



**UNIVERSIDAD
DE ANTIOQUIA**

INTEGRACIÓN DE MAVEN A APLICACIÓN LEGACY Y MIGRACIÓN A SERVIDOR WEBSHERE APPLICATION SERVER (WAS) DE IBM.

Autor(es)

Johnathan Arley Monsalve Bello

Universidad de Antioquia

Facultad De ingeniería

Medellín, Colombia

2021



INTEGRACIÓN DE MAVEN A APLICACIÓN LEGACY Y MIGRACIÓN A
SERVIDOR WEBSHERE APPLICATION SERVER (WAS) DE IBM.

Johanthan Arley Monsalve Bello

Tesis o trabajo de investigación presentada(o) como requisito parcial para optar al título de:

Ingeniero de sistemas

Asesores (a):

Asesora Interna:

Deisy Loaiza Berrío

Asesora Externa:

Dennis Alexandra Marin Marin

ingeniera de sistemas, especialista en ciencias electrónica e informática

Universidad de Antioquia

Facultad de Ingeniería

Medellín, Colombia

2021.

Contenido

Contenido	2
Resumen	3
Introducción	3
Objetivos General	4
Objetivos Específicos	4
Marco Teórico	4
Metodología	5
Resultados y análisis	7
Dificultades	10
Conclusiones	11
Referencias Bibliográficas	11
Archivos Adjuntos	12
Lecciones aprendidas Migración de SIIU_ADM.docx	12
Migración a JasperReport 6.15.0.docx	12

Resumen

La Universidad de Antioquia se encuentra en un proceso de transformación digital donde se busca automatizar y digitalizar todos los procesos posibles que actualmente se hacen a mano, actualizar las aplicaciones legadas y migrarlas a un servidor Websphere application Server, WAS por sus siglas. Entre las aplicaciones que se busca actualizar y migrar, se encuentra el sistema SIIU, el cual es un sistema compuesto de 4 aplicaciones web para gestionar investigaciones, desde la fase de convocatorias hasta la fase de finalización y entrega de resultados, por lo tanto, se debió migrar todas las aplicaciones que hacen parte de este sistema, entre las cuales se encuentra la aplicación SIIU_ADM que administra los proyectos desde que son aceptados por la universidad hasta la finalización y entrega de resultados la cual faltaba por migrar al servidor WAS. Teniendo como objetivo principal migrar la aplicación SIIU_ADM a un servidor WAS se siguió inicialmente una guía planteada por una empresa externa la cual hizo la primera migración donde indica cómo implementar Maven en la aplicación y hacer una configuración inicial para despliegue en servidor tomcat local y en el servidor al WAS en internet y se procedió a solucionar inconvenientes únicos de este sistema hasta poderse desplegar en un ambiente productivo y disponible para los usuarios de este.

Introducción

La Universidad de Antioquia es una institución de educación superior pública con sede principal en Medellín Antioquia y con seccionales en varias subregiones del departamento. La institución es reconocida entre las mejores universidades del país por su calidad educativa y enfoque investigativo. La universidad apoya y dirige una gran cantidad de investigaciones desarrolladas por personal de la institución como profesores, grupos de investigación y estudiantes investigadores. El proceso que conlleva el seguimiento en cada una de las etapas se hace muy complejo cuando se tiene gran cantidad de investigaciones; por lo tanto, se usa un aplicativo que facilita el seguimiento entre estados de cada proyecto de investigación brindando la información adecuada y rápidamente.

La aplicación anteriormente mencionada es una aplicación web desarrollada aproximadamente en el 2010 y soportada aún por la universidad, corre en un servidor Oracle Application Server (OAS) y tiene muchas dependencias o librerías obsoletas, inseguras o sin soporte. Por lo tanto, la Universidad de Antioquia necesita actualizar la arquitectura de esta aplicación implementando Maven como constructor de proyecto y gestor de librerías, pluggins y tareas. Además, se necesita crear un Flow de desarrollo basado en GitFlow para el despliegue de la aplicación en diferentes ambientes para desarrollar, probar y poner en producción la aplicación migrada que como último paso se deberá desplegar en un Servidor WebSphere Application Server (WAS).

Objetivos General

Migrar aplicación SIIU-ADM para gestión de proyectos de investigación de servidor Oracle Application Server (OAS) a WebSphere Application Server (WAS).

Objetivos Específicos

- Crear repositorio de aplicación en Gitlab y gestionar ramificación con metodología GitFlow
- Integrar Maven a aplicación SIIU-ADM con las tareas y plugins indicados por la división para la compilación y despliegue en WAS
- Actualizar la versión de Java JDK 6 a Java JDK8 y las librerías de JSF, IceFaces, SpringFramework y librerías propias de la universidad por las versiones indicadas por la división de desarrollo
- Resolver incompatibilidades que se presenten en el código al actualizar las librerías
- Desplegar en servidores WAS en ambiente de desarrollo, pruebas y producción
- Retroalimentar documento de lecciones aprendidas

Marco Teórico

En este proyecto se migró una aplicación desde un servidor OAS a un servidor WAS, es decir, se movió una aplicación de un entorno Oracle Application Server (OAS) a un entorno WebSphere Application Server (WAS) con una arquitectura actualizada, más sofisticada y de alta disponibilidad para no sólo garantizar la continuidad operativa del software, sino optimizar la aplicación y rendimiento de esta[1].

La aplicación se migró a WebSphere Application Server (WAS) de IBM porque ayuda a la agilidad empresarial con una base innovadora y basada en el rendimiento para crear, reutilizar, ejecutar, integrar y gestionar desde aplicaciones empresariales críticas para toda la empresa hasta las aplicaciones de nivel departamental más pequeñas, WebSphere Application Server ofrece confiabilidad, disponibilidad, seguridad y escalabilidad [2].

Para llevar a cabo la migración se utilizó la versión 8 de Java JDK el cual es un ambiente de desarrollo para construir aplicaciones y componentes usando el lenguaje de programación Java[3]. Además, se utilizó Maven como herramienta de gestión y comprensión de proyectos de software que, gracias a que está basado en el concepto de un Project Object Model (POM), permite describir fácilmente el proyecto de software a construir, sus dependencias, componentes externos y orden de construcción de los elementos[4] permitiendo también la ejecución de tareas como compilación y empaquetado.

Para el desarrollo de la lógica de negocio se utilizó Java EE, como estaba estipulado inicialmente, este permite desarrollar y ejecutar aplicaciones en lenguaje de programación

Java utilizando arquitectura de N capas distribuidas y se apoya ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones [5].

El desarrollo de la parte FrontEnd se desarrolló con JavaServer Faces y Icefaces. La primera tecnología es el componente estándar orientado a interface gráfica para proyectos de Java EE [6]. La segunda tecnología es un framework open-source de Rich Internet Application (RIA) para Java EE que mejora la eficiencia de desarrollo mientras reduce el tiempo de comercialización y costos operativos, además, promete entregar una gran cantidad de características para aplicativos viejos como el que se migró en este caso [7].

El versionamiento de código se gestionó mediante Git, el cual es un sistema de control de versiones gratuito y de código abierto diseñado para todo tipo de proyecto, desde pequeños hasta muy grandes, con velocidad y eficiencia [8]. Además, se utilizó GitLab, el cual es un servicio web de control de versiones y desarrollo colaborativo de software [9], como gestor y almacenamiento de repositorios en internet.

Metodología

Para la migración de este aplicativo se siguieron varios pasos, inicialmente, se estudió el funcionamiento de la aplicación desplegada en el servidor OAS con el fin de entender a cabalidad cada paso interno en la gestión de proyectos de investigación, esto porque la idea de realizar este tipo de migraciones, donde sólo se actualiza el tipo de proyecto, versiones de librerías y componentes necesarios, es que el impacto en el usuario sea el menor posible; por lo tanto, la aplicación debe funcionar exactamente igual minimizando los cambios.

Anterior a esta migración, se realizaron otras migraciones similares por parte de una empresa externa a la Universidad de Antioquia, los cuales dejaron unas guías básicas de configuración para migrar un proyecto JAVA EE a Maven, por lo tanto, se siguieron estas guías para este primer paso.

Una vez se integró integrado Maven al proyecto, se probó en local en un servidor Apache Tomcat donde inicialmente se obtuvieron resultados favorables, posterior a esto, se actualizaron las librerías a las versiones deseadas, en este caso en particular, se migró el Java JDK 6 a Java JDK 8, JavaServer Faces 1.2.0 a 2.1.7, Icefaces 1.8.2 a 3.3.0, SpringFramework de 2.5 a 4.3.25.RELEASE, además de éstas, se actualizaron todas las librerías propias de la universidad, las cuales se encargan de filtros de sesión, consumo de APIs de la universidad y utilidades.

Hasta este punto se trabajó la migración en todo el proyecto, es decir, los cambios realizados afectaron todo el proyecto JAVA, sin embargo, para la continuación de la migración, se dividió el proyecto en 5 módulos:

- Configuración
- Notificaciones
- Inicio Formal
- Gestión de compromisos

- Solicitudes

Esta división fue necesaria para realizar despliegues y pruebas más pequeñas y asegurarse a detalle que todas las funcionalidades cumplieran correctamente y también evitar puntos muertos de desarrollo donde se tuviera que esperar mucho para identificar y solucionar errores. La continuación de la migración de la aplicación se llevó a cabo módulo por módulo en el orden indicado anteriormente.

Al realizar los cambios en las librerías se solucionaron los inconvenientes ocasionados por esto, módulo por módulo, para esto fue necesario apoyarse en un documento de hallazgos creado al migrar aplicaciones similares (La cual también se fue actualizada y retroalimentada por el desarrollador en caso de encontrar errores nuevos), documentación oficial de las librerías y foros de programadores en línea. Al solucionar todos los errores de las librerías se probó nuevo en local para asegurarse del funcionamiento y similaridad con la anterior versión de la aplicación.

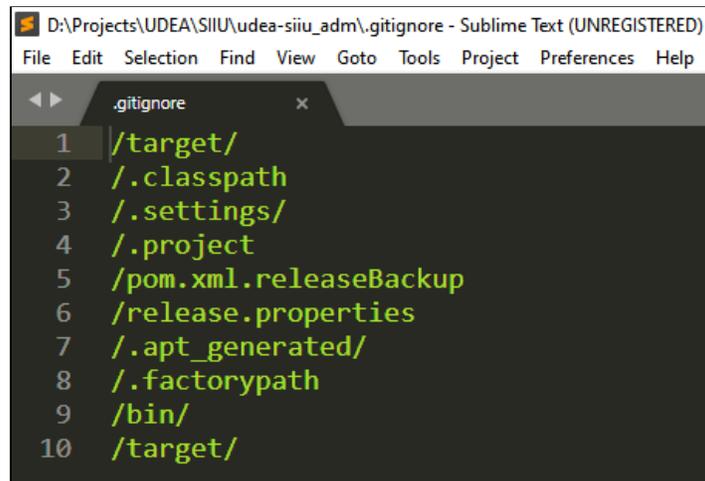
Desde el inicio de la integración se creó el repositorio y su respectivo flujo de ramas en GitLab que se utilizó hasta el final del proyecto para control de versiones y código. El repositorio, las ramas Master, Quality, Develop y las reglas de cada una de las ramas anteriormente mencionadas fueron creadas por Dennis Marín quien es la arquitecta del proyecto, sin embargo, el desarrollador tuvo la responsabilidad de la correcta gestión del repositorio.

Como penúltima etapa de la migración, se desplegó en cada ambiente en el servidor WAS designado, es decir, se desplegó en el servidor de desarrollo en WAS el cual el desarrollador tuvo toda la responsabilidad de subir, configurar y desplegar la aplicación. Después, se desplegó en el servidor de pruebas, el cual es gestionado por una empresa externa a la universidad y el desarrollador fue el responsable de enviar el ejecutable (WAR) y un documento con la información de despliegue y librerías usadas, posteriormente, la aplicación se probó por el usuario para verificar el correcto funcionamiento, se desplegó en el ambiente de producción y dió acceso libre a todos los interesados en esta aplicación. Para lo anterior, se creó un nuevo ejecutable (WAR) con la configuración del ambiente de producción y base de datos de producción, un documento similar al que se envía en pruebas y llenar un formulario con la información del despliegue para ser realizado por alguno de los arquitectos de la división conjunto a la empresa tercera encargada de la administración de los servidores WAS.

Como último requisito de la migración, se liberó cada módulo migrado a todos los usuarios que utilizan la aplicación, para esto, se sustituyeron los enlaces a la anterior aplicación (desplegados en servidor OAS) por los de la nueva versión, es decir, cuando en la universidad se busca alguna funcionalidad de SIIU_ADM, esta ya no cargará la aplicación desde el servidor anterior OAS, sino, que va a cargar la aplicación desde el servidor WAS. Esta configuración se hizo internamente por los arquitectos de la división de gestión de la universidad basándose en la documentación técnica otorgada por el desarrollador, por lo tanto, para los usuarios hubo algún cambio importante respecto a la experiencia.

Resultados y análisis

Como inicio de proyecto se creó el repositorio de la aplicación SIIU_ADM, este fue creado y configurado por la asesora externa e inicialmente sólo contenía un archivo readme.md por defecto, posteriormente, se procedió con la creación de código y relación con el repositorio, incluyendo el archivo .gitignore para evitar subir archivos indeseados a la aplicación, en este caso, se creó en base a una aplicación similar ya existente:



```
D:\Projects\UDEA\SIIU\udea-siiu_adm\.gitignore - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

.gitignore
1 /target/
2 /.classpath
3 /.settings/
4 /.project
5 /pom.xml.releaseBackup
6 /release.properties
7 /.apt_generated/
8 /.factorypath
9 /bin/
10 /target/
```

Imagen 1. GitIgnore.

A partir de este punto, se siguió una guía proporcionada por una empresa externa que anteriormente había hecho una migración similar para crear un proyecto maven y adaptar el código viejo a este tipo de proyecto. Como principal diferencia se tuvo que el proyecto nuevo tiene un archivo **pom.xml** con la configuración del proyecto Maven y una estructura de paquetes que separa los archivos Java (Lógica de negocio), Properties (Configuraciones, queries y constantes) y los de extensión xhtml y su configuración (Frontend y configuración).

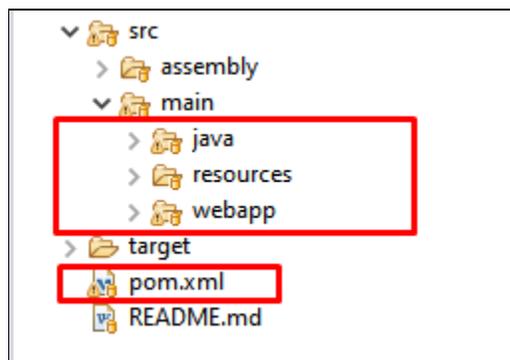


Imagen 2. Estructura de paquetes principal.

```
<java.version>1.8</java.version>
<jsf.version>2.1.7</jsf.version>
<icefaces.version>3.3.0</icefaces.version>
<spring.version>4.3.25.RELEASE</spring.version>
```

Imagen 3. Versiones actualizadas de librerías principales.

A partir de este punto, se actualizaron las dependencias mencionadas en la metodología y se comenzó a actualizar, probar y solucionar página por página de la aplicación, esta parte fue la más tediosa debido a que algunas páginas eran muy grandes, aproximadamente de 2000 - 3000 líneas y cada página tenía en ocasiones inconvenientes totalmente únicos, sin embargo, en los primeros 3 meses se logró migrar el módulo de configuración y el módulo de notificaciones, con despliegues en local, desarrollo, pruebas y producción. Incluso, se tuvieron estos módulos por 15 días en ambiente pre-productivo para probar a detalle cada función para finalmente dejar disponible a los usuarios la versión migrada sin que se notase alguna diferencia importante como se puede ver a continuación en la imagen 4 y 5.

Instancia Administrativa						
▶ Instancias Administrativas						
▼ Resultados de la Búsqueda						
Detalle	Código	Nombre	Nivel Administrativo	Administra Proyecto	Jefe	
(+ MOSTRAR)	67	Centro de Extensión - Escuela de Idiomas Centro de Extensión - Escuela de Idiomas	Centro de Extensión	Si	Claudia Patricia Díaz Mosquera jextensionidiomas@udea.edu.co 219 8783	 
(+ MOSTRAR)	68	Centro de Extensión - Escuela de Nutrición y Dietética Centro de Extensión - Escuela de Nutrición y Dietética	Centro de Extensión	Si	Uber Isaza Agudelo extensionnyd@udea.edu.co 8345	 
(+ MOSTRAR)	71	Centro de Extensión - Facultad de Artes Centro de Extensión - Facultad de Artes	Centro de Extensión	Si	Blanca Miriam Valencia Echavarría blanca.valencia@udea.edu.co 5888	 
(+ MOSTRAR)	72	Centro de Extensión - Facultad de Ciencias Agrarias Centro de Extensión - Facultad de Ciencias Agrarias	Centro de Extensión	Si	Libia Inés Martínez Hernández extensionfca@udea.edu.co 219 9178	 
(+ MOSTRAR)	73	Centro de Extensión - Facultad de Ciencias Económicas Centro de Extensión - Facultad de Ciencias Económicas	Centro de Extensión	Si	Gilma Fabiola Flórez Garcés gilma.florez@udea.edu.co 219 8824	 
(+ MOSTRAR)	74	Centro de Extensión - Facultad de Ciencias Exactas Centro de Extensión - Facultad de Ciencias Exactas	Centro de Extensión	Si	Fernando León Gutiérrez ceen@udea.edu.co, ceen@gmail.com 219 8901	 
(+ MOSTRAR)	51	CENQFAL Centro de Extensión - Facultad de Ciencias Farmacéuticas	Centro de Extensión	Si	Jorge Humberto Herrera Sierra jorgeh.herrera@udea.edu.co, cenqfal@udea.edu.co 2195457	 
(+ MOSTRAR)	53	Centro de extensión - Facultad de Ciencias Sociales y Humana Centro de extensión - Facultad de Ciencias Sociales y Humanas	Centro de Extensión	Si	Patricia Estrada Mejía patricia.estrada@udea.edu.co, extension.fosh@udea.edu.co 219-8787 - 300	 
(+ MOSTRAR)	77	Centro de Extensión - Facultad de Derecho y Ciencias Políticas Centro de Extensión - Facultad de Derecho y Ciencias Políticas	Centro de Extensión	Si	Juan Camilo Mejía Walker camilo.mejia@udea.edu.co, extensionderechopolitica@udea.edu.co 219 99 62	 
(+ MOSTRAR)	78	Centro de Extensión - Facultad de Educación Centro de Extensión - Facultad de Educación	Centro de Extensión	Si	Hugo Alberto Buitrago Montoya hugo.buitrago@udea.edu.co, jextensioneducacion@udea.edu.co 2195710 - 5734	 



▶ Creación/Edición Instancia Administrativa

Imagen 4. Página de configuración 1 en OAS ambiente de pruebas.

Instancia Administrativa						
▶ Instancias Administrativas						
▼ Resultados de la Búsqueda						
Detalle	Código	Nombre	Nivel Administrativo	Administra Proyecto	Jefe	
(+ MOSTRAR)	67	Centro de Extensión - Escuela de Idiomas Centro de Extensión - Escuela de Idiomas	Centro de Extensión	Si	Claudia Patricia Díaz Mosquera jextensionidiomas@udea.edu.co 219 8783	
(+ MOSTRAR)	68	Centro de Extensión - Escuela de Nutrición y Dietética Centro de Extensión - Escuela de Nutrición y Dietética	Centro de Extensión	Si	Uber Isaza Agudelo extensionnyd@udea.edu.co 8345	
(+ MOSTRAR)	83	Centro de Extensión - Facultad Nacional de Salud Pública Centro de Extensión - Facultad Nacional de Salud Pública	Centro de Extensión	Si	Gladis Irene Arboleda Posada cesaludpublica@udea.edu.co 219 68 07 - 219	
(+ MOSTRAR)	71	Centro de Extensión - Facultad de Artes Centro de Extensión - Facultad de Artes	Centro de Extensión	Si	Bianca Miriam Valencia Echavarría blanca.valencia@udea.edu.co 6888	
(+ MOSTRAR)	72	Centro de Extensión - Facultad de Ciencias Agrarias Centro de Extensión - Facultad de Ciencias Agrarias	Centro de Extensión	Si	Libia Inés Martínez Hernández extensionfca@udea.edu.co 219 9178	
(+ MOSTRAR)	73	Centro de Extensión - Facultad de Ciencias Económicas Centro de Extensión - Facultad de Ciencias Económicas	Centro de Extensión	Si	Gilma Fabiola Flórez Garcés gilma.florez@udea.edu.co 219 8824	
(+ MOSTRAR)	74	Centro de Extensión - Facultad de Ciencias Exactas Centro de Extensión - Facultad de Ciencias Exactas	Centro de Extensión	Si	Fernando León Gutiérrez oeeen@udea.edu.co, oeeen@gmail.com 219 8861	
(+ MOSTRAR)	51	CENQFAL Centro de Extensión - Facultad de Ciencias Farmacéuticas	Centro de Extensión	Si	Jorge Humberto Herrera Sierra jorgeh.herrera@udea.edu.co, cenqfal@udea.edu.co 2195457	
(+ MOSTRAR)	77	Centro de Extensión - Facultad de Derecho y Ciencias Político Centro de Extensión - Facultad de Derecho y Ciencias Políticas	Centro de Extensión	Si	Juan Camilo Mejía Walker camilo.mejia@udea.edu.co, extensionderechopolitica@udea.edu.co 219 99 62	
(+ MOSTRAR)	78	Centro de Extensión - Facultad de Educación Centro de Extensión - Facultad de Educación	Centro de Extensión	Si	Hugo Alberto Buitrago Montoya hugo.buitrago@udea.edu.co, jextensioneducacion@udea.edu.co 2195710 - 5734	

Imagen 5. Página de configuración 1 en WAS ambiente de pruebas.

Como las páginas anteriores, cada página migrada en el servidor WAS se comparó con la página en su versión anterior en el servidor OAS para asegurarse de no haber cambios bruscos que pudieran haber afectado la experiencia de los usuarios.

Los módulos de Configuración y Notificaciones fueron migrados y liberados a los usuarios al cabo de 3 meses de iniciar el proyecto, sin embargo, durante el mes de revisión y algunas configuraciones extras que se debieron hacer para desplegar en ambiente productivo también se estuvo trabajando en paralelo en el módulo de Inicio Formal, este módulo era un poco más largo que el anterior, sin embargo, se pudo finalizar la migración en aproximadamente 2 meses hasta el punto de desplegar en ambiente de pruebas donde se detuvo durante un tiempo debido a falta de personal para realizar las pruebas a detalle, sin embargo, en este punto iban aproximadamente 4 meses del proyecto y sólo había en un 20% de la aplicación migrada en ambiente productivo, por lo tanto, se procedió con la migración de los módulos Gestión de compromisos y Solicitudes. El primero de estos se finalizó en aproximadamente 15 días gracias a que era un módulo pequeño, sin embargo, el módulo de Solicitudes es un módulo bastante robusto y complicado, similar a Inicio Formal, por lo tanto, al finalizar los 6 meses de práctica se terminó a nivel de desarrollo

local pero no se pudo desplegar en ambiente WAS de desarrollo ni de pruebas por varias dificultades las cuales se explicarán en el siguiente punto del informe.

Durante el tiempo de ejecución del proyecto, unas de las responsabilidades de como desarrollador era crear informes sobre los hallazgos que se hayan obtenido en el proceso de migración, entre estos, se pueden destacar *Lecciones aprendidas Migración de SIIU_ADM.docx* y *Migración a JasperReport 6.15.0.docx* donde el primero son errores encontrados en la migración de los que no se tenían conocimientos en la división de gestión informática y su solución. El segundo es una guía de migración de JasperReport la cual es una librería de código abierto para crear informes para Java[10]. Esta se debió migrar a la versión 6.15.0 debido a la compatibilidad con el servidor WAS. Además, se mantuvo un archivo con los enlaces directos a las páginas principales migradas de cada módulo para seguimiento del estado de estas llamada *URLS migradas SIIU ADM.xlsx*. Los primeros dos documentos mencionados se adjuntará a este informe, sin embargo, el tercero no se hará debido a que expone información sensible sobre los servidores de la universidad de Antioquia.

Dificultades

Durante el desarrollo del proyecto se tuvieron varias dificultades las cuales fueron adaptarse al trabajo remoto, documentación y solución de errores al migrar código y reducción del equipo de trabajo. La primera de las dificultades mencionadas se produjeron principalmente problemas de comunicación con algunos compañeros ya que era un poco más difícil organizar un horario para una reunión o para salir de una duda precisa ya que no todos estábamos pendiente del correo y es que sinceramente sería algo que retrasaría el trabajo de todos, sin embargo, fue algo a lo que nos pudimos adaptar y poder trabajar programando reuniones y solucionar todas las dudas posibles encontradas hasta ese momento. En cuanto a las dificultades de documentación y solución de errores al migrar sucedieron porque la arquitectura del proyecto era única en toda la división informática de la universidad, lo que implicaba que si se tenía alguna duda se debía recurrir principalmente a información a internet y aún así era difícil ya que era difícil encontrar soluciones para problemas explícitos de la integración de JSF y SpringFramework, sin embargo, para algunos errores que no se encontraron las soluciones más adecuadas se tuvieron que adoptar estrategias diferentes como rehacer algunas páginas o limitar las funciones de Icefaces. La tercer dificultad mencionada fue la que más implicó en el desarrollo del proyecto porque en varios momentos se tuvo que detener la migración del proyecto por falta de alguna persona que pudiera probar las partes migradas, lo que implicó que hubiera mucho tiempo muerto para tener retroalimentación y decidir si desplegar a producción o no alguno de los módulos migrados, además, sobre el mes de enero de 2021 que volvimos de vacaciones de fin de año uno de los compañeros con más responsabilidades a cargo en la división de gestión informática renunció y se entregaron esos proyectos a los demás integrantes de la división de gestión informática y por supuesto, a mi también me tocó recibir porque como empleado debía hacerlo y esto me quitó bastante tiempo para el proyecto de migración, además, más o menos al mes renunció otra compañero dejando gran parte de sus responsabilidades a mi cargo porque era quien más tenía conocimiento en el área de desarrollo y a un compañero más en el área de base de datos y esto por supuesto

redujo aún más el tiempo disponible para el proyecto de migración durante el último mes e imposibilitando finalización del proyecto.

Conclusiones

- Los conocimientos adquiridos en la universidad durante la carrera fueron una base fundamental para resolver los retos afrontados en esta práctica y cumplir con todos los objetivos propuestos por la empresa.
- La migración de aplicativos legados es muy importante ya que mejora el rendimiento y la seguridad de cada uno de estos, sin embargo, se debe pensar correctamente la mejor estrategia para llevar a cabo este tipo de proyecto asegurándose de finalizar en un tiempo justo y usando las herramientas que más seguridad y rendimiento le den al aplicativo.
- La práctica académica es una oportunidad enorme para afrontar la transición de estudiante a ingeniero, ya que une los aspectos académicos e investigativos de la universidad con los requerimientos y necesidades del mundo laboral.

Referencias Bibliográficas

- [1] PowerData. (2016). Algunas mejores prácticas en migración de aplicaciones. [Online]. Available at: <https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/algunas-mejores-practicas-en-migracion-de-aplicaciones>
- [2] IBM Corp. WebSphere Application Server V8.5 Migration Guide. Primera edición, 2012.
- [3] Oracle. Java SE Development Kit 8 Downloads. [Online]. Available at: <https://www.oracle.com/co/java/technologies/javase/javase-jdk8-downloads.html>
- [4] Apache Maven Project. Welcome to Apache Maven. [Online]. Available at: <https://maven.apache.org/index.html>
- [5] Oracle. Java EE. [Online]. Available at: <https://www.oracle.com/java/technologies/java-ee-glance.html>
- [6] JavaServer Faces. Get started. [Online]. Available at: <http://www.java-serverfaces.org/>
- [7] Icesoft. Icefaces overview. [Online]. Available at: <http://www.icesoft.org/java/projects/ICEfaces/overview.jsf>
- [8] Angular. Introduction to the Angular Docs. [Online]. Available at: <https://angular.io/docs>
- [9] GitLab. GitLab is a complete DevOps platform. [Online]. Available at: <https://about.gitlab.com/>

[10] JasperReport. JasperReport Library- [Online]. Available at: <https://community.jaspersoft.com/>

Archivos Adjuntos

Lecciones aprendidas Migración de SIIU_ADM.docx

Excepción #1 durante ejecución:

GRAVE: Critical error during deployment:

[com.sun.faces.config.ConfigurationException](#):

Source Document:

file:/C:/Users/1214729117/NEONworkspaceSIIUAdm/.metadata/.plugins/org.eclipse.wst.s
erver.core/tmp0/wtpwebapps/udea-siiu_adm/WEB-INF/faces-config.xml

Cause: Unable to find class 'co.edu.udea.siiu.adm.listener.LifeCycleListener'

Solución Error 1:

Se agrega la siguiente dependencia:

```
<dependency>  
  <groupId>com.sun.facelets</groupId>  
  <artifactId>jsf-facelets</artifactId>  
  <version>1.1.14</version>  
  <scope>compile</scope>  
</dependency>
```

Excepción #2 durante ejecución:

GRAVE: Context initialization failed

[org.springframework.beans.factory.BeanDefinitionStoreException](#): Unexpected exception

parsing XML document from ServletContext resource

[/WEB-INF/applicationContext-dataSource.xml]; nested exception is

[org.springframework.beans.FatalBeanException](#): Class

[org.springframework.aop.config.AopNamespaceHandler] for namespace

[http://www.springframework.org/schema/aop] does not implement the

[org.springframework.beans.factory.xml.NamespaceHandler] interface

Solución error #2:

Este error se soluciona agregando unas librerías complementarias.

```
<dependency>  
  <groupId>org.springframework</groupId>  
  <artifactId>spring-aop</artifactId>
```

```
        <version>${spring.version}</version>
</dependency>
<dependency>
    <groupId>cglib</groupId>
    <artifactId>cglib</artifactId>
    <version>2.2</version>
</dependency>
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-aspects</artifactId>
    <version>${spring.version}</version>
</dependency>
<dependency>
    <groupId>commons-dbcp</groupId>
    <artifactId>commons-dbcp</artifactId>
    <version>1.3</version>
</dependency>
<dependency>
    <groupId>org.quartz-scheduler</groupId>
    <artifactId>quartz-oracle</artifactId>
    <version>1.8.6</version>
</dependency>
```

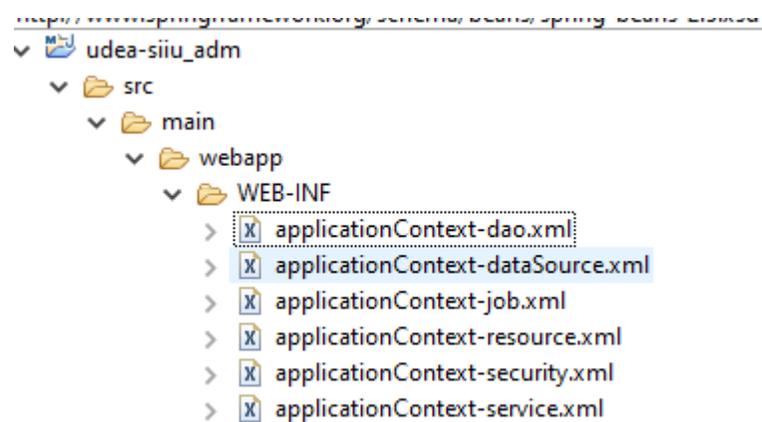
Excepción #3:

El valor de `schemaLocation` = ' <http://www.springframework.org/schema/beans>
<http://www.springframework.org/schema/beans/spring-beans-2.5.xsd>
<http://www.springframework.org/schema/aop>
<http://www.springframework.org/schema/aop/spring-aop-2.5.xsd>
<http://www.springframework.org/schema/context>
<http://www.springframework.org/schema/context/spring-context-2.5.xsd>
<http://www.springframework.org/schema/jms>
<http://www.springframework.org/schema/jms/spring-jms-2.5.xsd>

<http://www.springframework.org/schema/jee>
<http://www.springframework.org/schema/jee/spring-jee-2.5.xsd>
<http://www.springframework.org/schema/tx>
<http://www.springframework.org/schema/tx/spring-tx-2.5.xsd>
<http://www.springframework.org/schema/util>
<http://www.springframework.org/schema/util/spring-util-2.5.xsd>
<http://cxf.apache.org/jaxws> debe tener un número par de URI

Solución #3:

Se debe agregar <http://cxf.apache.org/jaxws.xsd> al final de la sentencia anterior, en los siguientes archivos:



La etiqueta `schemaLocation` debe quedar de la siguiente manera en todos los archivos mencionados anteriormente:

```
xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-2.5.xsd  
http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-2.5.xsd  
http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-2.5.xsd  
http://www.springframework.org/schema/jms http://www.springframework.org/schema/jms/spring-jms-2.5.xsd  
http://www.springframework.org/schema/jee http://www.springframework.org/schema/jee/spring-jee-2.5.xsd  
http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-2.5.xsd  
http://www.springframework.org/schema/util http://www.springframework.org/schema/util/spring-util-2.5.xsd  
http://cxf.apache.org/jaxws http://cxf.apache.org/jaxws.xsd">
```

Error durante ejecución #4:

Deformación de formularios:

Duración del proyecto: desde
24/01/2018
hasta
24/01/2020

Nombre etapa
*

Etapa única

Orden
*

1

Ejecución Presupuestal

Si

Duración

24

mes(es)

Inicio Vigencia
*

23/11/2016

dd/mm/aaaa

Fin Vigencia
*

23/11/2018

dd/mm/aaaa

Descripción
*

Etapa única creada automáticamente

Guardar

Cancelar

Los campos marcados con (*)
son obligatorios

Solución #4:

Este error se genera porque los componentes **ice:panelGroup** y **ice:panelGrid** modifican su comportamiento en la versión 3 y superiores de Icefaces, por lo tanto, se deben modificar todos los **ice:panelGroup** por **h:panelGroup** y cuando haya una composición de componentes como:

```
<ice:panelGroup>
```

```
    <ice:panelGrid>
```

```
        <table> </table>
```

```
</ice:panelGrid>
```

```
</ice:panelGroup>
```

Se debe corregir igualmente el **ice:panelGroup** por **h:panelGroup** y eliminar el **ice:panelGrid**, por lo tanto, quedaría de la siguiente forma:

```
<h:panelGroup>
```

```
    <table> </table>
```

```
</h:panelGroup>
```

Ejemplo en código:

Código anterior:

```
<h:panelGroup>  
    <ice:panelGrid columns="1" width="100%">  
        <table width="100%" border="0" cellpadding="3">  
            <tr>  
                <td colspan="6">
```

Resultado:

Creación/Edición Etapa : 2016-10705 - NOMBRE_CORTO 2016-10705

Duración del proyecto: desde
24/01/2018
hasta
24/01/2020

Nombre etapa
*

Etapa única

Orden
*

1

Código Nuevo:

```

        value="#{texts['Label.etapaProyectoFormTitle']} : #{proyectoController.proye
    </h:panelGroup>
</f:facet>
<h:panelGroup>
    <table width="100%" border="0" cellpadding="3">
        <tr>
            <td colspan="6"><ice:outputText
                value=" Duración del proyecto: desde" /> <h:outputText

```

Resultado:

Error durante ejecución #5:

Usar los componentes html `<p>` `<a>` `<ice:outputText/>` link `<a/>` `</p>` para instanciar un url o acción de navegación no funciona, es decir, se muestra el label del ouputText pero no funciona el link. ejemplo:

```

<p id="homePage">
    <a href="#{facesContext.externalContext.requestContextPath}/gotoPage?target=mainPage" >
        <ice:outputText value="#{texts['Label.volverInicio']}/>
    </a>
</p>

```

Solución #5:

Para que el link funcione se debe cambiar por el componente `<h:commandLink />` el cual debe ir dentro de un `<h:form>` `</h:form>` de esta manera:

```
<h:form>
  <h:commandLink action="mainPage" value="#{texts['Label.volverInicio']}" />
</h:form>
```

Instancias Administrativas

Buscar Instancias Administrativas

Palabra clave

Nivel administrativo

Administra proyecto

--Seleccione--

Si

Consultar Limpiar

Los campos marcados con (*) son obligatorios

Resultados de la Búsqueda

Creación/Edición Instancia Administrativa

[Volver a la página principal](#)

Error Durante Ejecución #6

El Mensaje de error “Los campos marcados con (*) son requeridos” fallan en su formato:

```
Los campos marcados con (<font color=#FF0000>*</font>) son obligatorios
```

Solución #6

Se ajustan componentes para recrear el mensaje en tres outputLabel

Código Anterior (Falla):

```
<tr>
  <td colspan="6" align="left">
    <ice:outputLabel value="#{texts['label.requiredFields']}" />
  </td>
</tr>
```

Código Nuevo:

```

<tr>
  <td align="left" valign="bottom" colspan="3">
    <ice:outputLabel value="#{texts['Label.requiredFields.parteA']}'"/>
    <ice:outputLabel style="color:#FF0000" value="*" />
    <ice:outputLabel value="#{texts['Label.requiredFields.parteB']}'"/>
  </td>
</tr>

```

Resultado:

Los campos marcados con (*) son obligatorios

Migración a JasperReport 6.15.0.docx

Excepción durante ejecución:

net.sf.jasperreports.engine.JRException: Class
com.jaspersoft.jrx.query.PISqlQueryExecuterFactory not found.

```

SEVERE: Fill 1: exception
net.sf.jasperreports.engine.JRException: Class com.jaspersoft.jrx.query.PISqlQueryExecuterFactory not found.
at net.sf.jasperreports.engine.util.JRSingletonCache.createInstance(JRSingletonCache.java:95)
at net.sf.jasperreports.engine.util.JRSingletonCache.getCachedInstance(JRSingletonCache.java:74)
at net.sf.jasperreports.engine.query.DefaultQueryExecuterFactoryBundle.getQueryExecuterFactory(DefaultQueryExecuterFactoryBundle.java:108)
at net.sf.jasperreports.engine.util.JRQueryExecuterUtils.getExecuterFactory(JRQueryExecuterUtils.java:108)
at net.sf.jasperreports.engine.fill.JRFillDataset.createQueryDatasource(JRFillDataset.java:1110)
at net.sf.jasperreports.engine.fill.JRFillDataset.initDatasource(JRFillDataset.java:691)
at net.sf.jasperreports.engine.fill.JRBaseFiller.setParameters(JRBaseFiller.java:1288)
at net.sf.jasperreports.engine.fill.JRBaseFiller.fill(JRBaseFiller.java:907)
at net.sf.jasperreports.engine.fill.JRBaseFiller.fill(JRBaseFiller.java:852)
at net.sf.jasperreports.engine.fill.JRFillSubreport.fillSubreport(JRFillSubreport.java:652)
at net.sf.jasperreports.engine.fill.JRSubreportRunnable.run(JRSubreportRunnable.java:59)
at net.sf.jasperreports.engine.fill.AbstractThreadSubreportRunner.run(AbstractThreadSubreportRunner.java:203)
at java.util.concurrent.ThreadPoolExecutor.runWorker(Unknown Source)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(Unknown Source)
at java.lang.Thread.run(Unknown Source)

```

Solución:

- Este error se genera porque JasperReport necesita un ejecutor de PISql para cuando se ejecutan consultas a procedimientos almacenados desde el reporte. En este caso, el problema se genera al usar la versión 5.5.1 de JasperReports con el mismo ejecutor de PISql que en la versión 3.X.X de esta misma librería, sin embargo, este ejecutor (que no es más que una clase) no existe en esta librería y por eso lanza la excepción vista anteriormente.
- Para solucionar este error se debe setear al ejecutor de PISql definido según la versión pero después de varias horas de investigación se llegó a la conclusión que la versión 5.5.1 de JasperReport no tenía un ejecutor exclusivo para ejecutar Procedimientos Almacenados, por lo tanto, se tendría que migrar a la versión 6.15.0.
- Para la migración se deben seguir los siguientes pasos:
 1. **Cambiar versión de JasperReport en POM por la 6.15.0**

```

<groupId>net.sf.jasperreports</groupId>
<artifactId>jasperreports</artifactId>
<version>6.15.0</version>
<!-- <version>5.5.1</version> -->

```

2. Cambiar el método `JRLoader.loadObject(string)`; por `(JasperReport) JRLoader.loadObjectFromFile(string)`; en todas las clases JAVA donde se necesite:

Antes de migrar:

```

if(getTipoProyectoNombre().equals("Investigación")){
   .jasperReport = (JasperReport) JRLoader.loadObject(appPath + "WEB-INF/jasper/Acta.jasper");
}
else if(getTipoProyectoNombre().equals("Extensión")){
   .jasperReport = (JasperReport) JRLoader.loadObject(appPath + "WEB-INF/jasper/Acta_extension.jasper");
}
else{
   .error("No se encuentra el tipo de proyecto");
}

```

Después de migrar 6.15.0:

```

if(getTipoProyectoNombre().equals("Investigación")){
   .jasperReport = (JasperReport) JRLoader.loadObjectFromFile(appPath + "WEB-INF/jasper/Acta.jasper");
}
else if(getTipoProyectoNombre().equals("Extensión")){
   .jasperReport = (JasperReport) JRLoader.loadObjectFromFile(appPath + "WEB-INF/jasper/Acta_extension.jasper");
}

```

3. Cambiar la clase `JRProperties` por `JRPropertiesUtil` en todas las clases JAVA que sea necesario de la siguiente manera:

- a. Eliminar importación de `JRProperties`: Se borra la siguiente línea.

```

94 import net.sf.jasperreports.engine.util.JRProperties;
95 //import org.apache.pdfbox.util.PDFMergerUtility;
96

```

- b. Crear objeto `JasperReportsContext` y `JRPropertiesUtil` con la instancia de `JasperReportsContext` :

```

However in my case the export to pdf fails without this line being executed. */
JasperReportsContext.jasperReportsContext = DefaultJasperReportsContext.getInstance();
JRPropertiesUtil jrPropertiesUtil = JRPropertiesUtil.getInstance(jasperReportsContext);

```

- c. Cambiar donde haya `JRProperties.setProperty(String, String)` por `jrPropertiesUtil.setProperty(String, String)`

Antes de migración:

```

JRPropertiesUtil jrPropertiesUtil = JRPropertiesUtil.getInstance(jasperReportsContext);

JRProperties.setProperty(JRQueryExecuterFactory.QUERY_EXECUTER_FACTORY_PREFIX+"plsc

/* Setting ignore missing font property */
JRProperties.setProperty("net.sf.jasperreports.awt.ignore.missing.font", "true");

/* Setting default font property */
JRProperties.setProperty("net.sf.jasperreports.default.font.name", "Sans Serif");

```

Después de migración:

```

jrPropertiesUtil.setProperty(JRQueryExecuterFactory.QUERY_EXECUTER_FACTORY_PREFIX+"plsql", "com.
/* Setting ignore missing font property */
jrPropertiesUtil.setProperty("net.sf.jasperreports.awt.ignore.missing.font", "true");

/* Setting default font property */
jrPropertiesUtil.setProperty("net.sf.jasperreports.default.font.name", "Sans Serif");

```

d. Cambiar la línea

```

jrPropertiesUtil.setProperty(JRQueryExecuterFactory.QUERY_EXECUTER_FACTORY_
PREFIX+"plsql", "com.jaspersoft.jrx.query.PlSqlQueryExecuterFactory");

```

por

```

jrPropertiesUtil.setProperty(QueryExecuterFactory.QUERY_EXECUTER_FACTORY_PR
EFIX + "plsql", "net.sf.jasperreports.engine.query.PlSqlQueryExecuterFactory");

```

```

/*
jrPropertiesUtil.setProperty(
    JRQueryExecuterFactory.QUERY_EXECUTER_FACTORY_PREFIX+"plsql",
    "com.jaspersoft.jrx.query.PlSqlQueryExecuterFactory");
*/

jrPropertiesUtil.setProperty(
    QueryExecuterFactory.QUERY_EXECUTER_FACTORY_PREFIX + "plsql",
    "net.sf.jasperreports.engine.query.PlSqlQueryExecuterFactory");

```

(Lo pongo en varias líneas para que se vea bien)

- e. **Guardar y correr:** Una vez se hayan llevado a cabo los pasos anteriores (En todas las clases donde sea necesario) ya debe funcionar JasperReport y debe levantar el reporte requerido:

Acta_1602864523073.pdf - Google Chrome

localhost:8080/udea-siiu_adm/archivos/Acta_1602864523073.pdf

Acta_1602864523073.pdf 1 / 5


VICERRECTORÍA DE INVESTIGACIÓN
COMITÉ PARA EL DESARROLLO DE LA INVESTIGACIÓN -CODI-
ACTA DE REGISTRO Y DE PROPIEDAD INTELECTUAL

 **Abrir en Acrobat**

Acta N°	2020-18565	Fecha	16/10/2020
1. Información general del Proyecto de Investigación			
Nombre del Proyecto	Pruebas avales en dos etapas completo		
Convocatoria / Proceso de Selección	Nombre	Fecha [dd/mm/yy]	
	Proceso de aval institucional (Dos etapas)	22/01/2020	
Centro de Investigación	Instituto de Investigaciones Médicas		
Documento de	Emisor	N°	Fecha Aprobación