

INTEGRACIÓN CONTINUA PARA EL DESARROLLO PÁGINA WEB
OFERTAS DE EMPLEO

NESTOR ANTONIO PINEDA OTERO
CARLOS EDUARDO ARIAS FERNANDEZ
LIZETH ALEXANDRA ROMERO CUERVO
ANGIE RUIZ PEÑA

INSTITUCIÓN UNIVERSITARIA POLITÉCNICO GRANCOLOMBIANO
FACULTAD DE INGENIERÍA Y CIENCIAS BÁSICAS
INGENIERÍA DE SOFTWARE - VIRTUAL
2018

**INTEGRACIÓN CONTINUA PARA EL DESARROLLO PÁGINA WEB
OFERTAS DE EMPLEO**

**NESTOR ANTONIO PINEDA OTERO
CARLOS EDUARDO ARIAS FERNANDEZ
LIZETH ALEXANDRA ROMERO CUERVO
ANGIE RUIZ PEÑA**

**PRESENTADO A:
DIEGO IVÁN OLIVEROS ACOSTA**

**INSTITUCIÓN UNIVERSITARIA POLITÉCNICO GRANCOLOMBIANO
FACULTAD DE INGENIERÍA Y CIENCIAS BÁSICAS
INGENIERÍA DE SOFTWARE - VIRTUAL**

2018

CONTENIDO

INTRODUCCIÓN	4
JUSTIFICACIÓN	5
OBJETIVOS	6
Objetivo General	6
Objetivos Específicos	6
MARCO TEÓRICO	7
CREACIÓN Y VINCULACIÓN DE CONTENEDORES DOCKER	10
BIBLIOGRAFÍA	12

INTRODUCCIÓN

En la actualidad los portales web se han convertido en sitios muy importantes usados para dar a conocer y encontrar todo tipo de información, el Portal Web para ofertas de empleo será una página dedicada especialmente al registro de ofertas de empleo vigentes con su respectivo detalle, se busca con esta solución abarcar todas las necesidades de las personas que están en búsqueda de empleo.

Con base al material de estudio del módulo de Integración continua, este portal web es susceptible para la aplicación de estos conceptos ya que el desarrollo va a ser por pequeñas tareas realizando la comprobación del código de cada uno de los integrantes del equipo de trabajo con el fin de compactar cada artefacto para la toma de decisiones ya sea para la construcción, mejora y/o detección de errores lo más frecuente posible.

JUSTIFICACIÓN

La integración continua es un concepto realizable con el fin de mejorar la calidad de software en los ámbitos de proceso, producto y equipo; para el portal web de ofertas de empleo se van a utilizar las herramientas como lo es Github, Docker, Jenkins, Travis CI y Codeship debidamente anidadas para cumplir el reto de hacer las integraciones, iteraciones, detección de errores y mejora continua del software con el fin de satisfacer las necesidades del cliente y/o usuario generando seguridad en el equipo de trabajo.

OBJETIVOS

Objetivo General

Aplicar los conocimientos de integración continua para el portal web de ofertas de empleo.

Objetivos Específicos

- Manejo básico de las herramientas Github, Docker, Jenkins, Travis CI y Codeship para el desarrollo y verificación del portal web de ofertas de empleo.
- Realizar la integración de cada uno de los artefactos entregados por cada integrante del equipo utilizando las herramientas establecidas.
- Identificar riesgos y oportunidad de mejora en cada integración para la toma de decisiones de cada escenario que se presente.

MARCO TEÓRICO

¿Qué es Docker? Esta herramienta permite crear contenedores (caja que permite almacenar información, programas o aplicaciones) con la finalidad de poder guardar los ejecutables de una aplicación y poder ejecutarla a través de Docker sin necesidad de instalarla localmente o de tener una versión específica por aplicación.

Al instalar Docker es posible ejecutar una aplicación software desde el contenedor de Docker, y dentro de él estarán todas las librerías y cosas que necesita dicha aplicación para funcionar correctamente.

Es posible compartir el contenedor para generar trabajos en aplicaciones de programación como java; anteriormente era necesario contar con las mismas versiones de cada herramienta para poder ejecutarla, por medio de Docker solo es necesario compartir el contenedor, abrirlo y ejecutarlo.

Una de las funciones más evidente, es posible encapsular todas las herramientas y ejecutables necesarios para una aplicación de esta forma el programador tendrá la seguridad de estar trabajando desde su servidor local, así mismo al momento de ejecutar los cambios la configuración se realiza en el mismo entorno en el que se han hecho todas las pruebas.

Ventajas y funciones de Docker

Virtualizar con Docker nos ofrece una serie de ventajas respecto a hacerlo con máquinas virtuales convencionales:

Portabilidad: Todos los contenedores creados en docker son portables, por esto es posible llevarlos fácilmente de un computador a otro sin necesidad de descargar el programa o aplicación que se vaya a utilizar.

Ligereza: Al no virtualizar un sistema completo, sino solo lo necesario, el consumo de recursos es mínimo. El ahorro de recursos es de alrededor de un 80%.

Autosuficiencia: Por medio de Docker es posible realizar casi todo, por esto los contenedores solo requieren el mínimo de información, por ejemplo, aquellas librerías, archivos y configuraciones necesarias para poder realizar su función.

Un sistema de contenedores Docker se compone principalmente de 5 elementos:

Demonio: Es el proceso principal de la plataforma.

Cliente: Es el binario que constituye la interfaz y que permite al usuario interactuar con el Demonio.

Imagen: Plantilla utilizada para crear el contenedor para la aplicación que queremos ejecutar.

Registros: Directorios donde se almacenan las imágenes, tanto de acceso público como privado.

Contenedores: Carpetas donde se almacena todo lo necesario (librerías, dependencias, binarios, etc) para que la aplicación pueda ejecutarse de forma aislada.

CREACIÓN Y VINCULACIÓN DE CONTENEDORES DOCKER

Para el desarrollo del sitio web creamos dos contenedores y los vinculamos mediante la línea de comandos.

Iniciamos con el contenedor encargado del manejo de la base de datos basado en MySQL, el cual se crea con el siguiente comando y en el mismo definimos la contraseña del usuario ROOT lo cual es obligatorio para la correcta instalación del módulo:

“docker run --name servidor_mysql -e MYSQL_ROOT_PASSWORD=123asd -d mysql”

El cual nos arroja la siguiente respuesta:

```
C:\>docker run --name servidor_mysql -e MYSQL_ROOT_PASSWORD=123asd -d mysql
Unable to find image 'mysql:latest' locally
latest: Pulling from library/mysql
f2aa67a397c4: Pull complete
1accf44cb7e0: Pull complete
2d830ea9fa68: Pull complete
740584693b89: Pull complete
4d620357ec48: Pull complete
ac3b7158d73d: Pull complete
a48d784ee503: Pull complete
f122eadb2640: Pull complete
3df40c552a96: Pull complete
da7d77a8ed28: Pull complete
f03c5af3b206: Pull complete
54dd1949fa0f: Pull complete
Digest: sha256:d60c13a2bfdbbeb9cf1c84fd3cb0a1577b2bbaeec11e44bf345f4da90586e9e1
Status: Downloaded newer image for mysql:latest
5af6e40678747626f0e65ac2e97b94fadf753422575b01a024141f04a490cd62
```

Luego creamos el segundo contenedor el cual proveerá el servidor http, en este comando definimos el puerto de escucha del contenedor, puerto 80 y el puerto del host, puerto 8081 y lo enlazamos mediante el parámetro **--link** al contenedor previamente creado para el manejo de las bases de datos:

“docker run --name servidor_httpd -p 8001:80 --link servidor_mysql:mysql -d httpd”

El cual nos arroja la siguiente respuesta:

```
C:\>docker run --name servidor_httpd -p 8001:80 --link servidor_mysql:mysql -d httpd
Unable to find image 'httpd:latest' locally
latest: Pulling from library/httpd
3d77ce4481b1: Pull complete
73674f4d9403: Pull complete
d266646f40bd: Pull complete
ce7b0dda0c9f: Pull complete
01729050d692: Pull complete
014246127c67: Pull complete
7cd2e04cf570: Pull complete
Digest: sha256:72f2b4aa99235509146bd12054d1a93c1c869ba60212d21729118c93ca4305d3
Status: Downloaded newer image for httpd:latest
ceeb4cf116a85f60190e95a753af946b91356c2177613452916b73026765150c
```

Verificamos ahora que los dos contenedores hayan quedado vinculados con el siguiente

comando:

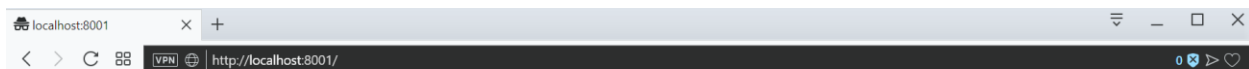
“docker inspect -f "{{ .HostConfig.Links }}" servidor_httpd”

con la siguiente respuesta verificando la vinculación:

```
C:\>docker inspect -f "{{ .HostConfig.Links }}" servidor_httpd
[/servidor_mysql:/servidor_httpd/mysql]
```

Finalmente comprobamos que el servidor http este habilitado en la máquina local y en el puerto

8081 como lo definimos anteriormente:



It works!

BIBLIOGRAFÍA

<https://www.redeszone.net/2016/02/24/docker-funciona-la-virtualizacion-contenedores/>

<http://www.javiergarzas.com/2015/07/que-es-docker-sencillo.html>