



**UNIVERSIDAD
DE ANTIOQUIA**

Trámites y Certificados

Autor(es)

Edison Bedoya García

Universidad de Antioquia

Ingeniería de Sistemas

Medellín, Colombia

2021



Trámites y Certificados

Edison Bedoya García

Informe de práctica presentado como requisito parcial para optar al título de:

Ingeniero de Sistemas

Asesores (a):

Deisy Loaiza Berrío - Ingeniera de Sistemas

Dennis Alexandra Marín Marín Ingeniera de Sistemas- Especialista en Telemática

Universidad de Antioquia

Ingeniería de Sistemas

Medellín, Colombia

2021

Trámites

Resumen

La Universidad de Antioquia posee un conjunto de aplicaciones web y web services que se encuentran en el ambiente productivo y hacen parte del conglomerado de herramientas informáticas para las labores administrativas que se llevan a cabo dentro de esta entidad pública. La problemática se manifestó cuando Oracle dejó de brindar soporte a su Oracle Application Server (OAS), desde ese momento se optó por otro proveedor en este caso IBM con su Websphere Application Server (WAS).

Este proyecto se centró en la migración de aplicaciones legacy a versiones soportadas por el nuevo servidor, específicamente la aplicación de Trámites y Certificados por parte de los arquitectos encargados de las mismas.

Se logró completar la migración de un total de 42 vistas en un 90% dentro del tiempo estipulado para la práctica empresarial, equivalente a 6 meses calendario, a medida que avanzaba la migración el sistema de control de versionamiento Gitlab sirvió para agregar una capa de seguridad en lo concerniente al almacenamiento del código fuente, además se alimentó el documento de lecciones aprendidas el cual es un insumo de gran importancia para futuros desarrollos, debido a que optimiza el tiempo de desarrollo al tener soluciones a errores que ya se han presentado, finalmente se desplegó la aplicación Trámites en los ambientes de desarrollo y pruebas dentro del WAS.

Introducción

La problemática que se plantea parte de que un grupo de aplicaciones propiedad de la Universidad de Antioquia, que se encuentran desplegadas dentro de un servidor Oracle Application Server, han dejado de tener soporte por parte del proveedor; a partir de esto se contacta con un nuevo proveedor, en este caso IBM, el cual brinda el WebSphere Application Server.

Las aplicaciones que carecen de soporte hacen parte de un conglomerado de web services y aplicaciones web que son esenciales dentro de la estructura administrativa de la Universidad de Antioquia, porque permiten realizar la gestión de diversos servicios que ofrece esta entidad educativa a las personas que tienen alguna relación con la universidad y a externos.

Si bien la idea inicial consiste en migrar las aplicaciones, también se puede dar el caso en el cual se requiere que sean construidas desde cero, esta decisión es tomada por los arquitectos teniendo en cuenta ciertas variables, luego de esto el proyecto se asigna a un integrante del equipo de la capa web, el cual será el encargado de realizar la implementación.

Una vez asignada la aplicación por parte de los arquitectos a cargo, el paso a seguir es realizar la integración con Maven, posteriormente es necesario actualizar algunas librerías para que el proyecto compile en el servidor Apache Tomcat (ambiente local) y así poder corregir bugs que surgen producto de las nuevas versiones de algunas librerías, que son imprescindibles para poder realizar de manera exitosa el despliegue en el servidor WAS. A medida que se avanza con el proceso de migración, son realizadas reuniones de seguimiento no superiores a 30 minutos una vez por semana con el arquitecto a cargo, donde se discute lo que se avanzó, estado actual y objetivos para la semana siguiente.

En el transcurso de esta práctica empresarial se aplican diversos conocimientos adquiridos en el campo de la ingeniería de sistemas como lo son las buenas prácticas de programación, administración de software y lógica de programación con el fin de tener las aplicaciones ejecutándose en el ambiente productivo y que puedan ser accedidas por la estructura administrativa de la universidad y terceros, que requieran los servicios brindados por estas.

Objetivo general

Migrar aplicación Trámites alojada en Oracle Application Server a WebSphere Application Server (WAS).

Objetivos específicos

- Integrar Maven a aplicación Trámites perteneciente al banco de aplicaciones heredadas de la Universidad de Antioquia.
- Actualizar librerías según sea requerido por el artefacto a medida de que se va avanzando con el proceso de migración.
- Manejar sistema de control de versionamiento el cual permita tener un historial de cambios y agregue una capa de seguridad del código fuente en términos de almacenamiento.
- Realizar pruebas unitarias y de integración que permitan tener certeza de que la migración fue exitosa en cuanto a funcionalidades.
- Desplegar la aplicación en el WebSphere Application Server validando que funcione en los ambientes de desarrollo, pruebas y producción respectivamente.
- Documentar lecciones aprendidas en un documento maestro el cual sea útil para futuras migraciones.

Marco Teórico

Hoy en día, la mayoría de las organizaciones se encuentran adaptando procesos de negocio a los nuevos escenarios económicos y tecnológicos, con el fin de estar a la vanguardia y ofrecer sus servicios de manera óptima. En ese orden de ideas, la modernización e integración de sus sistemas de software son tareas indispensables para lograr dicho objetivo. En algunas entidades como es el caso de la Universidad de Antioquia, existe un portafolio de sistemas de software que está compuesto por los llamados sistemas heredados, los cuales dan soporte a funciones críticas dentro de la entidad educativa, y se hacen necesario modernizar e integrar.

La modernización e integración de sistemas heredados no es una tarea fácil, en algunos casos es poco viable, dado que el tiempo y costos requeridos para realizar dicha labor se salen del alcance del proyecto, esto debido a algunas características que poseen estos sistemas, por ejemplo, tener una estructura monolítica, estar desarrollados con tecnología obsoleta, poseer poca o nula documentación, entre otras. Tradicionalmente, los esfuerzos de integración de estos sistemas apuntan al reemplazo total (rediseño) o gradual (migración) de estos sistemas.[1]

Un modelo en el cual se han basado una gran porción de aplicaciones dentro de las entidades, las cuales requieren ser migradas, es el de Cliente/Servidor, el cual se compone de dos actores fundamentales, el Cliente y el Servidor. El Cliente es el programa ejecutable que participa activamente en el establecimiento de las conexiones. envía una petición al servidor y se queda esperando por una respuesta. Su tiempo de vida es finito una vez que son servidas sus solicitudes, termina el trabajo. El servidor es un programa que ofrece un servicio que se puede obtener en una red, Acepta la petición desde la red, realiza el servicio y devuelve el resultado al solicitante. Al ser posible implantarlo como aplicaciones de programas, puede ejecutarse en cualquier sistema donde exista TCP/IP y junto con otros programas de aplicación, El servidor comienza su ejecución antes de comenzar la interacción con el cliente.[2]

Ahora, para realizar migraciones es muy útil Maven, que es una herramienta de software para la gestión de proyectos Java y la automatización de compilaciones. Maven está alojado en Apache Software Foundation, donde anteriormente formaba parte del Proyecto Jakarta. Maven utiliza una construcción conocida como Project Object Model (POM) para describir el proyecto de software que se está construyendo, sus dependencias en otros módulos y componentes externos y el orden de construcción. Viene con objetivos predefinidos para realizar ciertas tareas bien definidas, como la compilación de código y su empaque. Maven descarga dinámicamente las bibliotecas de Java y los complementos de Maven desde uno o más repositorios. Maven proporciona soporte integrado para recuperar archivos del Repositorio central de Maven 2 y otros repositorios de Maven, y puede cargar artefactos en repositorios específicos después de una compilación exitosa.[3]

Por último, las aplicaciones migradas van a ser desplegadas en el WebSphere Application Server. Que ha evolucionado rápidamente desde una simple extensión para servidores web y un tiempo de ejecución de servidor para objetos comerciales hasta el sistema operativo distribuido de IBM para computación de misión crítica y el servidor de aplicaciones líder en la industria. WebSphere Application Server juega un papel central en la

transformación de un sistema operativo distribuido a un sistema operativo distribuido bajo demanda. Esta transformación se logra mediante la creación de extensiones a la base de WebSphere Application Server para la infraestructura de computación en red, modelos de interacción enriquecidos basados en Web, arquitectura orientada a servicios, autonomía, gestión de procesos de negocio y aprovisionamiento dinámico y gestión de servicios.[4]

Metodología

En el momento de inicio de la práctica empresarial se realizó una inducción en la cual se brindó información sobre la entidad, como por ejemplo el IDE del cual se hace uso, que en este caso fue Eclipse, además de los ambientes que se manejan, desarrollo, pruebas y producción. Se hizo una breve descripción de la arquitectura de cada servidor y de cómo es la forma de proceder a la hora de realizar un despliegue, también se dió a conocer el repositorio que se usa para almacenar librerías (Nexus) y para hacer el control de versionamiento(GitLab). Luego la asesora externa brindó documentación sobre la arquitectura del software que maneja la entidad y en donde se pudo hallar información necesaria para llevar a cabo la migración de forma exitosa.

Para realizar el proceso de migración se requirió seguir una serie de pasos, en un principio el arquitecto a cargo eligió una aplicación heredada según su orden de prioridad dentro de la universidad, esta elección fue el aplicativo Trámites y Certificados, se designó la persona a cargo de ésta, la cual se encargó de dar una explicación detallada del funcionamiento tal y como está corriendo en el ambiente productivo, además de una breve introducción de la parte algorítmica del proyecto.

Una vez se asignó el proyecto, se procedió a trabajar en la configuración del entorno de desarrollo, para el cual existe un manual que contiene las recomendaciones y/o indicaciones para que cualquier integrante se acople a una integración continua ya establecida y ejecute la estrategia de migración sin ninguna dificultad posible, dentro de las herramientas que se hicieron uso están: Eclipse IDE for Java EE Developers, JDK, Apache

Maven, Plugin WebSphere Application Server Migration Toolkit, Git SDK, Plugin EGit- Git Team Provider.

Ahora bien, luego de haber configurado todo el ambiente de desarrollo se procedió a migrar la aplicación a un proyecto Maven, el cual debió contar con una configuración descrita en el manual de ejecución de una migración, que contiene indicaciones para la depuración y estabilización de código fuente para el proyecto. A medida de que se iba avanzando con el proyecto se debió hacer uso oportuno de GitLab para llevar un control de versionamiento.

Luego de que la aplicación se encontró funcional dentro de Maven se procedió a intentar el despliegue en el ambiente de desarrollo, que en un principio no fue posible debido a problemas que se fueron presentando por inestabilidad del servidor, se optó por crear un ticket para que la aplicación fuera desplegada en el ambiente de pruebas, esto por la entidad correspondiente, una vez se encontró allí, luego de solucionarse algunas dificultades por incompatibilidades de librerías con el servidor WAS, se validó que todo marchara de forma correcta, sin embargo, aparecieron algunos bugs que en el ambiente local no se presentaban, por lo cual se continuó la labor de resolver estos problemas dentro del ambiente local.

A medida de que se avanzó con la migración de la aplicación en la mayoría de casos se iban presentando una serie de errores, los cuales debían de ser registrados en un documento maestro que actúa como una especie de repositorio.

Resultados y análisis

Dentro del marco de esta práctica académica se logra integrar con Maven la aplicación legacy Trámites y Certificados pudiendo así acceder a todas las bondades que brinda este software.

Una vez la integración con Maven fue completada se procedió a realizar la actualización de librerías, se pudo evidenciar el éxito de dicho proceso

en términos de compilación del proyecto en el servidor Apache Tomcat (ambiente local), fue necesario usar el método de prueba y error a medida que se iban agregando librerías, debido a que las actualizaciones de algunas de ellas a versiones más recientes ocasionaban colisiones con las que debían permanecer en su versión original, en algunos casos fue necesario excluir librerías que vienen contenidas en algunas dependencias.

```
<dependency>
  <groupId>com.sun.faces</groupId>
  <artifactId>jsf-api</artifactId>
  <version>${jsf.version}</version>
</dependency>
<dependency>
  <groupId>com.sun.faces</groupId>
  <artifactId>jsf-impl</artifactId>
  <version>${jsf.version}</version>
</dependency>
<dependency>
  <groupId>org.icefaces</groupId>
  <artifactId>icefaces-ace</artifactId>
  <version>3.3.0</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.icefaces/icefaces-compat -->
<!-- Biblioteca con el conjunto completo de componentes personalizados
y extendidos de ICEfaces 1.8.x, actualizado para ejecutarse en JSF 2 y el
marco actual de ICEfaces. -->
<dependency>
  <groupId>org.icefaces</groupId>
  <artifactId>icefaces-compat</artifactId>
  <version>3.3.0</version>
  <exclusions>
    <exclusion>
      <groupId>javax.servlet</groupId>
      <artifactId>jstl</artifactId>
    </exclusion>
    <exclusion>
      <groupId>commons-logging</groupId>
      <artifactId>commons-logging</artifactId>
    </exclusion>
    <exclusion>
      <groupId>commons-beanutils</groupId>
      <artifactId>commons-beanutils</artifactId>
    </exclusion>
  </exclusions>
</dependency>
```

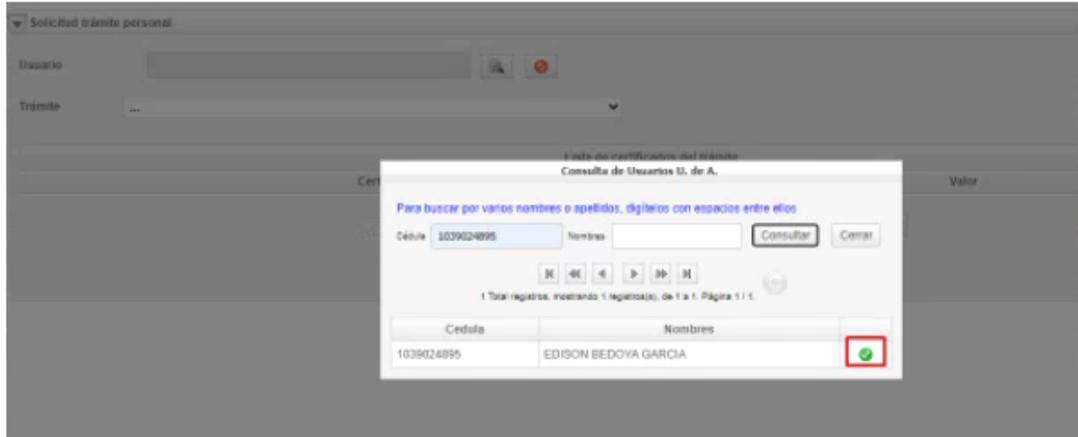
Ilustración 1. POM.xml

Como se puede observar en la ilustración 1 se tiene que las librerías con su correspondiente versión son jsf 2.1.7 y ICEfaces 3.3.0. Para jsf se declara e inicializa la versión al comienzo del archivo xml, en ICEfaces, cabe resaltar que son excluidas dentro de la dependencia icefaces-compat los artefactos jstl, commons-logging y commons-beanutils, estos son algunos que ocasionan errores de compilación debido a colisiones con otras librerías que ya se hallan dentro del archivo, no se muestran todas las librerías usadas debido a cuestiones de privacidad de la entidad.

Luego de que las dependencias se encuentran configuradas correctamente, se procedió a resolver problemas que surgen por temas de incompatibilidades que se van presentando con los fragmentos de código que no se ajustan a las nuevas versiones de las librerías, como caso particular se presenta la ilustración 2, donde se describe la problemática que se encontró en el proceso de migración, esta queda registrada en el documento maestro de lecciones aprendidas que sirve como insumo para futuras migraciones, una vez se halla la solución (ilustración 3) se describe de manera detallada el conjunto de pasos a seguir para resolver la problemática.

Error

Cuando hay un dataTable y dentro de las filas se encuentran botones como se muestra en el pantallazo, en el momento de presionar en el botón que se señala este no activa la acción que se debería ejecutar



en este caso se tiene la propiedad binding de el `ice:commandLink` la cual tiene un `actionListener` pendiente de un evento, que no se ejecuta cuando se presiona el botón de chulito verde.

```
<ice:column id="seleccionarColumn">
  <f:facet name="header">
    <ice:outputText value="" />
  </f:facet>
  <div align="center">
    <ice:commandLink binding="#{UsuariosIsvsBB.seleccionarEmpleado}" >
      <ice:graphicImage value="#{sesPathTramites}/img/ok.png" title="Seleccionar" style="border:none;" />
    </ice:commandLink>
  </div>
</ice:column>
```

```
usuarios.getSeleccionarEmpleado().addActionListener(new SeleccionarUsuarioActionListener());
```

Ilustración 2. Presentación del error

Solución

Se hallan 2 soluciones, la primera es poner el atributo **listaUsuarios** como **static**, esto con el fin de que esta propiedad pueda ser accedida globalmente por el sistema y su valor no se pierda entre peticiones

```
<ice:dataTable binding="{UsuariosWsBB.tblListaDatos}" width="100%" id="tblListaDatosPersonas" value="{UsuariosWsBB.listaUsuarios}" var="dato" rows="10" >
```

```
private static List<PersonaNatural> listaUsuarios = new ArrayList();;
```

La segunda solución consiste en modificar el scope del BB a session como se muestra en el pantallazo, en este caso esta solución no funcionó debido a que si se asignaba el scope a session no se mostraba el popup, pero en otro caso diferente debería funcionar

```
<managed-bean>
  <managed-bean-name>UsuariosWsBB</managed-bean-name>
  <managed-bean-class>co.edu.udea.tramites.bb.UsuariosWsBB</managed-bean-class>
  <managed-bean-scope>session</managed-bean-scope>
</managed-bean>
```

Se deja video totalmente relacionado, en el cual se explican los scopes de una manera muy entendible.

<https://www.youtube.com/watch?v=5gkQeCTIQ94>

Ilustración 3. Solución hallada

A medida que los errores fueron solventados se hizo uso de la herramienta GitLab con el fin de llevar el control de versionamiento, dentro de esta se manejaron 4 tipos de ramas estas son master para producción, develop para desarrollo, quality para pruebas y feature/migración que es donde se realizó los push a medida de que se avanzaba con la migración.

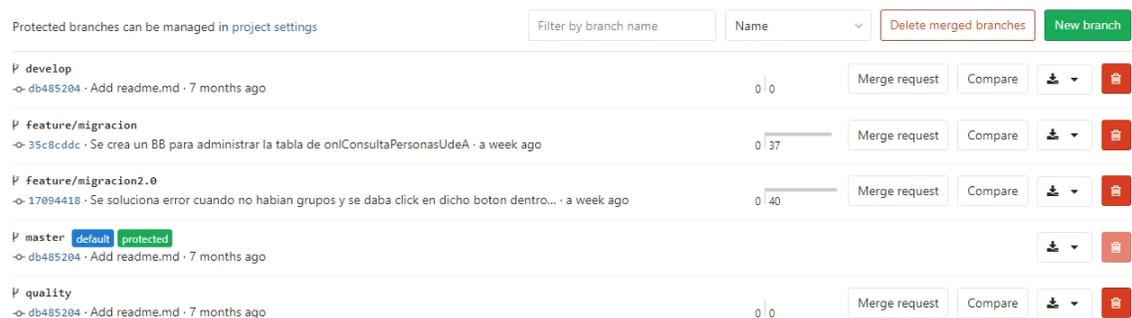


Ilustración 4. GitLab Branches

En el apartado gráfico cómo se logra apreciar en la ilustración 5 se tienen 16 vistas señaladas con recuadros rojos, a su vez en varias de estas hay vistas anidadas, lo que hace que la sumatoria arroje como resultado un total de 42.

Definición de certificados

Administrar trámites

Solicitud trámite personal

Solicitudes Por otro

Solicitud trámite reúne

Solicitudes de certificados

Consultar trámites usuario

Validar certificado

Prueba Generar Certificado

Gestionar grupo OID

Reporte de trámites

Código	Nombre	Tipo	Vigencia	Modo ejecución	Tiempo ejecución	Editar	Estado	Grupos
ACTIVIDADESLAB	DETALLE DE ACTIVIDADES LAB	Personal	3	AUTOMATICO	4			
ACTIVIDADESLABO	DETALLE DE ACTIVIDADES LABORALES	Personal	31	AUTOMATICO	4			
ACTIVIDADESLAB2	DETALLE DE ACTIVIDADES LABORALES 3	Academico	2	MANUAL	3			
ACTIVIDADESLAB4	DETALLE DE ACTIVIDADES LABORALES 4	Personal	3	MANUAL	2			
CATEDHISINVLRC	HISTORIA CATEDRA SIN VALOR CONTRATO	Personal	5	AUTOMATICO	5			
CATEDHISVLRCONT	HISTORIA CATEDRA VALOR CONTRATO	Personal	5	AUTOMATICO	5			
CATEDSINVLRCONT	Catedra sin valor contrato	Personal	5	AUTOMATICO	1			
CATEDVLRCONTRAT	CATEDRA CON VALOR CONTRATO	Personal	5	AUTOMATICO	1			
CERTEXPLAB	CERTIFICADO LABORAL DE EXPERIENCIA O HISTORIA LABORAL	Personal	30	MANUAL	3			
EXPEDIRTIP	RE-EXPEDICIÓN DE LA TIP	Personal	1	MANUAL	1			

Anegar Trámite / configurar certificados y dependencias

Ilustración 5. Aplicación Trámites y Certificados

Si bien se logra hacer funcionar en un 98% la aplicación Trámites y Certificados en el ambiente local, se presentaron dificultades a la hora de llevar al servidor de desarrollo luego de un arduo proceso. Esto surge debido a que se generaron diversos problemas con respecto a los despliegues por una inadecuada actualización del WAS; se presentaron incompatibilidades con librerías del servidor, lo que provocaba una baja disponibilidad, hasta tal punto que llegó a estar más de un mes fuera de servicio y se complicó la realización pruebas de la aplicación. Sin embargo, finalizando el tiempo estipulado para esta práctica se logra estabilizar el WAS y es posible realizar el despliegue, se detecta que en este ambiente funciona un 90% del total de la aplicación, por lo que se procede a dar solución desde el ambiente local, labor que continúa más allá del tiempo estipulado para esta práctica académica.

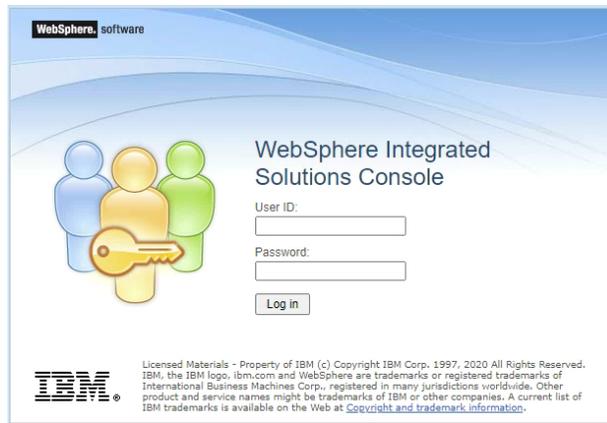


Ilustración 6. Interfaz de login WAS

Conclusiones

Migrar una aplicación permite que sea desplegada en un nuevo servidor y tener disponibilidad en un corto período de tiempo. En contraste, una reconstrucción desde cero de una aplicación legacy, incrementa de manera considerable el tiempo necesario para lograr esa disponibilidad.

Maven es una poderosa herramienta de software para gestionar y construir proyectos, dentro del alcance de esta práctica empresarial jugó un papel primordial al momento de gestionar las dependencias necesarias para que el proyecto pudiera compilar, esta labor se hizo relativamente sencilla gracias a las bondades de esta herramienta, también se usó la funcionalidad de generar un archivo .war que es requerido por el servidor WAS para su correspondiente despliegue.

Es de vital importancia llevar un control sobre el versionamiento del código que se va ajustando a medida que se avanza con la migración, por esto la herramienta Git juega un papel muy importante, ya que en muchas ocasiones se presentaban problemas con el código que se iba ajustando y era necesario el volver a una versión anterior, esta tarea era relativamente simple gracias a Git.

También a medida que se iban resolviendo los problemas que se hallaban en el código fuente era necesario documentar estas lecciones aprendidas

en el documento maestro, en realidad la utilidad de este tipo de archivos a la hora de realizar migraciones es muy elevada, ya que permitió avanzar de manera mucho más rápida en este proceso, con ayuda de la documentación generada en antiguos proyectos.

Si bien luego de un exhaustivo proceso de intentar desplegar la aplicación en el servidor del WAS de desarrollo, se evidenció que en muchas ocasiones debido a problemas con terceros, se puede generar un retraso considerable de las actividades que se están ejecutando, en este caso finalizando el tiempo destinado para la práctica empresarial, se logró el objetivo de desplegar en desarrollo, sin embargo por cuestiones de tiempo no fue posible llevar a cabo la implementación de pruebas unitarias y de integración, cuyo objetivo específico está definido dentro del marco de esta práctica empresarial.

Esta práctica académica resultó de gran utilidad debido a que se hizo un acercamiento al mundo laboral y así poner en práctica los conocimientos adquiridos durante el pregrado de Ingeniería de Sistemas, si bien se presentaron una serie de dificultades a lo largo de los 6 meses estipulados de esta práctica, fue posible resolverlas de manera paulatina gracias a las bases adquiridas durante la formación en la universidad.

Referencias Bibliográficas

[1] Rivas, H., 2006. *INTEGRACIÓN DE SISTEMAS HEREDADOS UTILIZANDO WEB SERVICES*. INGENIERO. UNIVERSIDAD RICARDO PALMA. Disponible en: <http://repositorio.urp.edu.pe/handle/urp/33>

[2] Oscar Lizama, Geordy Kindley, Javier Ignacio. *Redes de computadores Arquitectura Cliente - Servidor*. Disponible en: <http://profesores.elo.utfsm.cl/~agv/elo322/1s16/projects/reports/Proyecto%20Cliente%20-%20Servidor.pdf>

[3] F. P. Miller, A. F. Vandome and J. McBrewster, *Apache maven*, 2010. Disponible en: <https://dl.acm.org/doi/book/10.5555/1942838>

[4] E. N. Herness, R. J. High, and J. R. McGee, "WebSphere Application Server: A foundation for on demand computing," *IBM Systems Journal*, vol. 43, no. 2, 2004. Disponible en: <https://ieeexplore.ieee.org/abstract/document/5386788/citations#citations>