ningún trabajo de implementación por el nivel de complejidad en sus pre-requisitos y configuraciones de los servicios, por ejemplo: Como primer instancia se instala el S.O. base para la configuración de PaaS, por lo cual se actualiza los repositorios de Openshift Origin, se instala el paquete git para la configuración y soporte de ssh, el paquete Ntp para la sincronización de la hora en el nodo y el bróker, la instalación de Bind y Bind-utils para la configuración del servidor de dominio, la base de datos mongodb para la instalación de Openshift y la instalación de ActiveMq, Mcollective para la comunicación entre nodos, luego de la instalación de los paquetes se procederá a la creación de usuario y las cuotas y la instalación de los cartuchos. Todo lo antes expuesto da lugar a la investigación propuesta, siendo esta carencia de desarrollo el pilar fundamental para el trabajo de propuesto, en el cual se detalla paso a paso la implementación de la plataforma como servicio openshift origin y ofrecer un cimiento sólido para futura investigaciones.

2.2 Fundamentación teórica

2.2.1 Cloud Computing

Es una plataforma altamente escalable que promete un acceso rápido al recurso hardware o software y donde el usuario no necesita ser experto para su manejo y acceso.

Las nubes suelen apoyarse en tecnologías como la virtualización, técnicas de programación como el multitenancy y/o habilidades para la escalabilidad, balanceo de carga y rendimiento óptimo, para conseguir ofrecer el recurso de una manera rápida y sencilla. Además, en el caso de las nubes públicas, estas técnicas generan economías

de escala derivadas del aprovechamiento eficiente de los recursos hardware y humanos que terminan repercutiendo en el precio que paga el cliente.

El Cloud Computing lo podemos dividir en tres niveles, en función de los servicios que actualmente están ofreciendo las empresas. Desde el más interno hasta el más externo nos encontramos: infraestructura como servicio, plataforma como servicio y software como servicio. [7]

2.2.2 Características generales de la computación en nubes

Para alcanzar los objetivos mencionados con anterioridad, una plataforma de computación en nubes debe tener ciertas características generales. Existen aún algunas discrepancias con respecto a todas las características que deben tener la plataforma y su relevancia. La siguiente es una lista corta de las características elementales básicas que debe tener una plataforma de computación en nubes:

- **Acceso a la red:** es crucial para que la plataforma nube pueda operar. Este acceso a la red debe darse sin intervención alguna por parte del usuario final.
- **Servicio medido:** esta funcionalidad hace más interesante desde el punto de vista económico, la plataforma de computación en nubes. El servicio prestado por una plataforma nube se da por demanda, lo cual posibilita el "pago por uso" o por prestación de servicios. Usted puede optar por pagar el servicio con una tarifa fija o basada en otros factores como el número de usuarios, tiempo de uso, etc.
- **Elasticidad:** se refiere a rápida escalabilidad. Los recursos (hardware y software) son proveídos sin la intervención o sin que el usuario final se percate de ello por su lado. Todo ello está basado en la demanda de consumo.
- **Conjunto de recursos:** en la plataforma de computación en nube, todos los recursos están agrupados. Ello significa que todos los recursos son prestados a múltiples usuarios utilizando métodos para usar recursos virtuales y físicos, dependiendo de la demanda. En este caso el concepto de virtualización tiene gran relevancia para poder alcanzar este objetivo. La habilidad de agrupar los recursos está disponible dado el diseño multi-tenencia de la plataforma de computación en nube. [8]

2.2.3 Tipos De Nubes

- **Nubes públicas:** es el modelo más reconocido por los usuarios de internet. Puede ser un servicio gratuito (google drive, dropbox) o pago (Amazon Window Azure), dependiendo del servicio que se use. La información almacenada por parte del usuario se encuentra en servidores externos a este.
- **Nubes privadas:** se pueden encontrar en empresas, como medida de protección de información y aplicaciones dentro de la organización o entidad. No ofrece sus servicios a terceros. Generalmente se enfoca en la obtención de hardware, pero también trabaja con aplicaciones. La empresa decide quien hace uso y tiene acceso a su servicio.
- **Nubes híbridas:** posee las características tanto de las nubes públicas como de las privadas. También es usada por empresas, haciendo uso de la estructura privada para la protección de sus datos, mientras que da acceso al público a los aspectos que se consideren necesarios [9].

2.2.4. Computación en la nube

Software como Servicio (SaaS).

Software como Servicio (SaaS) es la distribución de software por parte de un proveedor de servicios, pero no como pieza de software que el usuario puede descargar en su sistema para usarlo de manera local e indefinida (dependiendo, eso sí, del modelo de licencia del producto), sino como servicio que el usuario puede contratar o utilizar de manera gratuita a través internet.

Lo cierto es que, si bien las soluciones Cloud basadas en SaaS empezaron a llegar al público en forma de espacios de almacenamiento online, hoy en día van mucho más allá, pudiendo ofrecer las mismas funcionalidades que muchas de las aplicaciones que podemos tener en nuestro sistema, pero sin la necesidad de poseer el software y ejecutarlo de manera local.

Tenemos hoy en día multitud de muestras de soluciones SaaS, y muchos de ellos probablemente los utilizas día a día prácticamente de forma inconsciente. Google Drive, Gmail, Dropbox, Apple iCloud, Microsoft One Drive y Microsoft Office 365, o el software CRM Salesforce son todos ejemplos de soluciones de software como

servicio [10].

Infraestructura como Servicio (IaaS).

Infraestructura como Servicio (IaaS) es el último escalón en cuanto a modelos de servicio Cloud, y representa en cierto modo la disponibilidad de recursos de hardware (CPU, RAM, almacenamiento de disco, red, etc.) en forma de servicio online, de modo que usuarios, o en este caso más bien desarrolladores o empresas, puedan construir sus plataformas o soluciones de software online, en forma de soluciones PaaS o SaaS.

Así, en el caso de adoptar un modelo basado en IaaS, el proveedor se hace cargo de la gestión de los recursos físicos, y los usuarios son los responsables de la plataforma adoptada y de las aplicaciones diseñadas o ejecutadas.

Ejemplos de soluciones IaaS los encontramos en Amazon Web Services (AWS), Microsoft Azure y la plataforma vCloud de VMWare [8].

Plataforma como Servicio (PaaS)

Plataforma como Servicio (**PaaS**) en castellano, es en cierto modo el siguiente escalón respecto a las soluciones SaaS, y representa el punto intermedio entre SaaS y IaaS. En las soluciones PaaS, el proveedor de servicios ofrece la plataforma o el sistema operativo en forma de servicio y se encarga de gestionarla, con todo lo que implica en forma de actualizaciones de sistema operativo, upgrades, etc. [8].

Las soluciones PaaS son plataformas de software para las cuales la herramienta de desarrollo en si misma está alojada en la nube y se accede a través de un navegador web. Con PaaS, los desarrolladores pueden construir aplicaciones web sin tener que instalar ninguna herramienta adicional en sus computadoras, y luego despliegan estas aplicaciones sin necesidad de tener ningún conocimiento administrativo especializado [9].

Un claro ejemplo de PaaS lo tenemos en **Google App Engine**, la plataforma de Google para que usuarios y desarrolladores puedan subir, probar sus aplicaciones y ofrecerlas en forma de SaaS. Lo puedes encontrar en plataformas Windows, Linux y OS X, y

soporta aplicaciones en Python, Java, PHP y Go [8].

Las aplicaciones alojadas pueden consumir una cantidad limitada de recursos de manera gratuita, y a partir de ahí puedes optar por una serie de suplementos de pago en forma de cuotas, dependiendo de los recursos adicionales que necesites [8].

McKinsey & Company, en su informe "La guerra de las plataformas emergentes" del 2008, definió a PaaS como "IDEs basados en la nube que no sólo incorporan a lenguajes de programación tradicionales, sino también que incluyen herramientas para el desarrollo con mashups" [9].

Las plataformas PaaS también tienen diferencias funcionales respecto a las plataformas de desarrollo tradicionales. Estas diferencias incluyen:

- **Herramientas de desarrollo múltiples inquilinos:** las herramientas de desarrollo tradicional son para un único usuario - un estudio basado en la nube debe soportar múltiples usuarios, cada uno de los cuales tiene múltiples proyectos activos.
- **Arquitectura de despliegue de múltiples inquilinos:** la escalabilidad suele no ser una preocupación durante el esfuerzo inicial de desarrollo y se deja para que lo gestionen los administradores de sistemas cuando se despliega el proyecto. En PaaS, la escalabilidad de la aplicación y de sus capas de datos debe ser intrínseca la solución (por ejemplo, el balanceo de carga y la disponibilidad tienen que ser elementos básicos de la plataforma de desarrollo).
- **Gestión integrada:** las soluciones de desarrollo tradicionales no suelen preocuparse con el monitoreo del entorno de ejecución, pero en PaaS, se tiene que soportar la habilidad de monitoreo desde la misma plataforma de desarrollo.
- **Facturación integrada:** las ofertas PaaS requieren mecanismos de facturación basados en la utilización, que son únicos para el mundo PaaS [9].

Beneficios de PaaS

Los beneficios y características de PaaS dan lugar a que nuevo usuarios se interesen en desarrollar, mantener y desplegar aplicaciones web.

PaaS ofrece el potencial para que desarrolladores puedan construir aplicaciones web sin ser un experto. Esto le abre la puerta a toda una generación de desarrolladores MS Access, Lotus Notes y PowerBuilder para que pueda comenzar a construir aplicaciones web sin una curva de aprendizaje enorme [9].

Ventajas y Desventajas

Una ventaja de los PaaS es la posibilidad de disponer de una plataforma adaptada, gestionada y mantenida por un proveedor externo, permitiendo que los usuarios únicamente se preocupen del desarrollo de su aplicación.

Las desventajas en este sentido vuelven derivar de la **dependencia con proveedor de servicios**. En PaaS esto se traduce básicamente en las limitaciones de la propia plataforma elegida por el proveedor de servicios, que ya de por si nos puede condicionar o limitar en gran medida en los desarrollos que queramos llevar a cabo.
[8]

CAPITULO III

METODOLOGIA

3.1. Modalidad de la investigación

El presente trabajo tiene las siguientes modalidades:

3.1.1 Modalidad de Campo La investigación tendrá la modalidad de campo porque el investigador acudirá al lugar en donde se producen los hechos para obtener información relacionada con los objetivos del trabajo de grado. Las técnicas a ser utilizadas serán: entrevistas y la observación.

3.1.2 Modalidad Bibliográfica o Documentada La investigación bibliográfica tendrá como propósito fortalecer la investigación se recurrirá a obtener investigación teórica de diferentes autores obtenidas en fuentes secundarias (Libros, revistas especializadas, publicaciones, internet, otros) y de ser necesario fuentes de información primaria a través de documentos válidos y confiables.

3.2 Población y muestra

La presente investigación por su característica no requiere población ni muestra, por tal motivo no se aplica.

3.3 Recolección de información

Para la recolección de la información se aplicará entrevistas al personal a cargo del Departamento de Informática de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

Además de las entrevistas también se aplicará la técnica de la observación para de esta forma conocer el funcionamiento de como los estudiantes reciben clases de desarrollo de software.

Se realizará una investigación profunda a través de la recopilación bibliográfica como son: artículos científicos papers, libros y tesis correspondientes al tema, los mismos que nos proporcionarán la información necesaria para la estructuración del presente tema de investigación.

Además, se recolectará datos los mismos que serán proporcionados por el departamento de redes de la Facultad con el fin de obtener información útil para la investigación

3.4 Procesamiento y análisis de datos.

Una vez recolectada la información se procederá a su respectivo análisis los cuales serán de gran importancia para la formulación de la propuesta. Los datos serán analizados y procesados en relación al problema para poder establecer las respectivas conclusiones asegurando que los datos sean lo más reales posibles, para así poder cumplir con los objetivos planteados anteriormente.

3.5 Desarrollo del proyecto

Analizar las Plataformas como Servicio que se ajusten a la infraestructura disponible.

- Análisis de infraestructura disponible.
- Análisis de lenguajes de programación, frameworks, IDEs SGBDs utilizados.
- Análisis de requisitos de instalación y despliegue de PaaS.
- Contratación de requerimientos PaaS con infraestructura disponible
- Selección de PaaS adecuada a la infraestructura disponible.

Realizar los procesos de instalación con sus correspondientes frameworks y soporte de lenguaje.

- Instalación de PaaS elegida
- Instalación de soporte de lenguajes de programación elegidos.
- Instalación de frameworks de desarrollo elegidos

Realizar pruebas de funcionamiento.

- Pruebas de creación y despliegue de proyectos de desarrollo.

CAPITULO IV

DESARROLLO DE LA PROPUESTA

4.1 Análisis de las plataformas como servicio (PaaS) que se ajusten a la infraestructura disponible.

4.1.1 Análisis de infraestructura disponible

Para la realización del proyecto se deben cumplir con ciertos requerimientos tanto de software como de hardware, con la entrevista realizada en el anexo N.-1, a la Ing. Encargada del departamento de administración de redes de la Facultad de Ingeniería en Sistemas Electrónica e Industrial se pudo obtener la información que se detalla en el cuadro N.-1 la cual servirá de base para el análisis de la infraestructura con que se cuenta y se necesita para la implementación de PaaS.

Infraestructura FISEI				
Hardware			Software	
Servidor			Sistemas Operativos	
	Procesador	Intel ® xeon de 11 núcleos		Windows
	Memoria RAM	16		Centos
	HDD	256		Fedora
			LENGUAJES DE PROGRAMACIÓN	
Laboratorios				Visual Studio

	Procesador	I5, I7		JAVA
	Memoria RAM	8GB		PHP
	HDD	1TB	FRAMEWOKS	
				.NET
				EMF
				YII
			IDES	
				NetBeans
				Eclipse
			SGBDs	
				Oracle
				Postgre
				Mysql
				Sql server
Cuadro N.-1 Infraestructura FISEI				

De la información detallada en el cuadro N.-1 se pudo verificar que los Paas más opcionados para la realización de este proyecto serian Windows azure y openshift origin

Windows azure

Windows Azure es la plataforma en la nube de Microsoft, es una **nube pública de pago** que permite compilar, implementar y administrar aplicaciones en una red global.

La plataforma de Windows Azure está formada por 4 partes:

- **Windows Azure:** Un entorno Windows que ejecuta aplicaciones y almacena datos en computadores de Microsoft Data Centers.
- **SQL Azure:** Servicios de datos relacionales en el cloud basados en SQL Server.
- **Windows Azure AppFabric:** Servicios de infraestructura basados en el cloud para aplicaciones que se ejecutan en el cloud o localmente.
- **Windows Azure Marketplace:** Un servicio en línea para la venta de aplicaciones de datos basados en el cloud.

Openshift origin

OpenShift es la Plataforma como servicio (PaaS) de cloud computing de Red Hat. OpenShift puede utilizarse de varias maneras: mediante su cloud público que se denomina OpenShift Online, mediante un producto de pago OpenShift Enterprise o bien instalando directamente en un cloud privado el software OpenShift Origin.

	OpenShift Online	OpenShift Enterprise	OpenShift Origin
¿Qué es?	Servicio PaaS en la web	PaaS privado	Software PaaS de código abierto
Funciones	Desarrollo rápido de aplicaciones escalables y alojadas en un cloud público	Acelera los servicios TIC y agiliza el desarrollo de aplicaciones	Usar un PaaS de código abierto y libre o ayudar a desarrollar OpenShift
¿Precio?	Free Plan o Premium Plan	Suscripción anual	Gratis y libre
Soporte	Comunidad(Free Plan);Red Hat(Premium Plan)	Red Hat	Comunidad
¿Dónde puedes ejecutarlo?	En el cloud público	En tus propios servidores o en tu cloud privado	En tu ordenador, en tus servidores o en cloud privada o pública
¿Para quién esta pensado?	Startups, desarrolladores, pequeñas empresas	Grandes empresas	Cualquiera que este interesado en lo último de software libre

Imagen N.1 versiones de openshift origin

4.1.2 Análisis de lenguaje de programación, frameworks, IDEs y SGBDs utilizados.

Con la información obtenida de los lenguajes de programación, frameworks , IDEs, SGBDs que utilizan los estudiantes para el desarrollo de aplicaciones en la cual podemos observar en el cuadro N.-2 se realizara una comparación con los pass que están siendo analizados como openshift origin y Windows azure.

LENGUAJES DE PROGRAMACIÓN	
	Visual Studio
	JAVA
	PHP
FRAMEWOKS	
	.NET
	EMF
	YII
IDES	
	NetBeans
	Eclipse
SGBDs	
	Oracle
	Postgre
	Mysql
	Sql server
Cuadro N.-2 lenguajes de programación, frameworks , IDEs, SGBDs, FISEI	

En el cuadro N.-3 y N.-4 se realizara una comparación de los lenguajes de programación, Frameworks, Ides y SGBDs utilizados por los estudiantes de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial y los que utiliza Windows Azure y Openshift Origin (PaaS)

	FISEI		Openshift Origin(PasS)	
Lenguajes de Programación	JAVA	X	PHYTON	
	PHP	X	PHP	X
	.NET		RUBY	
			Noe.js	X
			Pearl	
Frameworkos	.NET		DJANGO	
	EMF		CakePHP	
	YII		RAILS	
IDEs	NetBeans			
	Eclipse			
SGBDs	Oracle		Mongo	X
	Postgre	X	Postgre	X
	Mysql	X	Mysql	X
	Sql server			
	Mongo	X		
Cuadro N.- 3 Análisis Lenguaje de programación, Frameworks, IDEs, SGBDs				

	FISEI		Windows Azure(PaaS)	
Lenguajes de Programación	JAVA		Visual Studio	X
	PHP			
	Visual Studio	X		
Frameworks	.NET		.NET	
	EMF			
	YII			
IDES	NetBeans			
	Eclipse			
SGBDs	Oracle		Sqlserver	X
	Postgre	X	Postgre	X
	Mysql	X	Mysql	X
	Sql server			
	Mongo	X		
Cuadro N.-4 Análisis Lenguaje de programación, Frameworks, IDEs, SGBDs				

Luego de la comparación entre los diferentes tipos de PaaS en este caso Windows azure y openshift origin con los que utilizan los estudiantes para el desarrollo de aplicaciones, se puede corroborar que tienen más similitud con las herramientas brindadas por el PaaS de openshift Origin, por los lenguajes de programación en los que tienen más conocimientos ya que las SGBDs son las mismas en las dos distribuciones.

4.1.3 Análisis de requisitos de instalación y despliegue de PaaS.

Antes de la elección de la herramienta PaaS para la realización de este proyecto se realizara un análisis de los requerimientos que se necesitan para el correcto

funcionamiento

En los cuadros N.-5 y N.-6 se realizara una comparación de requisitos para la instalación y despliegue de PaaS.

REQUISITOS WINDOWS AZURE			
HARDWARE		SOFTWARE	
CPU	2	S.O.	Windows Server2012
MEMORIA RAM	8 GB		.NET FRAMEWOK 3.5
DISCO DURO	40 GB		IIS 8
Cuadro N.-5 requisitos Windows Azure			

REQUISITOS OPENSIFT ORIGIN			
HARDWARE		SOFTWARE	
CPU	2	S.O.	Fedora, CentOS o RHEL
MEMORIA RAM	8 GB		Selinux
DISCO DURO	40 GB		NTP
			DNS
			SSH
Cuadro N.-6 Requisitos Openshift Origin			

En la parte de hardware se puede verificar que los requisitos son los mismos tanto en Openshift Origin y Windows azure.

Para la instalación del software no se presentaría inconvenientes ya que los requisitos del hardware cumple con los requerimientos mínimos a parte del software, habría diferencia en los sistemas operativos, ya que Openshift Origin se necesita cualquier distribución de Fedora, CentOS o RHEL y Windows azure necesita Windows Server2012, requisitos con los que se cuenta en la facultad.

4.1.4 Contratación de requerimientos PaaS con infraestructura disponible.

Con la información obtenida de los requisitos necesarios para la instalación de PaaS

se realizara un análisis de la infraestructura requerida tanto en software como hardware en los cuadros N.-7 y N.-8 se detallara los requisitos necesarios

	INFRAESTRUCTURA (IaaS)	(PASS) OPENSIFT ORIGIN
PROCESADOR	CPU 1-2	CPU 2
MEMORIA RAM	8 GB	8 GB
HDD	100 GB	100 GB
S.O		Fedora, CentOS o RHEL
Cuadro N.-7 IaaS- PaaS Openshift Origin		

	INFRAESTRUCTURA (IaaS)	(PASS) WINDOWS AZURE
PROCESADOR	CPU 1-2	CPU 2
MEMORIA RAM	8 GB	8 GB
HDD	100 GB	100 GB
S.O		Windows Server2012
Cuadro N.-8 IaaS- PaaS Windows Azure		

Como se detalló en los cuadros N.-7 y N.-8 la infraestructura disponible en la Facultad de Ingenieria en Sistemas Electrónica e Industrial cumple con los requisitos para la instalación de PaaS tanto en hardware como software.

4.1.5 Selección de PaaS adecuada a la infraestructura disponible

Luego de realizar un análisis de la infraestructura con la que cuenta la Facultad de Ingeniería en Sistemas Electrónica e Industrial como los requerimientos que se necesitan para la instalación de los PaaS Openshift Origin y Windows Azure en la parte de software como hardware se llegó a las siguientes conclusiones:

- La infraestructura necesaria para la instalación en la parte de software y hardware cumplen con los requerimientos necesarios.
- Las herramientas para el desarrollo que brinda Openshift Origin tienen similitud con las herramientas utilizadas por los estudiantes dentro de la facultad como se pudo verificar en el cuadro N.-6.
- Revisado las características presentadas por cada uno de los PaaS tanto en requerimientos de hardware como software y el análisis que se hizo de la

infraestructura disponible en la Facultad de Ingeniería en Sistemas Electrónica e Industrial de la Universidad Técnica de Ambato y los leguajes de programación, frameworks y sgbd's más utilizados por los estudiantes el (PaaS) idóneo para la realización de este proyecto es Openshift Origin.

4.2 Procesos de instalación y configuración de PaaS.

4.2.1 Instalación de PaaS elegida.

Luego del análisis realizado de la infraestructura disponible tanto en hardware como software y cumplir con los requisitos mínimos que se necesitan para la instalación y configuración de PaaS se procede a instalar y configurar Openshift Origin

En el proyecto se presentara el siguiente esquema:

1 máquina "broker" con al menos 1Gb de RAM y 10 Gb de HDD, en la cual se instalará y configurara la parte administrativa de Openshift Origin.

1 máquina "node1" con al menos 1Gb de RAM y 30 Gb de HDD, en la cual se instalará y configurara las herramientas para el desarrollo de aplicaciones.

a) Instalar el Sistema Operativo

Se debe instalar cualquier distribución de Federa, CentOS o RHEL para realización del proyecto se instalará Fedora 19 en ambas máquinas para fines de demostración.

En el proceso de instalación, a la hora de elegir software se seleccionara instalación mínima para no ocupar demasiados recursos.

Configuración de la red:

La máquina bróker tendrá la 172.16.123.14

La máquina node1 tendrá la 172.16.123.15

b) Configuración de los host máquina virtual broker y máquina virtual node1

Una vez instalados los sistemas se deberá actualizarlos con el siguiente comando:

```
yum update -y
```

Se reinician las máquinas para que los cambios surjan efecto.

```
systemctl reboot
```

Lo primero es estar en modo root, luego se configurara los repositorios de OpenShift Origin, en ambas máquinas.

Se puede realizar de dos diferentes formas

La primera con el comando `cat<<EOF>` que nos permite sobre escribir directamente sobre el archivo.

O se puede modificar con cualquier editor, ingresando en la dirección que se encuentra el archivo por ejemplo `vi /etc/yum.repos.d/openshift.repo`.

Para demostración se utilizará `cat <<EOF>` en la mayoría de la configuración

```
[root@broker ~]# cat <<EOF> /etc/yum.repos.d/openshift.repo
> [openshift-origin]
> name=openshift-origin
> baseurl=http://mirror.openshift.com/pub/origin-server/release/3/fedora-19/packages/x86_64/
> gpgcheck=0
> enabled=1
>
> [openshift-origin-deps]
> name=openshift-origin-deps
> baseurl=http://mirror.openshift.com/pub/origin-server/release/3/fedora-19/dependencies/x86_64/
> gpgcheck=0
> enabled=1
> EOF
```

Imagen N.-2 Configuración de repositorio

Es importante que las dos máquinas tengan en los sistemas la hora sincronizada se procederá a instalar **ntp** y a sincronizar con los siguientes comandos:

```
yum install -y ntpdate ntp
ntpdate 0.rhel.pool.tp.org
systemctl enable ntpd.service
systemctl start ntpd.service
```

Se desinstalará el firewall que viene por defecto y en su lugar se instalará `system-config-firewall` en ambas máquinas, firewall que viene preconfigurado para la instalación de Openshift Origin, con el comando `yum` se removerá e instalará como se muestra a continuación:

```
yum remove -y firewalld
yum install -y system-config-firewall
```

Configuración del DNS

En la maquina broker se instalará y configurara el servidor DNS

Se instalarán los paquetes **bind** y **bind-utils**, que nos ofrecen las herramientas necesarias para configurar el servidor de nombres, digitando el siguiente comando.

```
yum install -y bind bind-utils
```

La máquina Broker será responsable de gestionar los nodos. Por lo que necesita acceder al servidor DNS y modificar las entradas, y para seguridad del DNS se utilizará DNSSEC.

Para facilidad de la configuración se utilizará la variable domain. Para ejemplo del proyecto se utilizará el dominio “fisei.ec”,

```
domain=fisei.ec
keyfile=/var/named/${domain}.key
```

Se creará un par de claves y se guardaran los valores de las claves privadas en una variable **KEY** para eso nos dirigimos a cd /var/named y se digitara los siguientes comandos.

```
dnssec-keygen -a HMAC-MD5 -b 512 -n USER -r /dev/urandom ${domain}
KEY="$(grep Key: K${domain}*.private | cut -d ' ' -f 2)"
echo $KEY
```

Es necesario crear otra clave para que el script del servicio pueda acceder al servidor DNS.

```
rndc-confgen -a -r /dev/urandom
```

Se configurara los permisos y el usuario propietario de los ficheros. Además restauramos los contextos de SELinux

```
chown -v root:named /etc/rndc.key
chmod -v 640 /etc/rndc.key
restorecon -v /etc/rndc.* /etc/named.*
```

Ahora se creará una configuración de reenvío de peticiones a otro servidor DNS (para la resolución externa)

```
echo "forwarders { 8.8.8.8; 8.8.4.4; } ;" >> /var/named/forwarders.conf
restorecon -v /var/named/forwarders.conf
chmod -v 640 /var/named/forwarders.conf
```

Ahora se modificara el directorio de resolución dinámica, y si no está instalado, se

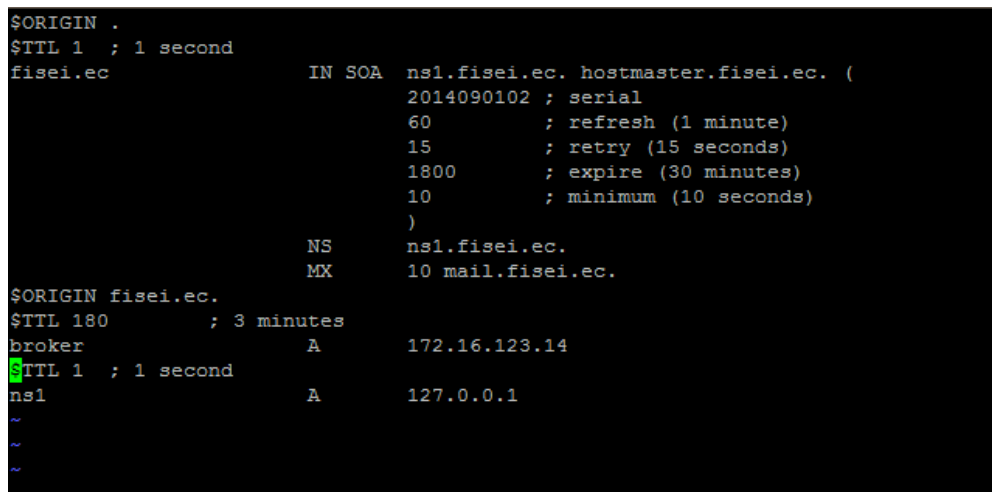
creará el fichero de configuración con el siguiente comando.

```
mkdir /var/named/dynamic
```

Se configurara el archivo dynamic con el comando `cat<<EOF>` o se editara con cualquier editor.

```
cat << EOF >/var/named/dynamic/${domain}.db  
\$ORIGIN .  
\$TTL 1 ; 1 seconds (for testing only)  
${domain} IN SOA ns1.${domain}. hostmaster.${domain}. (  
2014090100 ; serial  
60 ; refresh (1 minute)  
15 ; retry (15 seconds)  
1800 ; expire (30 minutes)  
10 ; minimum (10 seconds)  
)  
NS ns1.${domain}.  
MX 10 mail.${domain}.  
\$ORIGIN ${domain}.  
ns1 A 127.0.0.1  
EOF
```

Se verificara el contenido en `/var/named/dynamic/fisei.ec.db`



```
$ORIGIN .  
$TTL 1 ; 1 second  
fisei.ec IN SOA ns1.fisei.ec. hostmaster.fisei.ec. (  
2014090102 ; serial  
60 ; refresh (1 minute)  
15 ; retry (15 seconds)  
1800 ; expire (30 minutes)  
10 ; minimum (10 seconds)  
)  
NS ns1.fisei.ec.  
MX 10 mail.fisei.ec.  
$ORIGIN fisei.ec.  
$TTL 180 ; 3 minutes  
broker A 172.16.123.14  
$TTL 1 ; 1 second  
ns1 A 127.0.0.1  
~  
~  
~  
~
```

Imagen N.-3 Verificación contenido DNS

Se procederá a instalar la clave DNSSEC para el dominio con el comando `cat <<EOF>`

```
cat <<EOF> /var/named/${domain}.key  
key ${domain} {  
    algorithm HMAC-MD5;  
    secret "${KEY}";  
};  
EOF
```

Se actualiza el propietario y contexto SELinux

```
chown -Rv named:named /var/named  
restorecon -rv /var/named
```

Ahora se modificara el fichero de configuración principal de BIND que se encuentra en el siguiente directorio `/etc/named.conf`

```

> listen-on port 53 { any; };
> directory "/var/named";
> dump-file "/var/named/data/cache_dump.db";
> statistics-file "/var/named/data/named_stats.txt";
> memstatistics-file "/var/named/data/named_mem_stats.txt";
> allow-query { any; };
> recursion yes;
>
> /* Path to ISC DLV key */
> bindkeys-file "/etc/named.iscdlv.key";
>
> // set forwarding to the next nearest server (from DHCP response)
> forward only;
> include "forwarders.conf";
> };
>
> logging {
>     channel default_debug {
>         file "data/named.run";
>         severity dynamic;
>     };
> };
>
> // use the default rndc key
> include "/etc/rndc.key";
>
> controls {
>     inet 127.0.0.1 port 953
>     allow { 127.0.0.1; } keys { "rndc-key"; };
> };
>
> include "/etc/named.rfc1912.zones";
>
> include "${domain}.key";
>
> zone "${domain}" IN {
>     type master;
>     file "dynamic/${domain}.db";
>     allow-update { key ${domain} ; } ;
> };

```

Imagen N.- 4 Configuración del named.conf

chown -v root:named /etc/named.conf

restorecon -v /etc/named.conf

Se configura el firewall para que se inicie y permanezca habilitado

systemctl start iptables

systemctl enable iptables

Se configura y se inicia el servicio de DNS

lokit --service=dns

```
systemctl start named
```

Por último se modificara el hostname de la máquina con el editor en el siguiente directorio `vi /etc/hostname` o directamente con el siguiente comando:

```
hostnamectl broker.fisei.ec
```

c) **Instalación de MongoDB**

Se instalará la base de datos MongoDB que almacenará los datos de la infraestructura de OpenShift.

El servicio de mensajería ActiveMQ que se encargará de la comunicación entre el broker y los nodos.

El cliente de MCollective, que será el encargado de enviar y recibir los mensajes entre el broker y los nodos.

Mediante el comando yum se instalarán los paquetes `mongodb-server`, `mongodb` y `Libmongodb`

```
yum install -y mongodb-server mongodb libmongodb
```

MongoDB utiliza un fichero de configuración que se encuentra en la ruta `/etc/mongodb.conf` que se modificara en dos ocasiones.

La primera será para desactivar temporalmente la autenticación comentando la directiva `auth=true`.

MongoDB, hace uso de una cantidad enorme de espacio en disco, más del que OpenShift requiere, por ello se ingresará la directiva `smallfiles=true`, y con el editor vi se editara en archivo `mongodb.conf`.

```
vi /etc/mongodb.conf
bind_ip = 127.0.0.1
port = 27017
fork = true
pidfilepath = /var/run/mongodb/mongodb.pid
logpath = /var/log/mongodb/mongodb.log
dbpath = /var/lib/mongodb
journal = true
```

```
nohttpinterface = true
smallfiles=true
#auth=true
```

Ahora se iniciara el servicio con el siguiente comando:

```
systemctl start mongod.service
```

Creación del usuario de la base de datos de OpenShift.

Se creará el usuario y contraseña para la base de datos en nuestro caso estas serán las que están en negrillas.

```
mongo localhost/openshift_broker_dev --eval 'db.addUser("openshift",  
"mono")'
```

```
mongo localhost/admin --eval 'db.addUser("openshift", "mono")'
```

La segunda parte de la edición del fichero sera descomentar la autenticación **auth=true** y configurar la directiva *bind_ip=* para que apunte también a nuestra IP pública como se muestra a continuación.

Para esto se detendra el servicio

```
systemctl stop mongod.service
```

luego se editara con el editor vi como se muestra a continuación:

```
vi /etc/mongod.conf
```

```
bind_ip = 127.0.0.1,172.16.123.14  
port = 27017  
fork = true  
pidfilepath = /var/run/mongod/mongod.pid  
logpath = /var/log/mongod/mongod.log  
dbpath =/var/lib/mongod  
journal = true  
nohttpinterface = true  
smallfiles=true  
auth=true
```

Se habilitará el servicio para el arranque del sistema.


```
systemctl enable mongod.service
```

Además se debe configurar el firewall.

```
lokkit --port=27017:tcp
```

d) Instalar ActiveMQ

Para continuar con la configuración se instalará ActiveMQ, para lo cual con el comando yum se instalarán los paquetes necesarios que son activemq y activemq-client

```
yum install -y activemq activemq-client
```

El fichero de configuración principal es /etc/activemq/activemq.xml al que hay que añadir la configuración de queue y topic que requiere OpenShift.

OpenShift facilita la descarga de un fichero preconfigurado. Al cual se modificara unas pocas líneas. También hay que descargar el fichero del servidor donde ActiveMQ se ejecutará, para esto se digitara los siguientes comandos.

```
curl -o /etc/activemq/activemq.xml
```

```
https://raw.githubusercontent.com/openshift/origin-server/openshift-origin-release-3/documentation/files/activemq.xml
```

```
curl -o /etc/activemq/jetty.xml
```

```
https://raw.githubusercontent.com/openshift/origin-server/openshift-origin-release-3/documentation/files/jetty.xml
```

Lo primero será modificar estas líneas del activemq.xml que se encuentra en el siguiente directorio /etc/activemq/activemq.xml

```
<broker xmlns="http://activemq.apache.org/schema/core"
```

```
brokerName="activemq.example.com" dataDirectory="${activemq.data}">
```

Cambiando el bróker Name por nuestro broker.

```
<broker xmlns="http://activemq.apache.org/schema/core"
```

```
brokerName="broker.fisei.ec" dataDirectory="${activemq.data}">
```

Se cambiara el password tanto en el admin como en el usuario mcollective.

```
<simpleAuthenticationPlugin>
  <users>
    <authenticationUser username="mcollective" password="123456"
groups="mcollective,everyone"/>
    <authenticationUser username="admin" password="admin"
groups="mcollective,admin,everyone"/>
  </users>
</simpleAuthenticationPlugin>
```

El interfaz web de ActiveMQ es accesible sólo por localhost pero, podemos hacerlo accesible en el interfaz de red público ingresando el siguiente bloque del fichero */etc/activemq/jetty.xml*

```
<property name="connectors">
  <list>
    <bean id="Connector"
class="org.eclipse.jetty.server.nio.SelectChannelConnector">
      <property name="port" value="8161" />
      <property name="host" value="127.0.0.1" />
    </bean>

    <bean id="External-Connector"
class="org.eclipse.jetty.server.nio.SelectChannelConnector">
      <property name="port" value="8161" />
      <property name="host" value="172.16.123.14" />
    </bean>
  </list>
</property>
```

```

    <property name="connectors">
      <list>
        <bean id="Connector" class="org.eclipse.jetty.server.nio.SelectChannel
Connector">
          <property name="port" value="8161" />
          <property name="host" value="127.0.0.1" />
        </bean>
        <bean id="External-Connector" class="org.eclipse.jetty.server.nio.SelectChannelConn
ector">
          <property name="port" value="8161" />
          <property name="host" value="172.16.123.14" />
        </bean>
      </list>
    </property>

```

Imagen N.- 5 Configuración de la interfaz de ActiveMQ

ActiveMQ trabaja con el antiguo gestor de servicios SysV, así que para que pueda funcionar correctamente con systemd, hemos de crear un fichero para que ActiveMQ pueda almacenar los ficheros temporales

```

cat <<EOF >/etc/tmpfiles.d/activemq.conf
d /var/run/activemq 0755 activemq activemq -
EOF

```

Ahora se debe configurar el firewall, arrancar el servicio y habilitarlo

```

lokkit --port=61613:tcp --port=8161:tcp
service activemq start
chkconfig activemq on

```

Si existe algún error de sintaxis, podemos ver el log en

/var/log/activemq/activemq.log

Y se puede acceder a la consola web con la IP de nuestro broker ***http://172.16.123.14:8161*** con el usuario **admin** y password **admin**.

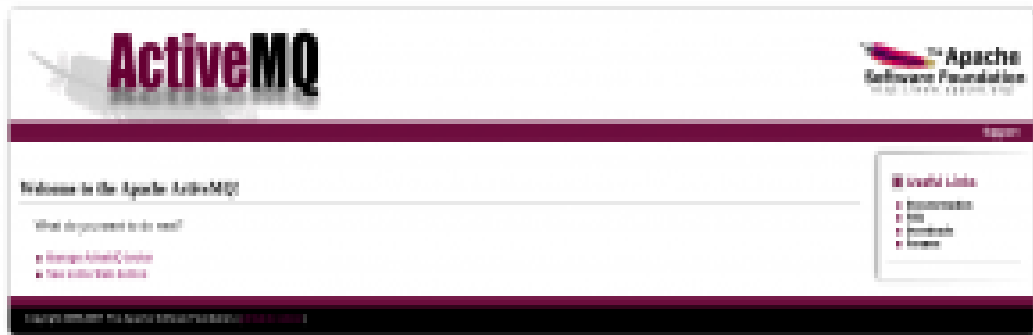


Imagen N.- 6 Interfaz gráfica de ActiveMQ

e) Instalar el cliente de MCollective

Continuando con la configuración se instalará el paquete `mcollective-client`

```
yum install -y mcollective-client
```

Después de instalar el paquete `mcollective-client`, se modificara su fichero de configuración que se encuentra `/etc/mcollective/client.cfg` para que apunte al servidor de ActiveMQ, con el comando `cat <<EOF>` y se ingresará el usuario y contraseña creados anteriormente y que el host apunte al dominio.

```
cat <<EOF > /etc/mcollective/client.cfg  
topicprefix = /topic/  
main_collective = mcollective  
collectives = mcollective  
libdir = /usr/libexec/mcollective  
logfile = /var/log/openshift/broker/mcollective-client.log  
loglevel = debug  
# Plugins  
securityprovider = psk  
plugin.psk = unset  
connector = activemq  
plugin.activemq.pool.size = 1  
plugin.activemq.pool.1.host = broker.fisei.ec
```

```
plugin.activemq.pool.1.port = 61613
plugin.activemq.pool.1.user = mcollective
plugin.activemq.pool.1.password = 123456
EOF
```

f) **Configurar el Broker**

Ahora llega el paso de configurar el broker con sus paquetes correspondientes.

Lo primero de todo es instalar los paquetes necesarios para el broker:

```
yum install -y openshift-origin-broker openshift-origin-broker-util \
rubygem-openshift-origin-auth-remote-user \
rubygem-openshift-origin-auth-mongo \
rubygem-openshift-origin-msg-broker-mcollective \
rubygem-openshift-origin-dns-avahi \
rubygem-openshift-origin-dns-nsupdate \
rubygem-openshift-origin-dns-route53 \
rubygem-passenger mod_passenger
```

Se debe configurar los puertos del firewall para que permanezcan abiertos con el siguiente comando:

```
lokkit --service=ssh --service=http --service=https
```

Para poder permitir que ciertos servicios se comuniquen con el broker, y evitar que dicha comunicación sea insegura, es necesario crear un par de claves de acceso. Para esto se utilizará el comando openssl de la siguiente forma:

```
openssl genrsa -out /etc/openshift/server_priv.pem 2048
openssl rsa -in /etc/openshift/server_priv.pem -pubout >
/etc/openshift/server_pub.pem
```

Para comunicación del broker y los nodos estos utilizan el protocolo SSH, por lo que se creará una clave RSA dentro del bróker.

```
ssh-keygen -t rsa -b 2048 -f /etc/openshift/rsync_id_rsa
```

Como SELinux está funcionando en modo Enforcing, lo que hara será habilitar una serie de booleanos necesarios para el correcto funcionamiento del bróker.

```
setsebool -P httpd_unified=on httpd_can_network_connect=on
httpd_can_network_relay=on \
```

```
    httpd_run_stickshift=on named_write_master_zones=on
allow_ybind=on \
    httpd_verify_dns=on httpd_enable_homedirs=on
httpd_execmem=on \
    httpd_read_user_content=on
```

Por último se generan los contextos necesarios para los directorios y los aplicamos:

```
semanage fcontext -a -t httpd_var_run_t
'/var/www/openshift/broker/httpd/run(/.*)?'
semanage fcontext -a -t httpd_tmp_t '/var/www/openshift/broker/tmp(/.*)?'
semanage fcontext -a -t httpd_log_t '/var/log/openshift/broker(/.*)?'
fixfiles -R rubygem-passenger restore
fixfiles -R mod_passenger restore
restorecon -RFvv /var/run
restorecon -RFvv /opt
restorecon -RFvv /var/www/openshift/
```

El comando fixfiles con la opción -R lee los ficheros que instala un paquete y restaura sus contextos.

g) **Fichero de configuración de OpenShift**

El fichero de configuración principal de OpenShift es el bróker.conf que se encuentra en el siguiente directorio /etc/openshift/broker.conf, y para ajustar la configuración se debe modificar una serie de directivas:

El primero será VALID_GEAR_SIZES: sirve para indicar el tamaño de los gears (que son los espacios de computación de CPU, RAM y HDD asignados)

```
VALID_GEAR_SIZES="small,medium,large"
```

El segundo será el CLOUD_DOMAIN definirá el dominio del cual dependerán los espacios de nombres de los usuarios de OpenShift

```
CLOUD_DOMAIN="fisei.ec"
```

Y por último se establece la configuración para contactar con la base de datos de

MongoDB

```
MONGO_HOST_PORT="broker.fisei.ec:27017"  
MONGO_USER="openshift"  
MONGO_PASSWORD="mono"  
MONGO_DB="openshift_broker_dev"
```

Se guardan los cambios y salimos del archivo.

Lo siguiente será generar los bits aleatorios para las huellas de autenticación y de sesión.

```
echo "AUTH_SALT=\"\$(openssl rand -base64 64/tr -d \"\n\")\"">>  
/etc/openshift/broker.conf
```

```
echo "SESSION_SECRET=\"\$(openssl rand -base64 64/tr -d \"\n\")\"">>  
/etc/openshift/broker.conf
```

h) Configuración de plugins

Los plugins se encargan de tareas como gestionar la autenticación, la actualización del DNS o la gestión de la mensajería de los nodos. Disponen de sus propios ficheros de configuración correspondientes, además se contara con unos ficheros de ejemplo que se tomaran como plantilla.

Lo primero será situarnos en el directorio de los plugins que se encuentra en /etc/openshift/plugins.d, donde se encontraran los ficheros de ejemplo: openshift-origin-auth-remote-user.conf.example, openshift-origin-dns-nsupdate.conf.example y openshift-origin-msg-broker-mcollective.conf.example.

Por seguridad para la configuración será dejar los ficheros de plantilla y trabajar sobre una copia de los mismos, para esto nos dirigimos a la siguiente dirección cd /etc/openshift/plugins.d y se realizaran las copias como se muestra a continuación:

```
cp openshift-origin-auth-remote-user.conf{.example,}  
cp openshift-origin-dns-nsupdate.conf{.example,}  
cp openshift-origin-msg-broker-mcollective.conf{.example,}
```

Del fichero `openshift-origin-msg-broker-mcollective.conf` hay que cambiar la directiva:

```
MCOLLECTIVE_CONFIG="%{scl_root}//etc/mcollective/client.cfg"
```

Y establecerla a:

```
MCOLLECTIVE_CONFIG="/etc/mcollective/client.cfg"
```

Ahora se configurara el plugin de actualización del DNS, así que para facilitar la configuración se utilizará de nuevo las variables que se ha usado previamente:

```
domain=fisei.ec  
keyfile=/var/named/${domain}.key  
KEY="$(grep Key: /var/named/K${domain}*.private | cut -d ' ' -f2)"
```

Y ahora se editara las directivas necesarias del plugin con el comando `cat << EOF >`.

```
cat << EOF > openshift-origin-dns-nsupdate.conf  
BIND_SERVER="127.0.0.1"  
BIND_PORT=53  
BIND_KEYNAME="${domain}"  
BIND_KEYVALUE="${KEY}"  
BIND_ZONE="${domain}"  
EOF
```

i) **Configurar autenticación y creación de usuarios**

OpenShift soporta diversos mecanismos de autenticación tales como MongoDB, Kerberos, LDAP, etc. Para desarrollo del proyecto se empleara el mecanismo Basic, que utiliza la autenticación de Apache, por lo que se utilizará la plantilla ofrecida por OpenShift:

```
cp /var/www/openshift/broker/httpd/conf.d/openshift-origin-auth-remote-  
user-basic.conf{.sample,}
```

En la directiva ***AuthUserFile*** `/etc/openshift/htpasswd` apunta al fichero donde se almacenarán los usuarios y los passwords. Para ello ejecutamos el comando de

Apache.

```
htpasswd -c /etc/openshift/htpasswd admin
```

Hay que tener en cuenta que la opción **-c** crea el fichero si no existe, pero si existe borra todo su contenido, así que para añadir nuevos usuarios simplemente se elimina esa opción.

OpenShift Admin Console

Se puede gestionar el broker desde una consola de administración, para ello se instalarán los paquetes necesarios.

```
yum install -y rubygem-openshift-origin-admin-console
```

Ruby Bundle

Dado que OpenShift está construido sobre Ruby on Rails, para que el broker pueda satisfacer las dependencias, se ubicara en el siguiente directorio `cd /var/www/openshift/broker` y luego se ejecutara el siguiente comando:

```
bundle --local
```

Arrancando los servicios

Se iniciara y habilitará el arranque de los servicios de openshift y httpd

```
systemctl start openshift-broker.service
```

```
systemctl start httpd.service
```

```
systemctl enable openshift-broker.service
```

```
systemctl enable httpd.service
```

Verificar el funcionamiento

Se puede realizar prueba de funcionamiento gracias a la herramienta **curl** y si para verlo en un formato más legible se usara **python** y un módulo de **json** con el siguiente comando:

```
curl -u admin:admin -k https://broker.fisei.ec/broker/rest/api.json|python -m json.tool
```

El cual despliega la siguiente ventana en la que se muestra el perfecto funcionamiento.

```

root@broker plugins.d]# curl -u admin:admin -k https://broker.fisei.ec/broker/rest/api.json|python -m json.tool
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100  5966  100  5966    0     0  14896      0  --:--:--  --:--:--  --:--:-- 14877

{
  "api_version": 1.6,
  "data": {
    "ADD_AUTHORIZATION": {
      "href": "https://broker.fisei.ec/broker/rest/user/authorizations",
      "method": "POST",
      "optional_params": [
        {
          "default_value": "userinfo",
          "description": "Select one or more scopes that this authorization will grant access to:\n\n* session\n Grants a client the authority to perform all API actions against your account. Valid for 1 day.\n* read\n Allow the client to access resources you own without making changes. Does not allow access to view authorization tokens. Valid for about 1 month.\n* userinfo\n Allows a client to view your login name, unique id, and your user capabilities. Valid for about 1 month.\n* domain/:id/view\n Grant read-only access to a single domain. Valid for about 1 month.\n* domain/:id/edit\n Grant edit access to a single domain and all its applications. Valid for about 1 month.\n* domain/:id/admin\n Grant full administrative access to a single domain and all its applications. Valid for about 1 month.\n* application/:id/view\n Grant read-only access to a single application. Valid for about 1 month.\n* application/:id/edit\n Grant edit access to a single application. Valid for about 1 month.\n* application/:id/admin\n Grant full administrative access to a single application. Valid for about 1 month.",
          "name": "scope",
          "type": "string",
          "valid_options": [
            "session",
            "read",
            "userinfo",
            "domain/:id/view",
            "domain/:id/edit"
          ]
        }
      ]
    }
  }
}

```

Imagen N.- 7 Verificación de funcionamiento con Json

Y mediante una conexión por ssh, podemos realizar una redirección de puertos para poder acceder a la consola

```
ssh -Nf -L 8080:localhost:8080 root@broker.fisei.ec
```

Configuración de la consola web

Lo primero de todo será instalar el paquete necesario

```
yum install -y openshift-origin-console
```

Lo siguiente será configurar la autenticación, se configura la consola web, aquella de acceso público para los usuarios.

La configuración, igual que para la admin-console, usará el mecanismo de autenticación BASIC de apache.

Lo primero se generará una serie de bits para la seguridad de la sesión que se guardará

en el fichero `/etc/openshift/console.conf`

```
echo "SESSION_SECRET=$(openssl rand -base64 64/tr -d "\\n")" >>
/etc/openshift/console.conf
```

Ahora se realizara una copia de la plantilla del fichero de configuración de autenticación en la cual se trabajara.

```
cp /var/www/openshift/console/httpd/conf.d/openshift-origin-auth-remote-user-
basic.conf{.sample,}
```

Se verifica que las dependencias de ruby se estén resolviendo, para lo cual se ubicara en el siguiente directorio `cd /var/www/openshift/console` y se ejecutara los siguientes comandos:

```
bundle --local
rake assets:precompile
chown -R apache:apache Gemfile.lock tmp
```

El comando **bundle --local** debe ofrecernos una salida similar a la siguiente

Your bundle is complete! Use ``bundle show [gemname]`` to see where a bundled gem is installed.

j) SELinux

La consola web también requiere una configuración de SELinux, así que se habilitaran los booleans y los contextos necesarios.

```
setsebool -P httpd_unified=on
setsebool -P httpd_can_network_connect=on
setsebool -P httpd_can_network_relay=on
setsebool -P httpd_run_stickshift=on
setsebool -P named_write_master_zones=on
setsebool -P allow_yplibind=on
setsebool -P httpd_verify_dns=on
setsebool -P httpd_enable_homedirs=on
setsebool -P httpd_execmem=on
```

```
setsebool -P httpd_read_user_content=on
```

```
semanage fcontext -a -t httpd_log_t '/var/log/openshift/console(/.*)?'  
semanage fcontext -a -t httpd_log_t '/var/log/openshift/console/httpd(/.*)?'  
semanage fcontext -a -t httpd_var_run_t  
'/var/www/openshift/console/httpd/run(/.*)?'
```

```
fixfiles -R rubygem-passenger restore  
fixfiles -R mod_passenger restore
```

```
restorecon -RFvv /var/run  
restorecon -RFvv /opt  
restorecon -RFvv /var/log/openshift/console  
restorecon -RFvv /var/www/openshift/console
```

Activando los servicios

Para activar y habilitar el servicio de la consola web se digitaran los siguientes comandos

```
systemctl start openshift-console.service  
systemctl enable openshift-console.service
```

Y si todo acido configurado correctamente se procederá a abrir el navegador e ingresar el DNS bróker.fisei.ec como se muestra en la imagen N.-8 en la cual pedirá el usuario y contraseña

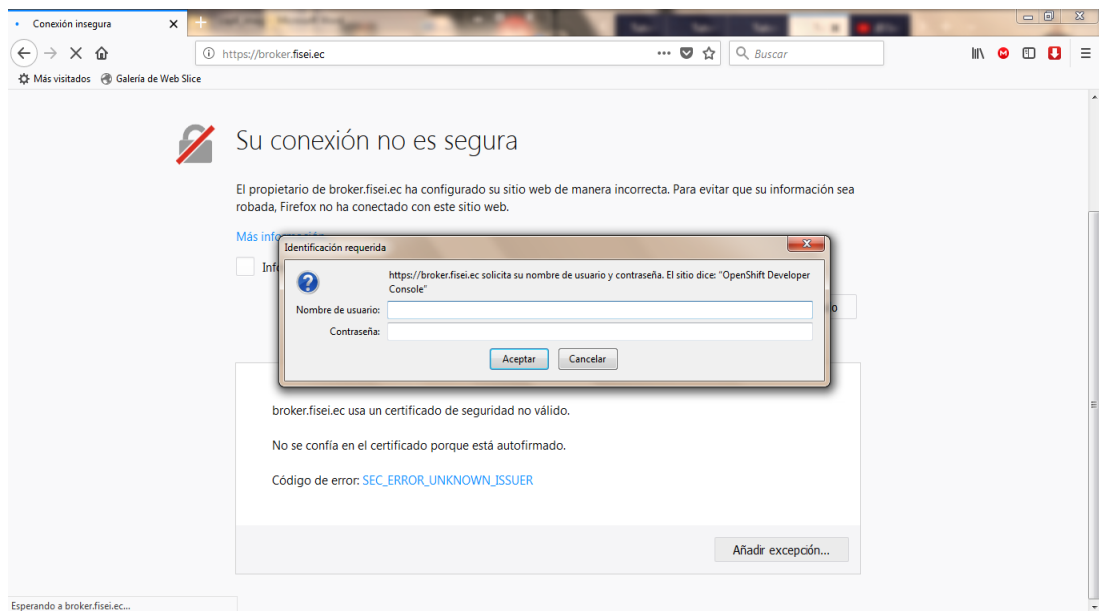


Imagen N.- 8 Verificación con la consola web

La cual nos dirigirá a la ventana principal de Openshift Origin imagen N.-9 la cual no se podrá realizar pruebas por el momento ya que no se han instalado los cartuchos necesarios para la realización de las pruebas.

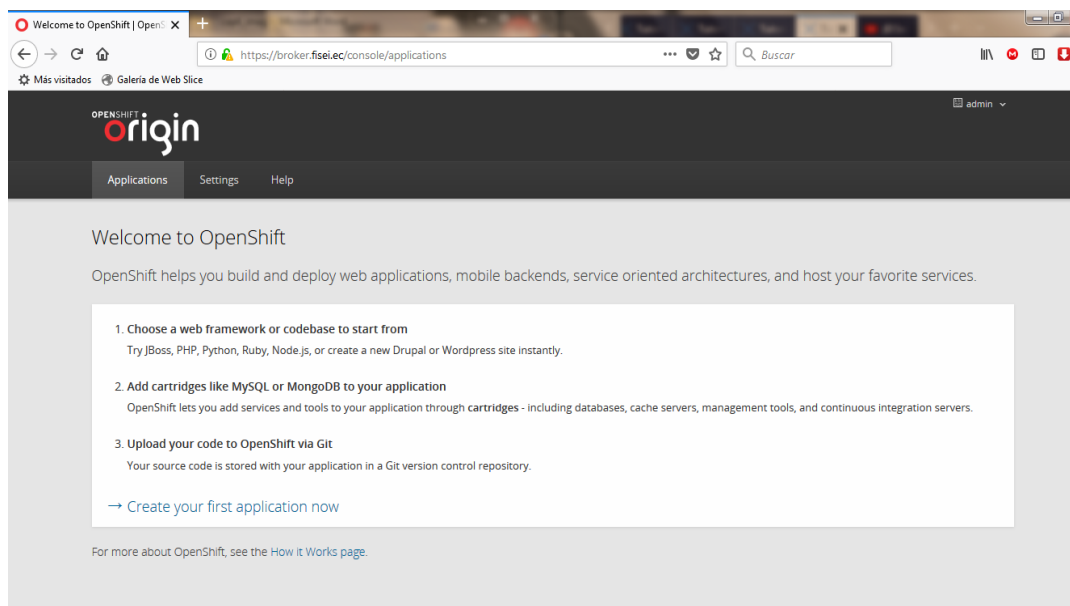


Imagen N.- 9 Verificación con la consola web ventana de Openshift

4.2.2 Instalación de frameworks, lenguajes de programación y SGBDs

Luego de realizar el análisis de los frameworks, lenguajes de programación y SGBDs utilizados por los estudiantes de la Facultad de Ingeniería en Sistemas Electrónica e industrial imagen N.-3 y las herramientas ofrecidas por Openshift Origin.

Se procederá con la instalación de las herramientas necesarias cumpliendo con las necesidades de los estudiantes para el desarrollo de sus aplicaciones.

Asumiendo el siguiente esquema de red:

- Broker: IP=172.16.123.14
- Node1: IP=172.16.123.15

Se creará la entrada en el DNS de node1, y para ello se a utilizarán las herramientas que tenemos disponibles en el broker.

Para lo cual se ubicara en la maquina bróker y se digitara el siguiente comando.

```
oo-register-dns -s 172.16.123.14 -h node1 -d fisei.ec -n 172.16.123.15 -k  
/var/named/fisei.ec.key
```

1. **-s** define la dirección del servidor DNS
2. **-h** determina el nombre del host a registrar
3. **-d** referencia el dominio donde se registra
4. **-n** indica la dirección IP
5. **-k** la clave para modificar el DNS

Ahora se copiara la clave SSH que permite al broker enviar archivos a los nodos, y se verifica que se pueda conectar con ella

```
ssh-copy-id -i /etc/openshift/rsync_id_rsa.pub node1.fisei.ec  
ssh -i /etc/openshift/rsync_id_rsa.pub node1.fisei.ec date
```

Configuración de Node1

Configuración de la red del node1.

Se configurara la red del node1 al servidor DNS en la maquina bróker para eso se ingresará al siguiente directorio /etc/sysconfig/network-scripts/ifcfg-enp0s3, y se ingresará la dirección ip del DNS.

```

IPV6_PEERDNS="yes"
IPV6INIT="yes"
UUID="e2d0a631-f813-4eeb-b21e-8c68914b0688"
IPADDR0="172.16.123.15"
DNS1="172.16.123.14"
PREFIX0="24"
DEFROUTE="yes"
IPV4_FAILURE_FATAL="no"
HWADDR="08:00:27:55:9A:29"
BOOTPROTO="none"
IPV6_DEFROUTE="yes"
GATEWAY0="172.16.123.254"
IPV6_AUTOCONF="yes"
IPV6_FAILURE_FATAL="no"
IPV6_PEERROUTES="yes"
TYPE="Ethernet"
ONBOOT="yes"
NAME="enp0s3"
~

```

Imagen N.- 10 Configuración de la red del node1

Se modificara el hostname de la máquina que se encuentra en el siguiente directorio /etc/hostname y se ingresará lo siguiente:

```
node1.fisei.ec
```

MCollective

Se instalará el paquete de MCollective en la máquina nodo con el comando yum:

```
yum install -y openshift-origin-msg-node-mcollective
```

Una vez instalado se modificara el fichero de configuración de MCollective que se encuentra en el siguiente directorio /etc/mcollective/server.cfg

```

cat <<EOF>/etc/mcollective/server.cfg
topicprefix = /topic/
main_collective = mcollective
collectives = mcollective
libdir = /usr/libexec/mcollective

```

```

logfile = /var/log/openshift/node/mcollective.log
loglevel = debug
daemonize = 0
direct_addressing = 1
registerinterval = 30

# Plugins
securityprovider = psk
plugin.psk = unset

connector = activemq
plugin.activemq.pool.size = 1
plugin.activemq.pool.1.host = broker.fisei.ec
plugin.activemq.pool.1.port = 61613
plugin.activemq.pool.1.user = mcollective
plugin.activemq.pool.1.password = 123456

# Facts
factssource = yaml
plugin.yaml = /etc/mcollective/facts.yaml
EOF

```

Para el funcionamiento en Federa 19 de MCollective con systemd se necesita crear el fichero de servicio.

```

cat <<EOF> /usr/lib/systemd/system/mcollective.service
[Unit]
Description=The Marionette Collective
After=network.target

[Service]
Type=simple
StandardOutput=syslog

```



```
StandardError=syslog
ExecStart=/usr/sbin/mcollectived --config=/etc/mcollective/server.cfg --
pidfile=/var/run/mcollective.pid
ExecReload=/bin/kill -USR1 \${MAINPID}
PIDFile=/var/run/mcollective.pid
KillMode=process

[Install]
WantedBy=multi-user.target
EOF
```

Luego de haber creado el fichero se debe reiniciar systemd, arrancar y habilitar el servicio

```
systemctl --system daemon-reload
systemctl enable mcollective.service
systemctl start mcollective.service
```

Instalando paquetes

Para que el nodo pueda trabajar correctamente, al igual que se hizo con el broker, se instalará una serie de paquetes.

```
yum install -y rubygem-openshift-origin-node rubygem-passenger-native
openshift-origin-port-proxy openshift-origin-node-util rubygem-openshift-
origin-container-selinux httpd
```

Ahora se realizará la configuración del enrutamiento de las peticiones HTTP/HTTPS y se habilitará el acceso a la información de los cartuchos

```
echo "ServerName node1.fisei.ec" >
/etc/httpd/conf.d/000001_openshift_origin_node_servername.conf
```

Se instalará los front-ends de Apache Virtual Hosts y de NodeJS WebSockets

```
yum install -y rubygem-openshift-origin-frontend-apache-mod-rewrite
yum install -y openshift-origin-node-proxy rubygem-openshift-origin-
frontend-nodejs-websocket
```

Se iniciaran los servicios necesarios y se configurara el firewall

```
systemctl start openshift-node-web-proxy
systemctl start httpd
systemctl enable openshift-node-web-proxy
systemctl enable httpd
iptables -N rhc-app-comm
iptables -I INPUT 4 -m tcp -p tcp --dport 35531:65535 -m state --state NEW -
j ACCEPT
iptables -I INPUT 5 -j rhc-app-comm
iptables -I OUTPUT 1 -j rhc-app-comm
/usr/libexec/iptables/iptables.init save
```

Instalar cartuchos

Se procederá a instalar los cartuchos que estarán disponibles desde el nodo.

Para ello se enlistaran los cartuchos que tenemos disponibles ejecutando

```
yum list openshift-origin-cartridge\*
```

```
openshift-origin-cartridge-haproxy.noarch 1.18.0.1-1.fc19 @openshift-origin
openshift-origin-cartridge-mongodb.noarch 1.17.1.1-1.fc19 @openshift-origin
openshift-origin-cartridge-mysql.noarch 1.19.0.1-1.fc19 @openshift-origin
openshift-origin-cartridge-nodejs.noarch 1.19.0.1-1.fc19 @openshift-origin
openshift-origin-cartridge-perl.noarch 1.17.0.3-1.fc19 @openshift-origin
openshift-origin-cartridge-php.noarch 1.18.0.3-1.fc19 @openshift-origin
openshift-origin-cartridge-postgresql.noarch 1.18.0.1-1.fc19 @openshift-origin
openshift-origin-cartridge-python.noarch 1.18.0.3-1.fc19 @openshift-origin
openshift-origin-cartridge-ruby.noarch 1.18.0.3-1.fc19 @openshift-origin
Paquetes disponibles
openshift-origin-cartridge-10gen-mms-agent.noarch
openshift-origin-cartridge-10gen-mms-agent.noarch 1.29.1.1-1.fc19 openshift-origin
openshift-origin-cartridge-abstract.noarch 1.5.9-1.fc19 fedora
openshift-origin-cartridge-abstract-jboss.noarch
openshift-origin-cartridge-abstract-jboss.noarch 1.5.9-1.fc19 fedora
openshift-origin-cartridge-cron.noarch 1.17.0.1-1.fc19 openshift-origin
openshift-origin-cartridge-cron-1.4.noarch 1.9.1-2.fc19 updates
openshift-origin-cartridge-diy.noarch 1.16.1.1-1.fc19 openshift-origin
openshift-origin-cartridge-diy-0.1.noarch 1.5.3-1.fc19 fedora
openshift-origin-cartridge-jbossas.noarch 1.18.0.1-1.fc19 openshift-origin
openshift-origin-cartridge-jbosseap.noarch 2.11.1.1-1.fc19 openshift-origin
openshift-origin-cartridge-jbossews.noarch 1.17.1.1-1.fc19 openshift-origin
openshift-origin-cartridge-jenkins.noarch 1.16.1.1-1.fc19 openshift-origin
openshift-origin-cartridge-jenkins-client.noarch
```

Imagen N.- 11 Lista de cartuchos disponibles en openshift origin

Cumpliendo los requerimientos de los estudiantes se instalarán los necesarios, pero de ser el caso se podría instalar todos los cartuchos que vienen por defecto en Openshift

Origin con el siguiente comando:

```
yum install -y openshift-origin-cartridge\*
```

```
node1 ~]# yum install -y openshift-origin-cartridge-haproxy.noarch openshift-origin-cartridge-mongodb.noarch openshift-origin-cartridge-mysql.noarch openshift-origin-cartridge-nodejs.noarch openshift-origin-cartridge-perl.noarch openshift-origin-cartridge-php.noarch openshift-origin-cartridge-postgresql.noarch openshift-origin-cartridge-python.noarch openshift-origin-cartridge-ruby.noarch
```

Imagen N.- 12 lista de cartuchos escogida para la instalación

Una vez instalados los paquetes, se pondrán disponibles los cartuchos en el node1

```
/usr/sbin/oo-admin-cartridge --recursive -a install -s
```

```
/usr/libexec/openshift/cartridges/
```

Firewall y servicios

Se configurará el firewall y habilitarán los servicios para el arranque

```
lokkit --service=ssh
```

```
lokkit --service=https
```

```
lokkit --service=http
```

```
lokkit --port=8000:tcp
```

```
lokkit --port=8443:tcp
```

```
systemctl enable network.service
```

```
systemctl enable sshd.service
```

```
systemctl enable oddjobd.service
```

Módulos PAM

OpenShift Origin hace uso de unos módulos PAM propios, pero como prevalecen los del sistema, se configurará los módulos del sistema para que se usen los de OpenShift. Augeas es un proyecto que pertenece Red Hat y que es una herramienta de edición de ficheros de configuración. Analiza los archivos de configuración en sus formatos nativos y los transforma en un árbol. Los cambios de configuración se realizan mediante la manipulación de este árbol y los guarda de nuevo en los archivos de configuración de origen.

Se procederá a instalar el paquete augeas

```
yum install -y augeas
```

Ya realizada la instalación se editara con el comando cat<<EOF>

```
cat <<EOF | augtool  
set /files/etc/pam.d/sshd/#comment[.='pam_selinux.so close should be the first  
session rule'] 'pam_openshift.so close should be the first session rule'  
ins 01 before /files/etc/pam.d/sshd/*[argument='close']  
set /files/etc/pam.d/sshd/01/type session  
set /files/etc/pam.d/sshd/01/control required  
set /files/etc/pam.d/sshd/01/module pam_openshift.so  
set /files/etc/pam.d/sshd/01/argument close  
set /files/etc/pam.d/sshd/01/#comment 'Managed by openshift_origin'  
  
set /files/etc/pam.d/sshd/#comment[.='pam_selinux.so open should only be  
followed by sessions to be executed in the user context'] 'pam_openshift.so  
open should only be followed by sessions to be executed in the user context'  
ins 02 before /files/etc/pam.d/sshd/*[argument='open']  
set /files/etc/pam.d/sshd/02/type session  
set /files/etc/pam.d/sshd/02/control required  
set /files/etc/pam.d/sshd/02/module pam_openshift.so  
set /files/etc/pam.d/sshd/02/argument[1] open  
set /files/etc/pam.d/sshd/02/argument[2] env_params  
set /files/etc/pam.d/sshd/02/#comment 'Managed by openshift_origin'  
  
rm /files/etc/pam.d/sshd/*[module='pam_selinux.so']  
  
set /files/etc/pam.d/sshd/03/type session  
set /files/etc/pam.d/sshd/03/control required  
set /files/etc/pam.d/sshd/03/module pam_namespace.so  
set /files/etc/pam.d/sshd/03/argument[1] no_unmount_on_close
```

```

set /files/etc/pam.d/sshd/03/#comment 'Managed by openshift_origin'

set /files/etc/pam.d/sshd/04/type session
set /files/etc/pam.d/sshd/04/control optional
set /files/etc/pam.d/sshd/04/module pam_cgrouop.so
set /files/etc/pam.d/sshd/04/#comment 'Managed by openshift_origin'

set /files/etc/pam.d/runuser/01/type session
set /files/etc/pam.d/runuser/01/control required
set /files/etc/pam.d/runuser/01/module pam_namespace.so
set /files/etc/pam.d/runuser/01/argument[1] no_unmount_on_close
set /files/etc/pam.d/runuser/01/#comment 'Managed by openshift_origin'

set /files/etc/pam.d/runuser-l/01/type session
set /files/etc/pam.d/runuser-l/01/control required
set /files/etc/pam.d/runuser-l/01/module pam_namespace.so
set /files/etc/pam.d/runuser-l/01/argument[1] no_unmount_on_close
set /files/etc/pam.d/runuser-l/01/#comment 'Managed by openshift_origin'

set /files/etc/pam.d/su/01/type session
set /files/etc/pam.d/su/01/control required
set /files/etc/pam.d/su/01/module pam_namespace.so
set /files/etc/pam.d/su/01/argument[1] no_unmount_on_close
set /files/etc/pam.d/su/01/#comment 'Managed by openshift_origin'

set /files/etc/pam.d/system-auth-ac/01/type session
set /files/etc/pam.d/system-auth-ac/01/control required
set /files/etc/pam.d/system-auth-ac/01/module pam_namespace.so
set /files/etc/pam.d/system-auth-ac/01/argument[1] no_unmount_on_close
set /files/etc/pam.d/system-auth-ac/01/#comment 'Managed by
openshift_origin'

save
EOF

```

```
cat <<EOF > /etc/security/namespace.d/sandbox.conf
#/sandbox    \${HOME}/.sandbox/    user:iscript=/usr/sbin/oo-namespace-
init    root,adm,apache
EOF
```

```
cat <<EOF > /etc/security/namespace.d/tmp.conf
/tmp    \${HOME}/.tmp/    user:iscript=/usr/sbin/oo-namespace-init
root,adm,apache
EOF
```

```
cat <<EOF > /etc/security/namespace.d/vartmp.conf
/var/tmp \${HOME}/.tmp/ user:iscript=/usr/sbin/oo-namespace-init
root,adm,apache
EOF
```

CGroups

CGroups permite ubicar recursos de cpu, de memoria, de red o una combinación de todos, para una tarea de usuario (proceso). Así que haremos uso de CGroups para establecer los límites en nuestro nodo.

En Fedora debería estar instalado por defecto, así que se arrancaran los servicios necesarios. Es importante el orden de arranque, ya que existe una dependencia entre ellos.

```
systemctl start cgconfig.service
systemctl start cgred.service
systemctl enable cgconfig.service
systemctl enable cgred.service
```

Estableciendo los límites “quotas”

En el fichero `/etc/openshift/resource_limits.conf` se establecen los límites para los gears. De todas las directivas sólo se configuraran dos, la que establece el límite de

almacenamiento en disco y la que establece el límite de RAM asignada:

```
# Quotas: File system limits
#
quota_files=80000
# 1GB. 1 block = 1024byte
quota_blocks=1048576
```

ImagenN.- 13 Configuración de límite de almacenamiento en disco

```
#
# memory
memory_limit_in_bytes=536870912      # 512MB
memory_mems_wlimit_in_bytes=641728512 # 512M + 100M (100M swap)
# memory_soft_limit_in_bytes=-1
# memory_swappiness=60
```

Imagen.- 14 Configuración de límite de almacenamiento memoria RAM

Para que las cuotas de disco funcionen, se habilitarán en el punto de montaje, así que se modificara la entrada del fichero /etc/fstab

```
/dev/mapper/fedora_node1-root /          ext4  defaults    1 1
```

Por esta otra como se muestra en la imagen N.-15

```
#
# /etc/fstab
# Created by anaconda on Fri Feb  2 05:31:02 2018
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/mapper/fedora-root /          ext4  defaults,usrquota
1 1
UUID=4417af98-b818-4d16-bb0b-7f6265275d2a /boot    ext4  default
ts      1 2
/dev/mapper/fedora-swap swap     swap  defaults      0 0
~
~
```

Imagen N.- 15 Habilitación de las cuotas de disco

Se remontara la unidad y habilitará la información de cuotas

```
mount -o remount /
```

```
quotacheck -cumgv /
```

SELinux

Se habilitarán varios booleanos de SELinux, y restaurar varios contextos

```
setsebool -P httpd_unified=on httpd_can_network_connect=on
httpd_can_network_relay=on httpd_read_user_content=on
httpd_enable_homedirs=on httpd_run_stickshift=on
allow_polyinstantiation=on httpd_run_stickshift=on httpd_execmem=on
restorecon -RFvv /var/run
restorecon -RFvv /var/lib/openshift/etc/openshift/node.conf
/etc/httpd/conf.d/openshift
```

System Control

Ya que el sistema va a gestionar multitud de peticiones http, no se puede permitir que se disparen más procesos httpd de los normales y por ello se establecerán una serie de semáforos en el kernel para evitar el colapso de los procesos, así que se modificará el parámetro del kernel *kernel.sem*. También se deberá incrementar el número de puertos en uso, así como también el tamaño de la tabla de seguimiento de conexiones del kernel, para ello se utilizarán los parámetros *net.ipv4.ip_local_port_range* y *net.netfilter.nf_conntrack_max* respectivamente.

Nuevamente se utilizará las augeas

```
cat <<EOF | augtool
set /files/etc/sysctl.conf/kernel.sem "250 32000 32 4096"
set /files/etc/sysctl.conf/net.ipv4.ip_local_port_range "15000 35530"
set /files/etc/sysctl.conf/net.netfilter.nf_conntrack_max "1048576"
save
EOF

sysctl -p /etc/sysctl.conf
```

SSH y GIT

Se permitirá que SSH pase a través de la variable de entorno *GIT_SSH*, así que se modificará el fichero */etc/ssh/sshd_config*.


```
cat <<EOF>> /etc/ssh/sshd_config
AcceptEnv GIT_SSH
EOF
```

Habilitar servicios y firewall

Se iniciaran los servicios para el control de tráfico de red, y se configurara el redireccionamiento de aplicaciones a su correspondiente contenido (porque lógicamente tendremos más de una aplicación en cada nodo)

```
lokkit --port=35531-65535:tcp
systemctl enable httpd.service
systemctl enable openshift-port-proxy.service
systemctl enable openshift-tc.service
systemctl enable openshift-gears.service
```

Configurando el dominio para el nodo

Se establecerán los parámetros del dominio de OpenShift para el nodo. Esto se realizara en el fichero `/etc/openshift/node.conf` mediante las siguientes directivas:

```
# These should not be left at default values, even for a demo.
# "PUBLIC" networking values are ones that end-users should be able to reach
PUBLIC_HOSTNAME="node1.fisei.ec" # The node host's public hostname
PUBLIC_IP="172.16.123.15" # The node
's public IP address
BROKER_HOST="broker.fisei.ec" # IP or DN
ame of broker server for REST API

# Usually (unless in a demo) this should be changed to the domain for your in
llation:
CLOUD_DOMAIN="fisei.ec" # Domain suffix to u
for applications (Must match broker config)

# You may want these, depending on the complexity of your networking:
EXTERNAL_ETH_DEU='enp0s3' # Specify the in
net facing public ethernet device
INTERNAL_ETH_DEU='eth1' # Specify the in
nal cluster facing ethernet device

# Uncomment and use the following line if you want to gear users to be member
f
# additional groups besides the one with the same id as the uid. The other gr
# should be an existing group.
```

Imagen N.- 16 Configuración del dominio para el node1

Por último se inicializa el *facter* que permitirá recolectar los metadatos de MCollective. Este es un trabajo que gestiona el cron, pero así aseguramos de su correcto funcionamiento

```
/etc/cron.minutely/openshift-facts
```

Reiniciar el nodo y se realizara las pruebas de funcionamiento.

```
systemctl reboot
```

4.3 Implementar PaaS y pruebas de funcionamiento.

4.3.1 Pruebas de creación y despliegue de proyectos de desarrollo.

Para la creación y despliegue de las aplicaciones, se la realizaron en la facultad de ingeniería en sistemas electrónica e industrial, en un laboratorio de computación con 20 equipos.

En cada equipo se configuro la dirección del dns para poder tenerlas en red y se puedan comunicar con la maquina bróker.

Se ingresó al browser y se digito la dirección bróker.fisei.ec en la cual nos desplegó la siguiente ventana

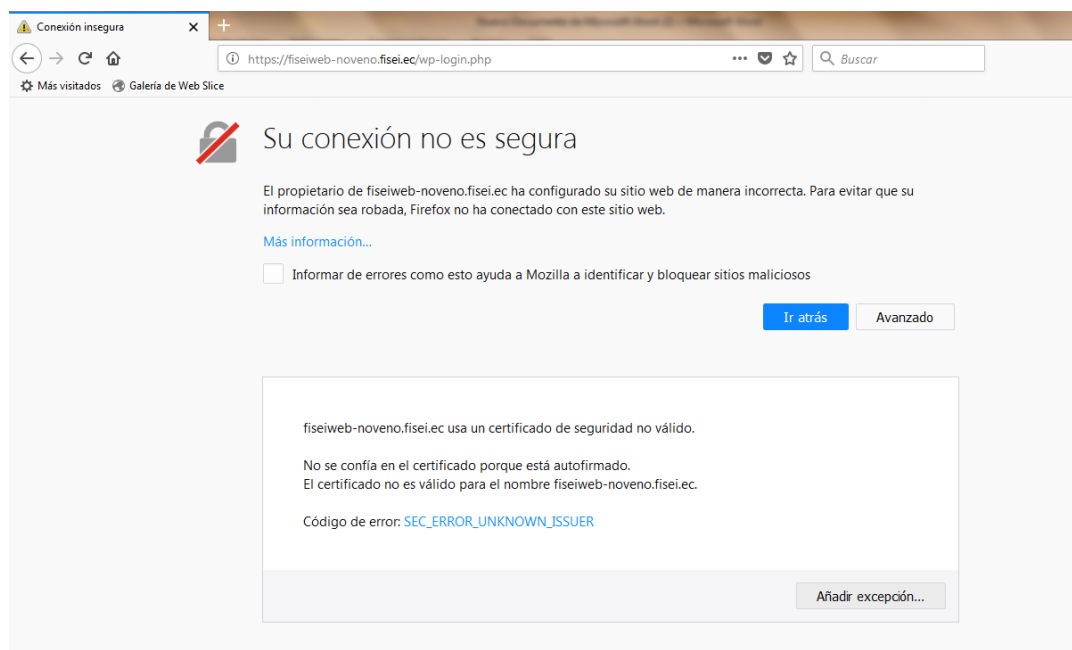


Imagen N.- 17 Ingreso al browser para pruebas

Se Añade la excepción la cual pide por motivos de seguridad

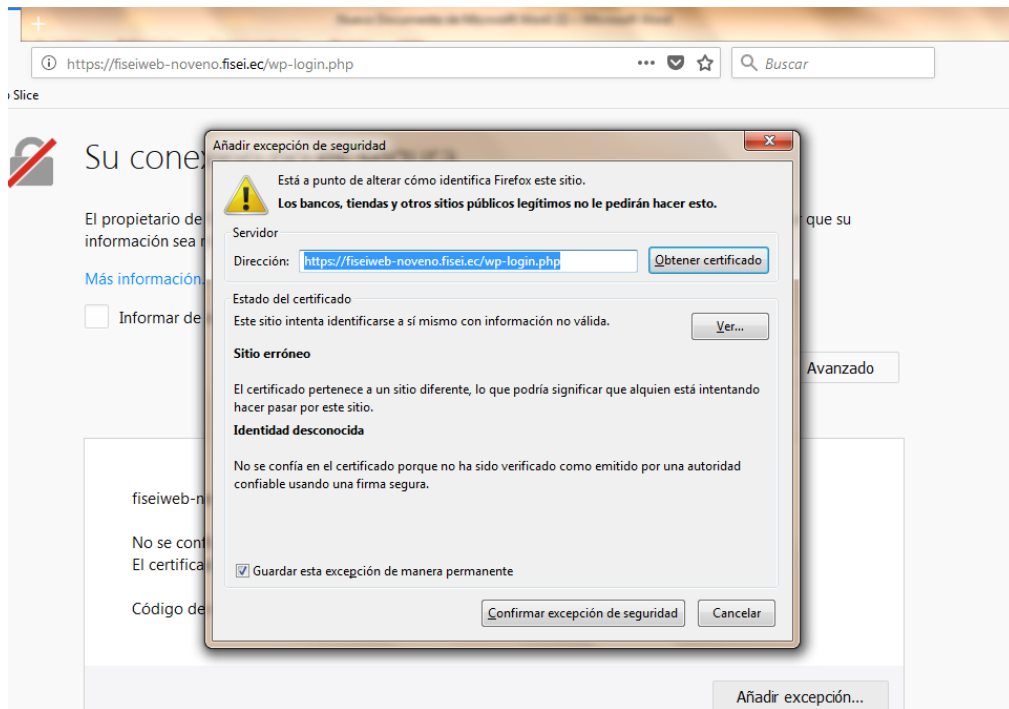


Imagen N.-18 Añadir excepción

Luego de añadir la excepción, se desplegará una ventana en la cual pedirá el usuario y contraseña, y se ingresará el usuario que se creó previamente

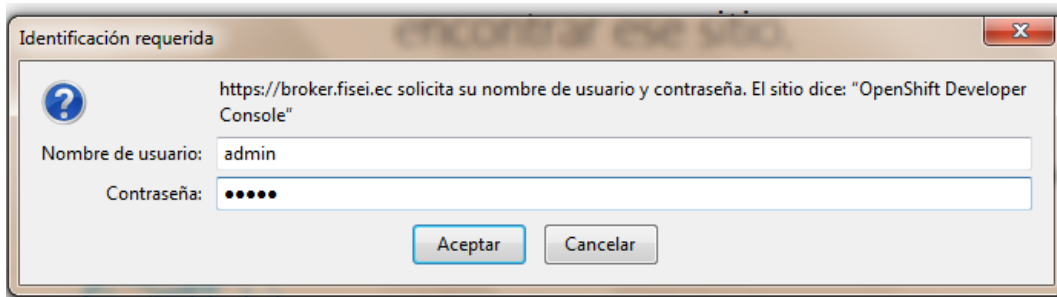


Imagen N.-19 Inicio de sesión

Ingresado usuario y contraseña se desplegará la página principal de openshift origin en la cual se puede verificar el usuario que ingreso o que se encuentra activo.

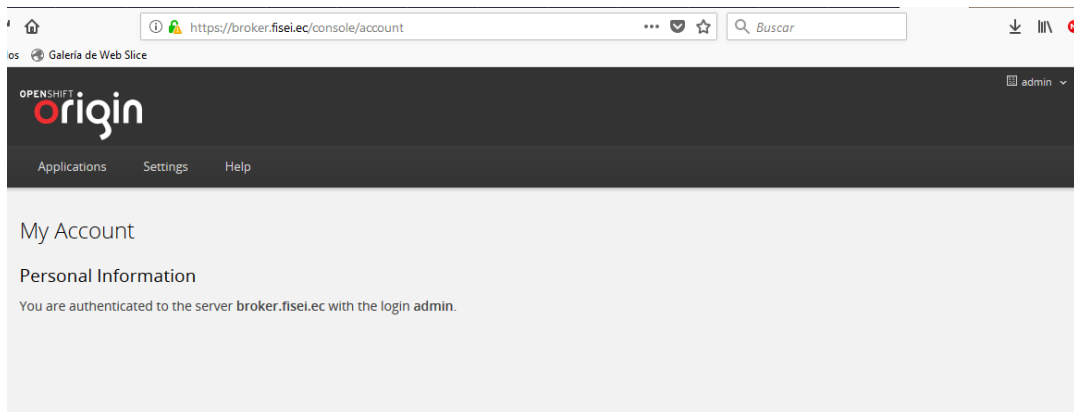


Imagen N.-20 verificación de usuario

Al ingresar en la página principal de OpenShift Origin se procede a escoger la opción **CRÉATE YOUR FIRST APLICACION NOW**

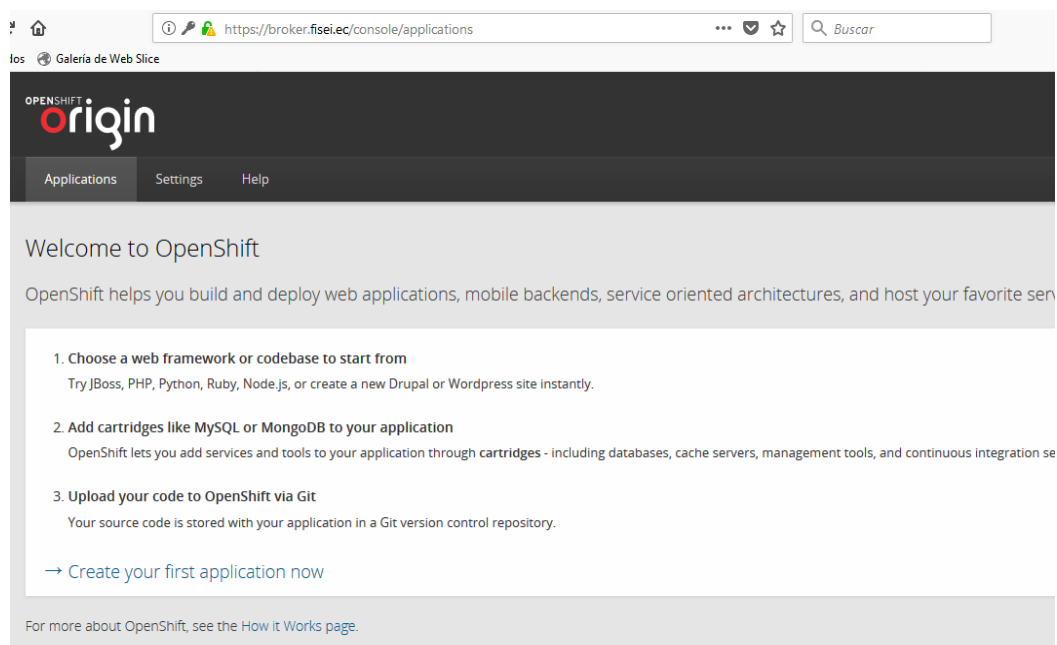


Imagen N.-21 Ventana principal de Openshift Origin

Se desplegará la siguiente ventana en la cual se escogerá el lenguaje de programación, frameworks, sgbds instalados previamente en la imagen N.-12 con el cual se empezara a desarrollar.

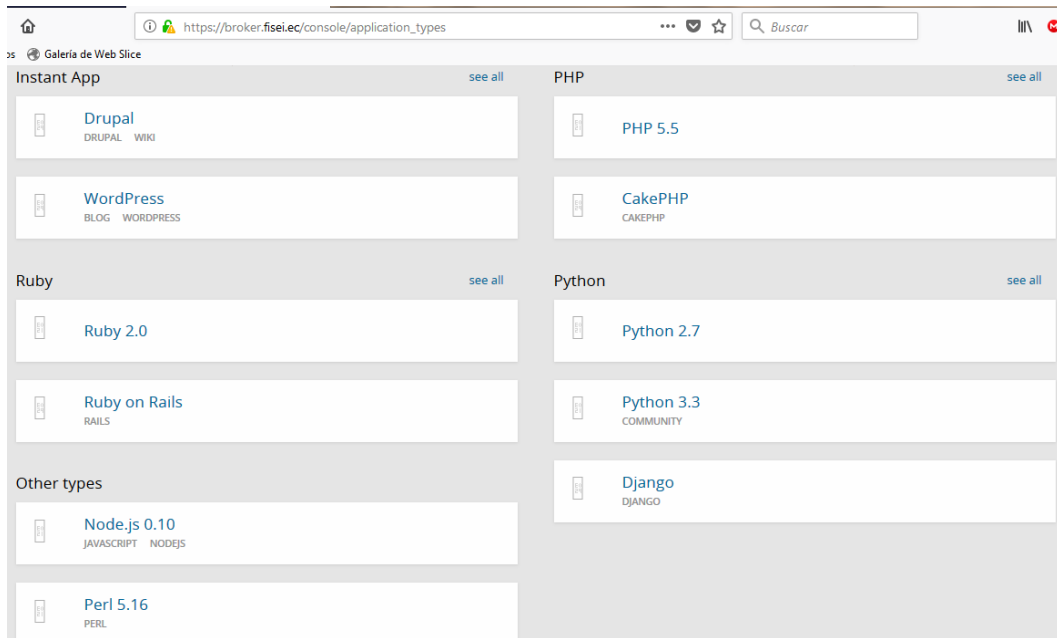


Imagen N.-22 lista de herramientas de desarrollo

Openshift Origin permite crear proyectos en wordPress el cual se realizara un ejemplo, se procede a dar clic en la opción WordPress

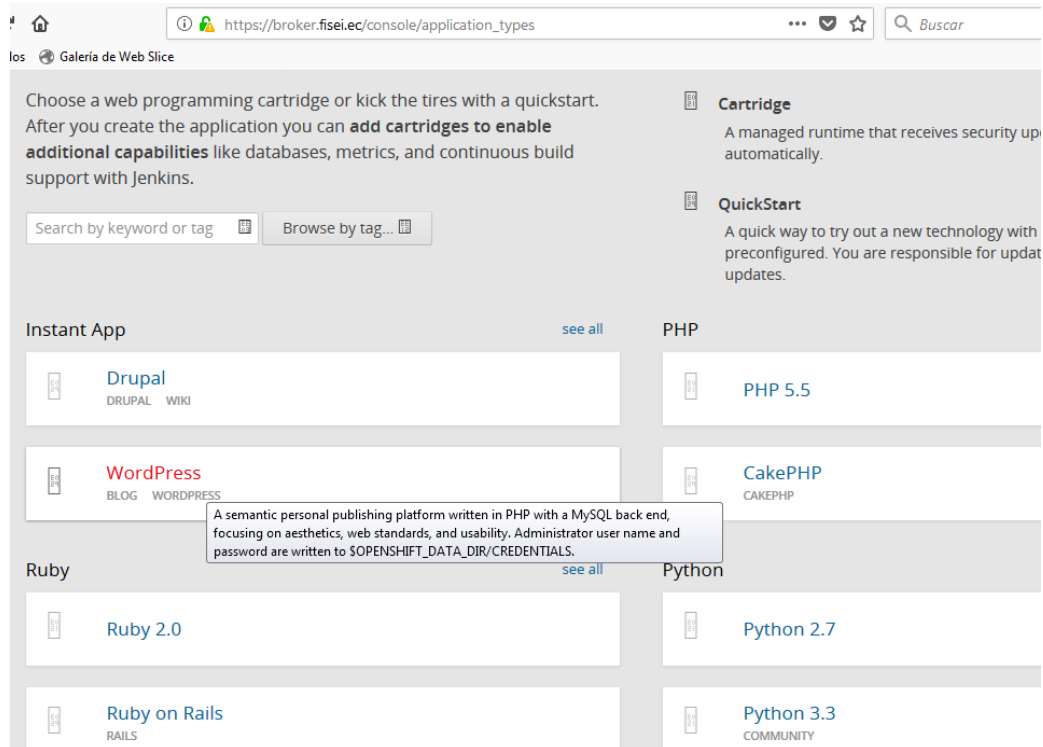


Imagen N.-23 proyecto de wordPress

La cual se desplegará la siguiente ventana en la cual va a pedir que se ingrese el nombre del proyecto y un subdominio el cual se lo ingresa una sola vez en este caso será NOVENO y el nombre del proyecto fiseiweb como se muestra en la pantalla

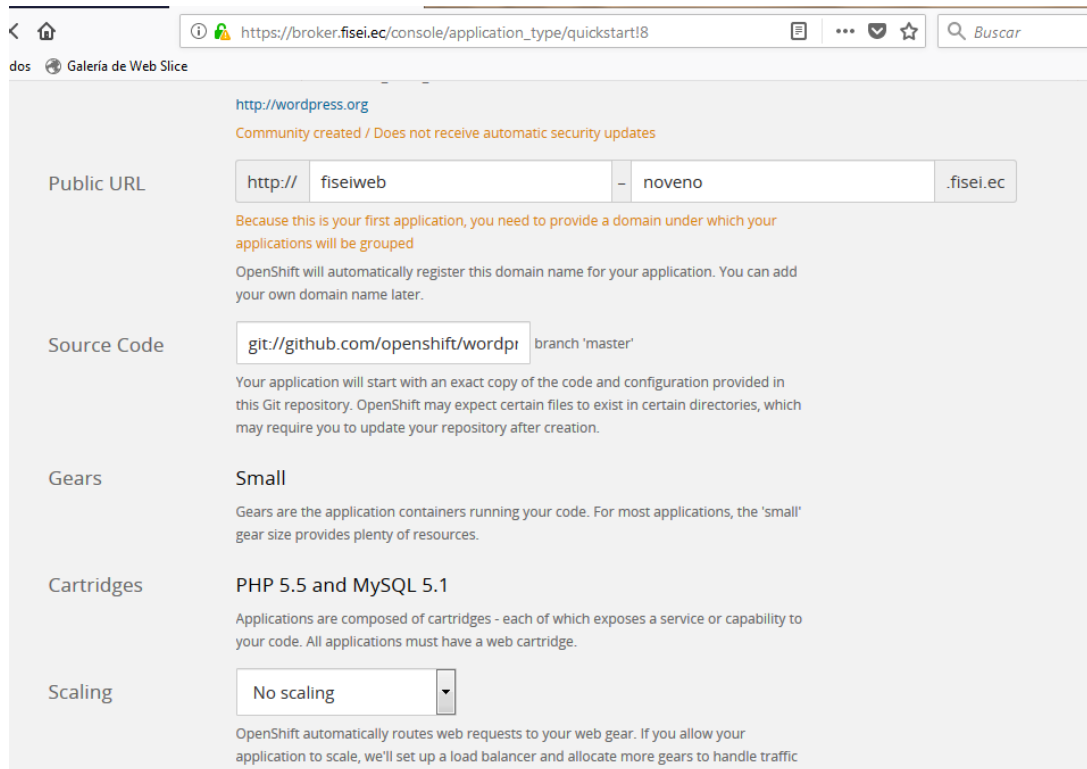


Imagen N.-24 Creación de plantilla WordPress

Se procede a dar clic en `create Application`.

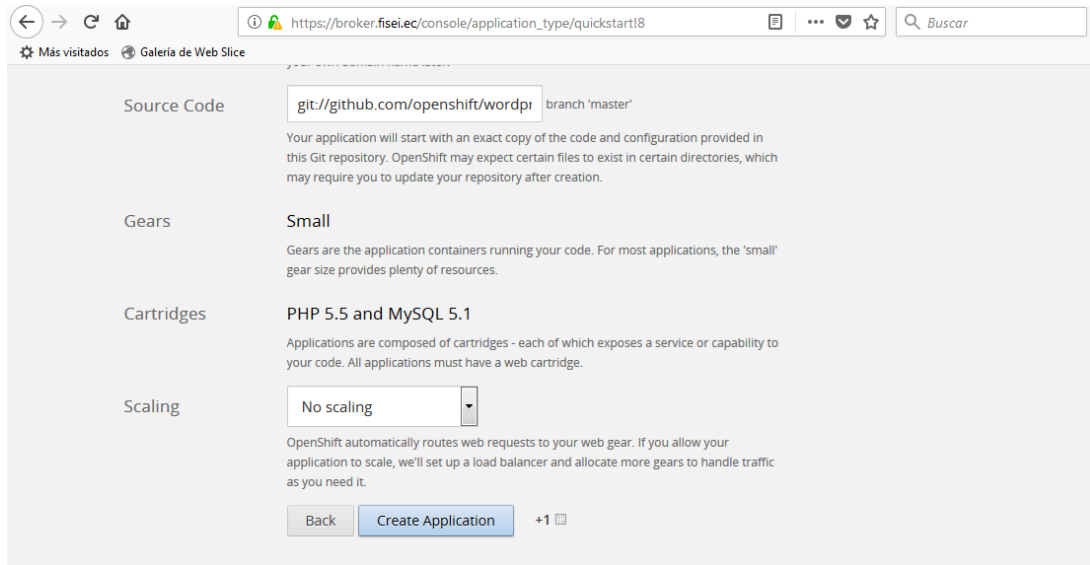


Imagen N.-25 Creación de proyecto WordPress

Se desplegará la siguiente ventana en la cual se procederá a dar clic en Continue to the application overview page

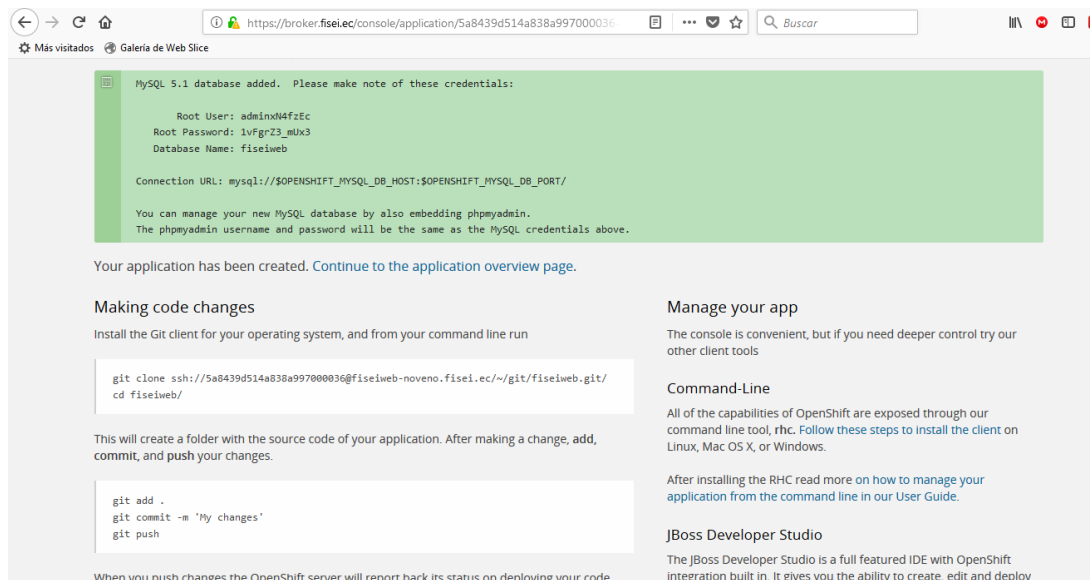


Imagen N.-26 proyecto WordPress

Se desplegará la siguiente ventana se dará clic en el nombre del proyecto

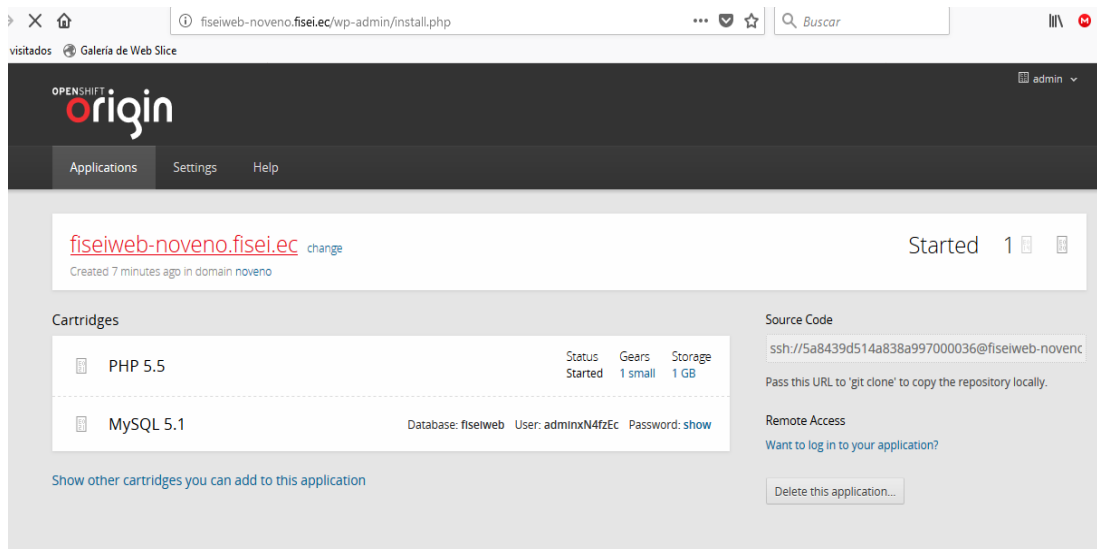


Imagen N.-27 proyecto WordPress paso 2

Esta se dirigirá a la siguiente ventana en la que se seleccionara el idioma con el que se desplegará el proyecto y damos clic en continuar

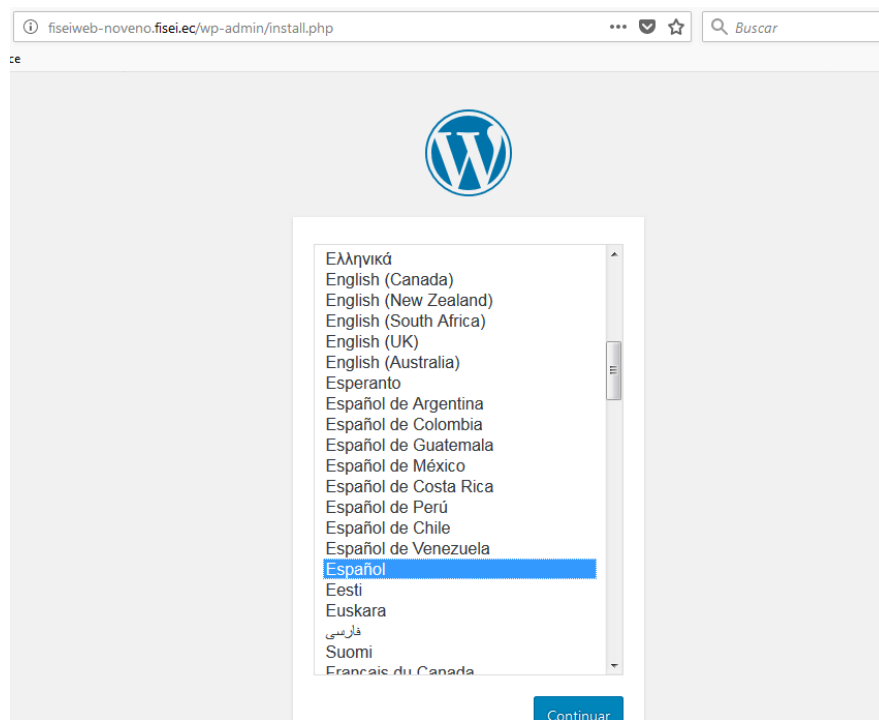
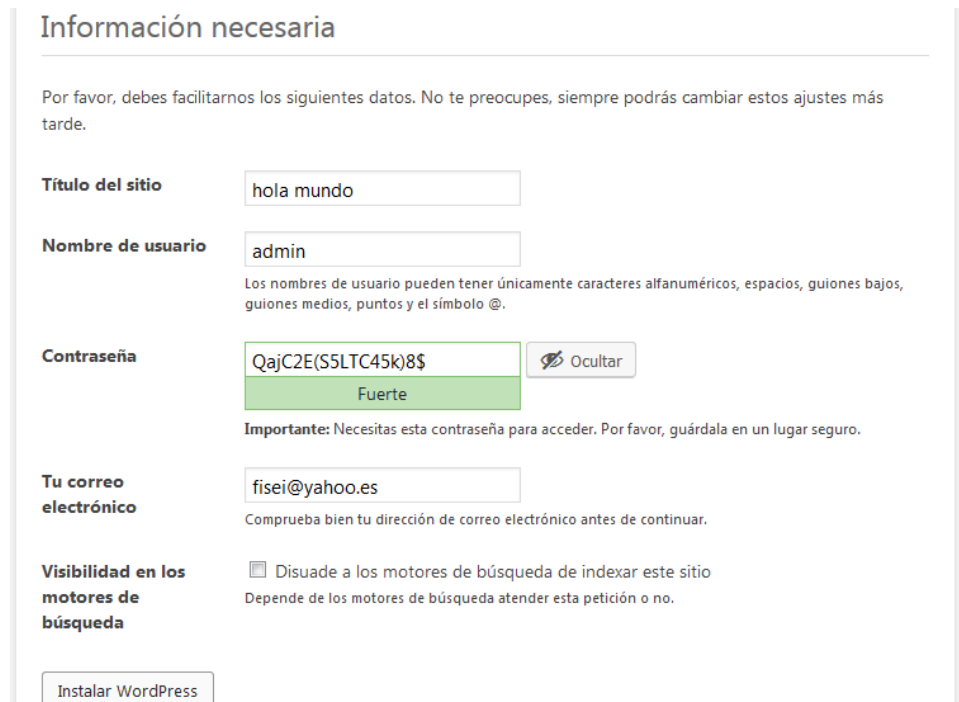


Imagen N.-28 selección de idioma para instalar wordpress

Se desplegará la siguiente ventana en la cual pedirá que se ingrese el nombre de usuario, contraseña y cualquier cuenta de correo, y clic en instalar WordPress.



The screenshot shows the 'Información necesaria' (Information needed) step of the WordPress installation process. It contains the following fields and instructions:

- Título del sitio:** A text input field containing 'hola mundo'.
- Nombre de usuario:** A text input field containing 'admin'. Below it, a note states: 'Los nombres de usuario pueden tener únicamente caracteres alfanuméricos, espacios, guiones bajos, guiones medios, puntos y el símbolo @.'
- Contraseña:** A text input field containing 'QajC2E(S5LTC45k)8\$'. To its right is an 'Ocultar' (Hide) button. Below the field, a green bar indicates the password strength as 'Fuerte' (Strong). Below this, an important note reads: 'Importante: Necesitas esta contraseña para acceder. Por favor, guárdala en un lugar seguro.'
- Tu correo electrónico:** A text input field containing 'fisei@yahoo.es'. Below it, a note says: 'Comprueba bien tu dirección de correo electrónico antes de continuar.'
- Visibilidad en los motores de búsqueda:** A checkbox labeled 'Disuade a los motores de búsqueda de indexar este sitio' (Discourage search engines from indexing this site). Below it, a note says: 'Depende de los motores de búsqueda atender esta petición o no.'

At the bottom of the form is a button labeled 'Instalar WordPress'.

Imagen N.-29 creación de usuario y contraseña WordPress

Se desplegará la siguiente ventana en la que pedirá el ingreso de usuario y contraseña en la que fue creado en el paso anterior

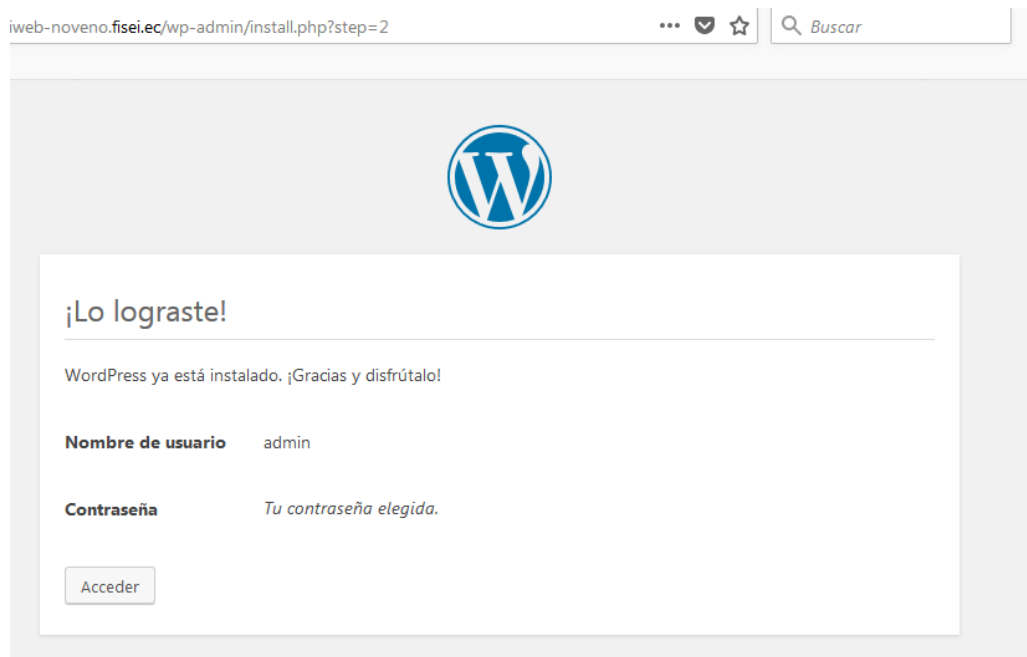


Imagen N.-30 Instalación correcta de WordPress

Se Ingresará con la cuenta de correo y contraseña generada y damos clic en acceder

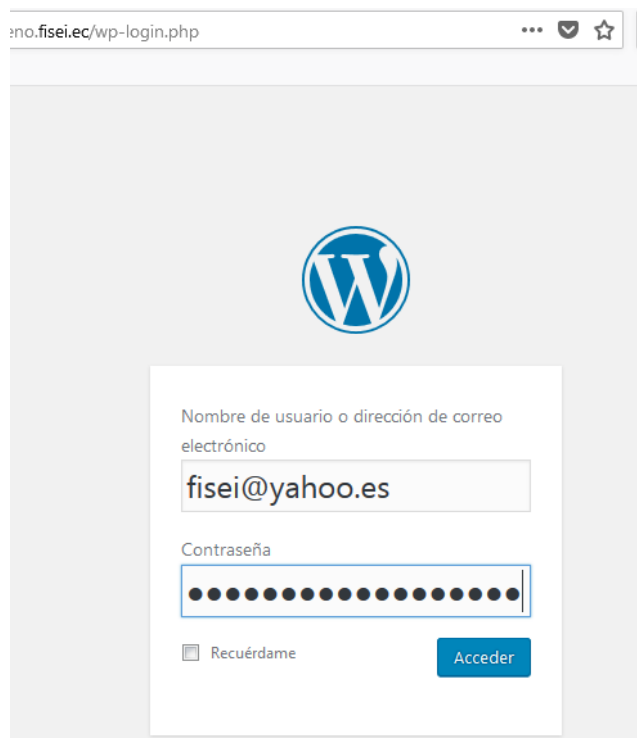


Imagen N.-31 Ingreso a WordPress

La cual dirigirá a la ventana principal de WordPress, en la cual mostrara las herramientas necesarias para el desarrollo de la interfaz.

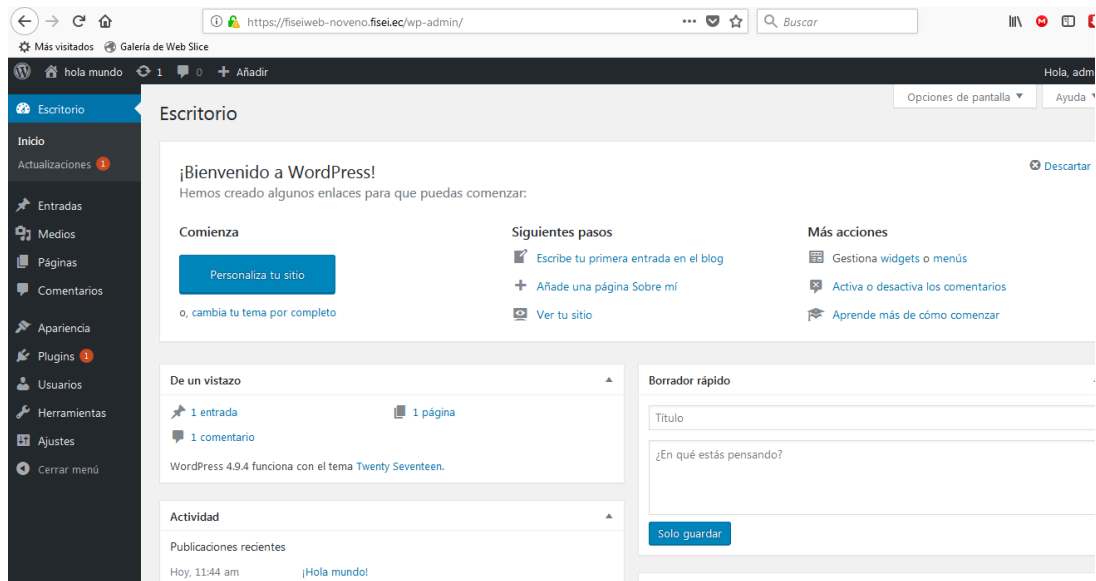


Imagen N.-32 ventana principal de WordPress

Se procede a dar clic en personalizar tu sitio la cual desplegará la siguiente ventana

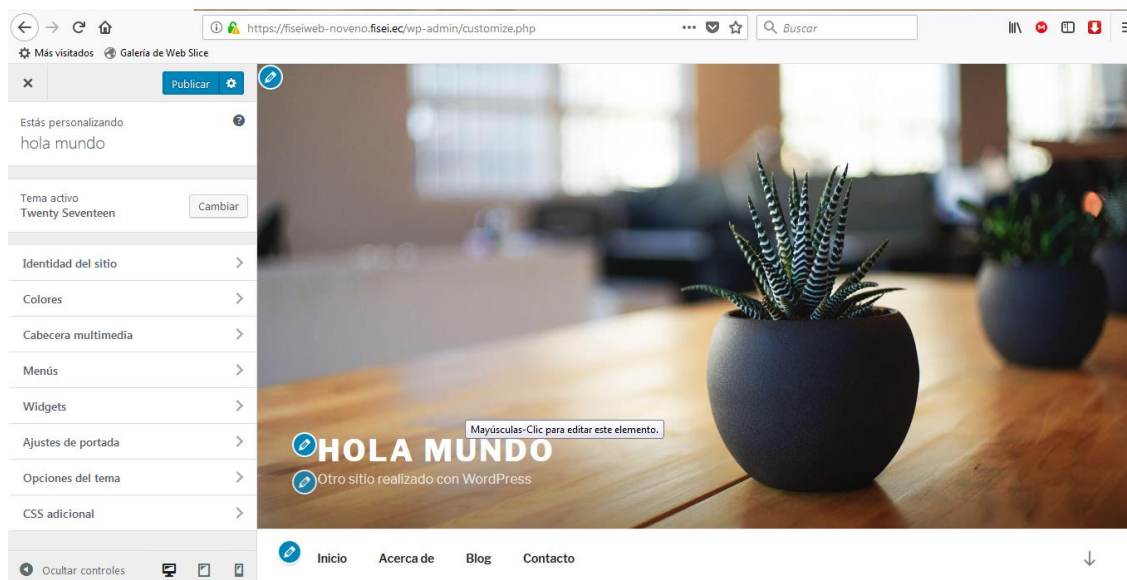


Imagen N.-33 interfaz del proyecto wordPress

La cual se puede modificar con las herramientas presentadas en la parte izquierda de la pantalla, en la cual permite personalizar a gusto del desarrollador.

Luego de acabar con la configuración se da clic en publicar para que se guarden los cambios

Luego se ingresará en el browser la dirección del programa que en este caso será fiseiweb-novenofisei.ec

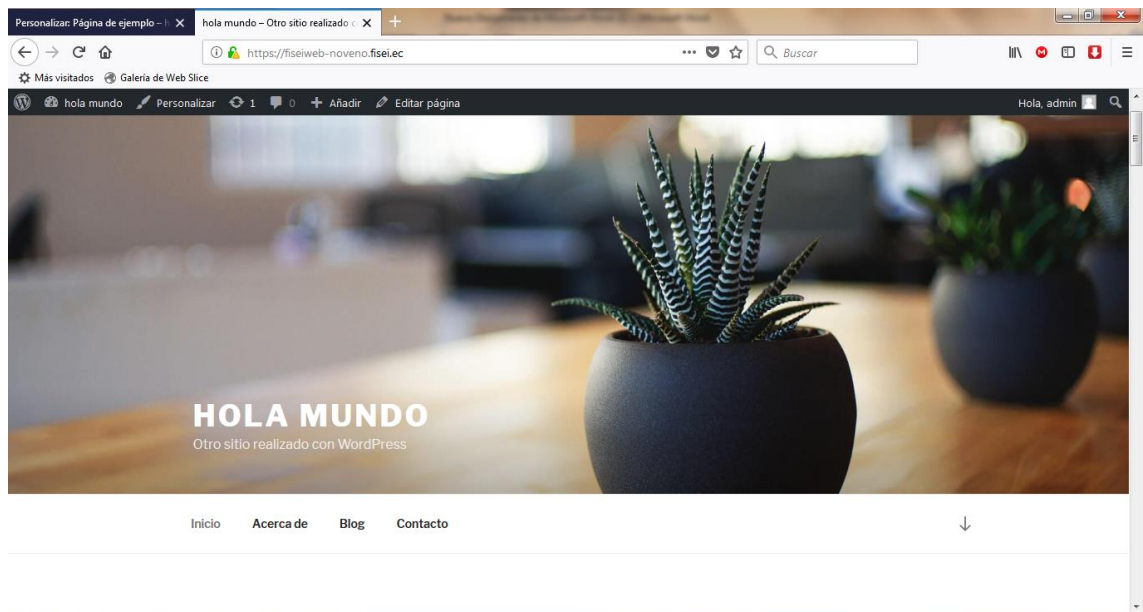


Imagen N.-34 Visualización del proyecto

Creación de aplicaciones con los lenguaje las herramientas ofrecidas por Openshift Origin

Para comenzar ya con el desarrollo de aplicaciones con los lenguajes de programación y base de datos configurados Openshift Origin

Lo primero será instalar git en cada una de las máquinas.

Git, es un software de control de versiones diseñado por Linus Torvalds. Que permite realizar versiones a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo es decir a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración.

Se procederá a descargar la aplicación y se procederá a instalar, se escoge la opción next, next, y finalizar.

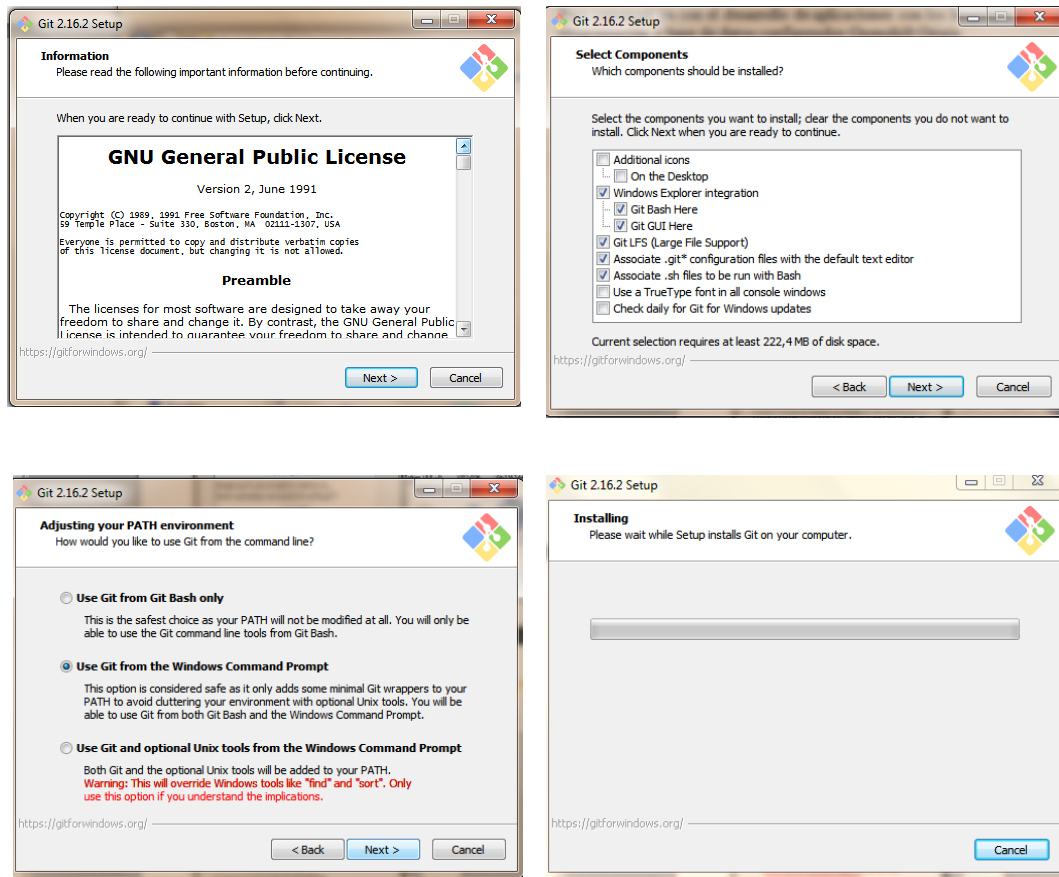


Imagen N.-35 Instalación de Git

Luego de la instalación se ejecuta la aplicación la cual se desplegará la siguiente ventana, en la cual se creará la clave o llave de acceso para integrar el proyecto con Openshift Origin, para lo cual se ingresará el siguiente código ssh-keygen.exe

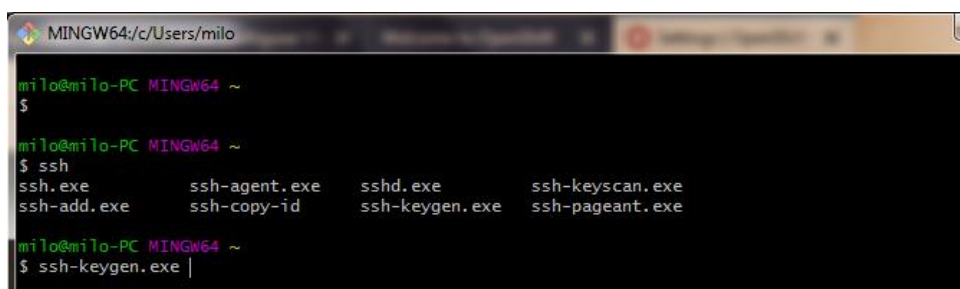


Imagen N.-36 Creación de la llave de acceso

La cual verifica si existe la llave o si se desea crear una nueva y pedirá confirmación para la creación se digitara yes y enter.

```
mi1o@mi1o-PC MINGW64 ~
$ ssh-keygen.exe
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/milo/.ssh/id_rsa):
/c/Users/milo/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/milo/.ssh/id_rsa.
Your public key has been saved in /c/Users/milo/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:5Y+aZ2iQQcH5S1wI174ovwwIQIYL3wXkrUWich1Eeh4 milo@mi1o-PC
The key's randomart image is:
+---[RSA 2048]-----+
|oo.o=*#*=|
|=o o.#0 .|
|oo..o*Eo .|
|o . .#=.o|
|. .o.=S .|
|... . = o|
|. .o . . .|
| o. 000|
| o. .00|
+---[SHA256]-----+
```

Imagen N.-37 Creación de la llave de acceso paso 2

Luego se podrá visualizar la clave con el comando cat id_rsa.pub

```
mi1o@mi1o-PC MINGW64 ~/.ssh
$ ls
id_rsa id_rsa.pub known_hosts

mi1o@mi1o-PC MINGW64 ~/.ssh
$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDLW1AB707pTVIrTv5pes1ZQ3dG6XaVhk6WrxWahKcxvn4UAXXerwDRdfqhAV5m3VDTb
vdsdr9CJkzF8k0pobSBqBg/DqcqzknDb69WUM/QzPxtWkMFgLMsrMrWoXPMJwucBJrFJ+Sueg0E6de61dLh6u705LX2k04H1E10q5kAwd
UFv95T42bmy/ZHBXxkWJZhEXn0VaabX/1YRk7TcHnX1NGC/HBT1Buw4E0bj1LyNPcEiJCeHayxs+T2T7aVP4t6DmNmaCaa/tIJ5sacIO+
iJ/+LhnkYrGYD45kcJk0mdvgRdmZKo2XkVzQdjLiGRGajRH0IJsqdI7nuseeJIZfR milo@mi1o-PC
```

Imagen N.-38 Creación de la llave de acceso paso 3

Luego de generar la clave se copia para luego ser pegada en Openshift Origin como se muestra a continuación

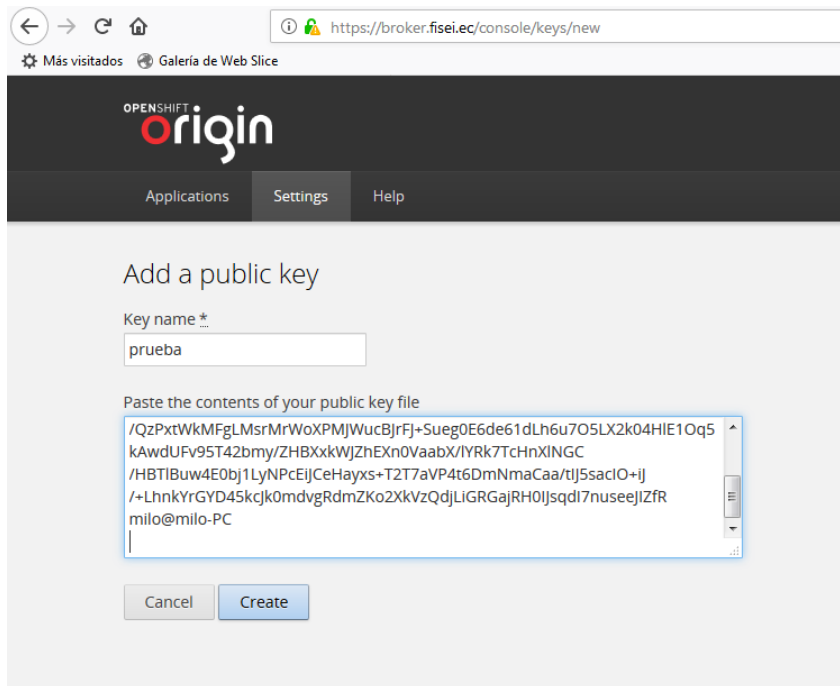


Imagen N.-39 vincular llave entre git y openshift

Se le dará un nombre y clic en crear, se podrá tener la cantidad de llaves que se requiera dependiendo del número de usuarios que necesiten conectarse a Openshift Origin.

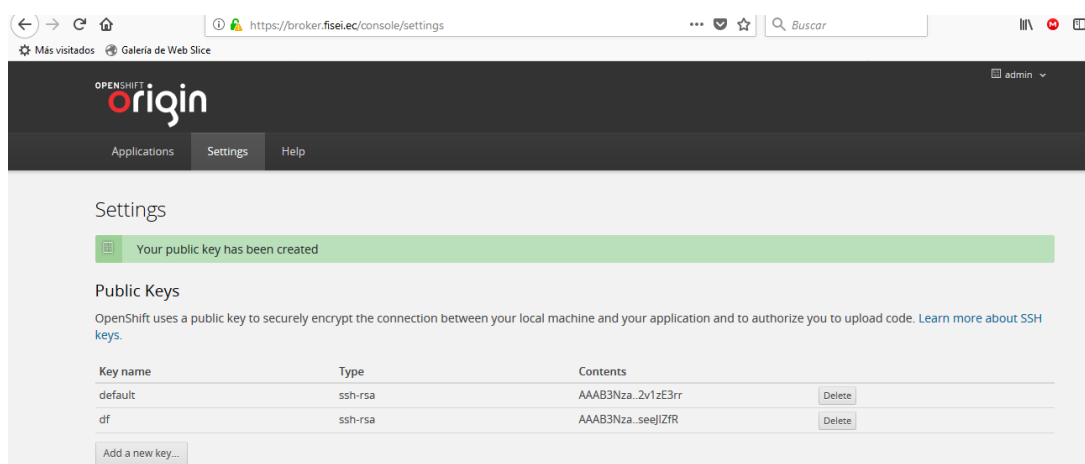


Imagen N.-40 visualización de llaves en openshift

Luego de a ver creado las llaves y estar vinculado entre git y openshift origin se procederá a crear las aplicaciones.

Para esto se realizara una aplicación en php, se escogerá de entre la lista que ofrece openshift origin se dará clic en PHP.5.5

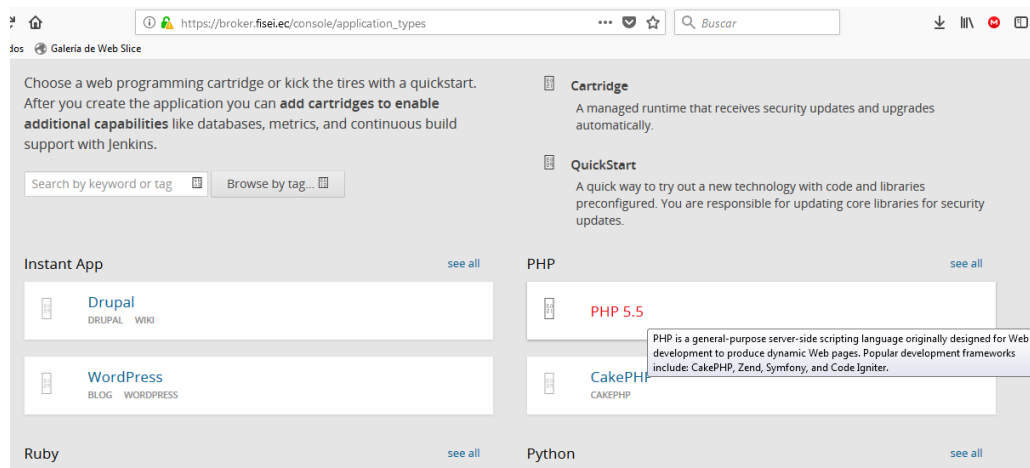


Imagen N.-41 Creación de aplicación PHP

Se procede a agregar un nombre y el subdominio que en este caso será noveno y clic en crear

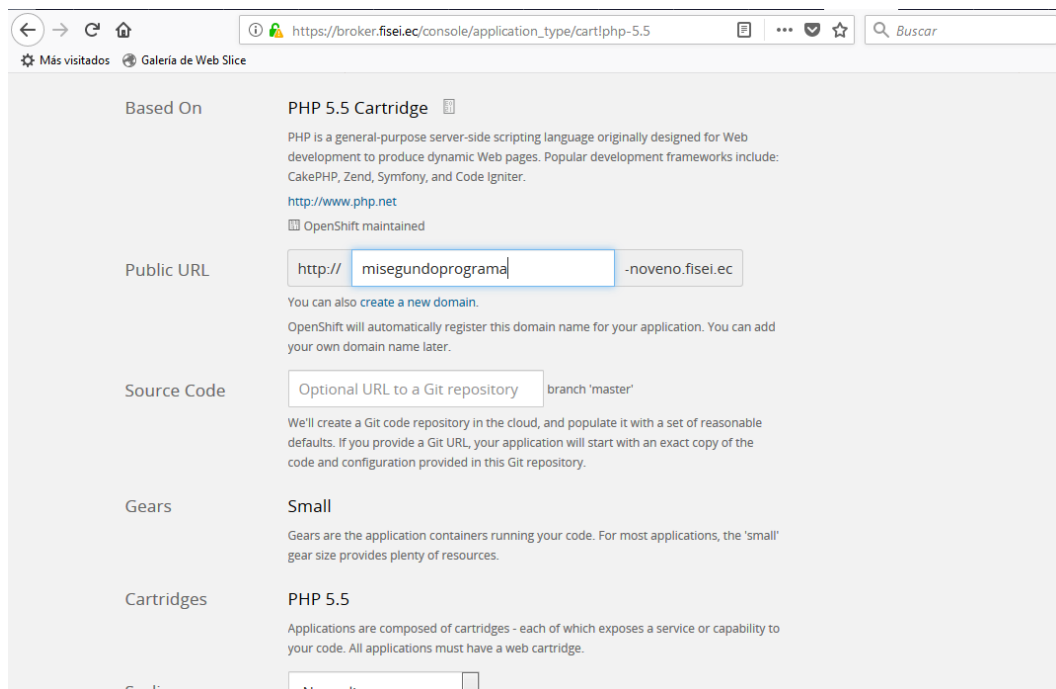


Imagen N.-42 Creación de aplicación PHP paso 2

Luego de creado la aplicación este generara un código el cual será copiado y pegado en la aplicación git

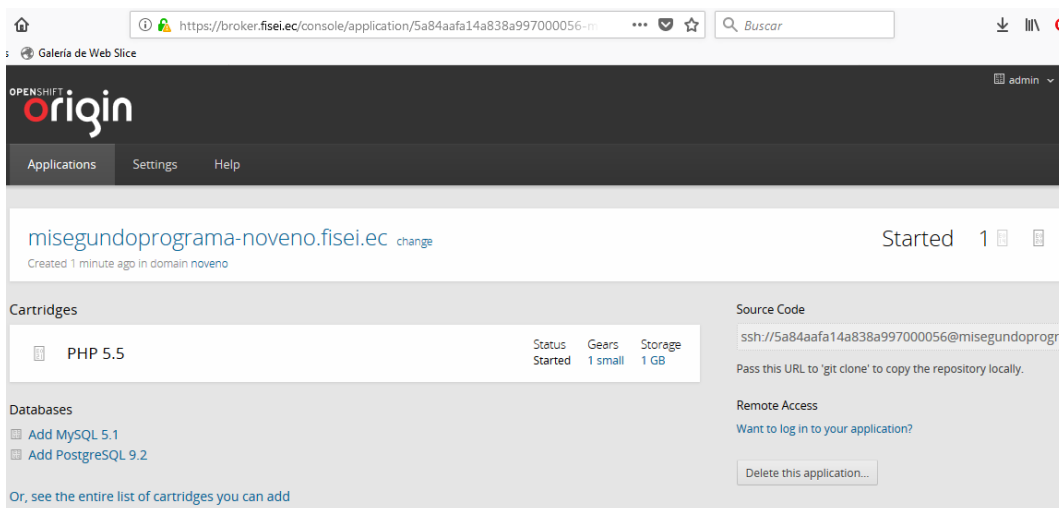


Imagen N.-43 Creación de aplicación PHP paso 3

Con el comando git clone más el código que se generó en Openshift Origin se ejecutara en la aplicación git.

Al ejecutar este comando se está clonando los repositorios de openshift origin a nuestro usuario local para poder comenzar a desarrollar la aplicación y de esta forma poder evitar daños en la aplicación principal del servidor

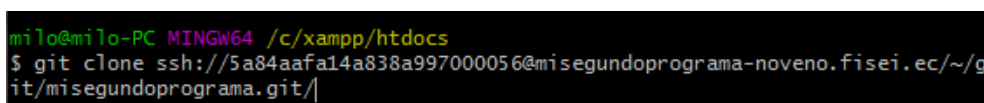


Imagen N.-44 Clonación de repositorios en git

```
Cloning into 'misegundoprograma'...
remote: Counting objects: 4, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 0), reused 0 (delta 0)
Receiving objects: 100% (4/4), 15.70 KiB | 1.21 MiB/s, done.
```

Imagen N.-45 Clonación de repositorios en git paso 2

Una vez realizado la clonación de los repositorios se podrá comprobar que se copiaron ingresando el browser la dirección 127.0.0.1/ misegundoprograma y se desplegará la siguiente ventana.

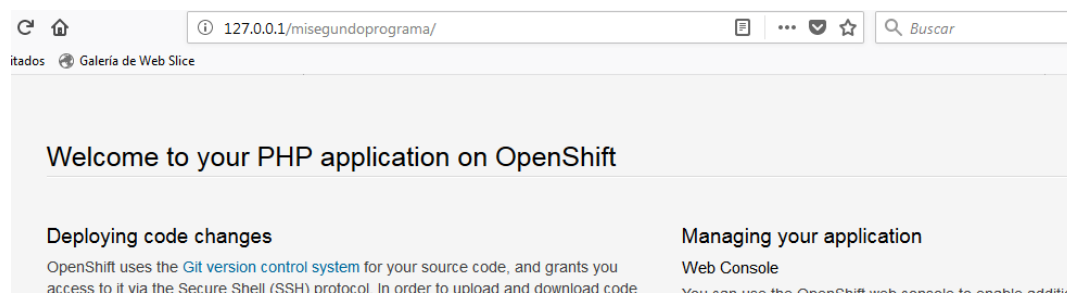


Imagen N.-46 Prueba en el browser de la clonación

Para poder realizar el desarrollo de las aplicaciones lo se realizara desde cualquier editor, para pruebas se realizara en el notepad en la cual se modificara el contenido del archivo index.php que se clono al repositorio.

```

201     font-family: Menlo, Monaco, "Liberation Mono", Consolas, monospace !important;
202 }
203
204 </style>
205
206 </head>
207 <body>
208
209 <section class='container'>
210     <hgroup>
211         <h1>Página modificada por el programador</h1>
212     </hgroup>
213
214
215     <div class="row">
216         <section class='col-xs-12 col-sm-6 col-md-6'>
217             <section>
218                 <h2>Cambios pedidos por el cliente</h2>
219                 <p>OpenShift uses the <a href="http://git-scm.com/">Git vers
220
221                 <h3>Working in your local Git repository</h3>
222                 <p>If you created your application from the command line and
223
224                 <p>If you created the application from the web console, you'
225
226 <pre>$ git clone &lt;git_url&gt; &lt;directory_to_create&gt;;
227
228 # Within your project directory
229 # Commit your changes and push to OpenShift
230
231 $ git commit -a -m 'Some commit message'
232 $ git push</pre>
233

```

Imagen N.-47 edición del fichero index.php

Luego de realizar los cambios y que remos que se reflejen se tendrá que ingresar el comando git init la primera vez que se crea un proyecto.

Con el comando git commit –a permite guardar la versión del proyecto y poner comentario para así llevar un orden cronológico de la edición de las aplicaciones

```

milo@milo-PC MINGW64 /c/xampp/htdocs/misegundoprograma (master)
$ git commit -a
[master ca703ce] Mi primer cambio sobre la plantilla de openshift
1 file changed, 2 insertions(+), 2 deletions(-)

```

```

MINGW64:/c:/xampp/htdocs/misegundoprograma
Mi primer cambio sobre la plantilla de openshift
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
# Your branch is up to date with 'origin/master'.
#
# Changes to be committed:
#   modified:   index.php
#

```

Imagen N.-48 ingreso de comentario para le versión

Y para que los cambios se reflejen en Openshift Origin se ingresa el comando git push

```
milo@milo-PC MINGW64 /c/xampp/htdocs/misegundoprograma (master)
$ git push
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 413 bytes | 59.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
To gitlab.com:31i6or/misegundoprograma.git
 a694883..ca703ce master -> master
```

Imagen N.-49 pruebas comando git push

Se actualiza el browser de Openshift y se mostraran los resultados.

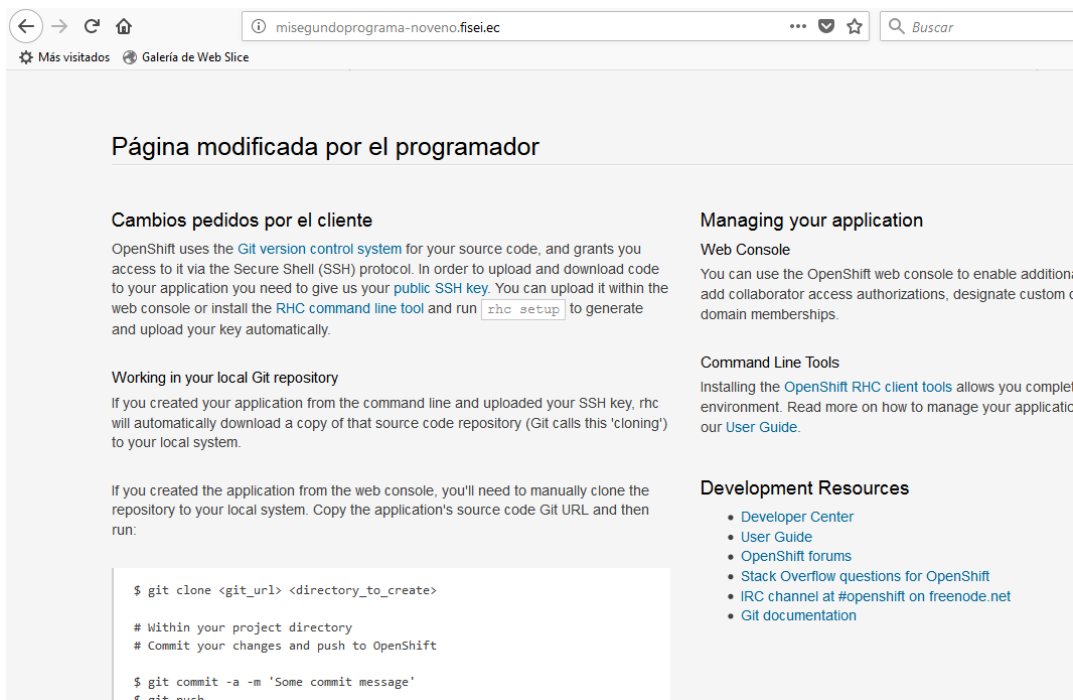


Imagen N.-50 vista de aplicación modificada

CAPITULO V

Conclusiones y Recomendaciones

5.1 Conclusiones

- Con la implementación de PaaS dentro de la Facultad de Ingeniería en Sistemas Electrónica e Industrial, será de gran apoyo para los estudiantes y docentes, ya que permitirá la rápida agregación de complementos, soporte y despliegue e eliminación de las aplicaciones creadas por los usuarios.
- Con la utilización de (PaaS) openshift origin se podrá elegir de entre los diferentes lenguajes de programación, Sgbds y framework con lo cual los estudiantes podrán desarrollar las aplicaciones y realizar aplicaciones simultaneas.
- Con la implementación de openshift origin dentro de la Facultad de Ingeniería en Sistemas Electrónica e Industrial nos permite un ahorro considerable de tiempo en configuración y recursos de hardware y software.
- Al ser (PaaS) un entorno en la nube nos permite compilar, implementar y administrar las aplicaciones en la misma herramienta.

5.2 Recomendaciones

- Se recomienda que al implementar PaaS openshift Origin en la facultad se desarrolle una socialización a los estudiantes y docentes para el correcto aprovechamiento de las herramientas ofrecidas.
- Se recomienda que al instalar los lenguajes de programación, Sgbds y framework se tome en cuenta los que utilizan los estudiantes con el fin de no consumir recursos instalando los paquetes que vienen por defecto en Openshiftf Origin.
- Se recomienda tener cuidado en la asignación de las claves de acceso para la clonación de los repositorios, ya que puede ocasionar problemas de identificación porque los nodos se comunican mediante el dominio.
- Se recomienda que al implementar PaaS se cree varios nodos Openshiftf

Origin ya que estos contienen el soporte de los cartuchos y la administración de todo el cluster, para lograr un balance en el procesamiento de datos y evitar la saturación de almacenamiento, red y procesamiento.

Bibliografía

1. Geek Heterodoxo. “El origen de el computo en la nube ” Internet : <https://www.fayerwayer.com/2012/01/el-origen-de-el-computo-en-la-nube/> ,Enero/06/2012 [Abril/20/2017].
2. Comstor. “Cómo surgió el Cloud Computing” Internet: <http://blogmexico.comstor.com/como-surgio-el-cloud-computing>, Sept./30/2014 [Abril/20/2017].
3. C. M. Gallardo Paredes, «Servicio de cloud computing para la comunicación bilateral en tiempo real, aplicada a brazos robóticos,» Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas, Electrónica e Industrial. Carrera de Ingeniería en Sistemas Computacionales e Informáticos, Ambato, 2016.
4. B. V. Ruiz Quispe, « Implementación de Prototipo de Tecnología de Cloud Computing para Servicios de Infraestructura (IaaS) en la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato,» Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas, Electrónica e Industrial. Carrera de Ingeniería en Sistemas Computacionales e Informáticos, Ambato, 2016.
5. “FAQ de Cloud Computing” Internet: <http://www.saasmania.com/faq-sobre-cloud-computing/> , Junio/13/2010[Abril/20/2017].
6. “Cuál es la diferencia entre un modelo tradicional de computación y una infraestructura en nube?” Internet: http://blog.itechcareer.com/index.php?option=com_content&view=article&id=39:icual-es-la-diferencia-entre-un-modelo-tradicional-de-computacion-y-una-infraestructura-en-nube&catid=17:tecnologia , [Abril/20/2017].
7. Nikolai Aldana “LA NUBE” Internet: http://informatica1bsecu.blogspot.com/2016/04/_,Abril/27/2016 [Abril/20/2017].
8. Uri “Cloud Computing: SaaS, PaaS y IaaS” Internet: <https://computernewage.com/2015/06/07/todo-lo-que-necesitas-saber-sobre-el-cloud-computing-saas-paas-y-iaas/> , Jun./07/215 [Abril/20/2017].
9. Leonardo De Seta “Qué es una Plataforma como Servicio (PaaS)” Internet:

- <https://dosideas.com/noticias/actualidad/504-ique-es-una-plataforma-como-servicio-paas> , Abril/09/2009 [Abril/20/2017].
10. “¿Qué son las bases de datos?” Internet:
<http://www.maestrosdelweb.com/que-son-las-bases-de-datos/>, Oct./26/2007 [Abril/20/2017].
 11. IBM “Bases de datos relacionales”
Internet: https://www.ibm.com/support/knowledgecenter/es/SSEPGG_8.2.0/com.ibm.db2.udb.doc/admin/c0004099.htm , [Abril/25/2017].
 12. “Funcionamiento de los Web services” Internet:
https://josh1982.gitbooks.io/programacion-web-en-cliente/content/concepto_de_framework.html , [Abril/25/2017].
 13. openshift.org “Prerequisites” Internet
https://docs.openshift.org/latest/install_config/install/prerequisites.html, May 2, 2017, [Abril/25/2017].
 14. Microsoft “Requisitos de hardware y software para una implementación distribuida” Internet <https://technet.microsoft.com/es-es/library/dn457749.aspx>, julio de 2016, [Abril/25/2017].
 15. Gonzalo Nazareno “Introducción a PaaS.OpenShift” Internet
<http://iesgn.github.io/cloud/cursos/u9/>, julio de 2017, [enero/25/2018].
 16. Isidro López “Que es la plataforma de Windows azure” Internet
<https://isazure.wordpress.com/2011/04/02/%C2%BFque-es-la-plataforma-de-windows-azure/>, 02/04/2011, [Abril/25/2017].
 17. José David Martín Nieto “Instalar OpenShift Origin en Fedora 19” Internet:
<http://www.dmartin.es/2014/09/instalar-openshift-origin-en-fedora-19-mega-tutorial-parte-1/>, 04/09/2014, [Enero/25/2018].
 18. José David Martín Nieto “Instalar OpenShift Origin en Fedora 19”
Internet: <http://www.dmartin.es/2014/09/instalar-openshift-origin-en-fedora-19-mega-tutorial-parte-2/>, 04/09/2014, [Enero/25/2018].

19. José David Martín Nieto “Instalar OpenShift Origin en Fedora 19”
Internet: <http://www.dmartin.es/2014/09/instalar-openshift-origin-en-fedora-19-mega-tutorial-parte-3/>,04/09/2014, [Enero/25/2018].
20. José David Martín Nieto “Instalar OpenShift Origin en Fedora 19”
Internet: <http://www.dmartin.es/2014/09/instalar-openshift-origin-en-fedora-19-mega-tutorial-parte-4/>,04/09/2014, [Enero/25/2018].
21. José David Martín Nieto “Instalar OpenShift Origin en Fedora 19”
Internet: <http://www.dmartin.es/2014/09/instalar-openshift-origin-en-fedora-19-mega-tutorial-parte-5/>,04/09/2014, [Enero/25/2018].
22. José David Martín Nieto “Instalar OpenShift Origin en Fedora 19”
Internet: <http://www.dmartin.es/2014/09/instalar-openshift-origin-en-fedora-19-mega-tutorial-parte-6/>,04/09/2014, [Enero/25/2018].
23. José David Martín Nieto “Instalar OpenShift Origin en Fedora 19”
Internet: <http://www.dmartin.es/2014/09/instalar-openshift-origin-en-fedora-19-mega-tutorial-parte-7/>,04/09/2014, [Enero/25/2018].
24. José David Martín Nieto “Instalar OpenShift Origin en Fedora 19”
Internet: <http://www.dmartin.es/2014/09/instalar-openshift-origin-en-fedora-19-mega-tutorial-parte-8/>,04/09/2014, [Enero/25/2018].

ANEXO N.-1

Entrevista realizada a la Ing. Cristina Frutos encargada del departamento de administración de redes de la Facultad de Ingeniería en Sistemas Electrónica e Industrial de la Universidad Técnica de Ambato.

1.- ¿Con que infraestructura en hardware cuenta la Facultad de Ingeniería en Sistemas Electrónica e Industrial al momento?

Con un servidor Intel ® xeon de 11 núcleos con memoria ram de 16 gigas y disco duro de 256 gb. Servidor virtualizado con centos 7

2.- ¿Qué sistemas operativos están instalados en los laboratorios de la facultad?

En las computadoras tenemos instalado Windows 10 de 64 bits y en el laboratorio de redes fedora 26.

3.-¿Qué características en hardware tienen las computadoras en los laboratorios de la facultad?

Son computadoras con procesadores I5, I7 con 8GB de memoria ram y un tera en disco duro.

4.-¿Qué tipos de Lenguajes de Programación, IDEs, Frameworks que utilizan los estudiantes de facultad para el desarrollo de aplicaciones?

Depende de los semestres en los que se reciba clases por ejemplo utilizan Java con Netbeans o Eclipse, visual Studio con .NET, PHP con YII regularmente son los que más utilizan.

5.-¿Cuál son las bases de datos más utilizadas por los estudiantes de facultad para el desarrollo de aplicaciones?

Depende de cada ingeniero que da clases por lo general utilizan Oracle, PostgreSQL, MySQL y SqlServer