

Experiencia en implementación de un proceso de despliegue continuo en un sistema basado en Wordpress

Heraldo Valenti

heraldovalenti@gmail.com

Laboratorio de Investigación en Ingeniería y Calidad de Software

<http://www.institucional.frc.utn.edu.ar/sistemas/lidicalso/>

Departamento de Ing. en Sistemas de Información

Universidad Tecnológica Nacional

Maestro M. López esq. Cruz Roja Argentina

(X50165ZAA) Ciudad Universitaria, Córdoba, Argentina

Resumen: El desarrollo que ha tenido internet en parte puede atribuirse a la generación de contenido por parte de los usuarios. La publicación de dicho contenido no hubiera sido posible sin la existencia de herramientas para facilitar dicha actividad, entre las que se encuentra Wordpress, la que mayor porción del mercado ocupa, no solo por su facilidad de uso para la gestión de contenido sino por su extensibilidad, que brinda un framework para el desarrollo de sistemas de negocio. En estos casos, resulta de vital importancia adoptar prácticas de ingeniería que apunten a asegurar la calidad de los mismos, sobre todo cuando la responsabilidad de llevar adelante tales desarrollos recae sobre un equipo de varias personas y no solo una. En este trabajo se expondrán los requerimientos para un desarrollo como el descrito, las dificultades que devienen de utilizar Wordpress como framework y las soluciones adoptadas para cada una de éstas.

Palabras claves: wordpress, docker, gestión de la configuración, integración continua, despliegue continuo.

1. Introducción

1.1. Wordpress como framework

Este trabajo se llevó adelante en el contexto de un proyecto el cual tiene como objetivo la implementación de un sistema web basado en Wordpress [1] para una conocida revista del medio local.

Una de las principales necesidades que recaen sobre el sistema es la capacidad de gestionar el contenido multimedia [2] generado por la organización, de la cual surge la decisión de utilizar un Sistema de Gestión de Contenido (CMS por sus siglas en

inglés) [3], sistemas cuyo principal objetivo es brindar la capacidad de aplicar principios de gestión al contenido de artículos [3]. La decisión de utilizar Wordpress [1] como framework [4] surge a partir de las principales alternativas disponibles en cuanto a productos CMS [3]. Wordpress [1] resulta una de las más utilizadas [5], con lo que se obtiene la ventaja de una gran cantidad de desarrolladores disponibles en el mercado capacitados en la herramienta. Además, Wordpress [1] cuenta con una gran cantidad de extensiones (plugins) desarrollados por terceros disponibles al público para ser reutilizados [3] [6]. Por otra parte, otro requerimiento funcional importante del sistema es la de permitir la publicación de servicios de proveedores que brinden la posibilidad a los usuarios navegar a través de ellos, consultar sus detalles específicos, revisar experiencias de otros usuarios y contactar al proveedor del servicio y posteriormente, contratar dichos servicios. A partir de este requerimiento es que se refuerza la decisión de seleccionar Wordpress [1], ya que entre los plugins que se encuentran disponibles para ser re-utilizados se encuentran numerosas opciones que apuntan a resolver esta necesidad, particularmente los plugins Listify [7], el cual permite la gestión de anuncios, y el plugin WooCommerce [8] que brinda funcionalidad de comercio electrónico.

1.2. Metodología de trabajo

El equipo encargado de llevar adelante el desarrollo de producto se constituyó por 5 personas que cumplieron los roles principales de gestor de proyecto, desarrollador y encargados de pruebas. Para organizar el equipo, y dado el tamaño del mismo, se decidió llevar adelante la gestión de las actividades bajo un marco de trabajo basado en Scrum [9], además de que la empresa cuenta con experiencia en la metodología ya que la mayoría de los proyectos que se desarrollan en la misma se llevan adelante siguiendo esta metodología.

Con la decisión de adoptar Scrum [9] lo que la empresa busca principalmente es lograr la entrega de software funcionando con un grado de calidad suficiente que permita cumplir las expectativas del cliente y lograr la satisfacción del mismo, tal y como se menciona en el manifiesto ágil [10].

1.3. Gestión de la configuración

La experiencia en proyectos anteriores de desarrollo de software en la empresa ha demostrado que una de las prácticas indispensables para lograr los objetivos de los mismos es la de Gestión de la Configuración (SCM por sus siglas en inglés), tal y como ya se menciona en el modelo de CCMi [11]. Por este motivo, uno de los objetivos primordiales a la hora de comenzar un nuevo proyecto es la definición de un plan de SCM, incluyendo las herramientas que se van a utilizar como soporte, y contribuir así a un producto con una calidad aceptable.

Dentro de las actividades específicas definidas en CMMI para SCM se encuentran las de:

- Identificación de los elementos de configuración.
- Control de cambios sobre los elementos de configuración.
- Definición de líneas base.

Como resultado de la identificación de los elementos de configuración, y dada la arquitectura de Wordpress [12], se pueden mencionar dos elementos principales que deben ser puestos bajo la gestión de configuración en un sistema basado en Wordpress [1]: el código fuente de la aplicación y su base de datos.

Para lograr la definición de líneas base de los elementos de configuración, es necesario llevar adelante el versionado de los mismos como parte del control de cambios de los mismos. Por un lado, el código fuente de la aplicación se constituye en gran parte por archivos de código PHP [13] dado que Wordpress [1] se encuentra implementado en este lenguaje. El versionado de estos elementos de configuración resulta trivial con el uso de herramientas versionado de código fuente como Git [14]. Sin embargo, la otra gran parte de los elementos de configuración que constituyen el producto se encuentran almacenados en la base de datos, MySQL [15] particularmente cuando se trata de Wordpress [1], con lo cual resulta necesario encontrar una forma adecuada de gestionar dicha configuración dentro de la base de datos.

1.4. Despliegue continuo

Dado que la entrega de software funcionando como principio resulta uno de los conductores en una metodología ágil [10], se debe prestar especial atención al despliegue de la aplicación, ya que constituye la actividad que fin al proceso iterativo y resulta crucial su éxito para lograr el objetivo de la iteración. La estrategia que se suele adoptar para realizar el despliegue es la de realizar el despliegue manualmente una vez finalizado el ciclo de desarrollo, lo cual constituye una mala práctica y resulta en potenciales riesgos que pueden causar demoras en los plazos de entrega [16]. Esto se potencia aún más en metodologías ágiles que en las cuales frecuentemente, típicamente entre cada 2 o 4 semanas, se genera una nueva versión del producto que debe ser desplegada.

Una práctica que se puede adoptar para mitigar esta problemática es la de despliegue continuo, la cual se sustenta en los principios de integración continua, y consiste básicamente en que cada cambio introducido en el código fuente de la aplicación dispara el ciclo de retroalimentación y conocer con de forma temprana los problemas que se pueden haber introducido como consecuencia de ese cambio [16].

2. Dificultades que presenta Wordpress

2.1. Datos de configuración en la base de datos

En primer lugar, es necesario establecer la diferencia entre los datos de la aplicación y los datos de configuración. Los datos de la aplicación son aquellos que son generados como resultado de la operación de las funcionalidades que brinda el sistema al usuario final, mientras que los datos de configuración son aquellos que determinan o modifican de alguna forma el comportamiento del sistema frente a determinada funcionalidad [16]. Cuando se trata de datos de configuración, resulta necesario controlar los cambios en ellos; por otro lado, el control de cambios no es necesario cuando se trata de datos de aplicación, al menos en nuestro caso de estudio, ya que pueden ser gestionados adecuadamente por los procedimientos de copia de respaldo y recuperación. Además, una consecuencia directa de versionar los datos de aplicación, es que obtiene una versión de los mismos por cada ambiente en el cual el sistema funcione, ya que cada ambiente tiene su propio estado de datos de acuerdo a las acciones que se fueran realizando en éste, lo cual resulta en la adición de complejidad para SCM.

El principal desafío con respecto a SCM cuando se utiliza Wordpress [1] como framework [4] para el desarrollo de un sistema web, resulta el hecho de que muchas configuraciones son almacenadas dentro de la base de datos, lo cual requiere adoptar alguna estrategia para extraer dichos datos desde la base de datos a un formato que permita su correcto versionado.

La alternativa directa es extraerlos estos datos desde la base de datos en forma de archivos de consultas SQL, los cuales pueden ser directamente gestionados por cualquier sistema de versionado de código fuente, como por ejemplo Git [14].

Para la adopción de la estrategia descrita, es necesario identificar en primer lugar las tablas en las cuales dichos datos de configuración son almacenados, ya que algunos los plugins que se pueden instalar en el sistema utilizar diversas tablas. A raíz de esto, se pueden identificar principalmente los siguientes casos:

- Tablas que contienen solamente datos de configuración, las cuales pueden ser versionadas directamente, tanto su estructura como sus datos. El caso más común es el de aquellas tablas que son creadas por los plugins de Wordpress y que almacenan la configuración de los mismos.
- Tablas que contienen datos de configuración y datos de aplicación, lo que implica la necesidad de separar unos de otros, ya que no es necesario que los datos de aplicación sean puestos bajo versionado.

El primer caso resulta relativamente sencillo de resolver, ya que basta con utilizar las utilidades de extracción provistas por la base de datos MySQL [15], tanto de la estructura como de los datos almacenados en cada una de estas tablas.

El segundo caso resulta un poco más complejo ya que requiere adoptar una estrategia que permita lograr la separación de los datos de aplicación de los datos de configuración.

2.2. Reproducción de ambientes

Una de las grandes ventajas que tiene Wordpress [1] es que con pocas configuraciones se puede obtener un ambiente totalmente funcional que sirve para comenzar a desarrollar. Esta ventaja en la simplicidad de crear un nuevo ambiente resulta también una desventaja, ya que parte de su configuración inicial es la de establecer el nombre del sitio, que incluye el nombre de dominio del servidor donde se ejecuta, con lo que resulta necesario al momento de cambiar de ambiente (por ejemplo, de desarrollo a pruebas) adecuar esta configuración con el nombre de dominio del nuevo servidor. Para lograr esto, es necesario llevar a cabo un procedimiento de renombrado del sitio, que, a pesar de ser posible de automatizar, el mismo muchas veces suele ser insuficiente [17]. Por lo tanto, es necesario apoyarse en una completa gestión de la configuración del sitio para poder identificar todos los cambios necesarios para lograr un procedimiento de renombrado correcto.

Este problema resulta un inconveniente a la hora de mantener varios ambientes de la aplicación, principalmente frente al hecho de que no es conveniente compartir un ambiente entre varios desarrolladores, sino que cada desarrollador debería realizar sus actividades en un ambiente totalmente aislado de los demás desarrolladores para evitar perder cambios realizados por otro desarrollador debido a sobre escrituras [16].

Para resolver esta problemática existen algunas alternativas, como el plugin Versionpress [18] disponible para Wordpress [1], pero el mismo no soporta algunas funcionalidades deseadas, como una configuración multi-sitio [19]. Además, durante las pruebas de concepto realizadas se comprobó que este plugin a veces resulta insuficiente en cuanto al versionado de los datos de configuración almacenados en la base de datos, resultando en la pérdida de datos de configuración en el despliegue de la aplicación en un nuevo ambiente. Además, adoptando esta estrategia resulta dificultosa la gestión de las líneas base de la aplicación dada la falta de un mecanismo de empaquetamiento que permita la distribución de la aplicación de manera uniforme entre ambientes, tal como se aborda en la siguiente sección.

2.3. Distribución de líneas base

Para lograr una definición completa de una línea base y poder realizar una distribución de la misma, una estrategia frecuentemente adoptada es la de generar un paquete auto contenido, el cual contiene todos los datos necesarios para instalar la aplicación, lo cual incluye, además de los binarios que constituyen la aplicación, todas sus configuraciones de ambiente [16].

Para adoptar la estrategia anteriormente descrita, y dado que Wordpress [1] está implementado en lenguaje de programación PHP [13], el cual es un lenguaje de scripting no compilado que además no provee mecanismos de empaquetamiento de los archivos de código fuente, resulta necesario adoptar una estrategia para poder realizar el empaquetado de los mismos.

3. Solución implementada

La solución implementada brinda un proceso de despliegue automatizado que facilitan tanto las actividades de desarrollo como las de las entregas incrementales del software al cliente.

El procedimiento completo puede visualizarse en la Figura 1. Los componentes principales de la solución son los siguientes:

- Un repositorio de código fuente Git para gestionar el versionado de los archivos de código fuente de la aplicación y sus configuraciones.
- Una herramienta implementada que permite automatizar la gestión de datos de configuración almacenados en la base de datos automáticamente.
- Un conjunto de scripts Bash para automatizar la ejecución de las tareas de versionado de configuración de la aplicación y su eventual despliegue.
- Imágenes de Docker que permiten la reproducción automática de ambientes y el empaquetado de la aplicación para facilitar la distribución de líneas base.
- Un conjunto de Jobs de Jenkins que conducen el despliegue de la aplicación en los distintos ambientes de forma automática.

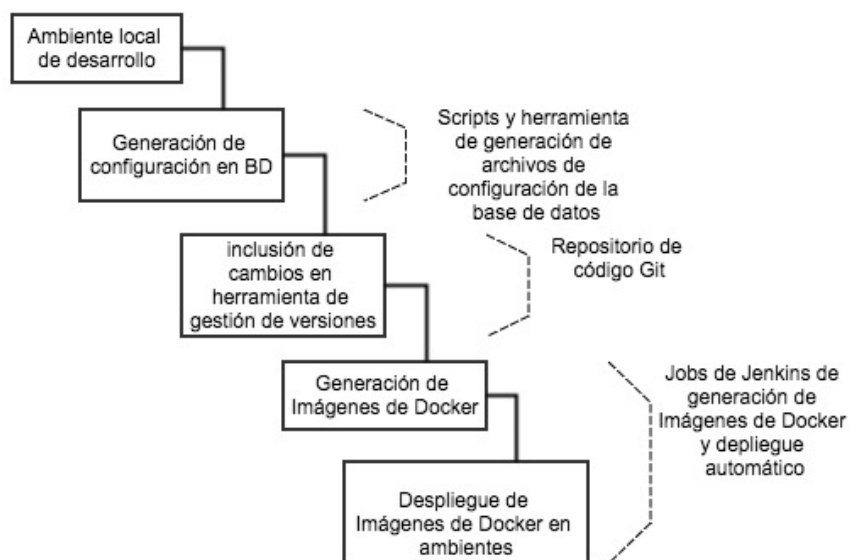


Figura 1- Proceso implementado

3.1. Herramienta de gestión de datos en la base de datos

Se trata de una herramienta implementada en lenguaje Java [20] sobre el framework Spring [21] para cubrir las necesidades planteadas que realiza dos actividades principales de forma automática: la extracción de los datos de configuración desde la base de datos hacia archivos de texto plano que pueden ser versionados como código fuente y la instalación de dichos datos cuando se realiza el despliegue de la aplicación.

Para la primera actividad, la funcionalidad desarrollada consiste en:

1. Mantener un registro de todas las tablas existentes en la base de datos y una estrategia de gestión para cada una de ellas. Este registro es implementado actualmente como un archivo en formato JSON [22] para facilitar su configuración, el cual puede ser a su vez versionado con Git [14]. Las estrategias para cada tabla pueden ser:
 - a. Versionado completo: se genera un archivo de estructura de la tabla de la base de datos y por otro lado un archivo con cada una de las filas que en ella existen.
 - b. Versionado parcial: al igual que la anterior, se genera un archivo con la estructura de la tabla, pero el archivo de datos contenidos en ella se realiza de forma parcial, seleccionando aquellos datos que cumplen con un criterio especificado para la tabla.
 - c. Estructural: solo se genera un archivo con la estructura de la tabla.
 - d. Ninguno: utilizado para indicar aquellas tablas que no son necesarias de versionar ni su estructura ni sus datos.
2. De acuerdo al registro de cada una de las tablas, se generan los archivos de estructura y de datos para cada una de ellas según la estrategia indicada.

De esta manera, la herramienta permita la gestión de los datos de configuración almacenados en la base de datos. La estrategia de “Versionado parcial” permite definir un criterio de selección de datos de la tabla para separar los datos de aplicación de los datos de configuración.

La segunda actividad para el despliegue de la aplicación en un nuevo ambiente consiste en cargar la estructura de cada una de las tablas definida en el archivo de estructura de la tabla y la posterior carga de los datos versionados en el archivo de datos.

3.2. Scripts de soporte para la gestión de la configuración y de despliegue

Estos scripts se encuentran implementados en lenguaje de scripting Bash [23]. Estos scripts permiten la ejecución automática de las tareas de versionado de los datos de configuración en la base de datos y el despliegue de la aplicación junto con dicha configuración. En la actualidad, los scripts Bash [23] pueden ser ejecutados en

sistemas operativos Windows, Linux o MacOS, lo cual brinda independencia en cuanto al sistema operativo, permitiendo al desarrollador seleccionar el que más le resulte conveniente.

Adicionalmente, estos scripts incluyen el procedimiento de renombrado del sitio para el despliegue de la aplicación en distintos ambientes, lo cual permite al desarrollador tener un ambiente completo localmente que puede utilizar para llevar adelante sus actividades de forma aislada, sin interrumpir el trabajo de otros desarrolladores.

Una vez que el desarrollador finaliza sus actividades y debe incluir sus cambios en el sistema de versionado Git [14], puede ejecutar el script de versionado de datos de configuración de la base de datos, lo cual genera los archivos en texto plano de dichas configuraciones y pueden ser incluidos en el sistema de versionado de igual manera que los archivos de código fuente PHP [13] del sitio.

3.3. Empaquetado mediante imágenes de Docker

Docker es una plataforma que permite la creación, ejecución y gestión de ambientes aislados para las aplicaciones de software, conocido generalmente como “conteneirización”, que permite el empaquetado de las aplicaciones y todas sus dependencias en Imágenes de Docker [24].

La arquitectura de Wordpress [12] incluye principalmente un servidor web capaz de interpretar código PHP [13] y una base de datos SQL [12]. Una de las implementaciones más comunes involucra un servidor web Apache httpd [25] y una base de datos MySQL [15].

Por otro lado, una de las herramientas provistas por Docker es el DockerHub [26], el cual contiene numerosas imágenes de Docker bien definidas de aplicaciones y mantenidas por los proveedores oficiales de dichas aplicaciones. Entre estas imágenes de Docker disponibles para ser utilizadas se encuentran las imágenes correspondientes al servidor de Wordpress [27] y de MySQL [28]. A partir de estas imágenes base, se generan 3 nuevas imágenes:

1. Una para el sitio Wordpress, que incluye los archivos PHP [13] que constituyen el sitio.
2. Una para la base de datos MySQL, a la cual se agregan los archivos de configuración necesarios para el despliegue.
3. Una imagen que incluye: las configuraciones del sitio; la herramienta de gestión de datos de la base de datos implementada para instalar automáticamente las configuraciones del sitio; y los scripts de soporte que realizar la instalación de las configuraciones de forma automática.

Las imágenes de Docker generadas son identificadas unívocamente por etiquetas, permitiendo distinguir cada una de las versiones de los componentes para mantener la trazabilidad. Estas imágenes pueden utilizarse para instalar y ejecutar de forma

completa la aplicación, lo que permite considerarlas los entregables de cada versión del producto.

Al finalizar cada Sprint, se seleccionan las versiones de las imágenes que han sido certificadas por el plan de pruebas de la iteración y se realiza la entrega y despliegue de cada una de ellas.

3.4. Despliegue automático mediante Jobs de Jenkins

El despliegue de la aplicación es conducido por Jobs¹ de Jenkins [29]. Estos Jobs realizan la instalación de las imágenes de Docker en los distintos ambientes y ejecutan los scripts contenidos en ellas para la correcta configuración del sitio de acuerdo al ambiente en el cual se realiza el despliegue, de forma independiente si se trata de un ambiente de desarrollo, pruebas o producción.

Adicionalmente, como cada línea base es totalmente auto contenida, es posible realizar el despliegue de una versión anterior en caso de que ocurra algún error, de forma tal que se minimiza el riesgo en el despliegue de la aplicación asegurando un sistema funcional en la mayor parte del tiempo.

4. Conclusiones

El procedimiento implementado permitió resolver las dificultades inherentes de utilizar Wordpress como framework para el desarrollo del sistema, a la vez de mantener las ventajas brindadas por el mismo. Principalmente, se consiguió la posibilidad de brindar un ambiente aislado para cada desarrollador de forma tal de no interrumpir las actividades de los demás miembros del equipo de desarrollo. Por otra parte, se consigue un procedimiento que permite la generación automática de líneas base y el despliegue automático de las mismas en el ambiente deseado, lo cual permite realizar la entrega al cliente de un producto correcto que incluye toda la funcionalidad desarrollada en cada iteración mitigando los riesgos que implican realizar estas actividades de forma manual, con lo cual se consiguen los objetivos planteados de entregar software funcionando al cliente para lograr la satisfacción del mismo.

5. Referencias

- [1] Wordpress, «Wordpress,» [En línea]. Available: <https://es.wordpress.org/>. [Último acceso: 30 April 2017].

¹ Un Job es la unidad de trabajo que provee Jenkins para la automatización de procedimientos.

- [2] wikipedia, «Multipedia,» [En línea]. Available: <https://es.wikipedia.org/wiki/Multimedia>. [Último acceso: 30 April 2017].
- [3] V. R. y. S. P. Savan K Patel, «Joomla, Drupal and WordPress - A Statistical Comparison of Open Source CMS,» IEEE, 2011.
- [4] I. Sommerville, «Application Frameworks,» de *Software Engineering*, 9th ed., vol. 16, Pearson, Ed., Addison-Wesley, 2011, pp. 431-434.
- [5] managewp.com, «More Surprising Statistics About WordPress Usage,» 28 April 2016. [En línea]. Available: <https://managewp.com/statistics-about-wordpress-usage>. [Último acceso: 11 April 2017].
- [6] Wordpress, «Plugins extend and expand the functionality of Wordpress,» [En línea]. Available: <https://es-ar.wordpress.org/plugins/>. [Último acceso: 30 April 2017].
- [7] Astoundify, «Listify Theme Documentation,» [En línea]. Available: <http://listify.astoundify.com/>. [Último acceso: 2017 April 2017].
- [8] WooCommerce, «The best eCommerce Platform for Wordpress,» [En línea]. Available: <https://woocommerce.com/>. [Último acceso: 30 April 2017].
- [9] M. Cohn, «Scrum Methodology and Project Management,» 2012. [En línea]. Available: <https://www.mountaingoatsoftware.com/agile/scrum>. [Último acceso: 30 April 2017].
- [10] K. Beck, M. Beedle, A. v. Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland y D. Thomas, «Manifiesto for agile software development,» 2001.
- [11] CMMI Product Development Team, «CMMI® for Development, Version 1.3,» Software Engineering Process Management Program, 2010.
- [12] Wordpress, «Installing Wordpress,» [En línea]. Available: https://codex.wordpress.org/Installing_WordPress#Detailed_Instructions. [Último acceso: 30 April 2017].
- [13] PHP, «Hipertext Preprocessor,» [En línea]. Available: <http://php.net/>. [Último acceso: 30 April 2017].
- [14] Git, «Git,» 30 April 2017. [En línea]. Available: <https://git-scm.com/>.
- [15] Oracle, [En línea]. Available: <https://www.mysql.com/>. [Último acceso: 30 April 2017].
- [16] J. Humble y D. Farley, Continuous Delivery, Addison-Wesley, Ed., 2011.
- [17] Wordpress, [En línea]. Available: https://codex.wordpress.org/Changing_The_Site_URL. [Último acceso: 30 April 2017].

- [18] Versionpress, «Git for Wordpress sites,» [En línea]. Available: <https://versionpress.net/>. [Último acceso: 30 April 2017].
- [19] Wordpress, «Create a Network,» 30 April 2017. [En línea]. Available: https://codex.wordpress.org/Create_A_Network.
- [20] Oracle, «java.com,» 30 April 2017. [En línea]. Available: <https://www.java.com/es/>.
- [21] Pivotal Software, «Spring,» [En línea]. Available: <https://spring.io/>. [Último acceso: 30 April 2017].
- [22] ecma-international, [En línea]. Available: <http://www.json.org/json-es.html>. [Último acceso: 30 April 2017].
- [23] Free Software Foundation, «Bash - GNU Project,» [En línea]. Available: <https://www.gnu.org/software/bash/>. [Último acceso: 30 April 2017].
- [24] C. S. Moreno, H. R. Valenti, D. M. Rubio y C. J. Gonzalez, «Experiencias de Implementación de Despliegue Continuo con una Infraestructura,» *IV CONGRESO NACIONAL DE INGENIERÍA EN INFORMÁTICA/SISTEMAS DE INFORMACIÓN*, 2016.
- [25] Apache Software Foundation, [En línea]. Available: <https://httpd.apache.org/>. [Último acceso: 30 April 2017].
- [26] Docker Hub, [En línea]. Available: <https://hub.docker.com/>. [Último acceso: 30 April 2017].
- [27] Docker Hub, «library/wordpress,» [En línea]. Available: https://hub.docker.com/_/wordpress/. [Último acceso: 30 April 2017].
- [28] Docker Hub, «library/mysql,» [En línea]. Available: https://hub.docker.com/_/mysql/. [Último acceso: 30 April 2017].
- [29] Jenkins, «Jenkins,» [En línea]. Available: <https://jenkins.io/>. [Último acceso: 30 April 2017].