

	<b>MACROPROCESO DE APOYO</b>	CODIGO: AAAr113
	<b>PROCESO GESTION APOYO ACADEMICO</b>	VERSION:1
	<b>DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL</b>	PAGINA: 8 de 8

**FECHA** | jueves, 1 de diciembre de 2016

Señores  
**UNIVERSIDAD DE CUNDINAMARCA**  
 BIBLIOTECA  
 Ciudad

<b>SEDE/SECCIONAL/EXTENSIÓN</b>	Sede Fusagasugá
<b>DOCUMENTO</b>	Otro
<b>FACULTAD</b>	Ingeniería
<b>NIVEL ACADÉMICO DE FORMACIÓN O PROCESO</b>	Pregrado
<b>PROGRAMA ACADÉMICO</b>	Ingeniería de Sistemas

El Autor(Es):

<b>APELLIDOS COMPLETOS</b>	<b>NOMBRES COMPLETOS</b>	<b>NO. DOCUMENTO DE IDENTIFICACIÓN</b>
Villalobos Delgado	Hector Alveiro	1.069.745.515

Director(Es) del documento:

	<b>MACROPROCESO DE APOYO</b>	CODIGO: AAAR113
	<b>PROCESO GESTION APOYO ACADEMICO</b>	VERSION:1
	<b>DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL</b>	PAGINA: 8 de 8

<b>APELLIDOS COMPLETOS</b>	<b>NOMBRES COMPLETOS</b>
Sotelo Cubillos	Jose Fernando

<b>TÍTULO DEL DOCUMENTO</b>
Sistema de información para el control y verificación de los soportes y evidencias de los grupos de investigación de la facultad de ingeniería

<b>SUBTÍTULO</b> (Aplica solo para Tesis, Artículos Científicos, Disertaciones, Objetos Virtuales de Aprendizaje)

<b>TRABAJO PARA OPTAR AL TITULO DE:</b> Aplica para Tesis/Trabajo de Grado/Pasantía

<b>AÑO DE EDICION DEL DOCUMENTO</b>	<b>NÚMERO DE PÁGINAS (Opcional)</b>
Haga clic aquí o pulse para escribir una fecha.	

**DESCRIPTORES O PALABRAS CLAVES EN ESPAÑOL E INGLES: (Usar como mínimo 6 descriptores)**

<b>ESPAÑOL</b>	<b>INGLES</b>
1. Programación	Programming
2. Java Web	Java Web
3. Patrones de diseño	Design Patterns
4. JSF (Java Server Faces)	JSF (Java Server Faces)
5. Sistematización	Systematization
6. Metodologías ágiles	Agile Methodologies
7. SCRUM	SCRUM

**RESUMEN DEL CONTENIDO EN ESPAÑOL E INGLES: (Máximo 250 palabras – 1530 caracteres):**

Ante la necesidad de la facultad de ingeniería de la universidad de Cundinamarca sede Fusagasugá de contar con un sistema de información que

	<b>MACROPROCESO DE APOYO</b>	CODIGO: AAAR113
	<b>PROCESO GESTION APOYO ACADEMICO</b>	VERSION:1
	<b>DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL</b>	PAGINA: 8 de 8

les permitirá el control y verificación de la documentación de los productos creados por los grupos de investigación se define la creación de un software que pueda cumplir con esta tarea, ya que actualmente este proceso no se realiza de la forma adecuada, debido a que los grupos de investigación en ocasiones no entregan las evidencias necesarias para que puedan soportar la existencia de los productos, provocando que estos no puedan ser tenidos en cuenta por Colciencias debido a que no cumplen con los requisitos de existencia necesarios para su soporte.

La creación de este sistema de información se realizó utilizando tecnologías de programación JAVA Web haciendo uso de Framework como JSF (Java Server Faces) Primefaces, Bootstrap y patrones de diseño como MVC (Modelo vista Controlador), Value Object, DAO (Data Access Object) y trabajando conjuntamente con metodologías ágiles como SCRUM y xp (eXtreme Programming) para su gestión y desarrollo.

Faced with the need for engineering of the University of Cundinamarca Fusagasugá headquarters to have an information system that control and verification of the documentation of products created by research groups defined the creation of software that can meet this This process is not currently carried out in the proper way, because research groups sometimes do not provide the evidence necessary for buyers to have the existence of the products, because they do not comply with the Requisites of existence necessary for its support.

The creation of this information system is done using JAVA Web programming technologies to use as the framework as JSF (Java Server Faces), value objects, data access object (DAO) AND working in conjunction with agile methodologies such as SCRUM and xp (EXtreme programming) for its management and development.

### AUTORIZACION DE PUBLICACIÓN

Por medio del presente escrito autorizo (Autorizamos) a la Universidad de Cundinamarca para que, en desarrollo de la presente licencia de uso parcial, pueda ejercer sobre mí (nuestra) obra las atribuciones que se indican a continuación, teniendo en cuenta que, en cualquier caso, la finalidad perseguida será facilitar, difundir y promover el aprendizaje, la enseñanza y la investigación.

	<b>MACROPROCESO DE APOYO</b>	CODIGO: AAAr113
	<b>PROCESO GESTION APOYO ACADEMICO</b>	VERSION:1
	<b>DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL</b>	PAGINA: 8 de 8

En consecuencia, las atribuciones de usos temporales y parciales que por virtud de la presente licencia se autoriza a la Universidad de Cundinamarca, a los usuarios de la Biblioteca de la Universidad; así como a los usuarios de las redes, bases de datos y demás sitios web con los que la Universidad tenga perfeccionado un alianza, son:

Marque con una "x":

<b>AUTORIZO (AUTORIZAMOS)</b>	<b>SI</b>	<b>NO</b>
1. La conservación de los ejemplares necesarios en la Biblioteca.	X	
2. La consulta física o electrónica según corresponda.	X	
3. La reproducción por cualquier formato conocido o por conocer.	X	
4. La comunicación pública por cualquier procedimiento o medio físico o electrónico, así como su puesta a disposición en Internet.	X	
5. La inclusión en bases de datos y en sitios web sean éstos onerosos o gratuitos, existiendo con ellos previa alianza perfeccionada con la Universidad de Cundinamarca para efectos de satisfacer los fines previstos. En este evento, tales sitios y sus usuarios tendrán las mismas facultades que las aquí concedidas con las mismas limitaciones y condiciones.	X	
6. La inclusión en el Repositorio Institucional.	X	

De acuerdo con la naturaleza del uso concedido, la presente licencia parcial se otorga a título gratuito por el máximo tiempo legal colombiano, con el propósito de que en dicho lapso mi (nuestra) obra sea explotada en las condiciones aquí estipuladas y para los fines indicados, respetando siempre la titularidad de los derechos patrimoniales y morales correspondientes, de acuerdo con los usos honrados, de manera proporcional y justificada a la finalidad perseguida, sin ánimo de lucro ni de comercialización.

Para el caso de las Tesis, Trabajo de Grado o Pasantía, de manera complementaria, garantizo(garantizamos) en mi(nuestra) calidad de estudiante(s) y por ende autor(es) exclusivo(s), que la Tesis, Trabajo de Grado o Pasantía en cuestión, es producto de mi(nuestra) plena autoría, de mi(nuestro) esfuerzo personal intelectual, como consecuencia de mi(nuestra) creación original particular y, por tanto, soy(somos) el(los) único(s) titular(es) de la misma. Además, aseguro (aseguramos) que no contiene citas, ni transcripciones de otras obras protegidas, por fuera de los límites autorizados por la ley, según los usos honrados, y en proporción a los fines previstos; ni tampoco contempla declaraciones difamatorias contra terceros; respetando el derecho a la imagen, intimidad, buen nombre y

	<b>MACROPROCESO DE APOYO</b>	CODIGO: AAAR113
	<b>PROCESO GESTION APOYO ACADEMICO</b>	VERSION:1
	<b>DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL</b>	PAGINA: 8 de 8

demás derechos constitucionales. Adicionalmente, manifiesto (manifestamos) que no se incluyeron expresiones contrarias al orden público ni a las buenas costumbres. En consecuencia, la responsabilidad directa en la elaboración, presentación, investigación y, en general, contenidos de la Tesis o Trabajo de Grado es de mí (nuestra) competencia exclusiva, eximiendo de toda responsabilidad a la Universidad de Cundinamarca por tales aspectos.

Sin perjuicio de los usos y atribuciones otorgadas en virtud de este documento, continuaré (continuaremos) conservando los correspondientes derechos patrimoniales sin modificación o restricción alguna, puesto que, de acuerdo con la legislación colombiana aplicable, el presente es un acuerdo jurídico que en ningún caso conlleva la enajenación de los derechos patrimoniales derivados del régimen del Derecho de Autor.

De conformidad con lo establecido en el artículo 30 de la Ley 23 de 1982 y el artículo 11 de la Decisión Andina 351 de 1993, “*Los derechos morales sobre el trabajo son propiedad de los autores*”, los cuales son irrenunciables, imprescriptibles, inembargables e inalienables. En consecuencia, la Universidad de Cundinamarca está en la obligación de RESPETARLOS Y HACERLOS RESPETAR, para lo cual tomará las medidas correspondientes para garantizar su observancia.

**NOTA:** (Para Tesis, Trabajo de Grado o Pasantía):

**Información Confidencial:**

Esta Tesis, Trabajo de Grado o Pasantía, contiene información privilegiada, estratégica, secreta, confidencial y demás similar, o hace parte de la investigación que se adelanta y cuyos resultados finales no se han publicado. **SI \_\_ NO X.**  
En caso afirmativo expresamente indicaré (indicaremos), en carta adjunta tal situación con el fin de que se mantenga la restricción de acceso.

**LICENCIA DE PUBLICACIÓN**

Como titular(es) del derecho de autor, confiero(erimos) a la Universidad de Cundinamarca una licencia no exclusiva, limitada y gratuita sobre la obra que se integrará en el Repositorio Institucional, que se ajusta a las siguientes características:

a) Estará vigente a partir de la fecha de inclusión en el repositorio, por un plazo de 5 años, que serán prorrogables indefinidamente por el tiempo que dure el derecho patrimonial del autor. El autor podrá dar por terminada la licencia solicitándolo a la Universidad por escrito. (Para el caso de los Recursos Educativos Digitales, la Licencia de Publicación será permanente).

	<b>MACROPROCESO DE APOYO</b>	CODIGO: AAAr113
	<b>PROCESO GESTION APOYO ACADEMICO</b>	VERSION:1
	<b>DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL</b>	PAGINA: 8 de 8

b) Autoriza a la Universidad de Cundinamarca a publicar la obra en formato y/o soporte digital, conociendo que, dado que se publica en Internet, por este hecho circula con un alcance mundial.

c) Los titulares aceptan que la autorización se hace a título gratuito, por lo tanto, renuncian a recibir beneficio alguno por la publicación, distribución, comunicación pública y cualquier otro uso que se haga en los términos de la presente licencia y de la licencia de uso con que se publica.

d) El(Los) Autor(es), garantizo(amos) que el documento en cuestión, es producto de mi(nuestra) plena autoría, de mi(nuestro) esfuerzo personal intelectual, como consecuencia de mi (nuestra) creación original particular y, por tanto, soy(somos) el(los) único(s) titular(es) de la misma. Además, aseguro(aseguramos) que no contiene citas, ni transcripciones de otras obras protegidas, por fuera de los límites autorizados por la ley, según los usos honrados, y en proporción a los fines previstos; ni tampoco contempla declaraciones difamatorias contra terceros; respetando el derecho a la imagen, intimidad, buen nombre y demás derechos constitucionales. Adicionalmente, manifiesto (manifestamos) que no se incluyeron expresiones contrarias al orden público ni a las buenas costumbres. En consecuencia, la responsabilidad directa en la elaboración, presentación, investigación y, en general, contenidos es de mí (nuestro) competencia exclusiva, eximiendo de toda responsabilidad a la Universidad de Cundinamarca por tales aspectos.

e) En todo caso la Universidad de Cundinamarca se compromete a indicar siempre la autoría incluyendo el nombre del autor y la fecha de publicación.

f) Los titulares autorizan a la Universidad para incluir la obra en los índices y buscadores que estimen necesarios para promover su difusión.

g) Los titulares aceptan que la Universidad de Cundinamarca pueda convertir el documento a cualquier medio o formato para propósitos de preservación digital.

h) Los titulares autorizan que la obra sea puesta a disposición del público en los términos autorizados en los literales anteriores bajo los límites definidos por la universidad en las "Condiciones de uso de estricto cumplimiento" de los recursos publicados en Repositorio Institucional, cuyo texto completo se puede consultar en [biblioteca.unicundi.edu.co](http://biblioteca.unicundi.edu.co)

i) Para el caso de los Recursos Educativos Digitales producidos por la Oficina de Educación Virtual, sus contenidos de publicación se rigen bajo la Licencia Creative Commons : Atribución- No comercial- Compartir Igual.

	<b>MACROPROCESO DE APOYO</b>	CODIGO: AAAR113
	<b>PROCESO GESTION APOYO ACADEMICO</b>	VERSION:1
	<b>DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL</b>	PAGINA: 8 de 8



j) Para el caso de los Artículos Científicos y Revistas, sus contenidos se rigen bajo la Licencia Creative Commons Atribución- No comercial- Sin derivar.



**Nota:**

Si el documento se basa en un trabajo que ha sido patrocinado o apoyado por una entidad, con excepción de Universidad de Cundinamarca, los autores garantizan que se ha cumplido con los derechos y obligaciones requeridos por el respectivo contrato o acuerdo.

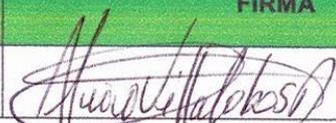
La obra que se integrará en el Repositorio Institucional, está en el(los) siguiente(s) archivo(s).

Nombre completo del Archivo Incluida su Extensión (Ej. Título Trabajo de Grado o Documento.pdf)	Tipo de documento (ej. Texto, imagen, video, etc.)
1. ACTA No1 _ director.pdf	Texto
2. ACTA No2 _ Facultad.pdf	Texto
3. ACTA No3 _ Facultad.pdf	Texto
4. ACTA No4 _ Facultad.pdf	Texto
5. ACTA No5 _ Facultad.pdf	Texto
6. ACTA No6 _ Facultad.pdf	Texto
7. ACTA No7 _ Facultad.pdf	Texto
8. ACTA No8 _ Facultad.pdf	Texto
9. ACTA No1 _ Director.pdf	Texto
10. ACTA No2 _ Director.pdf	Texto
11. ACTA No3 _ Director.pdf	Texto
12. ACTA No4 _ Director.pdf	Texto
13. ACTA No5 _ Director.pdf	Texto
14. ACTA No6 _ Director.pdf	Texto
15. ACTA No7 _ Director.pdf	Texto
16. ACTA No8 _ Director.pdf	Texto
17. ACTA No9 _ Director.pdf	Texto
18. ACTA No10 _ Director.pdf	Texto

	<b>MACROPROCESO DE APOYO</b>	CODIGO: AAAr113
	<b>PROCESO GESTION APOYO ACADEMICO</b>	VERSION:1
	<b>DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL</b>	PAGINA: 8 de 8

19. Artículo_ Sistema de información para el control y verificación de los soportes y evidencias de los grupos de investigación de la facultad de ingeniería.doc	Texto
20. Artículo_ Sistema de información para el control y verificación de los soportes y evidencias de los grupos de investigación de la facultad de ingeniería.pdf	Texto
21. Sistema de información para el control y verificación de los soportes y evidencias de los grupos de investigación de la facultad de ingeniería.pdf	Texto
22. Sistema de información para el control y verificación de los soportes y evidencias de los grupos de investigación de la facultad de ingeniería.doc	Texto
23. modelo relacional.png	Imagen
24. modelo entidad relación.png	Imagen
25. GrupControl.war	Ejecutable
26. grupcontrol.sql	Texto
27. Manual de usuario.pdf	Texto
28. Manual técnico.pdf	Texto
29. Manual implementación servidor.pdf	Texto

En constancia de lo anterior, Firmo (amos) el presente documento:

APELLIDOS Y NOMBRES COMPLETOS	FIRMA
Villalobos Delgado Hector Alveiro	

SISTEMA DE INFORMACION PARA EL CONTROL Y VERIFICACION DE LOS  
SOPORTES y EVIDENCIAS DE LOS GRUPOS DE INVESTIGACION DE LA  
FACULTAD DE INGENIERIA

HECTOR ALVEIRO VILLALOBOS DELGADO

CÓDIGO: 161211240

DIRECTOR

JOSE FERNANDO SOTELO CUBILLOS

C.C 79384003



UNIVERSIDAD DE CUNDINAMARCA

FUSAGASUGÁ

2016

# Contenido

Introducción .....	3
Antecedentes .....	4
Dipòsit Digital – Universidad de Barcelona .....	4
Proyecto Academia Institucional – Universidad de los andes .....	4
Alfresco.....	4
Athenea .....	4
Tecnologías utilizadas .....	5
JSF (Java Server Faces) .....	5
Bootstrap.....	6
PrimeFaces .....	6
Mysql.....	7
Apache tomcat .....	7
JUnit.....	7
Metodología.....	8
Aplicación de la metodología XP (Extreme programing) .....	11
Valores de XP.....	11
Fase 1 – planeación.....	13
Fase 2 – Diseño.....	15
Fase 3 – Desarrollo .....	18
Utilización de patrones de diseño.....	19
Patrón MVC (modelo visto controlador).....	19
Patrón value object .....	21
Patrón DAO (Data Access object).....	22
Fase 4 –pruebas .....	24
Pruebas de aceptación .....	24
Pruebas unitarias.....	25
Conclusiones .....	31
Bibliografía .....	33
Anexos.....	34

## **Introducción**

Ante la necesidad de la facultad de ingeniería de la universidad de Cundinamarca sede Fusagasugá de contar con un sistema de información que les permitirá el control y verificación de la documentación de los productos creados por los grupos de investigación se define la creación de un software que pueda cumplir con esta tarea, ya que actualmente este proceso no se realiza de la forma adecuada, debido a que los grupos de investigación en ocasiones no entregan las evidencias necesarias para que puedan soportar la existencia de los productos, provocando que estos no puedan ser tenidos en cuenta por Colciencias debido a que no cumplen con los requisitos de existencia necesarios para su soporte.

La creación de este sistema de información se realizó utilizando tecnologías de programación java haciendo uso de framework y patrones de diseño y trabajando conjuntamente con metodologías ágiles como scrum y xp para su gestión y desarrollo.

## **Antecedentes**

### **Dipòsit Digital – Universidad de Barcelona**

El Dipòsit Digital de la Universitat de Barcelona es el repositorio institucional que contiene en formato digital las publicaciones en acceso abierto derivadas de la actividad docente, investigadora e institucional del profesorado y otros miembros de la comunidad universitaria. [1]

### **Proyecto Academia Institucional – Universidad de los andes**

El Proyecto Academia Institucional tiene como propósito apoyar procesos de gestión y divulgación de la investigación en Uniandes. Profesores, estudiantes de doctorado y personal administrativo de la Universidad de los Andes podrán consultar, generar reportes y administrar la información relacionada con la producción intelectual de los profesores a través de Academia. [2]

### **Alfresco**

Es un software de gestión documental para documentos, páginas web, registros, imágenes y desarrollo colaborativo de contenido. Es un sistema de administración de contenidos de código fuente libre, desarrollado en Java.

En la universidad uniminuto, es utilizado como un repositorio de contenidos que contiene bases de datos referentes a actas, hojas de vida y otros soportes públicos con información desde el año 2011, sobre la investigación a nivel de toda la Sede, desde las políticas y resoluciones que rigen la investigación hasta las actas de los comités de investigación de cada unidad académica, lo anterior para la investigación aplicada como formativa. [3]

### **Athenea**

Es una plataforma de información piloto del Sistema Universitario Minuto de Dios, que funciona como una base de datos de consulta de los registros de la investigación de la institución.

Aquí el usuario encontrará información referente a investigadores y estudiantes de semilleros, como hojas de vida, perfiles de grupos de investigación, perfiles

de semilleros de investigación, registro de producto de investigación, tales como libros, capítulos de libro, artículos, ponencias y proyectos, entre otros. [3]

## **Tecnologías utilizadas**

### **JSF (Java Server Faces)**

El proyecto está desarrollado en lenguaje de programación JAVA así que para su desarrollo se utilizó el framework JSF, se utilizó este framework debido que este implementa directamente el patrón MVC (modelo vista controlador) además de contar con una buena documentación y una comunidad de desarrollo activa.

Además de los framework de trabajo web de JAVA, JSF es uno con la curva de aprendizaje de más fácil comprensión, permitiendo así adaptarse a él más fácilmente a comparación de frameworks como sprint mvc o strust.

Entre las principales ventajas de JSF se encuentran:

1. Una clara separación entre vista y modelo.
2. Desarrollo basado en componente, no en peticiones.
3. Las acciones del usuario se ligan muy fácilmente al código en el servidor.
4. Creación de familias de componentes visuales para acelerar el desarrollo.

Las aplicaciones JSF están formadas por los siguientes elementos principales:

- Páginas JSP, XHTML, HTML que incluyen los formularios JSF. Estas páginas generarán las vistas de la aplicación
- Beans java que se conectan con los formularios JSF
- Clases java para la lógica de negocio y utilidades.
- Ficheros de configuración, componentes a medida y otros elementos del framework.
- Resto de recursos de la aplicación web: recursos estáticos, JavaScript y otros elementos. [4]

## **Bootstrap**

Para el apartado gráfico de la aplicación se utilizó el framework de diseño bootstrap en su versión 3

Bootstrap, es un framework originalmente creado por Twitter, que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice.

A la hora de crear interfaces web, los diseños creados con Bootstrap son simples, limpios e intuitivos, esto les da agilidad a la hora de cargar y al adaptarse a otros dispositivos. El Framework trae varios elementos con estilos predefinidos fáciles de configurar: Botones, Menús desplegables, Formularios incluyendo todos sus elementos e integración jQuery para ofrecer ventanas y tooltips dinámicos. [5]

## **PrimeFaces**

Es una librería de componentes para JavaServer Faces (JSF) de código abierto que cuenta con un conjunto de componentes enriquecidos que facilitan la creación de las aplicaciones web. Primefaces está bajo la licencia de Apache License V2

Las principales características de Primefaces son:

- soporte nativo de Ajax, incluyendo Push/Comet.
- kit para crear aplicaciones web para móviles.
- es compatible con otras librerías de componentes, como JBoss RichFaces.
- uso de javascript no intrusivo (no aparece en línea dentro de los elementos, sino dentro de un bloque <script>).
- es un proyecto open source, activo y bastante estable entre versiones. [6]

## **Mysql**

MySQL es el servidor de bases de datos relacionales más popular, desarrollado y proporcionado por MySQL AB. MySQL AB es una empresa cuyo negocio consiste en proporcionar servicios en torno al servidor de bases de datos MySQL. [7]

## **Apache tomcat**

Tomcat es un contenedor de servlets que se utiliza en la Referencia oficial de la implementación para Java Servlet y JavaServer Pages (JSP). Las especificaciones Java Servlet y JavaServer Pages son desarrolladas por Sun Microsystems cuyas especificaciones vienen dadas por la JCP (Java Community Process). Apache Tomcat es desarrollado en un entorno abierto y participatorio, bajo la licencia de Apache Software License.

Tomcat es mantenido y desarrollado por miembros de la Apache Software Foundation y voluntarios independientes. Los usuarios disponen de libre acceso a su código fuente y a su forma binaria en los términos establecidos en la Apache Software License. Las primeras distribuciones de Tomcat fueron las versiones 3.0.x. Las versiones más recientes son las 7.x, que implementan las especificaciones de Servlet 3.0 y de JSP 2.2. A partir de la versión 4.0, Jakarta Tomcat utiliza el contenedor de servlets Catalina.

Tomcat es un servidor web con soporte de servlets y JSPs. Tomcat no es un servidor de aplicaciones, como JBoss o JOnAS. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor web Apache. [8]

## **JUnit**

JUnit es un conjunto de bibliotecas creadas por Erich Gamma y Kent Beck que son utilizadas en programación para hacer pruebas unitarias de aplicaciones Java.

JUnit es un conjunto de clases (framework) que permite realizar la ejecución de clases Java de manera controlada, para poder evaluar si el funcionamiento de

cada uno de los métodos de la clase se comporta como se espera. Es decir, en función de algún valor de entrada se evalúa el valor de retorno esperado; si la clase cumple con la especificación, entonces JUnit devolverá que el método de la clase pasó exitosamente la prueba; en caso de que el valor esperado sea diferente al que regresó el método durante la ejecución, JUnit devolverá un fallo en el método correspondiente [9]

## **Metodología**

Para el desarrollo del proyecto se hizo uso de las metodologías ágiles como SCRUM y XP (Extreme programming), la razón de la utilización de estas metodologías en lugar de usar las metodologías tradicionales son las ventajas que esta proporcionan en la gestión del proyecto

Debido a que el objetivo principal del proyecto es el diseño y desarrollo de un software, en palabras más específicas un sistema de información para la verificación y control de los grupos de investigación el uso de estas metodologías proporcionan una serie de instrumentos que facilitan la realización de las diferentes fases y tareas correspondientes al desarrollo de un software.

La asignación de roles que tendrá cada miembro en la ejecución del proyecto en el uso la metodología ágil SCRUM, en este orden de ideas los roles dentro del desarrollo del proyecto quedaron asignados de la siguiente forma

Product owner (dueño de producto) este rol en el proyecto es desempeñado por la facultad de ingeniería de la universidad Cundinamarca sede Fusagasugá la cual a representación de esta está el ingeniero HUMBERTO NUNPAQUE director de programa de ingeniería electrónica y que a su vez desempeña funciones de líder de grupo investigación como de verificador de la documentación de grupos de investigación correspondientes a su programa por este motivo es la persona que cuenta con los conocimientos más acertados en razón del cliente sobre las tareas que debe de realizar el software.

Scrum master este rol es desempeñado por el ingeniero Fernando Sotelo y el estudiante Alveiro Villalobos Delgado el cual está en constante comunicación con el product owner y team de desarrollo en los temas referentes a la gestión

del proyecto, llevando la voz de cada una de estas partes y gestionando lo mejor posible las tareas para obtener un producto que la satisfaga los requisitos del cliente.

Team desarrollador este rol es cumplido por el estudiante Alveiro Villalobos Delgado el cual se encargó del desarrollo del software plasmando los requisitos recolectados del cliente ante su necesidad, bajo las especificaciones del cliente en términos de requisitos y uso de tecnologías indicadas para la creación del software (java web, servidor apache tomcat, base de datos Mysql).

Se hizo uso de las historias usuario como instrumento para levantamiento de requisitos del software, ya que estas, se realizan con un lenguaje no técnico haciéndolas entendibles para cualquier persona, esto permite que tanto como el cliente como el equipo desarrollador entiendan fácilmente los requisitos implícitos en estas, además de incorporar características complementarias como los criterios mínimos de aceptación y el nivel de importancia que estas tiene dentro del proyecto , esto permite que el equipo desarrollador conozca de los criterios de aceptación plasmados por el cliente para este requisito y la importancia que este tiene a nivel de realización del proyecto, y de esta forma poder delegar los recursos necesarios para la realización optima de este requisito.

A comparación de su equivalente en las metodologías tradicionales que seria los casos de uso, las historias de usuario expresan lo más claro posible y en lenguaje muy coloquial para ambas partes lo que el cliente quiere de esta parte de proyecto permitiendo poder obtener el resultado correcto con lo que el cliente desea. Por estos motivos se implementó este instrumento para el levantamiento de información en la realización del proyecto a través del siguiente formato:

3 - Creación de login	
Como cliente quiero que una interfaz de logeo donde el usuario ingresara al sistema con su correspondiente rol	
<b>Criterios de aceptación</b>	
<ul style="list-style-type: none"> <li>• Cada usuario podrá ingresar al sistema con su correspondiente nombre de usuario y contraseña</li> <li>• No permitirá que usuarios no registrados ingresen al sistema</li> </ul>	
<b>Estimación:</b>	<b>Dependencia:</b> N/A

En donde vemos reflejado en un lenguaje no técnico y de fácil comprensión los requisitos y criterios que tendría esta parte del sistema de información a realizar.

Seguidamente al levantamiento de requisitos a través de las historias de usuario pasamos a creación del product backlog en este priorizamos todos los requisitos recolectados y ordenados a razón de la prioridad de realización que tendrá cada uno de los elementos identificados, el product backlog del proyecto tiene la siguiente estructura

Figura 1 – Product Backlog (Anexos)

Ya identificados y ordenados los requisitos continuamos con la creación de los sprints, en estos se dividen las tareas a realizar en lista de tareas más pequeñas llamadas iteraciones en las cuales se establecen las actividades a realizar y el tiempo estimado que están tendrán para su realización, como recomendación se expresa que el tiempo establecido no supere las 4 semanas ya que se estima que este tiempo ya se puede presentar un incremento en la realización del proyecto debemos de tener en cuenta que con la terminación de cada iteración se debe de obtener una parte del proyecto ya funcional la cual se presentará al cliente como avance en el desarrollo del proyecto.

Figura 2 – Sprint Backlog (Anexos)

## **Aplicación de la metodología XP (Extreme programming)**

La metodología XP se aplicó conjuntamente con la metodología SCRUM, como observamos anteriormente SCRUM fue utilizada mayormente como metodología de organización del proyecto en cambio XP está enfocada como metodología de desarrollo, de esta se hizo uso de algunos de sus componentes como:

### **Valores de XP**

Principalmente la metodología xp tiene como fundamento 5 valores los cuales tiene como objetivo el crear un ambiente de trabajo cómodo para las partes que participan en ella y de esta forma entender que el tener una relación óptima entre las partes brinda la posibilidad de lograr un producto que sea satisfactoria para todos.

#### Comunicación

Es de suma importancia que exista una buena comunicación entre el equipo desarrollador y el cliente debido a que a través de esta, que se obtiene la mayor cantidad de información útil para el desarrollo óptimo del producto.

#### Simplicidad

Lo ideal es mantener la sencillez en el proceso de codificación de esta forma se puede agilizar desarrollo y facilitar el mantenimiento del producto brindándole al equipo desarrollador la facilidad de comprender el código implementado de una forma sencilla.

#### Retroalimentación feedback

La retroalimentación debe funcionar en forma permanente. El cliente debe brindar retroalimentación de las funciones desarrolladas, de manera de poder tomar sus comentarios para la próxima iteración, y para comprender, cada vez más, sus necesidades.

Los resultados de las pruebas unitarias son también una retroalimentación permanente que tienen los desarrolladores acerca de la calidad de su trabajo.

### Coraje

El equipo debe tener la valentía de tomar decisiones sobre la realización del proyecto desde decisiones que tengan que ver directamente con el proceso de codificación como con decisiones que incluyan al cliente esto con fin de siempre velar con la mejor realización posible del proyecto

### Respeto

Debe existir el mayor respeto posible entre las partes involucradas en el proyecto, esto con el fin de garantizar el mejor ambiente de trabajo posible. Siempre teniendo en cuenta que lo importante es la realización óptima del proyecto

Debido al número reducido de personas en el proyecto solo se hizo uso de algunas de las prácticas de xp, gracias a la flexibilidad de xp como metodología esta permite ser modificada según las necesidades que presente el proyecto pero con la premisa de conservar su esencia.

## **Fase 1 – planeación**

### Historias de usuario – iteraciones

Las historias de usuario fueron levantadas anteriormente a través de la metodología scrum para la creación del product backlog y el sprint backlog, se utilizó esta misma organización de los Sprints para la creación de las iteraciones de XP

### Velocidad del proyecto

Se indica cantidad de historias de usuario que serán desarrolladas y tiempo estimado que estas pueden tardar en su realización, esto con fin de tener un estimado del tiempo de realización del proyecto y así mismo poder ofrecerle al cliente fechas de entrega de avance del proyecto en donde se le presentan resultados ya funcionales e incrementales

### Velocidad de tiempo estimada inicial

Como se puede visualizar en la tabla las iteraciones se agruparon de parejas, esto es debido al número limitado de personas en el proyecto por ende se modificó esta parte de la metodología, esta modificación también está apoyada en la utilización a su par de la metodología SCRUM la cual nos permite realizar varias tareas o incluso sprint completos en el mismo periodo de tiempo.

Al realizar la estimación de la velocidad del proyecto se les dio un valor similar a las semanas y horas semanales invertidas por cada pareja de iteración, pero esto se ve modificado, en los resultados reales de la velocidad de proyecto obtenidos en el proceso de realización de este

Iteraciones	Semanas	Horas semanales	Historias de usuario
Iteración 1 y 2	1	30	12
Iteración 3 y 4	1	30	17
Iteración 5 y 6	1	30	16
Iteración 7 y 8	1	30	12
Iteración 9 y 10	1	30	16
Iteración 11 y 12	1	30	12
Iteración 13 y 14	1	30	8

### Velocidad del proyecto real

En la velocidad del proyecto real se puede observar que la cantidad de semanas en algunas de las iteraciones se ve aumentado esto es debido a la dificultad que presentaba la realización de las historias también aumentado la cantidad de horas de programación semanales invertidas por el team de desarrollador en estas para lograr su realización.

Iteraciones	Semanas	Horas semanales	Historias de usuario
Iteración 1 y 2	1	30	12
Iteración 3 y 4	2	45	17
Iteración 5 y 6	2	40	16
Iteración 7 y 8	1	35	12
Iteración 9 y 10	1	34	16
Iteración 11 y 12	1	37	12
Iteración 13 y 14	1	30	8

### Reuniones

Las reuniones con el cliente fueron en promedio cada 2 semanas, estas con fin de mostrar al cliente los avances incrementales que ha tenido el software y a su vez retroalimentarse de las opiniones del cliente en base al avance presentado del proyecto.

## **Fase 2 – Diseño**

El diseño de los elementos del software se realizaron con las especificaciones brindadas por el cliente conservando siempre la esencia de XP de mantener la simplicidad cumpliendo con los requisitos plasmados para estos.

### **Diseño del modelo de la base datos**

Por medio de la información levantada a través de la historias de usuario y conjuntamente con el documento proporcionado por el cliente (facultad de ingeniería de universidad de Cundinamarca sede Fusagasugá) el cual está realizado por Colciencias y contiene los requisitos de existencia necesarios para cada uno de los productos que pueden ser producidos por los grupos de investigación en base a esto se realizó el modelo correspondiente de la base datos.

Iniciamos creando el modelo entidad relación de la base datos en cual se identifica cada una de las entidades que tendrá el modelo con los correspondientes atributos de cada entidad y las relaciones entre las entidades

Figura 3 - Modelo entidad relación (Anexos)

Ya realizado el modelo entidad relación continuamos con la construcción del modelo relacional en el cual identificamos las llaves primarias y foranas de las tablas y las tablas que son resultado de la relación de las entidades

Figura 4 - Modelo relacional (Anexos)

Luego de haber realizado el modelo relacional se continuó con la creación de las tablas correspondientes en motor de bases de datos

Para la elección del motor de base de datos se tomaron en cuenta las características de los principales motores utilizados, realizando una tabla comparativa entre estos

	<b>Mysql</b>	<b>postgresql</b>	<b>oracle</b>
<b>Ventajas</b>	<p>MySQL software es Open Source</p> <p>Bajo costo en requerimientos para debido a su bajo consumo puede ser ejecutado en una máquina con escasos</p> <p>Facilidad de configuración e instalación. Soporta gran variedad de Sistemas Operativos</p> <p>Su conectividad, velocidad, y seguridad hacen de MySQL Server altamente apropiado para acceder bases de datos en Internet</p>	<p>Puede operar sobre distintas plataformas, incluyendo Linux, Windows, Unix, Solaris y MacOS X.</p> <p>Gran capacidad de almacenamiento</p> <p>Soporta los tipos de datos, cláusulas, funciones y comandos de tipo estándar SQL92/SQL99 y extendidos propios de PostgreSQL.</p>	<p>Oracle es el motor de base de datos relacional más usado a nivel mundial</p> <p>El software del servidor puede ejecutarse en multitud de sistemas operativos.</p> <p>Un aceptable soporte</p>
<b>Desventajas</b>	<p>Un gran porcentaje de las utilidades de MySQL no están documentadas</p>	<p>En comparación con MySQL es más lento en inserciones y actualizaciones, ya que cuenta con cabeceras de intersección que no tiene MySQL.</p> <p>Soporte en línea: Hay foros oficiales, pero no hay una ayuda obligatoria.</p>	<p>El mayor inconveniente de Oracle es quizás su precio. Incluso las licencias de Personal Oracle son excesivamente caras,</p> <p>Otro problema es la necesidad de ajustes</p> <p>Un error frecuente consiste en pensar que basta instalar el Oracle en un servidor y enchufar directamente las aplicaciones</p>

		<p>Consume más recursos que MySQL.</p> <p>La sintaxis de algunos de sus comandos o sentencias no es nada intuitiva.</p>	<p>clientes. Un Oracle mal configurado puede ser desesperantemente lento.</p> <p>También es elevado el coste de la formación, y sólo últimamente han comenzado a aparecer buenos libros sobre asuntos técnicos distintos de la simple instalación y administración.</p>
--	--	---	---

Tras el análisis de las ventajas y desventajas que presenta cada uno de los motores de bases datos comparados y teniendo en cuenta que uno de los requisitos del proyecto es la utilización de software libre se decide trabajar con el motor MYSQL

Para la creación de la base de datos y las tablas se utilizó el asistente de interfaz gráfica adminer, asignado como nombre a la base de datos “grupcontrol”

### **Creación de interfaces de usuario**

Las interfaz de usuario están construida en HTML utilizando para esto el framework de diseño bootstrap en su versión 3 y complementando con la librería UI para JSF primefaces en su versión 5.0 la cual ya trae por defecto una serie de elementos prediseñados para trabajar con JSF y sus especificaciones.

¿Por qué escoger bootstrap?

Aunque como se dijo anteriormente primefaces ofrece una serie de componentes listo para trabajar con JSF incluyendo plantillas ya pre diseñadas , éstas no cumplen con todos requisitos necesarios para creación de las interfaces de usuario que son necesarias para el sistema de información como formularios personalizados, colores de interfaz, colores y tamaño de los botones y demás características para la creación de los componentes de la interfaz, el trata de modificar el código de componentes de PRIMEFACES para trabajar con ellos en la aplicación no era una opción viable debido esta es una librería ya compilada es decir es un archivo .JAR que se agregar al proyecto el tratar de descompilar

esta librería sería un trabajo de ingeniería a la inversa lo cual implicaría un gasto de tiempo enorme e ineficaz, por estos motivos se decide trabajar con bootstrap ya que como framework de diseño creado en HTML5, CSS3 y JavaScript cuenta con una gran flexibilidad, permitiéndonos manejar el código de sus elementos de la forma que nos convenga, pudiendo manipular desde los colores utilizados hasta las funciones JavaScript como la utilización de ventanas modales para brindar una mejor interacción del usuario con el software.

En la utilización de colores para las interfaces de usuario se decidió trabajar con colores institucionales aunque este no era un requisito obligatorio para la construcción de interfaces se decide la utilización de estos, para que el sistema mantenga una línea acorde al diseño de las demás interfaces que hacen parte del ambiente virtual actual de la universidad de Cundinamarca

No solucionar antes de tiempo

El equipo desarrollador solo se enfocó en la solución de los requisitos actuales por iteración esto debido a que la metodología nos propone solucionar solo los inconvenientes que se van presentando y que son directamente relacionados con los requisitos ya que el pensar en solucionar errores o problemas que no han surgido en el momento provoca el gasto de tiempo y recursos en algo que no ha sucedido y que probablemente no se vaya a utilizar.

### **Fase 3 – Desarrollo**

Cliente siempre presente

Aunque la metodología sugiere que el cliente este siempre en el desarrollo del proyecto en este caso se modificó un poco este aspecto de la metodología debido a que el cliente no contaba con el tiempo, para estar siempre presente en el desarrollo se acordó el mostrar avances en periodos de tiempo de 2 semanas en donde ya se tendrían resultados de las iteraciones programadas para este periodo de tiempo.

## Integración frecuentes

Tras la terminación de cada historia de usuario esta iba siendo integrada al proyecto cumpliendo con una integración frecuente y un avance incremental de este.

## Programación en parejas

Esta pauta no pudo ser implementada debido a que el team desarrollador solo conto con un solo programador el estudiante Alveiro Villalobos delgado

## Estándares y propiedad colectiva del código

Debido a que el team desarrollador solo está conformado por una sola persona no se tuvo problema de versiones diferentes de implementación de código,.

Como también se trabajó apoyándose en frameworks como JSF y BOOTSTRAP y conjuntamente implementando patrones de diseño como MVC (modelo – vista - controlador), DAO estos aportaron ala estandarización del código desde su arquitectura

## **Utilización de patrones de diseño**

Para la creación de la aplicación se hizo uso de los siguientes patrones de diseño esto con el fin de tener una arquitectura de software lo más solidad posible.

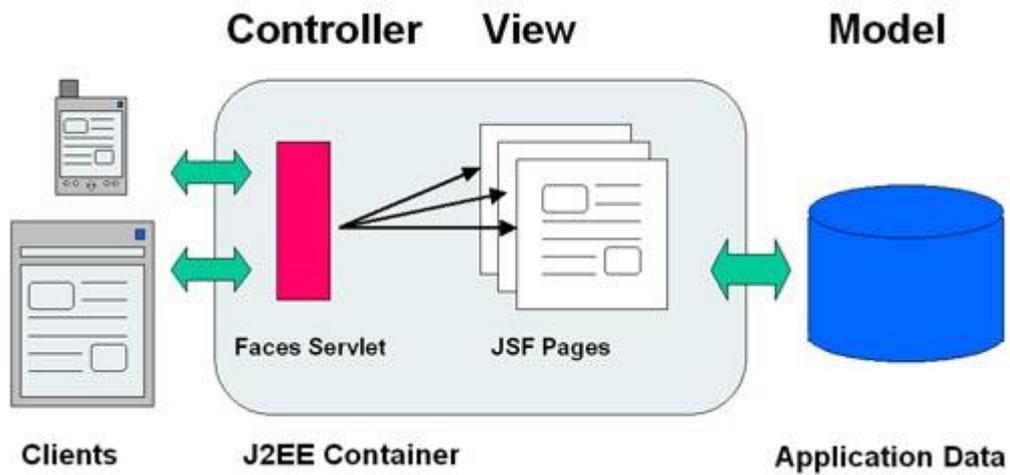
### **Patrón MVC (modelo visto controlador)**

Como mencionamos anteriormente para la construcción del software se hizo usó del framework para JAVA JSF el cual trabaja con el patrón MVC (modelo visto controlador), este nos permite separar los componentes del aplicativo desde las vistas del usuario es decir las interfaces con las cuales este interactúa con el sistema, pasando por los controladores del aplicativo los cuales actúan como intermediarios entre la vista y los modelos quienes son los que se comunican con nuestra base de datos implementado la lógica de negocio

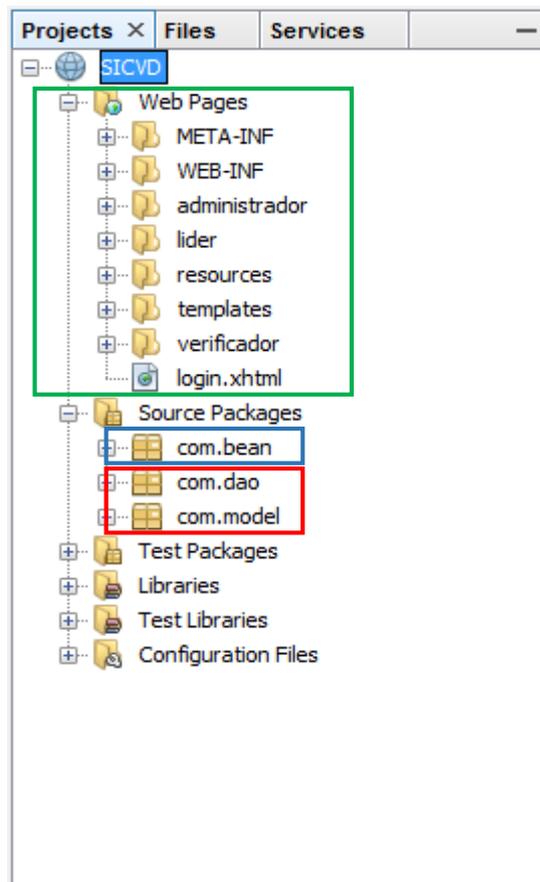
Vista = interfaces usuario

Controladores = implementa lógica para comunicar las vistas con los modelos de datos

Modelos = capa a donde se trabaja con los datos (base de datos select, insert, update, delete)



Este patrón lo podemos ver reflejado en la estructura del proyecto como se muestra en la siguiente imagen



Como podemos observar en el proyecto se encuentran separados los componentes, las vistas conformadas por la interfaces de usuarios creadas en el xhtml, los controladores en el paquete bean programados en java y los modelos y lógica del negocio contenida en los paquetes model y dao los cuales realizan las operaciones sobre la base de datos

## Patrón value object

Este patrón consiste en mapear las tablas correspondientes de nuestra base de datos en clases para la creación de objetos sobre estas, esto con el fin de facilitar la creación y mantenimiento de las funciones de transacción que tenga el aplicativo sobre la base de datos, puesto que ya no se trabajarían con valores serados e independientemente si no en cambio la función podrá recibir todo el objeto el cual ya contiene la información necesaria para realizar la operación correspondientes sobre las tablas.

Este patrón lo vemos reflejado en el proyecto específicamente en los modelos estos se encuentran en la paquete model, en este están el mapeo de cada una de las tablas utilizadas en el aplicativo convertidas en clases, como ejemplo tenemos la tabla articulo\_electronico y su correspondiente clase en donde podemos observar que la clase contiene los atributos correspondientes a su tabla conjuntamente con sus métodos get y set de cada uno de los atributo para el acceso a estos.

Tabla: articulo\_electronico

[Visualizar contenido](#) [Mostrar estructura](#)

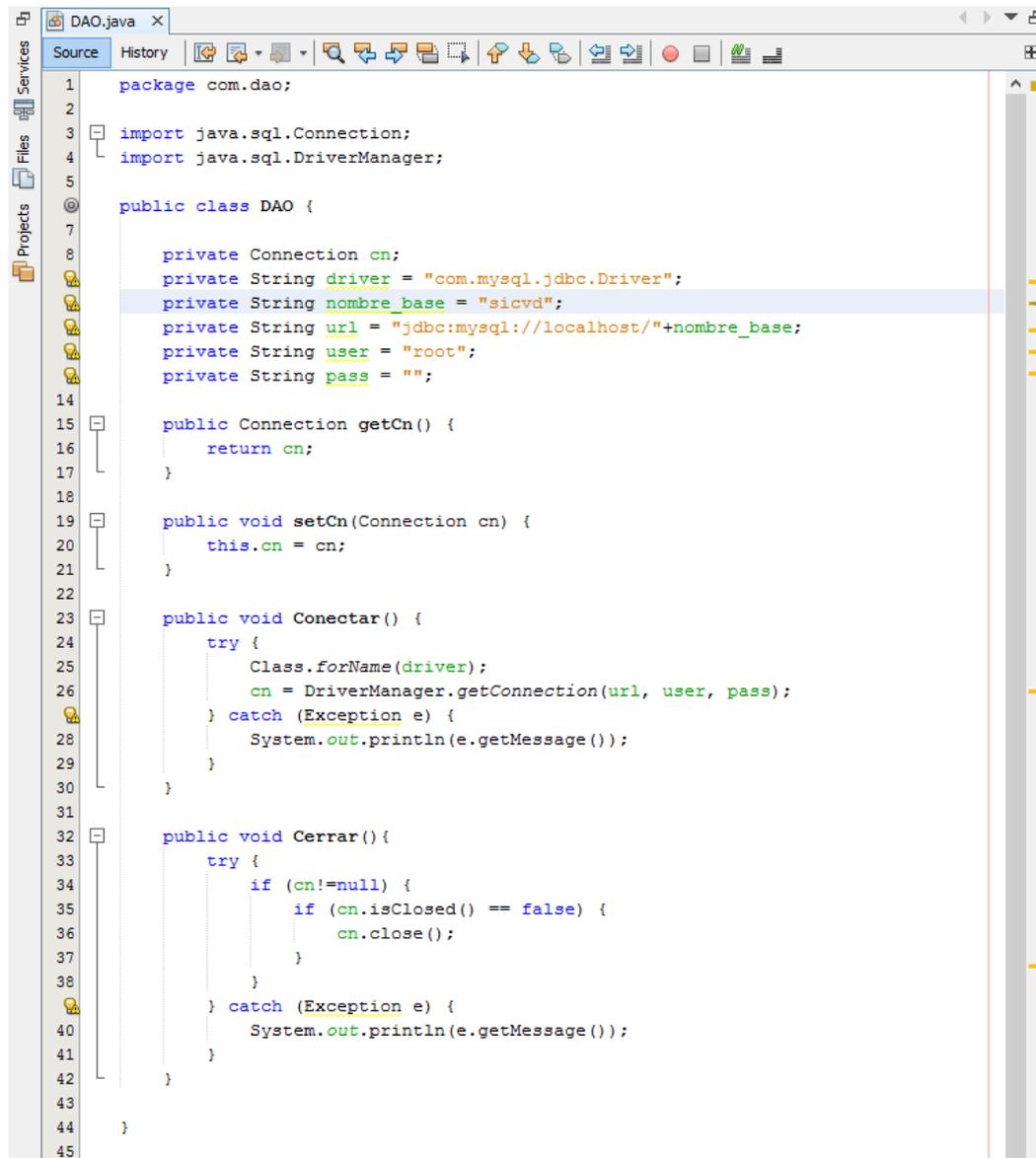
Columna	Tipo	Comentario
id	int(11)	
codigo	varchar(100)	
titulo	varchar(300)	
revista	varchar(100)	
volumen	varchar(10)	
pagina_inicial	int(11)	
pagina_final	int(11)	
issn	varchar(20)	
fecha	varchar(20)	
url	varchar(100)	
doi	varchar(100)	

```
Articulo_electronico.java X
Source History
1 package com.model;
2
3 public class Articulo_electronico {
4
5     private int id;
6     private String codigo;
7     private String titulo;
8     private String revista;
9     private String volumen;
10    private int pagina_inicial;
11    private int pagina_final;
12    private String issn;
13    private String fecha;
14    private String url;
15    private String doi;
16    private String numero_observaciones;
17
18    public int getId() {
19        return id;
20    }
21
22    public void setId(int id) {
23        this.id = id;
24    }
25
26    public String getCodigo() {
27        return codigo;
28    }
29
30    public void setCodigo(String codigo) {
31        this.codigo = codigo;

```

## Patrón DAO (Data Access object)

Este patrón trabaja sobre la accesibilidad de los datos en este caso como el aplicativo se conecta con la base de datos, su uso lo podemos ver reflejado en la creación del archivo DAO el cual es el encargado de realizar las funciones de abrir la conexión y cerrar la conexión con la base de datos especificando los parámetros para realizar estas tareas



```
1 package com.dao;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5
6
7 public class DAO {
8
9     private Connection cn;
10    private String driver = "com.mysql.jdbc.Driver";
11    private String nombre_base = "sicvd";
12    private String url = "jdbc:mysql://localhost/"+nombre_base;
13    private String user = "root";
14    private String pass = "";
15
16    public Connection getCn() {
17        return cn;
18    }
19
20    public void setCn(Connection cn) {
21        this.cn = cn;
22    }
23
24    public void Conectar() {
25        try {
26            Class.forName(driver);
27            cn = DriverManager.getConnection(url, user, pass);
28        } catch (Exception e) {
29            System.out.println(e.getMessage());
30        }
31    }
32
33    public void Cerrar(){
34        try {
35            if (cn!=null) {
36                if (cn.isClosed() == false) {
37                    cn.close();
38                }
39            }
40        } catch (Exception e) {
41            System.out.println(e.getMessage());
42        }
43    }
44 }
45
```

El objetivo principal de la utilización del patrón dao es brindar al aplicativo la flexibilidad de desacoplar la conectividad de los datos de la lógica del negocio

Por ejemplo

Si cambiamos el nombre de base de datos o alguno de los atributos necesarios para la conexión como usuario o contraseña del motor base de datos solo con hacer los cambios respectivos en el archivo DAO el aplicativo seguirá con su funcionamiento normal, en caso contrario que este no estuviese aplicado se tendría que realizar este cambio en cada uno de los códigos que realizan procesos de transacción con la base de datos volviendo más engorroso el proceso de creación y mantenimiento del sistema

## Fase 4 –pruebas

### Pruebas de aceptación

Las pruebas de aceptación son realizadas con el usuario en las cuales, este evalúa el software según los criterios de aceptación estipulada para cada historia de usuario.

Para la realización de estas pruebas el tester junto con usuario crean casos de prueba en los cuales se estipulan las entradas y salidas que deberá de tener el modulo evaluado

A continuación observaremos la aplicación de la prueba de aceptación sobre el módulo de login con su correspondiente caso de prueba.

#### Caso de prueba login

Caso de prueba	
Nombre de la prueba	Login
Numero de referencia	1
Historia de usuario	3
Iteración	1
Funcionalidad critica	Si
¿Qué se testeara?	Prueba interfaz de login, validación de usuario y contraseña
Categoría	Gestión de usuario
Prerrequisitos	Usuarios registrados en base de datos

#### Resumen del caso de prueba

paso	Comando/url	acción	datos	Expectativa
1	http://localhost:8084/SIDCVD/faces/login.xhtml	Ingresar nombre de usuario y contraseña	Nombre usuario contraseña	Ingreso a la interfaz correspondiente al tipo de usuario logeado

## Pasos y acciones de los casos de prueba login

usuario	contraseña	resultado
Ingenium suta	1234	Login correcto a interfaz de grupo
Verificador-1	1234	Login correcto a interfaz de verificador
admin	admin	Login correcto a interfaz de administrador
null	null	Usuario incorrecto / contraseña incorrecta
usuario	vacío	Usuario incorrecto / contraseña incorrecta
Vació	contraseña	Usuario incorrecto / contraseña incorrecta

## Pruebas unitarias

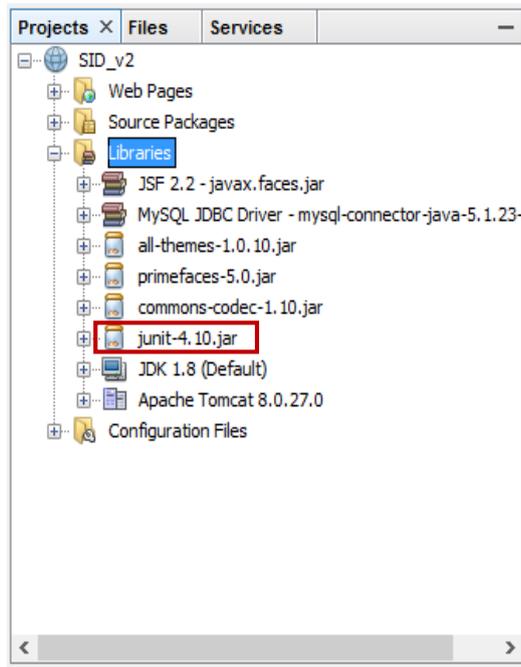
Las pruebas unitarias son pruebas realizadas por los desarrolladores sobre el código estas pruebas se realiza con el fin de obtener un código lo más limpio posible de errores

Las pruebas unitarias en proyecto fueron aplicadas atreves de la utilización del framework de pruebas JUNIT este es una herramienta que permite la realización de pruebas del código sobre proyecto creados en java

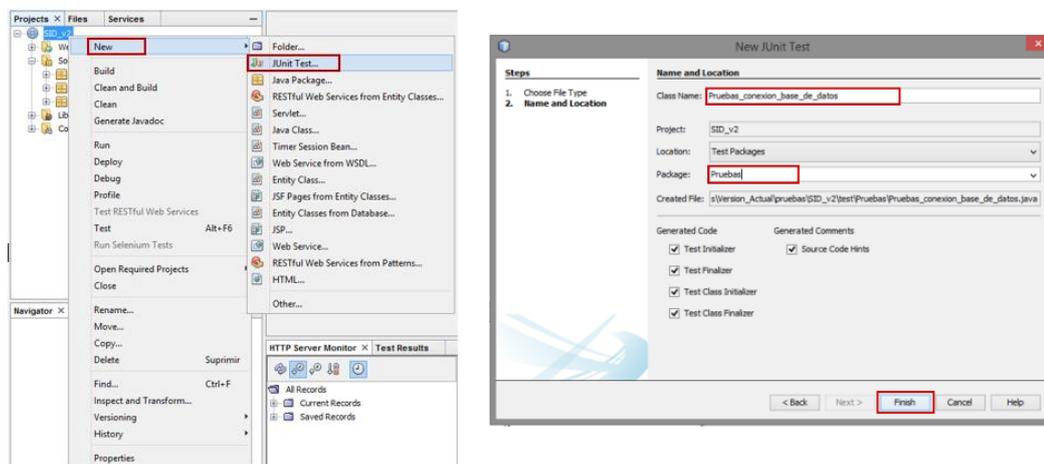
A continuación mostraremos los pasos correspondientes para la realización de estas pruebas sobre la conexión de la aplicación a la base de datos

1. lo primero es agregar al proyecto la librería JUNIT la cual nos permitirá la aplicación de la pruebas

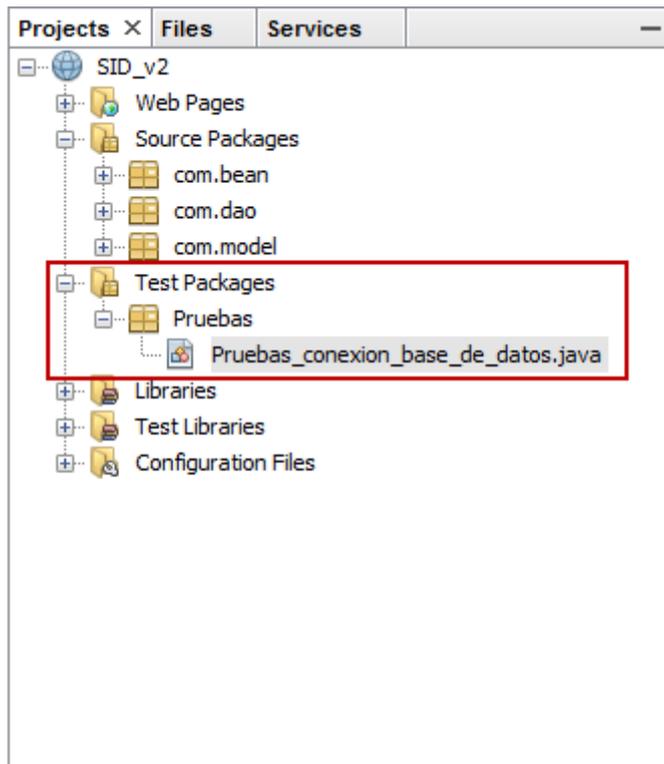
En este caso utilizaremos JUNIT en su versión 4.10 este archivo .JAR lo podremos encontrar en la página oficial del proyecto JUNIT



2. Crearemos un nueva clase de tipo JUnit Test con el nombre “Pruebas\_conexion\_base\_de\_datos ” y especificamos el paquete con el nombre pruebas



Como observamos se crea una carpeta con el nombre “Test Package” y dentro de esta el paquete que definimos “pruebas” y la clase “Pruebas\_conexion\_base\_de\_datos”



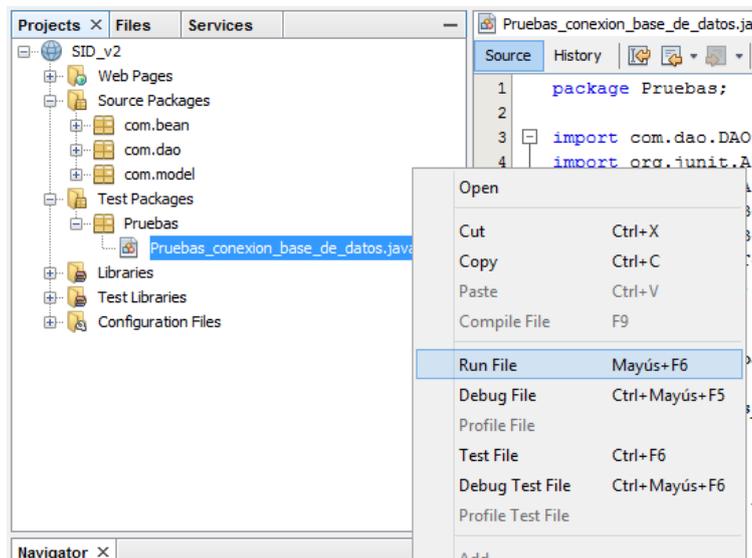
3. escribiremos el código correspondiente a la prueba, como la prueba se realizara sobre la conexión de la base de datos esta la realizaremos sobre el código escrito en la clase DAO la cual es la encargada de definir los parámetros de conexión sobre la base de datos como el driver de conexión, el nombre la base de datos, usuario y contraseña definidos.

La prueba específicamente se realizara sobre los métodos de abrir conexión y cerrar conexión entre la aplicación y la base de datos

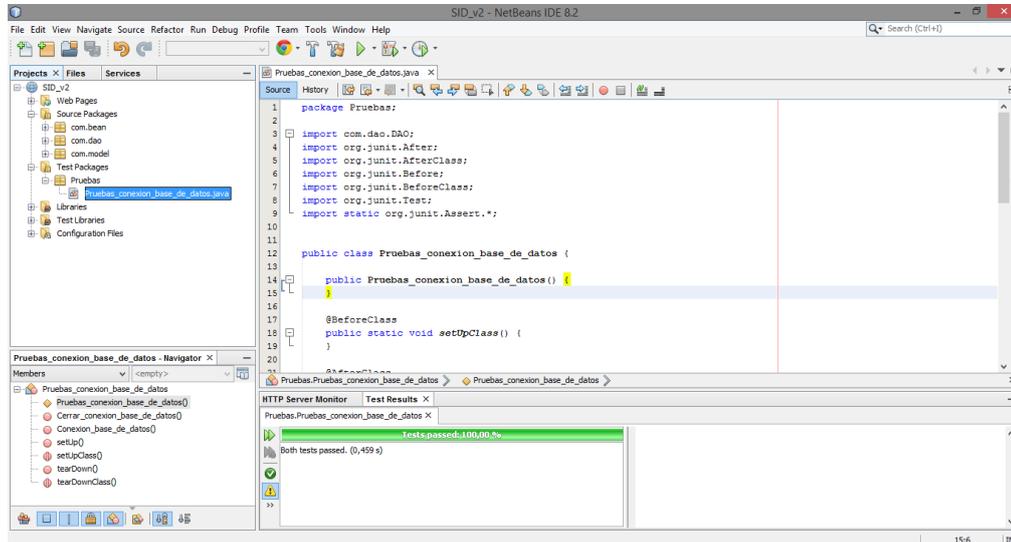
```
1 package Pruebas;
2
3 import com.dao.DAO;
4 import org.junit.After;
5 import org.junit.AfterClass;
6 import org.junit.Before;
7 import org.junit.BeforeClass;
8 import org.junit.Test;
9 import static org.junit.Assert.*;
10
11
12 public class Pruebas_conexion_base_de_datos {
13
14     public Pruebas_conexion_base_de_datos() {
15     }
16
17     @BeforeClass
18     public static void setUpClass() {
19     }
20
21     @AfterClass
22     public static void tearDownClass() {
23     }
24
25     @Before
26     public void setUp() {
27     }
28
29     @After
30     public void tearDown() {
31     }
32
33     @Test()
34     public void Conexion_base_de_datos(){
35         DAO dao = new DAO();
36         dao.Conectar();
37         assertEquals("conectado",dao.getEstado_conexion());
38     }
39
40     @Test()
41     public void Cerrar_conexion_base_de_datos(){
42         DAO dao = new DAO();
43         dao.Conectar();
44         dao.Cerrar();
45         assertEquals("conexion cerrada",dao.getEstado_conexion());
46     }
47 }
```

Como observamos los métodos de test se colocan anticipados de la anotación @test esto indica al framework que el método a continuación es la ejecución de una prueba en el cual dentro de este indicamos el método a ejecutar de la clase a la que realizaremos la prueba y el resultado esperado, de esta manera sabremos si la prueba realizada finalizo correctamente o si el código evaluado contiene un error y no devuelve el valor esperado.

#### 4. ejecución de la prueba, esta se realiza corriendo el archivo que ejecuta la prueba



5. una vez ejecutado el archivo este mostrara los resultados obtenidos de la ejecución de la prueba



Como podemos observar la prueba fue realizada correctamente ya que nos muestra un porcentaje de 100% esto indica que el código ejecutado de los métodos de la clase DAO en este caso retornaron los valores esperados es decir el código cumple correctamente con su propósito.

Si cambiamos alguno de los parámetros que afecten la conexión con la base de datos observaremos que el resultado al ejecutar la prueba, esta arrojará un 0% indicando que el código no se está ejecutando correctamente y existe un error en este.

En este caso cambiaremos la contraseña de la conexión para demostrar este punto

```
private Connection cn;
private String driver = "com.mysql.jdbc.Driver";
private String nombre_base = "sid";
private String url = "jdbc:mysql://localhost/" + nombre_base;
private String user = "root";
private String pass = "pass_incorrecta";
private String estado_conexion;
```

Screenshot of NetBeans IDE 8.2 showing a Java project named 'SID\_v2'. The main editor displays the source code for the 'DAO' class in 'Pruebas\_conexion\_base\_datos.java'. The code defines a 'DAO' class with private fields for 'cn', 'driver', 'nombre\_base', 'url', 'user', 'pass', and 'estado\_conexion'. It includes methods 'getCn()', 'setCn()', and 'getEstado\_conexion()'. The 'url' is constructed as 'jdbc:mysql://localhost/' + nombre\_base.

```
public class DAO {
    private Connection cn;
    private String driver = "com.mysql.jdbc.Driver";
    private String nombre_base = "sid";
    private String url = "jdbc:mysql://localhost/" + nombre_base;
    private String user = "root";
    private String pass = "pass_incorrecta";
    private String estado_conexion;

    public Connection getCn() {
        return cn;
    }

    public void setCn(Connection cn) {
        this.cn = cn;
    }

    public String getEstado_conexion() {
        return estado_conexion;
    }
}
```

The 'Members' view shows the class structure with fields like 'cn', 'driver', 'estado\_conexion', 'nombre\_base', and 'pass'. The 'Test Results' window shows a failed test for 'Pruebas.Pruebas\_conexion\_base\_datos' with the message: 'Exception: Access denied for user 'root'@'localhost' (using password)'. The test summary indicates 'Tests passed: 0.00 %' and 'No test passed, 2 tests failed. (1,1632 s)'.

## Conclusiones

La aplicación de patrones de diseño en el desarrollo del software permite la estandarización del código facilitando las tareas de creación y mantenimiento del sistema ya que la utilización de estos permite que el código del aplicativo sea comprendido más fácilmente por los integrantes del grupo desarrollador o los encargados de realizar el mantenimiento a este

La implementación del patrón de diseño dao le brinda la capacidad al software de que si se ha de realizar algún cambio sobre los parámetros de acceso a los datos, en este caso la conexión con la base de datos estos cambios solo se deben de realizar en el archivo de configuración de acceso a datos facilitando el proceso de mantenimiento del aplicativo

El trabajar con Frameworks como JSF(Java Server Faces) y Bootstrap facilita bastante el desarrollo del proyecto, ya que estos brindan orden a la estructura del código, JSF proporcionando una arquitectura de software a través de la utilización del modelo vista controlador separando las diferentes capas del proyecto para que estén puedan ser independientes una de otra es decir separando lo visual de lo lógico y bootstrap utilizado para la creación de las interfaces gráficas ofrece una flexibilidad y facilidad para implementar y adaptar su código según las necesidades de personalización de la interfaz propias del software

La utilización conjunta del marco de trabajo ágil SCRUM y la metodología XP brinda una serie de instrumentos que facilitaron la elaboración de proyecto, SCRUM utilizada como metodología de gestión y organización del proyecto proporciona los roles, el medio de levantamiento de requisitos (historias de usuario), la definición de prioridad de los requisitos a través del Product backlog y la organización de tareas a realizar a través de la creación del Sprint backlog, XP utilizada principalmente para la fase de desarrollo del software aporta pautas como la comunicación continua con el cliente, la retroalimentación a través de

esta, la realización de pruebas tanto de aceptación con el cliente como pruebas sobre el código, esto con el fin de obtener el mejor producto posible

La utilización del *framework* JUNIT como herramienta de aplicación de pruebas unitarias facilita en gran medida la realización de estas pruebas, ya que esta herramienta proporciona una estructura sobre la cual se escriben las pruebas que se realizarán al código del aplicativo, y de esta forma verificar si el código escrito realiza correctamente la tarea o tareas programadas en este, y así garantizar que se está construyendo un software con la mejor calidad posible a nivel de código

La creación del aplicativo brinda la posibilidad de mejorar el proceso por el cual se realiza verificación de los productos resultados de investigación, ya que este no se tendrá que realizar de forma manual como se realizaba normalmente, si no ahora el software permitirá sistematizar la realización de esta tarea. Mejorando y simplificando el trabajo realizado por la persona encargada de realizar el proceso de verificación de los productos realizados por los grupos de investigación de facultad de ingeniería de la universidad de Cundinamarca sede Fusagasugá

## Bibliografía

- [1] Universidad de Barcelona, (2004). Repositorios digitales de la Universidad de Barcelona]. Disponible en: <http://crai.ub.edu/es/recursos-de-informacion/repositorios-digitales>
- [2] Universidad de Los Andes, (2016), Introducción Academia Institucional. Disponible en: <https://academia.uniandes.edu.co/webhelp/>
- [3] Uniminuto, (2016), Dirección de Investigaciones SP. Disponible en: [http://www.uniminuto.edu/web/investigaciones/investigacionesp/-/asset\\_publisher/MDqzrfX6Auu5/content/repositorios?inheritRedirect=false&redirect=http%3A%2F%2Fwww.uniminuto.edu%2Fweb%2Finvestigaciones%2Finvestigacionesp%3Fp\\_p\\_id%3D101\\_INSTANCE\\_MDqzrfX6Auu5%26p\\_p\\_lifecycle%3D0%26p\\_p\\_state%3Dnormal%26p\\_p\\_mode%3Dview%26p\\_p\\_col\\_id%3Dcolumn-2%26p\\_p\\_col\\_count%3D1](http://www.uniminuto.edu/web/investigaciones/investigacionesp/-/asset_publisher/MDqzrfX6Auu5/content/repositorios?inheritRedirect=false&redirect=http%3A%2F%2Fwww.uniminuto.edu%2Fweb%2Finvestigaciones%2Finvestigacionesp%3Fp_p_id%3D101_INSTANCE_MDqzrfX6Auu5%26p_p_lifecycle%3D0%26p_p_state%3Dnormal%26p_p_mode%3Dview%26p_p_col_id%3Dcolumn-2%26p_p_col_count%3D1)
- [4] Oracle, JavaServer Faces Technology, Disponible en: <http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html>
- [5] ARWEB, (2014), ¿QUÉ ES BOOTSTRAP Y CÓMO FUNCIONA EN EL DISEÑO WEB? Disponible en: <http://www.arweb.com/chucherias/editorial/%C2%BFque-es-bootstrap-y-como-funciona-en-el-diseno-web.htm>
- [6] Primefaces, (2015), PrimeFaces, Disponibles en: <http://www.primefaces.org/>
- [7] Eduardo, (2007), ¿Qué es MySQL?. Disponible en: <http://indira-informatica.blogspot.com.co/2007/09/qu-es-mysql.html>
- [8] Alonso Javier Pérez Díaz, (), Apache Tomcat, <http://www.ajpdsoft.com/modules.php?name=Encyclopedia&op=content&tid=769>
- [9] Junit Project, (2002 - 2016), Junit. Disponible en: <http://junit.org/junit4/>

# Anexos

Figura 1 – Product Backlog

Lista de producto (Product Backlog)	
ID	Description
1	Diseño de modelo la base de datos
2	Creación de perfiles de usuario
3	Creación de login
4	Creación de grupos de investigación
5	Modificación de grupos de investigación
6	Eliminación de grupos de investigación
7	Creación de líderes del grupo investigación
8	Modificación de líderes
9	Eliminación de líderes
10	Creación de miembros del grupo de investigación
11	Modificación de miembros del grupo de investigación
12	Eliminación de miembros
13	Ingreso de artículos de investigación
14	Modificación de artículos de investigación
15	Eliminación de artículos de investigación
16	Modulo de requerimientos de existencia de artículos de investigación
17	Ingreso de libros resultados de investigación
18	Modificación de libros resultados de investigación
19	Eliminación libros resultados de investigación
20	Modulo de requerimientos de existencia de libros resultados de investigación
21	Ingreso de capítulos en libros resultados de investigación
22	Modificación de capítulos en libros resultados de investigación
23	Eliminación de capítulos en libros resultados de investigación
24	Modulo de requerimientos de capítulos en libros resultados de investigación
25	Ingreso de productos tecnológicos patentados o en proceso de concesión
26	Modificación de productos tecnológicos patentados o en proceso de concesión
27	Eliminación de productos tecnológicos patentados o en proceso de concesión
28	Modulo de requerimientos de productos tecnológicos patentados o en proceso de concesión
29	Ingreso de variedad vegetal y nueva raza animal
30	Modificación de variedad vegetal y nueva raza animal
31	Eliminación de variedad vegetal y nueva raza animal
32	Modulo de requerimientos de variedad vegetal y nueva raza animal
33	Ingreso de productos tecnológicos certificados o validados
34	Modificación de productos tecnológicos certificados o validados
35	Eliminación de productos tecnológicos certificados o validados
36	Modulo de requerimientos de productos tecnológicos certificados o validados
37	Ingreso de productos empresariales
38	Modificación de productos empresariales
39	Eliminación de productos empresariales
40	Modulo de requerimientos de productos empresariales
41	Ingreso de productos empresariales
42	Modificación de productos empresariales
43	Eliminación de productos empresariales
44	Modulo de requerimientos de productos empresariales
45	Ingreso de regulación, normas, reglamentos o legislación
46	Modificación de regulación, normas, reglamentos o legislación
47	Eliminación de regulación, normas, reglamentos o legislación
48	Modulo de requerimientos de regulación, normas, reglamentos o legislación
49	Ingreso de consultorías e informes técnicos finales
50	Modificación de consultorías e informes técnicos finales
51	Eliminación de consultorías e informes técnicos finales
52	Modulo de requerimientos de consultorías e informes técnicos finales
53	Ingreso de acuerdos de licencias para la explotación de obras protegidas por derechos de autor
54	Modificación de acuerdos de licencias para la explotación de obras protegidas por derechos de autor
55	Eliminación de acuerdos de licencias para la explotación de obras protegidas por derechos de autor
56	Modulo de requerimientos de acuerdos de licencias para la explotación de obras protegidas por derechos de autor
57	Ingreso de tesis de grado
58	Modificación de tesis de grado
59	Eliminación de tesis de grado
60	Modulo de requerimientos de tesis de grado
61	Ingreso de trabajo de grado de maestría
62	Modificación de trabajo de grado de maestría
63	Eliminación de trabajo de grado de maestría
64	Modulo de requerimientos de trabajo de grado de maestría
65	Ingreso de trabajo de grado de pregrado
66	Modificación de trabajo de grado de pregrado
67	Eliminación de trabajo de grado de pregrado
68	Modulo de requerimientos de trabajo de grado de pregrado
69	Ingreso de proyectos de investigación y desarrollo
70	Modificación de proyectos de investigación y desarrollo
71	Eliminación de proyectos de investigación y desarrollo
72	Modulo de requerimientos de proyectos de investigación y desarrollo
73	Ingreso de proyectos de investigación- creación
74	Modificación de proyectos de investigación- creación
75	Eliminación de proyectos de investigación- creación
76	Modulo de requerimientos de proyectos de investigación- creación
77	Ingreso de proyectos de investigación, desarrollo e innovación (ID+)
78	Modificación de proyectos de investigación, desarrollo e innovación (ID+)
79	Eliminación de proyectos de investigación, desarrollo e innovación (ID+)
80	Modulo de requerimientos de proyectos de investigación, desarrollo e innovación (ID+)
81	Ingreso de proyecto de extensión y responsabilidad social en CTI
82	Modificación de proyecto de extensión y responsabilidad social en CTI
83	Eliminación de proyecto de extensión y responsabilidad social en CTI
84	Modulo de requerimientos de proyecto de extensión y responsabilidad social en CTI
85	Ingreso de apoyo a programas de formación
86	Modificación de apoyo a programas de formación
87	Eliminación de apoyo a programas de formación
88	Modulo de requerimientos de apoyo a programas de formación
89	Pruebas
90	Documentacion
91	Manual de usuario
92	Capacitación al usuario

Figura 2 – Sprint Backlog

Lista de tareas de la iteración (Sprint Backlog)					
ID	Descripción	Estimación	Prioridad	Estado	Sprint
1	Diseño de modelo la base de datos	1 semana	1	Pendiente	1
2	Creación de perfiles de usuario		2	Pendiente	1
3	Creación de login		3	Pendiente	1
4	Creación de grupos de investigación		4	Pendiente	1
5	Modificación de grupos de investigación		5	Pendiente	1
6	Eliminación de grupos de investigación		6	Pendiente	1
7	Creación de líderes del grupo investigación	1 semana	7	Pendiente	2
8	Modificación de líderes		8	Pendiente	2
9	Eliminación de líderes		9	Pendiente	2
10	Creación de miembros del grupo de investigación		10	Pendiente	2
11	Modificación de miembros del grupo de investigación		11	Pendiente	2
12	Eliminación de miembros		12	Pendiente	2
13	Ingreso de artículos de investigación	1 semana	13	Pendiente	3
14	Modificación de artículos de investigación		14	Pendiente	3
15	Eliminación de artículos de investigación		15	Pendiente	3
16	Modulo de requerimientos de existencia de artículos de investigación		16	Pendiente	3
17	Ingreso de libros resultados de investigación		17	Pendiente	3
18	Modificación de libros resultados de investigación		18	Pendiente	3
19	Eliminación libros resultados de investigación	19	Pendiente	3	
20	Modulo de requerimientos de existencia de libros resultados de investigación	20	Pendiente	3	
21	Ingreso de capítulos en libros resultados de investigación	1 semana	21	Pendiente	4
22	Modificación de capítulos en libros resultados de investigación		22	Pendiente	4
23	Eliminación de capítulos en libros resultados de investigación		23	Pendiente	4
24	Modulo de requerimientos de capítulos en libros resultados de investigación		24	Pendiente	4
25	Ingreso de productos tecnológicos patentados o en proceso de concesión		25	Pendiente	4
26	Modificación de productos tecnológicos patentados o en proceso de concesión		26	Pendiente	4
27	Eliminación de productos tecnológicos patentados o en proceso de concesión	27	Pendiente	4	
28	Modulo de requerimientos de productos tecnológicos patentados o en proceso de concesión	28	Pendiente	4	
29	Ingreso de variedad vegetal y nueva raza animal	1 semana	29	Pendiente	5
30	Modificación de variedad vegetal y nueva raza animal		30	Pendiente	5
31	Eliminación de variedad vegetal y nueva raza animal		31	Pendiente	5
32	Modulo de requerimientos de variedad vegetal y nueva raza animal		32	Pendiente	5
33	Ingreso de productos tecnológicos certificados o validados	1 semana	33	Pendiente	6
34	Modificación de productos tecnológicos certificados o validados		34	Pendiente	6
35	Eliminación de productos tecnológicos certificados o validados		35	Pendiente	6
36	Modulo de requerimientos de productos tecnológicos certificados o validados		36	Pendiente	6
37	Ingreso de productos empresariales		37	Pendiente	6
38	Modificación de productos empresariales		38	Pendiente	6
39	Eliminación de productos empresariales	39	Pendiente	6	
40	Modulo de requerimientos de productos empresariales	40	Pendiente	6	
41	Ingreso de regulación, normas, reglamentos o legislación	1 semana	41	Pendiente	7
42	Modificación de regulación, normas, reglamentos o legislación		42	Pendiente	7
43	Eliminación de regulación, normas, reglamentos o legislación		43	Pendiente	7
44	Modulo de requerimientos de regulación, normas, reglamentos o legislación		44	Pendiente	7
45	Ingreso de consultorías e informes técnicos finales	1 semana	45	Pendiente	8
46	Modificación de consultorías e informes técnicos finales		46	Pendiente	8
47	Eliminación de consultorías e informes técnicos finales		47	Pendiente	8
48	Modulo de requerimientos de consultorías e informes técnicos finales		48	Pendiente	8
49	Ingreso de acuerdos de licencias para la explotación de obras protegidas por derechos de autor		49	Pendiente	8
50	Modificación de acuerdos de licencias para la explotación de obras protegidas por derechos de autor		50	Pendiente	8
51	Eliminación de acuerdos de licencias para la explotación de obras protegidas por derechos de autor	51	Pendiente	8	
52	Modulo de requerimientos de acuerdos de licencias para la explotación de obras protegidas por derechos de autor	52	Pendiente	8	
53	Ingreso de tesis de doctorado	1 semana	53	Pendiente	9
54	Modificación de tesis de doctorado		54	Pendiente	9
55	Eliminación de tesis de doctorado		55	Pendiente	9
56	Modulo de requerimientos de existencia de tesis de doctorado		56	Pendiente	9
57	Ingreso de trabajo de grado de maestría		57	Pendiente	9
58	Modificación de trabajo de grado de maestría		58	Pendiente	9
59	Eliminación de trabajo de grado de maestría	59	Pendiente	9	
60	Modulo de requerimientos de existencia de trabajo de grado de maestría	60	Pendiente	9	
61	Ingreso de trabajo de grado de pregrado	1 semana	61	Pendiente	10
62	Modificación de trabajo de grado de pregrado		62	Pendiente	10
63	Eliminación de trabajo de grado de pregrado		63	Pendiente	10
64	Modulo de requerimientos de existencia de trabajo de grado de pregrado		64	Pendiente	10
65	Ingreso de proyectos de investigación y desarrollo		65	Pendiente	10
66	Modificación de proyectos de investigación y desarrollo		66	Pendiente	10
67	Eliminación de proyectos de investigación y desarrollo	67	Pendiente	10	
68	Modulo de requerimientos de existencia de proyectos de investigación y desarrollo	68	Pendiente	10	
69	Ingreso de proyectos de investigación- creación	1 semana	69	Pendiente	11
70	Modificación de proyectos de investigación- creación		70	Pendiente	11
71	Eliminación de proyectos de investigación- creación		71	Pendiente	11
72	Modulo de requerimientos de proyectos de investigación- creación		72	Pendiente	11
73	Ingreso de proyectos de investigación, desarrollo e innovación (ID+I)		73	Pendiente	11
74	Modificación de proyectos de investigación, desarrollo e innovación (ID+I)		74	Pendiente	11
75	Eliminación de proyectos de investigación, desarrollo e innovación (ID+I)	75	Pendiente	11	
76	Modulo de requerimientos de proyectos de investigación, desarrollo e innovación (ID+I)	76	Pendiente	11	
77	Ingreso de proyecto de extensión y responsabilidad social en CTI	1 semana	77	Pendiente	12
78	Modificación de proyecto de extensión y responsabilidad social en CTI		78	Pendiente	12
79	Eliminación de proyecto de extensión y responsabilidad social en CTI		79	Pendiente	12
80	Modulo de requerimientos de proyecto de extensión y responsabilidad social en CTI		80	Pendiente	12
81	Ingreso de apoyo a programas de formación		81	Pendiente	12
82	Modificación de apoyo a programas de formación		82	Pendiente	12
83	Eliminación de apoyo a programas de formación	83	Pendiente	12	
84	Modulo de requerimientos de apoyo a programas de formación	84	Pendiente	12	
85	Pruebas	1 semana	85	Pendiente	
86	Documentacion		86	Pendiente	14
87	Manual de usuario		87	Pendiente	14
88	Capacitación al usuario		88	Pendiente	14

Figura 3 - Modelo relacional

