



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA EN SISTEMAS

**“ANÁLISIS COMPARATIVO DE SERVIDORES DE APLICACIÓN PARA LA INTEGRACIÓN DE
SISTEMAS MEDIANTE SERVICIOS WEB XML. CASO PRÁCTICO: COLEGIO CAPITÁN
EDMUNDO CHIRIBOGA”**

**“TESIS DE GRADO PREVIA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN SISTEMAS INFORMÁTICOS”**

AMANDA JOSEFINA ADRIANO MOYOTA

JOSÉ GREGORIO PINTO BARRETO

RIOBAMBA – ECUADOR

-2014-

AGRADECIMIENTO

Agradezco primeramente a Dios por derramar sus bendiciones en todo momento de mi vida, permitiéndome cumplir mis sueños y objetivos anhelados, a mis padres Rodrigo y Rosa por su apoyo incondicional y sacrificio para darme la educación, por sus palabras de aliento y fortaleza para estar siempre de pie ante las adversidades de la vida, a toda mi familia que con su apoyo incondicional.

Amanda Adriano Moyota

Agradezco totalmente a mi madre Mercedes Pinto y a mi tío Félix Pinto por todo el apoyo que me han brindado durante toda la vida.

José Pinto Barreto.

FIRMAS RESPONSABLES Y NOTAS

NOMBRES	FIRMA	FECHA
ING. GONZALO SAMANIEGO ERAZO DECANO DE LA FACULTAD DE INFORMÁTICA Y ELECTRÓNICA.	_____	_____
ING. JULIO SANTILLÁN DIRECTOR DE LA ESCUELA DE INGENIERÍA EN SISTEMAS.	_____	_____
ING. JORGE MENÉNDEZ DIRECTOR DE TESIS.	_____	_____
ING. PAÚL PAGUAY MIEMBRO DE TESIS.	_____	_____
DIRECTOR DEL CENTRO DE DOCUMENTACIÓN	_____	_____
NOTA: _____		

RESPONSABILIDAD DEL AUTOR

Nosotros, Amanda Josefina Adriano Moyota y José Gregorio Pinto Barreto somos los responsables de las ideas, doctrinas y resultados expuestos en esta Tesis y el patrimonio intelectual de la misma pertenecen a la Escuela Superior Politécnica de Chimborazo.

Amanda Josefina Adriano Moyota

José Gregorio Pinto Barreto

ÍNDICE DE ABREVIATURAS

XML	Lenguaje de Marcas Extensible
J2EE	Java 2 Enterprise Edition
HTTP	Protocolo de Transferencia de Hipertexto
JSP	Java Server Pages
EJB	Enterprise Java Beans
API	Interfaz de Programación de Aplicaciones
HTML	Lenguaje Hipertexto de Marcas
CDDL	Common Development and Distribution License
MPL	Mozilla Public License Version
JB1	Java Business Integration
IDE	Entorno de Desarrollo Integrado
JMX	Java Management Extensions
DAS	Servidor de Administración de Dominio
IBM	International Business Machines
EAR	Enterprise Archive
JMS	Java Message Service
FTP	Protocolo de Transferencia de Archivos
SOA	Arquitectura Orientada a Servicios
JVM	Java Virtual Machine
SMTP	Protocolo para la Transferencia Simple de Correo Electrónico

WSDL	Lenguaje de Descripción de Servicios Web
WSIF	Web Service Integration Framework
WSIF	Web Services Invocation Framework
SAR	Supervisión de Actividades del Sistema
RAM	Random Access Memory
CPU	Unidad Central de Procesamiento

ÍNDICE GENERAL

FIRMAS RESPONSABLES Y NOTAS

RESPONSABILIDAD DE AUTOR

ÍNDICE DE ABREVIATURAS

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE GRÁFICOS

INTRODUCCIÓN

CAPÍTULO I - MARCO REFERENCIAL

1.1. ANTECEDENTES	15
1.2. JUSTIFICACIÓN DEL PROYECTO DE TESIS O MEMORIA	18
1.2.1. Justificación Teórica	18
1.2.2. Justificación Práctica	19
1.3. Objetivos	22
1.3.1. Objetivos Generales	22
1.3.2. Objetivos Específicos.....	22
1.4. Hipótesis.....	22

CAPÍTULO II - MARCO TEÓRICO

2.1. Servidor de Aplicaciones	23
2.1.1. Características de los servidores de aplicación.....	24
2.1.2. Principios de los servidores de aplicación.....	25
2.1.3. Ventaja de los servidores de aplicación	26
2.1.4. Arquitectura del servidor de aplicaciones	27
2.2. Servicios Web	33
2.2.1. Características de un servicio web	34
2.2.2. Ventajas de los Servicios Web respecto aplicaciones distribuidas	35
2.2.3. Arquitectura de un servicio web	36

2.2.4.	Tecnología de servicios web.....	38
2.3.	SERVIDOR DE APLICACIONES GLASSFISH	41
2.3.1.	Características	41
2.3.2.	Arquitectura de GlassFish	42
2.4.	SERVIDOR DE APLICACIONES GERÓNIMO.....	46
2.4.1.	Características	46
2.4.2.	Arquitectura de Gerónimo	47
2.5.	SERVIDOR DE APLICACIONES JONAS.....	49
2.5.1.	Características	49
2.5.2.	Arquitectura de JOnAS	50

CAPÍTULO III - Análisis comparativo

3.	Análisis comparativo de servidores de aplicación Glassfish, JOnAS y Gerónimo.....	55
3.1.	Definición de parámetros de comparación.....	56
3.2.	Especificación del hardware donde se realizaron las pruebas	58
3.3.	Especificación del software donde se realizaron las pruebas.....	59
3.4.	Descripción del experimento	59
3.5.	Operacionalización metodológica y conceptual	60
	Tipo de hipótesis	60
	Operacionalización conceptual	60
	Operacionalización metodológica.....	60
3.6.	Descripción de las herramientas utilizadas.....	61
3.6.1.	Comando sar	62
3.6.2.	Comando sar -r.....	62
3.6.3.	Comando sar -u	63
3.6.4.	Comando sar -d	64
3.6.5.	Comando time.....	65
3.7.	Análisis de los parámetros de comparación	66
3.7.1.	Uso de RAM.....	66
3.7.2.	Uso de CPU.....	67
3.7.3.	Uso de Disco	69
3.7.4.	Tiempo de respuesta.....	70
3.8.	Análisis de resultados y demostración de hipótesis	71
3.8.1.	Uso de RAM.....	72
3.8.2.	Uso de CPU	75
3.8.3.	Uso de Disco	78

3.8.4.	Tiempo de respuesta.....	81
3.9.	Resultado del análisis y demostración de la hipótesis.....	83
CAPÍTULO IV - Sistema de Integración		
4.	Implementación del sistema de integración y administración de aplicaciones	86
4.1.	Descripción del sistema	86
4.2.	Historias de usuario.....	88
4.3.	Estimación de tiempo de historias usuario.....	89
4.4.	Plan de entregas de cada iteración al cliente	92
4.5.	Formato de historia usuario, tareas de ingeniería y pruebas de aceptación.....	95
4.6.	Desarrollo de historias de usuario y metáforas	98
4.6.1.	Iteración 1	98
4.6.2.	Iteración 2	104
4.6.3.	Iteración 3	110
4.6.4.	Iteración 9	117
CONCLUSIONES		
RECOMENDACIONES		
RESUMEN		
SUMARY		
GLOSARIO		
ANEXOS		
BIBLIOGRAFÍA		

ÍNDICE DE FIGURAS

Figura I. 1. Estructura académica para migración de Datos a Moodle	21
Figura II. 2. Servidor de Aplicaciones	24
Figura II. 3. Arquitectura del servidor de aplicaciones.....	27
Figura II. 4. Ciclo de vida de un servlet.....	29
Figura II. 6. Ciclo de vida de una página JSP.....	31
Figura II. 7. Funcionamiento de los componentes EJB.....	33
Figura II. 8. Descripción del proceso del servicio web	34
Figura II. 9. Arquitectura de Servicios Web.....	37
Figura II. 10. Esquema Protocolo SOA.....	40
Figura II. 13 Arquitectura Servidor de Aplicaciones GlassFish	43
Figura II. 14. Arquitectura Servidor de Aplicaciones Gerónimo.....	47
Figura II. 15. Ciclo de vida de un gbeans.....	49
Figura II. 16. Arquitectura servidor de aplicaciones JOnAS	51
Figura III. 17. Sintaxis comando SAR	62
Figura III. 18. Análisis parámetro uso de RAM	67
Figura III. 19. Análisis del parámetro uso de CPU	68
Figura III. 20. Análisis del parámetro uso de Disco	70
Figura III. 21. Análisis del parámetro tiempo de respuesta	71
Figura III. 22. Resultado final del análisis	84
Figura IV. 23. Solución.....	88
Figura IV. 24. Puntos estimados vs Puntos reales.....	92
Figura IV. 25. Formato historia de usuario.....	95
Figura IV. 26. Formato tarea de ingeniería	96
Figura I. 27. Formato prueba de aceptación.....	97

Figura IV. 28. Diagrama entidad relación DER	100
Figura IV. 29. Entidades usadas para la creación de usuarios en Moodle	105
Figura IV. 30. Entidades usadas para la creación de categorías en Moodle	106
Figura IV. 31. Entidades usadas para la creación de cursos en Moodle	107
Figura IV. 32. Entidades usadas para matriculación de estudiantes y docentes en Moodle....	108
Figura IV. 33. Tablas sistema académico.....	109
Figura IV. 34. Diagrama entidad relación.....	111
Figura IV. 35. Modelo de base de datos para la integración.....	114
Figura IV. 36. Arquitectura de despliegue.....	115
Figura IV. 37. Plantilla de menú principal	116

ÍNDICE DE TABLAS

Tabla III. I. Definición de los parámetros de comparación	57
Tabla III. II. Ponderación de parámetros	58
Tabla III. III. Especificación hardware del equipo	58
Tabla III. IV. Especificación software	59
Tabla III. V. Operacionalización conceptual	60
Tabla III. VI. Operacionalización metodológica	60
Tabla III. VII. Salida comando sar -r	62
Tabla III. VIII. Detalle de la salida del comando sar -u	63
Tabla III. IX. Salida del comando sar -d	64
Tabla III. X. Salida del comando time	65
Tabla III. XI. Parámetro uso de RAM.	66
Tabla III. XII. Parámetro uso de CPU	67
Tabla III. XIII. Parámetro uso de disco	69
Tabla III. XIV. Parámetro tiempo de respuesta	70
Tabla III. XV. Estadística descriptiva– Uso de RAM	73
Tabla III. XVI. Estadística inferencial – Uso de RAM	73
Tabla III. XVII. Ponderación - Uso de RAM	75
Tabla III. XVIII. Estadística descriptiva - Uso de CPU	76
Tabla III. XIX. Estadística inferencial - Uso de CPU	76
Tabla III. XX. Ponderación - Uso de CPU	78
Tabla III. XXI. Estadística descriptiva - Uso de Disco	79
Tabla III. XXII. Estadística inferencial - Uso de Disco	79
Tabla III. XXIII. Ponderación - Uso de Disco	80
Tabla III. XXIV. Estadística descriptiva - Tiempo de Respuesta	82

Tabla III. XXV. Estadística inferencial - Tiempo de Respuesta.....	82
Tabla III. XXVI. Ponderación - Tiempo de Respuesta	83
Tabla III. XXVII. Ponderación de los parámetros evaluados.....	84
Tabla IV. XXVIII. Historias de usuario y Metáforas	90
Tabla IV. XXIX. Plan de entregables.....	93
Tabla IV. XXX. Descripción de historia de usuario.....	95
Tabla IV. XXXI. Descripción de tarea de ingeniería	96
Tabla IV. XXXII. Descripción de pruebas de aceptación	97
Tabla IV. XXXIII. Tamaño inicial	102
Tabla IV. XXXIV. Proyección de crecimiento	103
Tabla IV. XXXV. Tamaño Actual	112

INTRODUCCIÓN

En la actualidad las instituciones ejecutan aplicaciones de vital importancia que automatizan los procesos de toda la empresa, estas se encuentran alojadas en servidores de aplicación, ya sea por reducir costes, compartir recursos o simplificar la gestión de la infraestructura. Por lo que surge la necesidad de realizar un estudio de los servidores de aplicaciones para determinar cuál es el de mejor rendimiento al publicar servicios web, para lo cual se determinó parámetros de comparación que serán analizados en los prototipos desarrollados para el cumplimiento de los objetivos y para la elección de uno u otro servidor.

En el Capítulo I Marco referencial, se detalla los antecedentes, la justificación de la investigación los objetivos y la hipótesis planteada, mismos que deberán ser cumplidos y comprobados al final de la investigación.

El Capítulo II Marco Teórico, se mencionará los conceptos relacionados con los servidores de aplicación: GlassFish, JOnAS y Gerónimo, conceptos de servicios web, que servirá de ayuda para el desarrollo de la presente investigación.

En el Capítulo III Comparación de los Servidores de Aplicación, contempla una breve descripción de los prototipos realizados para el análisis de los servidores de aplicación, una descripción de los parámetros para determinar el rendimiento y realizar la comprobación de hipótesis planteada.

En el Capítulo IV se describe el desarrollo del módulo de integración de sistemas del Colegio Chiriboga, publicado con el servidor de aplicaciones de mejor rendimiento.

CAPÍTULO I

MARCO REFERENCIAL

1.1. ANTECEDENTES

“Un servidor es un ordenador dedicado a ejecutar una aplicación servidora y como aplicación servidora, se conoce al software preparado para aceptar conexiones de otras máquinas con peticiones de servicio y resolver dichas peticiones enviando respuestas de vuelta a las máquinas que originaron las conexiones” (1).

Según Jesús Fernández (2), define que “un servidor de aplicaciones consiste en un contenedor que abarca la lógica de negocio de un sistema, y que provee respuestas a las peticiones de distintos dispositivos que tienen acceso a ella. Son un claro ejemplo del modelo cliente-servidor, cuyo lado cliente ejecuta

requerimientos de procesamiento al otro lado, donde el servidor se encarga de procesar y responder”.

“Un servicio Web XML es una unidad programable a la que sistemas muy dispares pueden tener acceso a través de Internet. Estos servicios dependen fundamentalmente de la aceptación generalizada de XML, HTTP y otros estándares de Internet que admiten la interoperabilidad” (3)

Actualmente el Colegio Capitán Edmundo Chiriboga cuenta con dos sistemas, que son: el sistema académico y la plataforma virtual de aprendizaje (Moodle), que trabajan de forma aislada y que comparten información referente a estudiantes, docentes, cursos, paralelos, materias, etc., por lo que surge la necesidad de integrar dichos sistemas a través de servicios web. Los servicios web son capaces de proporcionar información a diferentes aplicaciones con independencia de la plataforma y del lenguaje de programación con el objetivo de integrar sistemas y funcionar en la heterogeneidad de la red

El Colegio Capitán Edmundo Chiriboga es una institución educativa pública que tiene el propósito de forjar juventudes con alto espíritu de servicio cooperativo, en la actualidad según el decreto **1014** el software libre es una política de estado que debe ser adoptada por las entidades públicas por esta razón la solución informática que se propone para la institución estará desarrollada en base a esta política.

En investigaciones realizadas se observa que existe un trabajo de servidores de aplicación cuyo objetivo fue la creación de un sistema para la Cooperativa Patria, motivo por el cual se tiene la necesidad de investigar otros servidores de aplicación tales como: GlassFis, JOnAS y Gerónimo. Por lo que esta investigación tendrá un

enfoque totalmente diferente en cuanto a objetos de estudio y a nivel de complejidad, al momento de integrar aplicaciones.

1.2. JUSTIFICACIÓN DEL PROYECTO DE TESIS O MEMORIA

1.2.1. Justificación Teórica

En el mercado actualmente existe una gran cantidad de servidores de aplicación libres y propietarios como por ejemplo: JBoss, TomEE, WebLogic, GlassFish, JOnAS, Gerónimo, WebSphere, Resin, etc. En la presente investigación los servidores de aplicaciones a ser analizados son: GlassFish, JOnAS, Geronimo, cuya robustez, prestación y por ser gratuitos se enmarcan en lo requerido por la unidad educativa.

Los servidores de aplicaciones descritos anteriormente están hechos para trabajar con tecnología java, su principal ventaja es ser multiplataforma, lo que quiere decir que se puede utilizar la aplicación en arquitecturas Windows, Linux Unix sin necesidad de grandes cambios o configuraciones.

Los servicios Web son capaces de proporcionar un servicio a las diferentes aplicaciones con independencia de la plataforma y del lenguaje de programación.

Se realizaran prototipos para cada uno de los servidores de aplicación anteriormente mencionados los que servirá para establecer el más idóneo entre ellos.

El rendimiento será medido en base a los servicios web realizados en cada uno de los servidores de aplicación, teniendo en cuenta diferentes cantidades de carga de datos.

1.2.2. Justificación Práctica

El Colegio Capitán Edmundo Chiriboga, necesita generar un proceso automático de migración de datos de profesores, estudiantes, materias y niveles, mismos que serán tomados de la aplicación web Académica, por esta razón se propone una solución informática que permitirá migrar los datos de la aplicación web Académica a la Plataforma Virtual de Aprendizaje (Moodle). Con la integración de estas tecnologías la unidad educativa reducirá la duplicidad de información, minimizará tiempo, aportando un gran beneficio para la institución.

El ámbito de la solución será:

Implementar una aplicación la misma contendrá servicios web y constará con los siguientes módulos:

- Módulo de autenticación.- Este módulo permitirá autenticar a los usuarios dentro del sistema.
- Módulo de Administración.- Este módulo permitirá administrar usuarios y claves.
- Módulo de sincronización de Estudiantes.- Este módulo permitirá realizar la migración de los estudiantes a la Plataforma Virtual de Aprendizaje.

- Módulo de sincronización de Materias.- Este módulo permitirá realizar la migración de las materias a la Plataforma Virtual de Aprendizaje.
- Módulo de sincronización de Cursos/Paralelos.- Este módulo permitirá realizar la migración de los niveles a la Plataforma Virtual de Aprendizaje.
- Módulo de sincronización de Docentes.- Este módulo permitirá realizar la migración de los docentes a la Plataforma Virtual de Aprendizaje.

En la **Figura I. 1** se observa la estructura a generarse al realizar la migración de los datos al Sistema Virtual de Aprendizaje.

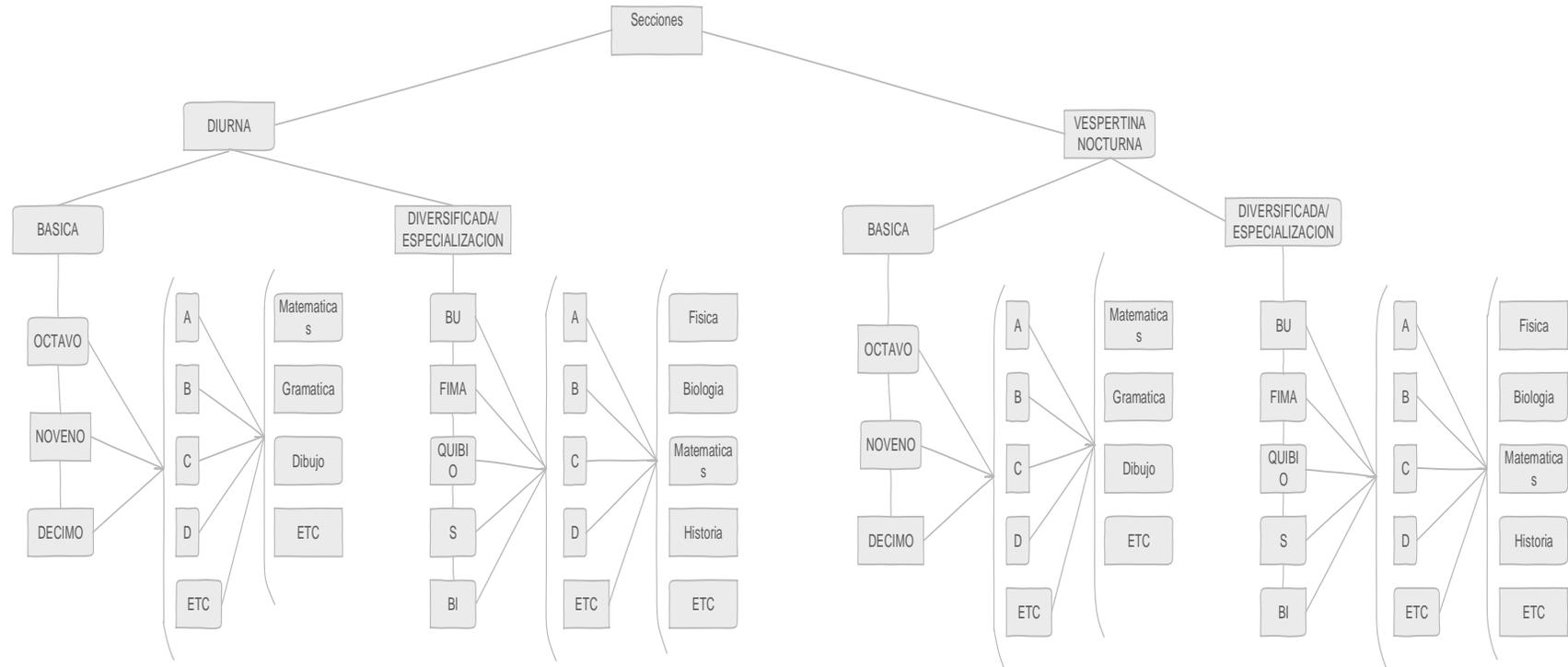


Figura I. 1. Estructura académica para migración de Datos a Moodle

1.3. Objetivos

1.3.1. Objetivos Generales

Realizar un análisis comparativo de servidores de aplicación para la integración de sistemas mediante servicios web xml, aplicado al Colegio Capitán Edmundo Chiriboga.

1.3.2. Objetivos Específicos

- ✓ Analizar los componentes y características de los servidores de aplicación para la publicación de servicios web.
- ✓ Determinar parámetros de comparación entre los servidores de aplicación en base al rendimiento.
- ✓ Construir prototipos para realizar el análisis comparativo de los servidores de aplicación.
- ✓ Seleccionar el servidor de aplicación idóneo al publicar servicios web en base a lo anteriormente establecido.
- ✓ Desarrollar una aplicación utilizando servicios web xml, usando el servidor de aplicaciones con mejor rendimiento que permita publicar servicios.

1.4. Hipótesis

El Servidor de Aplicación JOnAS obtendrá un mejor rendimiento en relación a GlassFish Server y Gerónimo al publicar servicios web.

CAPÍTULO II

MARCO TEÓRICO

2.1. Servidor de Aplicaciones

Según David Serra Manchado (4), en su memoria de proyecto señala que “un servidor de aplicaciones se trata de un dispositivo de software que proporciona servicios de aplicación a computadoras cliente. Un servidor de aplicaciones gestiona las funciones de lógica de negocio y de acceso de datos a la aplicación. Dentro de los principales beneficios tenemos la centralización y la disminución de la complejidad en el desarrollo de aplicaciones, según la referencia”.

“El Servidor de Aplicaciones soporta servicios de publicación Web en procesamiento de transacciones de escala empresarial, además que permite a los desarrolladores construir

aplicaciones basadas en tecnologías Java Server Pages (JSP's), Java Servlets, y Enterprise Java Beans (EJB's)" (5).

En la **Figura II. 2**, se observa que los clientes realizan peticiones por medio de un dispositivo al servidor de aplicaciones, las cuales dependiendo de las configuraciones o reglas de firewall pueden llegar o no al destino, las peticiones pueden ser a nivel de base de datos o lógica de negocio.

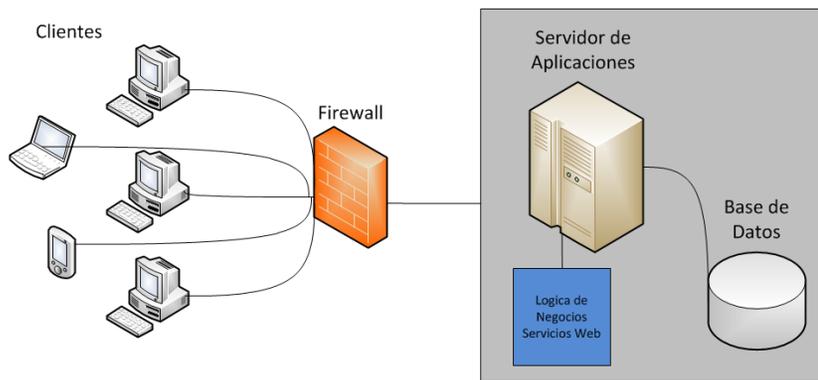


Figura II. 2. Servidor de Aplicaciones

2.1.1. Características de los servidores de aplicación

Los servidores de aplicación poseen cuatro características importantes:

- “Proporcionan middleware que les permite comunicarse con varios servicios, para efectos de confiabilidad y seguridad. Los servidores de aplicación también proporcionan a los desarrolladores una Interfaz API¹, de tal manera que no tengan

¹**Interfaz de Programación de Aplicaciones:** es un conjunto de funciones que permite al programador acceder a servicios de una aplicación a través del uso de un lenguaje de programación.

que preocuparse por el sistema operativo o por la variedad de interfaces requeridas en una aplicación web” (6).

- “Brindan soporte a una gran variedad de estándares, tales como HTML, XML, IIOP, JDBC, SSL, etc., que les permiten su funcionamiento en ambientes web y la conexión a una gran variedad de fuentes de datos, sistemas y dispositivos” (6).
- “Tiene como función convertir los datos, información y aplicaciones desde la estructura lógica interna de la empresa a una lógica externa o lógica del negocio” (6).
- “Proporciona servicios que soportan la ejecución y disponibilidad de las aplicaciones desplegadas. Es el corazón de un gran sistema distribuido” (6).

2.1.2. Principios de los servidores de aplicación

Según el sitio jtech, los tres principios fundamentales de los servidores de aplicación son:

- **“La alta disponibilidad:** se refiere a que un sistema debe estar trabajando las 24 horas del día los 365 días al año. Para cumplir con esto, es indispensable el uso de técnicas de balanceo de carga y de recuperación ante fallos” (7).
- **“La escalabilidad:** se refiere a la capacidad de desplegar un sistema cuando exista gran cantidad de carga de trabajo (número de peticiones). Cada máquina tiene una capacidad finita de recursos y por lo tanto sólo puede servir un número limitado de peticiones” (7).

- **“El mantenimiento:** se refiere a la versatilidad a la hora de actualizar, depurar fallos y mantener un sistema. La solución al mantenimiento es la edificación de la lógica de negocio en unidades reusables y modulares” (7).

2.1.3. Ventaja de los servidores de aplicación

Luis Manuel Guerrero, indica que los servidores de aplicación proporcionan las siguientes ventajas:

- **“Integridad de datos y códigos:** Al estar centralizada en una o un pequeño número de máquinas servidoras, las actualizaciones están garantizadas para todos sus usuarios” (8).
- **“Configuración centralizada:** Los cambios en la configuración de la aplicación, como mover el servidor de base de datos o la configuración del sistema, pueden ser hechos centralmente” (8).
- **“Seguridad:** Un punto central a través del cual los proveedores de servicios pueden administrar el acceso a los datos y de la aplicación en sí se considera como una prestación de seguridad, delegando la responsabilidad de la autenticación a la potencialmente insegura capa de cliente, sin exponer la capa de base de datos” (8).
- **“Rendimiento:** Limitando el tráfico de la red solamente al tráfico de la capa de presentación, es percibido como un modelo cliente/servidor que mejora el desempeño de grandes aplicaciones” (8).

2.1.4. Arquitectura del servidor de aplicaciones

La **Figura II. 3**, representa la arquitectura de un servidor de aplicaciones java, en la que se diferencia 6 componentes muy importantes que son: contenedores web, contenedores EJB, acceso del cliente, servicios web, servicios para aplicaciones y gestión de transacciones.

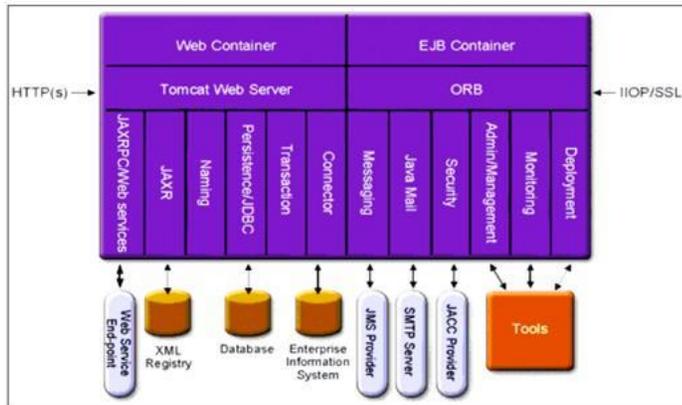


Figura II. 3. Arquitectura del servidor de aplicaciones

Fuente: <http://jano.unicauca.edu.co/apliweb/aservers/sunas/admin2.htm>

A continuación se describe cada uno de los componentes disponibles en la arquitectura de los servidores de aplicación.

Contenedores

“Un contenedor es un entorno en tiempo de ejecución que suministran servicios tales como seguridad y gestión de transacciones para componentes J2EE²” (5). La **Figura II. 3** muestra “los tipos de contenedores J2EE: Web y EJB” (5).

1. Contendor web

Los contenedores web contienen dos componentes que se detallan a continuación:

“**Servlets:** son clases de java que aumentan la funcionalidad de un servidor web mediante la concepción eficiente de páginas web. Un ambiente de ejecución llamado motor de servlets, se encarga de la carga y descarga de servlet, y trabaja con el servidor web HTTP para enviar las peticiones de los clientes a los servlets” (9).

Ciclo de vida de un servlet: Un servlet consta de tres estados, que se detallan a continuación:

- **Inicialización:** “Cuando un servidor carga un servlet ejecuta el método `init`, propio del servlet. Esta etapa se completa antes de administrar peticiones de clientes y antes de que el servlet esté destruido” (9).

²**Java 2 Enterprise Edition:** es una arquitectura multicapa para implementar aplicaciones de tipo empresarial y aplicaciones basadas en la Web. El objetivo principal de la tecnología J2EE es crear un modelo de desarrollo para aplicaciones empresariales utilizando componentes basados en el modelo de aplicación.

- **Servicio:** “Después de la primera etapa, el servlet puede operar peticiones de clientes. Las peticiones son manejadas por la misma instancia del servlet” (9).
- **Destrucción:** “Los servlets se ejecutan hasta que el servidor los destruye, ya sea por cierre del servidor o bien a petición del administrador del sistema. Cuando un servlet se destruye, ejecuta el método destroy del propio servlet” (9).

En la **Figura II. 4**, se observa el ciclo de vida de un servlet, el primer proceso es el de inicialización empieza con la carga de un servlet que es enviado a un servidor, el segundo proceso es el de servicio, el cual el opera las peticiones de los clientes y tercer proceso es el de destrucción por cierre del servidor o por petición del administrador del sistema.

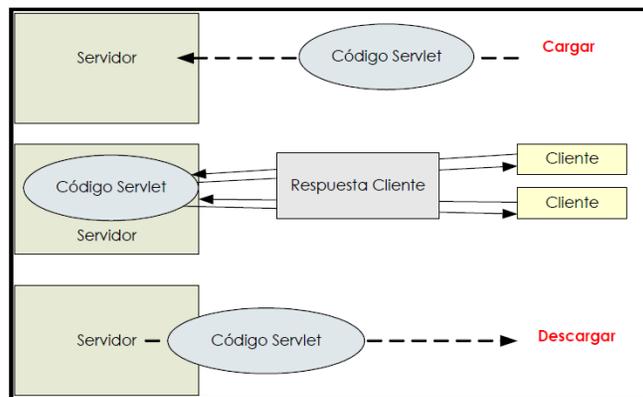


Figura II. 4. Ciclo de vida de un servlet

Fuente: <http://repositorio.utn.edu.ec/bitstream/123456789/625/3/CAPITULO%20II.pdf>

Java server pages: “permite la construcción de páginas web interactivas. De forma muy equivalente al estándar HTML, con la diferencia de que un JSP permite incrustar código Java en el desarrollo de las páginas web” (10).

Ciclo de vida de un java server pages

El ciclo de vida de un JSP consta de tres fases que se detallan a continuación:

- **“Fase de traducción:** después de la primera petición de la página jsp, se traduce en un servlet” (11).
- **“Fase de compilación:** el servlet traducido en la fase de traducción es compilado en esta fase para poder atender las peticiones de los clientes” (11).
- **“Fase de ejecución:** Se crea una instancia del servlet, que estará en memoria de forma permanente mientras el servidor siga en funcionamiento. Para peticiones posteriores se emplea la misma instancia del servlet” (11).

La **Figura II. 5**, representa el ciclo de vida de una página jsp, desde que un navegador invoca una página, la petición llega a un contenedor jsp que lee la página jsp, genera una clase java que es interpretada, compilada y enviada al petición procesada al contenedor jsp, este a su vez entrega la petición al cliente.

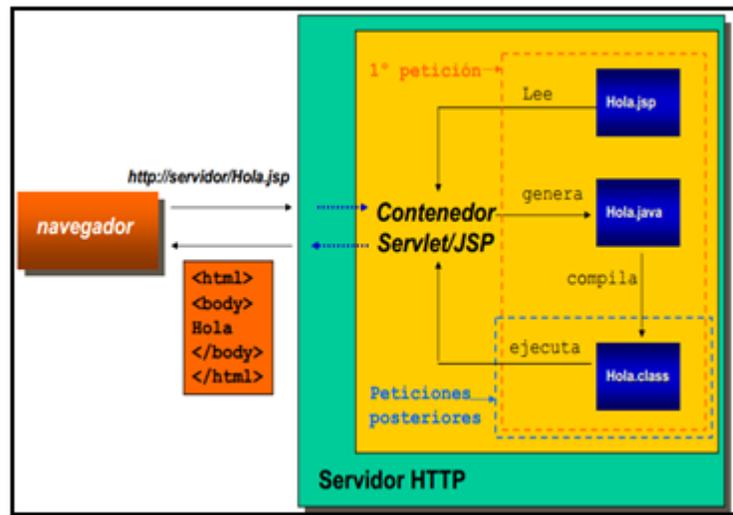


Figura II. 5. Ciclo de vida de una página JSP

Fuente: http://informatica.uv.es/it3guia/ARS/transparencias_2c/Tema13_Servlet-JSP.pdf

2. Contenedor Enterprise Java Beans

“Los EJB son componentes con interfaces definidas que están alojados en un contenedor y que pueden ser colocados en un entorno multicapa distribuido. Se utilizan para resolver problemas de negocios y permite acceder a bases de datos y a otros sistemas. Se caracteriza por utilizar componentes distribuidos, donde diferentes tipos de clientes pueden acceder a ellos” (10).

Tipos de beans

La tecnología EJB define tres tipos de beans que se detallan a continuación:

- **Beans de sesión:** “representa un proceso o una acción de negocio. Normalmente, cualquier llamada a un servicio del servidor debería comenzar con una llamada a

un Bean de sesión. Mientras que un Bean de entidad representa una cosa que se puede representar con un nombre, al pensar en un Bean de sesión se debería pensar en un verbo” (9).

- **Beans de entidad:** “Los beans de entidad representan “cosas”: objetos del mundo real como un estudiante estudiantes. Un bean de entidad puede representar incluso cosas abstractas como una reserva. Los beans de entidad describen tanto el estado como la conducta de objetos del mundo real y permiten a los desarrolladores encapsular las reglas de datos y de negocio asociadas con un concepto específico” (12).
- **Beans dirigidos por mensajes:** “pueden escuchar mensajes de un servicio de mensajes JMS. Los clientes de estos beans nunca los llaman directamente, sino que es necesario enviar un mensaje JMS para comunicarse con ellos” (12).

La **Figura II. 6**, representa el funcionamiento básico de los Enterprise JAVA beans, donde se puede observar que, el cliente realiza peticiones al bean³ y el servidor está ejecutando en máquinas virtuales Java. El cliente nunca se comunica directamente con el enterprise bean, sino que el contenedor EJB proporciona un EJBObject que hace de interfaz. Las peticiones se hacen a través de un objeto EJB, mismo que solicita al contenedor EJB una serie de servicios y se comunica con el enterprise bean. Por último, el bean realiza las peticiones correspondientes a la base de datos.

³ **JavaBean o bean:** es un componente hecho en software que se puede reutilizar y que puede ser manipulado visualmente por una herramienta de programación en lenguaje Java.

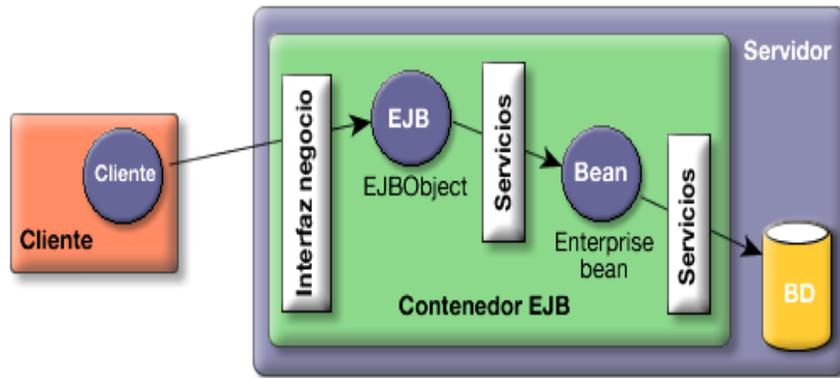


Figura II. 6. Funcionamiento de los componentes EJB

Fuente: <http://www.jtech.ua.es/j2ee/2003-2004/abierto-j2ee-2003-2004/ejb/sesion01-apuntes.htm>

2.2. Servicios Web

Según Mónica Cecilia Gallegos (13), “un servicio Web es una unidad de la lógica del negocio que proporciona datos y servicios para otras aplicaciones. Las aplicaciones acceden a los servicios web vía protocolos Web como HTTP y SOAP y formatos de datos universales como XML, sin necesidad de preocuparse de cómo cada Web Service es implementado. Los servicios del Web combinan los mejores aspectos del desarrollo basado en componentes y el Web, y son la piedra angular del modelo de programación”.

En la **Figura II.7**, se observa la descripción del proceso de un servicio web que es iniciado por el usuario haciendo una búsqueda del servicio requerido en un registro,

el mismo que proporciona la descripción del servicio web y URL⁴ al cliente y envía una solicitud por medio de un http al servidor donde se encuentra alojado el servicio web, el mismo que procesa y devuelve el resultado al cliente.



Figura II. 7. Descripción del proceso del servicio web

2.2.1. Características de un servicio web

Según Mónica Cecilia Gallegos (13), se describe siete características importantes de los servicios web:

- **“Comunicación ubicua:** La conexión de cualquier sistema o dispositivo a Internet debe garantizar la disponibilidad para cualquier sistema o dispositivo conectado a Internet” (13).
- **“Formato de datos universal:** Cualquier sistema compatible con estándares abiertos como mensajes de texto auto-descriptivos puede acertar y

⁴**Localizador Uniforme de Recursos:** sirve para nombrar recursos en Internet, tiene como propósito asignar una dirección única a cada uno de los recursos disponibles en Internet, como por ejemplo textos, imágenes, vídeos, etc.

compartir los servicios Web XML y permitir la comunicación entre sistemas heterogéneos” (13).

- **“Interoperabilidad:** Un servicio web debe permitir su utilización por clientes de otras plataformas” (13).
- **“Amigabilidad con Internet:** El sistema debe soportar clientes que accedan a los servicios desde Internet” (13).
- **“Interfaces fuertemente tipadas:** No debe haber ambigüedad sobre el tipo de dato enviado y recibido desde un servicio” (13).
- **“Aprovechar los estándares de Internet existentes:** Al implementar el servicio WEB, se debe evitar reinventar soluciones a problemas que ya han sido resuelto” (13).
- **“Soporte para cualquier lenguaje:** Un servicio Web es independiente del lenguaje de programación en el que haya escrito el cliente” (13).
- **“Soporte para cualquier infraestructura de componente distribuida:** La solución no debe estar ligada solo a una infraestructura de componentes en particular” (13).

2.2.2. Ventajas de los Servicios Web respecto aplicaciones distribuidas

Según Mónica Cecilia Gallegos (13), Los servicios Web XML presentan las siguientes ventajas:

- “Son la puerta a nuevas oportunidades empresariales, facilitan la comunicación entre aplicaciones asociadas” (13).
- “Permiten que las aplicaciones compartan información” (13).
- “Son unidades de código discretas, cada una de las cuales se encarga de un conjunto limitado de tareas” (13).
- “Están basados en XML, el lenguaje universal del intercambio de información de Internet y pueden utilizarse en cualquier plataforma o sistema operativo, independientemente del lenguaje de programación utilizado” (13).
- “Aumenta el flujo de ingreso, ya que ponen sus propios servicios Web XML a disponibilidad de otros” (13).
- “Cualquier servicio Web puede interactuar con cualquier otro servicio Web, a través del protocolo estándar SOAP” (13).
- “Permite que se comuniquen utilizando HTTP y XML. Cualquier dispositivo que trabaje con estas tecnologías puede ser huésped y acceder a los servicios Web” (13).

2.2.3. Arquitectura de un servicio web

“Un servicio Web es un servicio proporcionado por una aplicación, que expone su lógica a clientes de cualquier plataforma mediante una interfaz accesible, a través de la red utilizando tecnologías (protocolos) estándar de Internet” (13).

En la **Figura II. 8**, se puede observar la arquitectura de un servicio web, donde el cliente realiza una solicitud de servicio, que es receptada por el agente que escucha, procesando en la lógica empresarial y la lógica de negocio, envía las solicitudes a nivel de base de datos, de la misma manera este proceso es invertido y finalmente las solicitudes procesadas son enviadas al cliente.

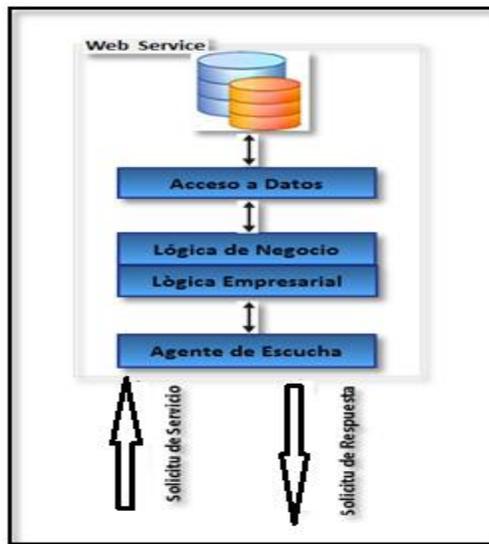


Figura II. 8. Arquitectura de Servicios Web

Según Mónica Cecilia Gallegos (13), La arquitectura de un servicio web se divide en cuatro capas que son:

- **“Capa de acceso a datos:** Presenta una vista lógica de los datos físicos a la capa de negocios, aísla la lógica de negocios de los cambios realizados a los almacenes de datos y garantiza la integridad de los datos” (13).
- **“Capa de negocios:** Implementa la lógica de negocios del servicio Web” (13).

- **“La Lógica Empresarial:** Proporciona una interfaz sencilla que se asigna a las operaciones expuestas por el servicio Web” (13).
- **“El agente de escucha:** Recibe los mensajes entrantes que contienen solicitudes de servicios, analiza los mensajes y envía la solicitud al método apropiado en la capa de negocios. Si el servicio devuelve una respuesta, el agente de escucha empaqueta la respuesta de la capa de negocios en un mensaje y su envío al cliente” (13).

2.2.4. Tecnología de servicios web

Según HUGO CÉSAR PONCE SUÁREZ (14), los servicios web requieren de varias tecnologías basadas en XML que se describe a continuación:

- **Extensible Markup Language:** “XML para los servicios web es una de las bases fundamentales de los servicios, gracias a este lenguaje de etiquetado es posible realizar los servicios Web. XML es un mecanismo para la descripción de datos, que puede ser utilizado con cualquier tipo de dato, independientemente del tipo” (14). “Todos los parámetros de un servicio web se encuentran definidos en XML, estos parámetros pueden ser de entrada y salida, así como su estructura. Para que las aplicaciones entiendan los datos en formato XML necesitan un guía, esto es, un nivel de inteligencia mutua que lo comparten tanto el proveedor como el consumidor” (14).

- **Web Services Description Language:** “es un esquema XML que define una estructura para describir las interfaces de los servicios Web” (14).

“WSDL es el núcleo de la estructura de los servicios Web, proporcionan un sentido común en el que los tipos de datos son pasados a mensajes, así como las operaciones que son ejecutadas en los mensajes y el mapeo de mensaje sobre la red. Inicialmente el documento WSDL estaba definido en tres elementos principales: definición de los tipos de datos, operaciones abstractas, enlace de servicios” (14).

“Los mensajes pueden tener múltiples partes y pueden ser definidas para usarse con la orientación a procedimientos o la interacción orientada a documento. A través de las capas de abstracción, los mismos mensajes pueden ser definidos y usados por múltiples puertos” (14).

“WSDL juega un rol importante que facilita las ventajas de los servicios Web y la orientación a servicio” (14).

- **Simple Object Access Protocol:** “define la estructura para el intercambio de información en XML a través de Internet y de esta manera provee el mecanismo de comunicación para conectar los servicios Web. La estructura del mensaje es simple, fácil de desarrollar y completamente neutral con respecto al sistema operativo, lenguaje de programación. Este protocolo se encuentra en constante desarrollo y el encargado de su evolución es el consorcio W3C” (14).

“SOAP está fundamentado en el modelo de comunicación de un solo sentido, que asegura el envío de un mensaje desde el transmisor al receptor, incluye intermediarios que pueden procesar o agregar partes al mensaje” (14).

“SOAP es un conjunto de acuerdos que especifican el formato de un mensaje a través de un conjunto de reglas que gobiernan su procesamiento a lo largo de su ruta. Estas convenciones describen como un mensaje es ensamblado y que nodos los procesan” (14).

En la **Figura II. 9**, se puede observar que “un mensaje SOAP es la unidad básica de comunicación entre nodos y está compuesto por tres elementos principales: Envelope, Header y Body. Cabe señalar que las etiquetas Envelope y Body son obligatorias dentro del cuerpo del mensaje SOAP” (14).

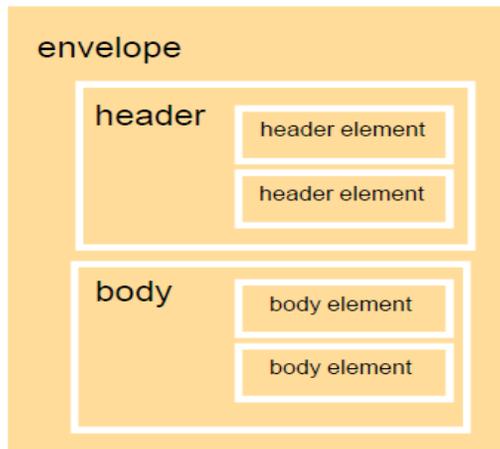


Figura II. 9. Esquema Protocolo SOA

Fuente: http://www.arcos.inf.uc3m.es/~acaldero/UC3M/lib/exe/fetch.php?media=teach:ii-dad:l10-servicio_web-v1b.pdf

A continuación se describe los tres elementos que componen un mensaje SOA:

- **“Envelope:** Define el inicio y el fin del mensaje, puede contener uno o más elementos header” (14).
- **“Header:** Contiene cualquier atributo opcional del mensaje usado en el procesamiento del mismo” (14).
- **“Body:** Contiene los datos XML que abarca el mensaje cuando es enviado, también conocida como información de negocio” (14).

2.3. SERVIDOR DE APLICACIONES GLASSFISH

Según David Serra Manchado (4) “glassFish es un servidor de aplicaciones desarrollado por Sun Microsystems que implementa las tecnologías definidas en la plataforma Java EE y permite ejecutar aplicaciones que siguen esta especificación. La versión comercial es denominada Sun GlassFish Enterprise Server. Soporta las últimas versiones de tecnologías como: JSP, Servlets, EJBs, Java API para Servicios Web (JAX-WS)”.

2.3.1. Características

Según José Torres y Jinson Villagomez, las principales características de GlassFish son:

- **“Clustering:** permite a las empresas agrupar servidores para lograr un mayor nivel de escalabilidad y replicar datos en memoria para su protección y alta disponibilidad en caso de caídas del sistema” (1).

- **“Administración Centralizada:** permite gestionar clusters de servidores de aplicaciones y despliegues de aplicaciones desde una consola de administración centralizada” (1).
- **“Java Business Integration (JBI):** permite proporcionar un enfoque centralizado para suministrar SOA utilizando servicios Web” (1).
- **“Integración con el Entorno de Desarrollo Integrado NetBeans(TM) IDE.** Permite a los desarrolladores desplegar aplicaciones SOA” (1).

2.3.2. Arquitectura de GlassFish

La **Figura II. 10**, representa la arquitectura del servidor de aplicaciones GlassFish, donde se observa que “está basado en una estructura de clusters flexibles, además brinda el soporte para gestión remota multimáquina y multi-dominio segura, lo cual pretende mejorar el grado de disponibilidad y la escalabilidad en el rendimiento del servidor de aplicaciones para sistemas de alto volumen y Servicios Web.” (15).

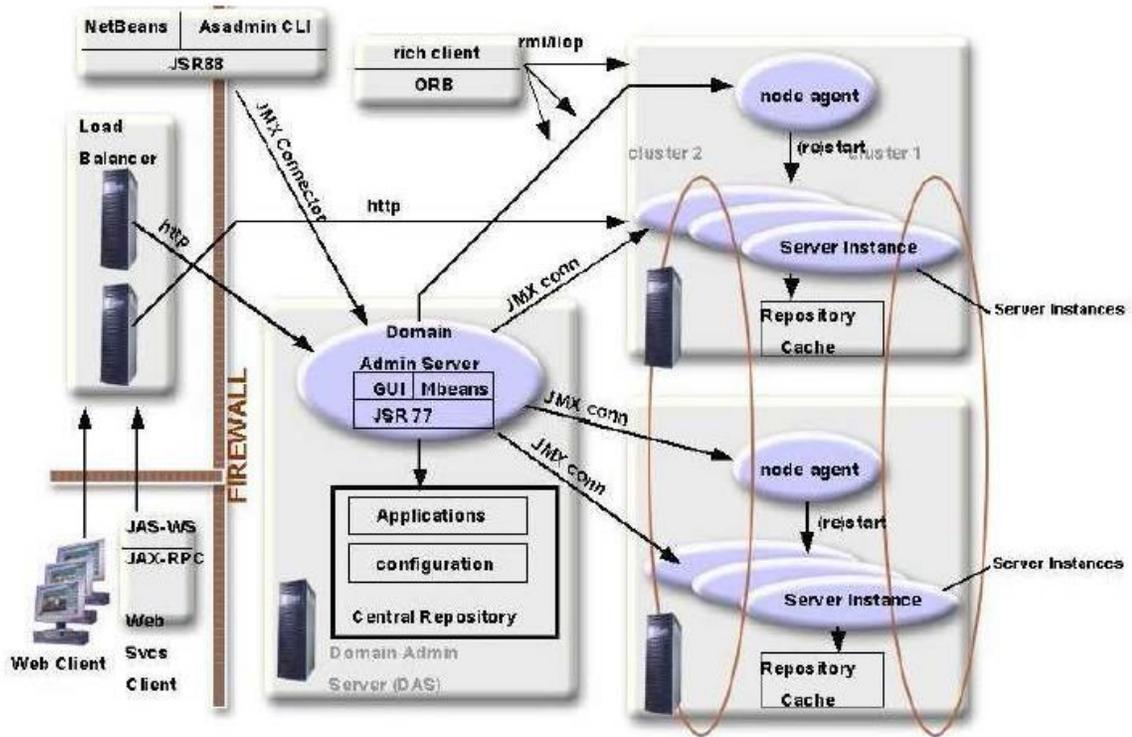


Figura II. 10 Arquitectura Servidor de Aplicaciones GlassFish

Fuente: <http://bibdigital.epn.edu.ec/bitstream/15000/1513/1/CD-2217.pdf>

A continuación se describe los componentes de la arquitectura del servidor GlassFish.

- **“Instancia:** Una instancia de servidor de aplicaciones es un motor del servidor de aplicaciones JavaEE donde se alojan las aplicaciones Java ejecutándose sobre una única Java Virtual Machine (JVM)” (15). “Cada instancia funciona como un servidor JMX MBean, es decir cuenta con los recursos administrativos para todos los servicios que brinda el servidor, puede existir dos tipos de instancias, las instancias que pertenecen a un cluster y las instancias independientes; las instancias que pertenecen a un cluster son homogéneas recibiendo los mismos recursos y

configuración de su cluster padre en cambio la instancia independiente tiene su propio set de aplicaciones, recursos y configuración” (15).

- **“Cluster:** Un cluster es una entidad virtualizada que pertenece exactamente a un solo dominio y que contiene un grupo de instancias homogéneas trabajando en conjunto, es decir trabajan con las mismas aplicaciones, recursos e información de configuración” (15). “Un cluster es una importante entidad de administración ofreciendo a los usuarios un servicio transparente en el manejo de las instancias y sus recursos, por ejemplo una aplicación puede ser desplegada en un cluster en una operación agregada sin tener la necesidad de desplegar la aplicación en cada instancia perteneciente al cluster, el usuario no necesita estar consciente de cuantas instancias provean el servicio. Además un cluster facilita el equilibrio de la carga al ser escalable horizontalmente ya que se pueden agregar instancias adicionales de servidor” (15).
- **“Dominio:** Un dominio es un punto unificado de la administración para cualquier número de clusters o instancias del servidor así como sus configuraciones y aplicaciones que son administradas en conjunto” (15). “Este actúa como un único punto de autenticación administrativa y autorización, además en una instalación del servidor pueden agregarse múltiples dominios, actuando como mínimo un dominio que trabaje en una máquina, cada dominio puede definirse como un ente independientemente manejado, restringiendo a un desarrollador la administración de cada instancia de dominio” (15). “Por último cada dominio cuenta con un

Servidor de Administración de Dominio (DAS) que se emplea para administrar instancias del servidor de aplicaciones en el dominio” (15).

- **“Servidor de Administración de Dominio (DAS):** es una instancia del servidor de aplicaciones que administra la configuración del dominio y alberga dicha configuración en un Repositorio (ficheros en el File System). El DAS utiliza una consola para la administración del depósito central que es una interfaz basada en explorador” (15).
- **“Nodo Agente:** El nodo agente es un proceso ligero que facilita la gestión del ciclo de vida remoto de las instancias, se ejecuta en cada Host físico o nodo en el cual funciona por lo menos una instancia del servidor” (15). “El propósito del nodo agente será administrar características de las instancias como: arrancar, parar, crear y reiniciar instancias caídas, así como lectura de logs y especialmente sincronizar la configuración de todo el dominio en conexión con el DAS, por último el nodo agente corre sobre una JVM recibiendo solamente una API JMX en tiempo de ejecución” (15).
- **“Balanceador De Carga:** Es un subsistema que distribuye los requerimientos de varios servidores de aplicaciones y servidores Web dentro de las instancias del servidor es decir provee balanceo de carga para clientes HTTP, RMI/IIOP y JMS, revisando sus operaciones si es necesario y manteniendo una conexión con un host particular con el que se ha establecido una sesión” (15).

2.4. SERVIDOR DE APLICACIONES GERÓNIMO

Segun David Serrano (4), define a Gerónimo como “un servidor de aplicaciones de código abierto⁵ desarrollado por la Apache Software Foundation, actualmente es compatible con la especificación Java Enterprise Edition (Java EE) 5.0. Se caracteriza por un diseño arquitectónico que se basa en el concepto de Inversión of Control (COI), lo que significa que el núcleo no tiene dependencia directa de cualquiera de sus componentes”. “El núcleo es un framework para los servicios que controla el ciclo de vida de servicio y el registro y está basado en JavaEE” (4).

2.4.1. Características

Las características del servidor de aplicaciones Gerónimo son:

- Es de código abierto.
- “El tiempo de ejecución es bueno para satisfacer las necesidades de desarrolladores, administradores e integradores de sistemas” (4).
- “Actualizaciones frecuentes con nuevas características y corrección de errores” (4).
- “Existe una comunidad que desarrolla nuevas herramientas constantemente” (4).

⁵**Código abierto:** es aquel código distribuido bajo una licencia que permite su uso, modificación y redistribución. Como su nombre lo indica, el requisito principal para que una aplicación sea considerada bajo esta categoría es que el código fuente se encuentre disponible.

2.4.2. Arquitectura de Gerónimo

La **Figura II. 11** representa la arquitectura de servidor de aplicaciones Gerónimo que está compuesto por: gbeans y mbeans.

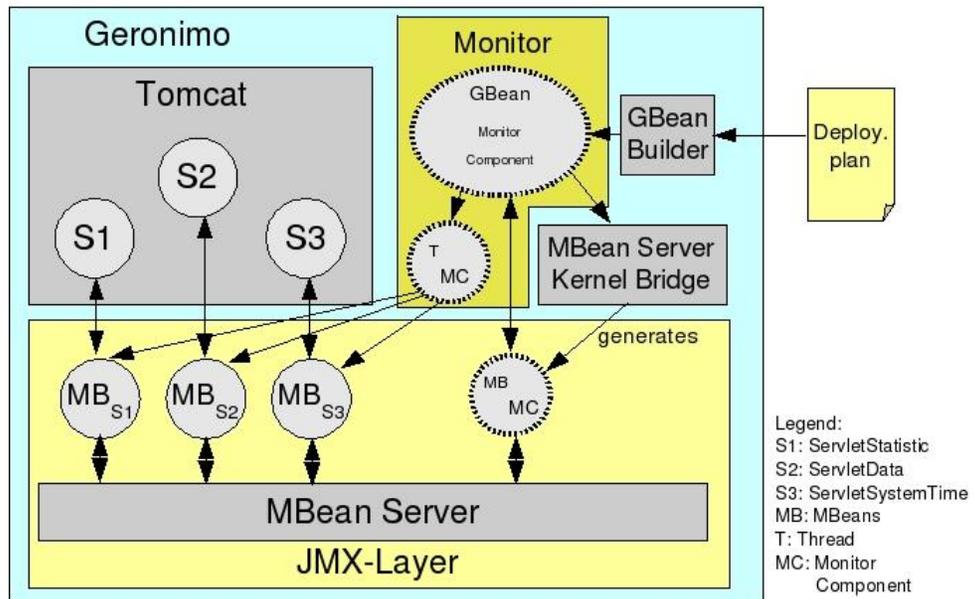


Figura II. 11. Arquitectura Servidor de Aplicaciones Gerónimo

Fuente: <http://geronimo.apache.org/GMOxDOC30/geronimo-architecture.html>

A continuación se detallan los componentes de la arquitectura del servidor de aplicaciones Gerónimo.

Gbean

GBean es una unidad gestionable de la infraestructura Gerónimo. Puede haber múltiples instancias activas de un mismo GBean. Gerónimo asigna un nombre exclusivo a cada GBean en el momento de iniciarse.

“GBeans son los cimientos a partir de los cuales está hecho Apache Gerónimo, casi todo en Gerónimo es un GBean, esto es, contenedores, aplicaciones, conectores, etc, tal que un GBean pueda agregar con certeza a otros GBeans, en especial a los de gran dimensión. Una instancia de servidor Gerónimo es básicamente una colección de GBeans, tal que el kernel de Gerónimo pudiese ser, en esencia, un contenedor de GBean” (16).

“Un GBeans puede mantener estados, definir dependencias (internas y externas) y reaccionar a ciertos eventos del ciclo de vida. Esta capacidad puede ser usada para guardar el estado actual del GBean y recobrar a dicho estado en un tiempo posterior. GBeans también puede ser administrados a través de una interfaz de administración (línea de comandos o la Consola Gerónimo), permitiéndoles poder ser iniciados, detenidos y editados, al vuelo” (16).

La siguiente **Figura II. 12**, se observa el ciclo de vida de un GBeans cuando está en línea es: iniciado, ejecutado, detenido, deteniendo o en fallo.

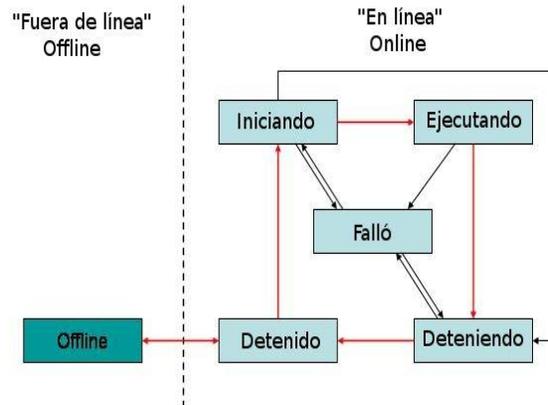


Figura II. 12. Ciclo de vida de un gbeans

Fuete: <http://geronimo.apache.org/GMOxDOC10/arquitectura.html>

2.5. SERVIDOR DE APLICACIONES JONAS

“JOnAS es un servidor de aplicaciones JEE de código abierto implementado en Java que forma parte de la iniciativa de código abierto de ObjectWeb con colaboración de empresas como Bull, France Telecom, Red Hat y MySQL, entre otras asociaciones” (17) .

2.5.1. Características

Las principales características del servidor de aplicaciones JOnAS son:

- Es de código abierto (18).
- Proporciona alta modularidad y capacidad de configuración del servidor (18).
- Permite aplicar un enfoque de componentes-modelo a nivel de middleware, y hace fácil la integración de nuevos módulos (18).

- Proporciona una manera de empezar sólo los servicios necesarios para una aplicación en particular, lo que ahorra valiosos recursos del sistema” (18).
- Según el sitio oficial de jonas (18) “Está diseñado con servicios, normalmente proporciona los recursos del sistema a los contenedores. La mayoría de los componentes del servidor de aplicaciones JOnAS están predefinidos en servicios JOnAS. Sin embargo, es posible y fácil para un usuario de JOnAS definir un servicio e integrarlo en JOnAS. Dado que las aplicaciones J2EE no necesariamente tienen todos los servicios, es posible definir, en el momento de configuración el servidor JOnAS, el conjunto de servicios que se pondrán en marcha en el arranque del servidor”.

2.5.2. Arquitectura de JOnAS

En la **Figura II. 13**, se observa la arquitectura de servidor de aplicaciones JOnAS que está compuesto por: contenedor web, servicios EAR, servicio de transacciones, servicio de base de datos, servicio de mensajería, servicio de servicios web y clustering y rendimiento.

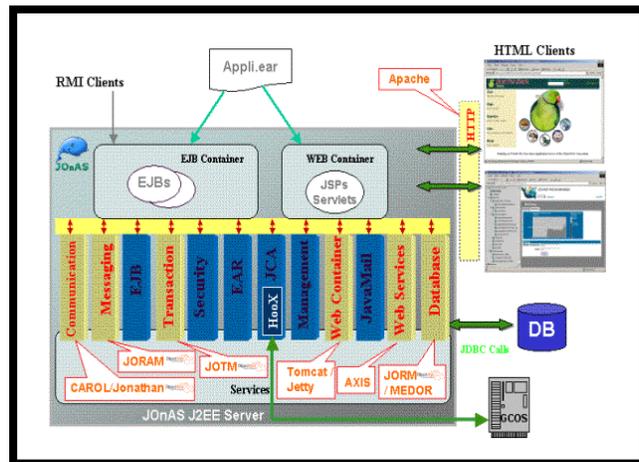


Figura II. 13. Arquitectura servidor de aplicaciones JOnAS

Fuente: <http://jonas.ow2.org/current/doc/doc-en/integrated/howto/JonasArchi.html>

A continuación se detalla cada uno de los componentes de la arquitectura del servidor de aplicaciones JOnAS.

- **Servicio de Contenedor Web**

Según el sitio oficial de jonas (19), el servicio de contenedor web “es el encargado de dirigir un motor de servlet/JSP en la JVM del servidor JOnAS y de la carga de aplicaciones web dentro de este motor. Actualmente, este servicio se puede configurar para utilizar Tomcat o Jetty. Los servlet/jsp están integrados en JOnAS como "contenedores Web", es decir, proporcionan los componentes Web con el acceso a los recursos del sistema (del servidor de aplicaciones) y los componentes EJB, de una manera compatible con J2EE”.

“La interfaz de este servicio define los métodos comunes como iniciar, detener el servidor incorporado, o desplegar. Además, estos métodos públicos están disponibles a través de un MBean” (19).

- **Servicio EAR**

Según el sitio oficial de jonas (19), el servicio EAR “se utiliza para el despliegue de aplicaciones J2EE completas, es decir, aplicaciones empaquetadas en archivos EAR, que a su vez contienen archivos ejb-jar. Este servicio se encarga de los archivos EAR y delega la implementación de los archivos al servicio de contenedor web y los ficheros ejb-jar al servicio de contenedor EJB. Se ocupa de la creación de los cargadores de clases apropiadas, a fin de que la aplicación J2EE se ejecute correctamente”.

- **Servicio de Transacciones**

Según el sitio oficial de jonas (19), el servicio de transacciones “encapsula un monitor de transacciones Java llamado JOTM (un proyecto de ObjectWeb). Se trata de un servicio obligatorio que se ocupa de las transacciones distribuidas. Proporciona la gestión de transacciones para los componentes EJB como se define en sus descriptores de despliegue”.

- **Servicio de base de datos**

“Este servicio es responsable de manejar los objetos de origen de datos. Un origen de datos es un objeto estándar JDBC administrativo para el manejo de las conexiones a una base de datos” (19).

- **Servicio de mensajería**

“Para apoyar beans controlados por mensajes JMS y operaciones codificadas dentro de los componentes de la aplicación, el servidor de aplicaciones JOnAS se basa en una implementación de JMS⁶” (19).

- **Servicio de servicios web**

Según el sitio oficial de jonas (19), el servicio de servicios web “se utiliza para el despliegue de servicios Web. Sólo una parte muy pequeña del trabajo de implementación se realiza durante el tiempo de instalación. Este servicio de JOnAS se utiliza para establecer una propiedad de Axis en JOnAS y está a cargo de algunas operaciones de implementación de servicios web”.

- **Clustering y Rendimiento**

Según el sitio oficial de jonas (19), el clusterin“de servidor de aplicaciones en general hace uso de tres características: Equilibrio de carga (LB), de alta

⁶**Servicio de Mensajes Java(JMS):**es un estándar de mensajería java, diseñado para aplicaciones basadas en componentesJ2EE / JEE

disponibilidad (HA) y la conmutación por error. Estos mecanismos se pueden proporcionar en el nivel de contenedor Web mediante el envío de solicitudes a varios Servlet / JSP, en el nivel de contenedor EJB mediante el envío de peticiones EJB a varias instancias del contenedor EJB, y en el nivel de base de datos mediante el uso de varias bases de datos”.

Los servidores de aplicación tanto GlassFish, JOnAS y Gerónimo presentan diferentes esquemas y componetes en sus arquitecturas para el despliegue de las aplicaciones, pero mantienen un mismo conjunto de servicios como manejo de transacciones, contenedores EJB entre otros, en cuanto al soporte GlassFish posee gran cantidad de documentación y ofrecen una interfaz muy amigable para la publicación y despilgüe de las aplicaciones.

CAPÍTULO III

3. Análisis comparativo de servidores de aplicación Glassfish, JOnAS y Gerónimo

Con la finalidad de determinar el servidor de aplicaciones con mejor rendimiento se realizaron prototipos, se establecieron parámetros de comparación y se definió una hipótesis que estará sujeta a comprobación.

En el presente capítulo se plantea realizar un conjunto de pruebas sobre los servidores de aplicación GlassFish, JOnAS y Gerónimo que permitirán determinar cual es el que cumple con el mejor rendimiento al momento de proporcionar información por medio de servicios web.

En base al estudio realizado la unidad educativa implementará en el servidor de aplicaciones ganador los servicios web y aplicaciones que permitirán a la institución el intercambio de información entre sistemas.

3.1. Definición de parámetros de comparación

Los parámetros seleccionados para el análisis comparativo se basan en la correcta administración que tengan los servidores de aplicaciones sobre los recursos hardware y el tiempo que tomen los mismos en responder solicitudes.

Importancia de los recursos Hardware

Según el sitio ehowenespanol (20), indica que “la velocidad y el rendimiento de una computadora ya no es simplemente la medida de la capacidad de la CPU para procesar números. Como la tecnología ha avanzado, otros factores de hardware, como la memoria RAM (Random Access Memory - Memoria de acceso aleatoria) disponible y el rendimiento del disco duro, han venido a apoyar a la CPU lo suficiente para justificar su importancia como significativos”.

Importancia del tiempo de respuesta

El tiempo de respuesta que brinde un servidor de aplicaciones corresponde el éxito o el fracaso de la aplicación o servicio web. “Desde un punto de vista de rendimiento, su objetivo es asegurarse de que el usuario final o cliente haga clic en el ratón y se cumpla a tiempo su petición” (21) . En la **Tabla III. I**, se define los parámetros a ser evaluadas en el presente análisis.

Tabla III. I. Definición de los parámetros de comparación

PARÁMETROS	DESCRIPCIÓN
USO DE RAM	Proporciona la información sobre de “cantidad de memoria RAM utilizada en” (22) la ejecución de los procesos del servidor de aplicaciones.
USO DE CPU	Proporciona el “porcentaje de procesador que está siendo utilizado por los procesos en ejecución en el servidor de aplicaciones” (22)
USO DE DISCO	Proporciona la información sobre el espacio de disco utilizado por el servidor de aplicaciones al ejecutar los procesos.
TIEMPOS DE RESPUESTA	“Es la cantidad de tiempo que le toma a un servidor el procesar la llegada de información a través de un paquete, y posteriormente enviarlo al origen. Este parámetro sirve para determinar el desempeño del servidor”. (22)

Ponderación

En la **Tabla III. II**, se determina la ponderación para cada uno de los parámetros considerados en el análisis de rendimiento de los servidores de aplicación, que han sido asignados de acuerdo al nivel de importancia dentro de un servidor de aplicaciones y a las necesidades de la institución, teniendo como prioridad el tiempo de respuesta, debido a que es un aspecto muy importante para el correcto funcionamiento de servicios web y aplicaciones alojadas en un servidor de aplicaciones. Esto implica que, cuando el tiempo de respuesta sea menor a “0,1s el usuario siente que el sistema está reaccionando instantáneamente. Cuando es menor a 1,0s la experiencia del usuario no se ve comprometida. Aunque el usuario

no está satisfecho con la espera pero aún se centra en la página web actual” (23). Cuando es menor a “10s como los tiempos de espera se acercan a 10 segundos, los estudios han demostrado que la posibilidad de distracción de usuario aumenta considerablemente. Cuando es mayor a 10s el usuario está más probable a distraerse de la actual página web y pierde interés” (23).

Tabla III. II. Ponderación de parámetros

VARIABLES	Porcentaje (%)
Uso de RAM	20%
Uso de CPU	25%
Uso de Disco	10%
Tiempo de Respuesta	45%
TOTAL	100%

3.2. Especificación del hardware donde se realizaron las pruebas

Las características físicas del computador que dispone la institución para realizar las pruebas en los prototipos desarrollados para cada servidor de aplicación son: Disco SATA de 250 Gb, Memoria RAM DDR2 de 2Gb y CPU AMD Athlon X2 Dual Core de 2.6Ghz como se detalla en la **Tabla III. III.**

Tabla III. III. Especificación hardware del equipo

HARDWARE	CARACTERÍSTICAS
DISCO	SATA(250Gb)
MEMORIA RAM	DDR2 (2 Gb)
CPU	AMD Athlon X2 Dual Core (2.6Ghz)

3.3. Especificación del software donde se realizaron las pruebas

El servidor donde se desarrollaran las pruebas constara de un sistema operativo Centos 6.0, el JDK de java, Glassfish, JOnAS y Geronimo, en la **Tabla III. IV** se encuentra especificado el software utilizado.

Tabla III. IV. Especificación software

SOFTWARE	DESCRIPCION
Sistema Operativo	Centos 6.0
GlassFish	v. 3.1.2
JOnAS	v. 5.3
Geronimo	v. 3.0
Java Development Kit	v. 1.7.0_55
Java Runtime Environment	v. 1.7.0_55

3.4. Descripción del experimento

Para determinar el servidor de aplicaciones con mejor rendimiento se realizó pruebas de carga. Para lo cual se envió diferentes cantidades que son: 20000, 40000 y 60000 solicitudes para estresar al servidor, con lo que se analizó el comportamiento de los parámetros: RAM, CPU, disco y tiempo de respuesta, “mientras menor sea el porcentaje obtenido de cada uno de los parámetros mejor será el rendimiento del servidor de aplicaciones” (24).

Para las pruebas de carga se utilizó el comando ab (Apache Benchmark), una herramienta para el testeado de rendimiento. Esta herramienta permite simular las solicitudes contra el servidor de aplicaciones.

3.5. Operacionalización metodológica y conceptual

- **Hipótesis**

El servidor de aplicación JOnAS obtendrá un mejor rendimiento en relación a GlassFish Server y Gerónimo al publicar servicios web.

- **Tipo de hipótesis**

La Hipótesis de esta investigación es de tipo Causa - Efecto.

Causa

Número de peticiones.

Efecto

Seleccionar el servidor de aplicaciones con mejor rendimiento.

- **Operacionalización conceptual**

Tabla III. V. Operacionalización conceptual

VARIABLE	TIPO
Servidores de Aplicación	Independiente
Rendimiento	Dependiente

- **Operacionalización metodológica**

Tabla III. VI. Operacionalización metodológica

VARIABLES	INDICADORES	TÉCNICAS	FUENTES DE VERIFICACIÓN
Servidores de	GlassFish	Revisión de	<ul style="list-style-type: none">• Internet

Aplicación		documentos	<ul style="list-style-type: none"> • Manuales • Tutoriales • Libros
	JOnAS		<ul style="list-style-type: none"> • Internet • Manuales • Tutoriales • Libros
	Gerónimo.		<ul style="list-style-type: none"> • Internet • Manuales • Tutoriales • Libros
Rendimiento	Uso de RAM	Observación Experimentación	<ul style="list-style-type: none"> • Envío de solicitudes • Línea de comandos (sar -r)
	Uso de CPU	Observación Experimentación	<ul style="list-style-type: none"> • Envío de solicitudes • Línea de comandos (sar -u)
	Uso de disco	Observación Experimentación	<ul style="list-style-type: none"> • Envío de solicitudes • Línea de comandos (sar -d)
	Tiempos de respuesta	Observación Experimentación	<ul style="list-style-type: none"> • Envío de solicitudes • Línea de comandos (time)

3.6. Descripción de las herramientas utilizadas

Para la comprobación de la hipótesis se utilizó el comando sar que se detalla a continuación:

3.6.1. Comando sar

El comando **sar** “permite monitorear el uso de CPU, memoria, discos, red y otros proceso, obteniendo métricas en tiempo real” (25).

En la **Figura III.14** se detalla la sintaxis del comando sar, donde en el parámetro <nro. de mediciones> indica cuantos valores tomar y el intervalo de medición indica cada cuantos segundos se realizará una medición.

SINTAXIS
sar <opción> <intervalo de medición> <nro. De mediciones>
sar -r 1 1

Figura III. 14. Sintaxis comando SAR

Para el presente análisis se utilizó el valor de 1308 como número de mediciones para monitorear el comportamiento de los servidores de aplicación, detallado en el **Anexo 1 – Sección B**.

3.6.2. Comando sar -r

El parámetro **-r** del comando sar especifica el monitoreo de la RAM, que es obligatorio ya que con este parámetro estamos delimitando que el monitoreo sea solamente para RAM. En la **Tabla III. VII**, se detalla la salida comando sar -r.

Tabla III. VII. Salida comando sar -r

NOMBRE CAMPO	DESCRIPCIÓN
Kbmemfree	Representa la cantidad de memoria libre en kilobytes
Kbmemused	Representa la cantidad de memoria utilizada en kilobytes

%memused	Representa el porcentaje utilizado de memoria
Kbbuffers	Representa la cantidad de espacio de búfer utilizado por el kernel en kilobytes
Kbcached	Representa la cantidad de espacio en caché utilizado por el kernel en kilobytes
Kbcommit	Representa la cantidad de memoria en kilobytes necesarios para la carga de trabajo actual. Esta es una estimación de la cantidad necesaria de RAM/swap para garantizar que nunca se quede sin memoria.
%commit	Representa el porcentaje de memoria necesaria para la carga de trabajo actual en relación con la cantidad total de memoria (RAM + swap).

Para la obtención de los resultados se ejecutó el comando `sar -r 1 1308`, del cual se utilizó el campo `%memused` para el análisis de los resultados obtenidos, que se encuentran detallados en el **Anexo 1 – Sección A**.

3.6.3. Comando `sar -u`

El parámetro `-u` del comando `sar` especifica el monitoreo de CPU este es obligatorio ya que con este parámetro se está delimitando que el monitoreo sea solamente para CPU. En la **Tabla III. VIII**, se detalla cada uno de los campos desplegados al ejecutar el comando.

Tabla III. VIII. Detalle de la salida del comando `sar -u`

NOMBRE CAMPO	DESCRIPCIÓN
%user	Porcentaje de utilización de la CPU que se produjo durante la ejecución a nivel de usuario (aplicación).

%nice	El % de utilización de CPU mientras se ejecuta procesos con prioridad.
%system	Porcentaje de utilización de la CPU que se produjo durante la ejecución a nivel de sistema (kernel).
%iowait	Porcentaje de tiempo que el CPU o CPUs está inactivo en un proceso de requerimiento de E/S del disco duro
%idle	El % de CPU inactiva en espera de trabajos

Para la obtención de los resultados se ejecutó el comando `sar -u 1 1308`, del cual se utilizó el campo `%user` para el análisis de los resultados obtenidos, detallados en el **Anexo 1 – sección A**.

3.6.4. Comando `sar -d`

El parámetro `-d` especifica la utilización del monitoreo del disco este es obligatorio ya que con este parámetro se está delimitando que el monitoreo sea solamente para el disco duro. En la **Tabla III. IX**, se detalla la salida comando `sar -d`.

Tabla III. IX. Salida del comando `sar -d`

NOMBRE CAMPO	DESCRIPCIÓN
Device	Nombre del dispositivo de disco que se está supervisando
% busy	Parte del tiempo que el dispositivo estuvo ocupado atendiendo una solicitud de transferencia.
Avque	Número medio de solicitudes durante el tiempo que el dispositivo estuvo ocupado atendiendo una solicitud de transferencia.
r+w/s	El número de transferencias de lectura y escritura al dispositivo

	por segundo.
blks/s	Número de bloques de 512 bytes que se transfieren al dispositivo por segundo.
Avwait	Tiempo medio, en milisegundos, que las solicitudes de transferencia esperan de manera inactiva en la cola. Este tiempo se mide únicamente cuando la cola está ocupada.
%util	Porcentaje de utilización del disco duro

Para la obtención de los resultados se ejecutó el comando `sar -d 1 1308`, del cual se utilizó el campo `%util` para el análisis de los resultados obtenidos, detallados en el **Anexo 1 – Sección A**.

3.6.5. Comando time

El comando `time` sirve para medir de forma precisa el tiempo que tarda el servidor en responder solicitudes. En la **Tabla III. X**, se detalla la salida del comando `time`.

Tabla III. X. Salida del comando `time`

NOMBRE CAMPO	DESCRIPCIÓN
t real	El tiempo total transcurrido desde que ha invocado una solicitud al servidor de aplicaciones hasta que la solicitud es atendida en totalidad
t user	La cantidad de tiempo actualmente consumido en la CPU fuera del tiempo <code>sys</code>
t sys	La cantidad de tiempo consumido en el kernel, que es el tiempo empleado en contestar peticiones del sistema.

Se ejecutará el comando `time java -jar archivo.jar`, del cual se utilizó el campo `t real` para realizar los cálculos del tiempo de respuesta.

3.7. Análisis de los parámetros de comparación

A continuación se analiza cada uno de los parámetros considerados para el estudio del rendimiento de los servidores de aplicaciones: GlassFish, JOnAS y Gerónimo.

3.7.1. Uso de RAM

En la **Tabla III. XI**, se refleja las medias de los resultados obtenidos del parámetro uso de RAM al enviar diferentes cantidades de carga a los servidores de aplicación GlassFish, JOnAS y Gerónimo.

Tabla III. XI. Parámetro uso de RAM.

% Uso de RAM	GlassFish	JOnAS	Gerónimo
20000 solicitudes	57,24%	76,40%	96,6%
40000 solicitudes	57,76%	77,10%	97,83%
60000 solicitudes	58,56%	77,45%	97,14%

La **Figura III.15**, representa el comportamiento de los servidores de aplicación en cuanto al uso de RAM dependiendo de la cantidad de carga enviada, en el comportamiento de los servidores de aplicación GlassFish, JOnAS y Gerónimo se puede observar que mientras mayor sea el número de carga enviada mayor será el porcentaje de uso de RAM, prevaleciendo el comportamiento del servidor aplicaciones GlassFish por debajo del comportamiento de los servidores de

aplicación JOnAS y Gerónimo independientemente de la cantidad de carga enviada.

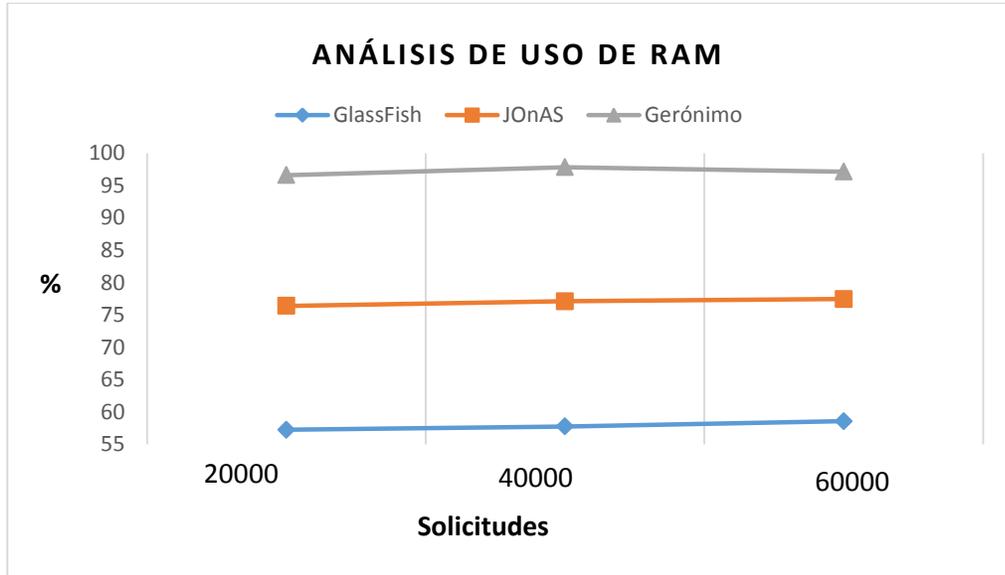


Figura III. 15. Análisis parámetro uso de RAM

3.7.2. Uso de CPU

En la **Tabla III. XII**, se refleja las medias de los resultados obtenidos del parámetro uso de CPU al enviar diferentes cantidades de carga a los servidores de aplicación GlassFish, JOnAS y Gerónimo.

Tabla III. XII. Parámetro uso de CPU

% Uso de CPU	GlassFish	JOnAS	Gerónimo
20000 solicitudes	8,25%	18,14%	32,98%
40000 solicitudes	13,61%	33,87%	33,15%
60000 solicitudes	20,10%	49,75%	36,62%

La **Figura III. 16**, representa el comportamiento de los servidores de aplicación en cuanto al uso de CPU dependiendo de la cantidad de carga enviada, en el comportamiento de los servidores de aplicación GlassFish, JOnAS y Gerónimo se puede observar que mientras mayor sea el número de carga enviada mayor será el porcentaje de uso de CPU. El servidor de aplicaciones JOnAS presenta una recta creciente, mientras mayor sea la cantidad de peticiones enviadas mayor será el porcentaje de CPU usado. El servidor de aplicaciones Gerónimo al enviar cantidades pequeñas de carga mayor es el porcentaje de uso de CPU y para cantidades más altas el porcentaje de uso de CPU tiende a estabilizarse. El servidor de aplicaciones GlassFish presente una recta creciente en cuanto al uso de CPU que se incrementa dependiendo de la cantidad de peticiones enviadas, prevaleciendo el comportamiento del uso de CPU por debajo del comportamiento del servidor de aplicaciones JOnAS y Gerónimo.

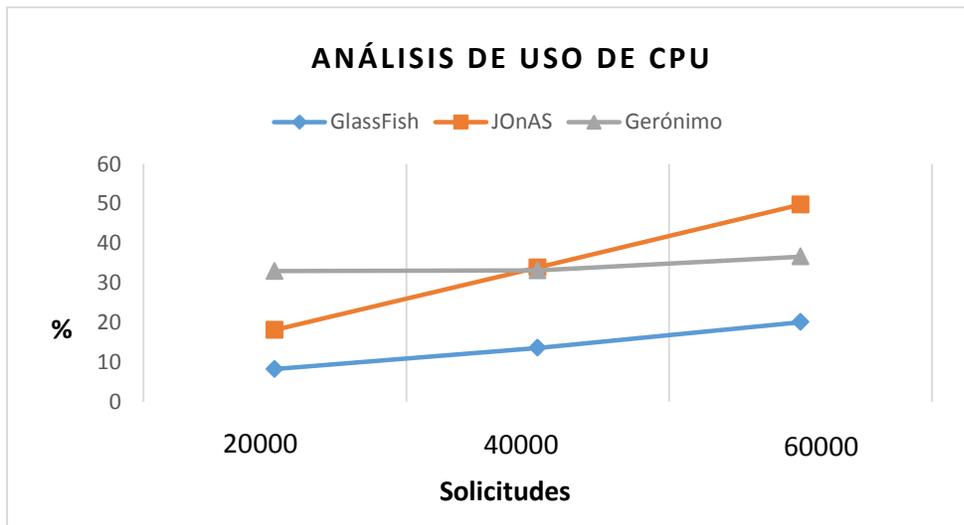


Figura III. 16. Análisis del parámetro uso de CPU

3.7.3. Uso de Disco

En la **Tabla III. XIII**, se refleja las medias de los resultados obtenidos del parámetro uso de disco al enviar diferentes cantidades de carga a los servidores de aplicación GlassFish, JOnAS y Gerónimo.

Tabla III. XIII. Parámetro uso de disco

% Uso de Disco	GlassFish	JOnAS	Gerónimo
20000 solicitudes	0,34%	0,09%	0,09%
40000 solicitudes	0,07%	0,08%	0,10%
60000 solicitudes	0,08%	0,08%	0,11%

La **Figura III. 17**, representa el comportamiento de los servidores de aplicación en cuanto al uso de Disco dependiendo de la cantidad de carga enviada, en el comportamiento del servidor de aplicación GlassFish y JOnAS se puede observar que al enviar menor cantidades de solicitudes, mayor es el consumo de disco y al enviar mayor cantidad de solicitudes el consumo disco es menor. El servidor de aplicaciones Gerónimo tiene un comportamiento normal, el consumo de disco va incrementando dependiendo de las cantidades de solicitudes enviadas.

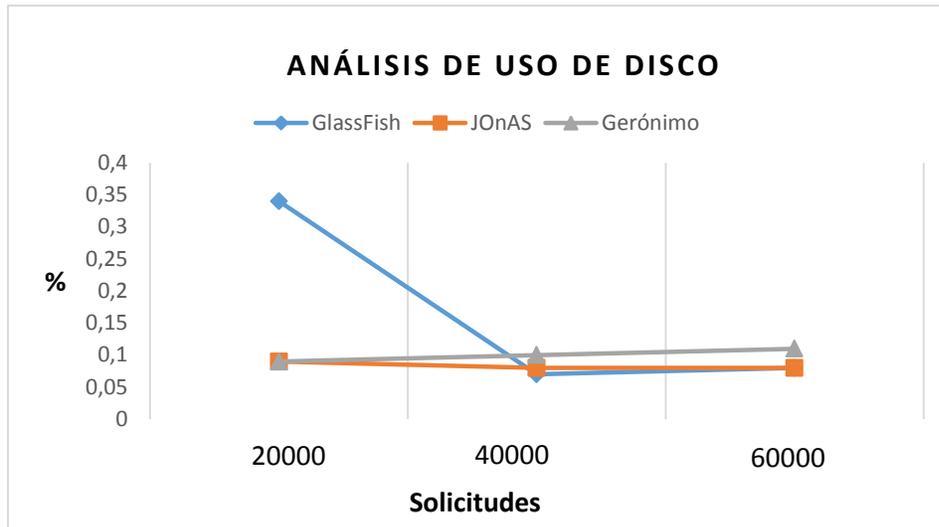


Figura III. 17. Análisis del parámetro uso de Disco

3.7.4. Tiempo de respuesta

En la **Tabla III. XIV**, se refleja las medias de los resultados obtenidos del parámetro tiempo de respuesta al enviar diferentes cantidades de carga con los servidores de aplicación GlassFish, JOnAS y Gerónimo.

Tabla III. XIV. Parámetro tiempo de respuesta

Tiempo de respuesta	GlassFish	JOnAS	Gerónimo
20000 solicitudes	482,68seg	732,97seg	885,54seg
40000 solicitudes	853,56seg	1377,42seg	1608,16seg
60000 solicitudes	1201,93seg	1854,89seg	2070,12seg

La **Figura III. 18**, representa el tiempo de respuesta que los servidores de aplicación tardan en responder las solicitudes enviadas, el tiempo que se tardan en responder las solicitudes los servidores de aplicación GlassFish, JOnAS y Gerónimo varía de

acuerdo a la cantidad de solicitudes enviada, mientras mayor sea la cantidad de solicitudes enviadas mayor será el tiempo que el servidor de aplicaciones se demore en atender todas las solicitudes. El servidor de aplicaciones GlassFish prevalece su comportamiento en cuanto al tiempo de respuesta por debajo del comportamiento de los servidores de aplicación JOnAS y Gerónimo.

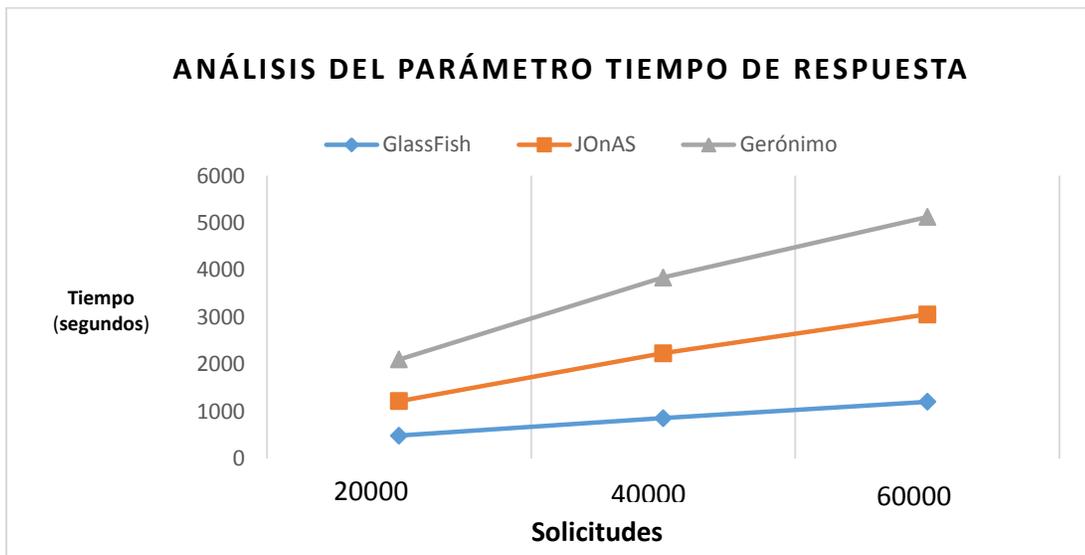


Figura III. 18. Análisis del parámetro tiempo de respuesta

3.8. Análisis de resultados y demostración de hipótesis

Para el análisis de los resultados y demostración de la hipótesis se analizó el comportamiento de los servidores de aplicación con una cantidad de 40000 solicitudes, considerando como una cantidad de solicitudes futuras que atenderá el servidor de aplicaciones que obtenga el mejor rendimiento tras el análisis realizado.

Para determinar si existe diferencia significativa entre los servidores de aplicación en estudio y aceptar o rechazar la hipótesis nula, se utilizó el coeficiente de Pearson donde indica que si el valor t calculado con el coeficiente de Pearson es menor igual que valor t de la tabla de Pearson de acuerdo al grado de libertad y el margen de error que se encuentran en el **Anexo1 - Sección B** entonces “se acepta la hipótesis nula” (26), caso contrario si el valor de t calculado es mayor que el valor t de la tabla de Pearson se rechaza la hipótesis nula.

3.8.1. Uso de RAM

Para determinar el servidor de aplicaciones con mejor rendimiento en el uso de RAM se plantea la hipótesis nula (H_0) y alternativa (H_1) como se detalla a continuación.

H_0 : El uso de RAM del servidor de aplicaciones GlassFish es igual al uso de RAM del servidor de aplicaciones JOnAS y Gerónimo.

$$H_0: \mu_1 = \mu_2 = \mu_3$$

H_1 : El uso de RAM del servidor de aplicaciones GlassFish es diferente al uso de RAM del servidor de aplicaciones JOnAS y Gerónimo.

$$H_1: \mu_1 \neq \mu_2 \neq \mu_3$$

En la **Tabla III. XV**, se detalla la estadística descriptiva del uso de RAM de los servidores de aplicación en estudio, donde se puede observar que el servidor de aplicaciones con mejor media es el servidor de aplicaciones GlassFish con un valor de 57,76. Debido a que en el punto 3.4 se definió que mientras menor sea el

porcentaje obtenido de cada uno de los parámetros mejor será el rendimiento del servidor de aplicaciones, de igual forma el valor de la varianza del servidor GlassFish y JOnAS es el que más tiende a 0, es decir más concentrados están los valores de la serie alrededor de la media, además podemos exponer que los datos recolectados no se encuentran muy dispersos uno de otro de acuerdo al valor de la desviación típica que no es muy alto, por otro lado tenemos al servidor de aplicaciones Gerónimo obtuvo una de las medias más alta en el uso de RAM y los datos recolectados se encuentran más dispersos uno de otro de acuerdo al valor de la varianza y desviación típica que tienden alejarse de 0.

Tabla III. XV. Estadística descriptiva– Uso de RAM

	N	Mínimo	Máximo	Media	Desv. típ.	Varianza
GlassFish	1308	57,25	59,04	57,76	0,22	0,051
JOnAS	1308	76,26	78,39	77,10	0,28	0,079
Gerónimo	1308	96,62	99,08	97,83	0,56	0,320

Utilizando los datos de la estadística descriptiva y el coeficiente de Pearson representados en la **Tabla III. XVI**, donde se observa la correlación entre cada pareja de variables y para aceptar o rechazar la hipótesis nula o alternativa se calculó el valor de t y se compara con el valor existente de la tabla de Pearson en este caso es de 1,96 de acuerdo al **Anexo 1 - Sección B**, para un margen de error α (0,03) y N-2 (1306) grados de libertad.

Tabla III. XVI. Estadística inferencial – Uso de RAM

		GlassFish	JOnAS	Gerónimo
--	--	------------------	--------------	-----------------

GlassFish	Correlación de Pearson	1	0,230	0,637
	T. (bilateral)	-	8,54	29,86
	Diferencia Significativa	-	0,061	0,02
	N	1308	1308	1308
JOnAS	Correlación de Pearson	0,230	1	-0,098
	T. (bilateral)	8,54	-	3,56
	Diferencia Significativa	0,061	-	0,025
	N	1308	1308	1308
Gerónimo	Correlación de Pearson	0,637	-0,098	1
	T (bilateral)	29,86	3,56	-
	Diferencia Significativa	0,02	0,025	-
	N	1308	1308	1308

El valor de t calculado entre GlassFish y JOnAS es de 8,54. Entre GlassFish y Gerónimo es de 29,86 y entre JOnAS y Gerónimo es de 3,56. De los cuales los valores de t calculados son mayores al valor t de la tabla de Pearson que es de 1,96. Entonces se rechaza la hipótesis nula con un riesgo de 0.03% y se acepta la hipótesis alternativa concluyendo que el uso de RAM del servidor de aplicaciones GlassFish es diferente al uso de RAM del servidor de aplicaciones JOnAS y Gerónimo.

Una vez analizado los datos con la estadística descriptiva e inferencial y al observar que existe diferencia significativa entre los valores del uso de RAM de los servidores de aplicación en estudio, se procede a realizar la ponderación de

acuerdo al porcentaje asignado como se detalla en la **Tabla III. XVII**. Para lo cual se aplicó la regla de tres invertida detallada en el **Anexo 1 - Sección B**.

Tabla III. XVII. Ponderación - Uso de RAM

	GlassFish	JOnAS	Gerónimo
Media	57,76%	77,10%	97,83%
Ponderación	20%	14,98%	11,81%

3.8.2. Uso de CPU

Para determinar el servidor de aplicaciones con mejor rendimiento en el uso de CPU se plantea la hipótesis nula (**H₀**) y alternativa (**H₁**) como se detalla a continuación.

H₀: El uso de CPU del servidor de aplicaciones GlassFish es igual al uso de CPU del servidor de aplicaciones JOnAS y Gerónimo.

$$H_0: \mu_1 = \mu_2 = \mu_3$$

H₁: El uso de CPU del servidor de aplicaciones GlassFish es diferente al uso de CPU del servidor de aplicaciones JOnAS y Gerónimo.

$$H_1: \mu_1 \neq \mu_2 \neq \mu_3$$

La **Tabla III. XVIII**, se detalla la estadística descriptiva del uso de CPU de los servidores de aplicación en estudio, donde se puede observar que el servidor de aplicaciones con mejor media es el servidor de aplicaciones GlassFish con un valor de 13,62. Debido a que en el punto 3.4 se definió que mientras menor sea el porcentaje obtenido de cada uno de los parámetros mejor será el rendimiento del servidor de aplicaciones, sin embargo el valor de la varianza y desviación típica

tiende alejarse de 0, es decir los valores recolectados se encuentran dispersos al igual que el servidor de aplicaciones JOnAS, cabe mencionar que Gerónimo no tiene tanta dispersión de los datos recolectados en relación a GlassFish y JOnAS.

Tabla III. XVIII. Estadística descriptiva - Uso de CPU

	N	Mínimo	Máximo	Media	Desv. típ.	Varianza
GlassFish	1308	0,00	65,00	13,62	19,39	376,00
JOnAS	1308	0,00	86,87	33,87	26,88	723,04
Geronimo	1308	3,96	63,00	33,15	9,65	93,29

Utilizando los datos de la estadística descriptiva y el coeficiente de Pearson representados en la **Tabla III. XIX**, se observa la correlación entre cada pareja de variables y para aceptar o rechazar la hipótesis nula o alternativa, se calculó el valor de t y se compara con el valor t de la tabla de Pearson en este caso 1,96 de acuerdo al **Anexo 1 - Sección B**, para un margen de error α (0,03) y N-2 (1306) grados de libertad.

Tabla III. XIX. Estadística inferencial - Uso de CPU

		GlassFish	JOnAS	Gerónimo
GlassFish	Correlación de Pearson	1	0,465	0,542
	T. (bilateral)	-	18,98	23,30
	Diferencia Significativa	-	0,035	0,028
	N	1308	1308	1308
JOnAS	Correlación de Pearson	0,465	1	0,297
	T. (bilateral)	18,98	-	1,07
	Diferencia Significativa	0,035	-	0

	N	1308	1308	1308
Gerónimo	Correlación de Pearson	0,542	0,297	1
	T. (bilateral)	23,30	1,07	-
	Diferencia Significativa	0,02	0	-
	N	1308	1308	1308

El valor de t calculado entre GlassFish y JOnAS es de 18,98. Entre GlassFish y Gerónimo es de 23,30. Esto implica que los valores son mayores al valor t de la tabla de Pearson que es de 1,96 mientras el valor de t calculado entre JOnAS y Gerónimo es de 1,07 es menor al valor t de la tabla de Pearson que es de 1,96. En este caso de los tres servidores evaluados por lo menos uno de ellos es diferente uno de otro. Entonces se rechaza la hipótesis nula con un riesgo de 0.03% y se acepta la hipótesis alternativa concluyendo que el uso de CPU del servidor de aplicaciones GlassFish es diferente al uso de CPU del servidor de aplicaciones JOnAS y Gerónimo.

Una vez analizado los datos con la estadística descriptiva e inferencial y al observar que existe diferencia significativa entre los valores del uso de CPU del servidor Glassfish - JOnAS y GlassFish - Geronimo, pero entre los servidores JOnAS y Gerónimo no existe diferencia significativa teniendo en cuenta este aspecto se procede a realizar la ponderación de acuerdo al porcentaje asignado como se detalla en la **Tabla III. XX**. Para lo cual se aplicó la regla de tres invertida detallada en el **Anexo 1 - Sección B**.

Tabla III. XX. Ponderación - Uso de CPU

	GlassFish	JOnAS	Gerónimo
Media	13,62%	33,87%	33,15%
Ponderación	25%	10,05%	10,05%

3.8.3. Uso de Disco

Para determinar el servidor de aplicaciones con mejor rendimiento en el uso de disco se plantea la hipótesis nula (H_0) y alternativa (H_1) como se detalla a continuación.

H_0 : El uso de disco del servidor de aplicaciones GlassFish es igual al uso de disco del servidor de aplicaciones JOnAS y Gerónimo.

$$H_0: \mu_1 = \mu_2 = \mu_3$$

H_1 : El uso de disco del servidor de aplicaciones GlassFish es diferente al uso de disco del servidor de aplicaciones JOnAS y Gerónimo.

$$H_a: \mu_1 \neq \mu_2 \neq \mu_3$$

En la **Tabla III. XXI**, se detalla la estadística descriptiva del uso de disco de los servidores de aplicación en estudio, donde se puede observar que el servidor de aplicaciones con mejor media es el servidor de aplicaciones GlassFish con un valor de 0,0710. Dado que en el punto 3.4 se definió que mientras menor sea el porcentaje obtenido de cada uno de los parámetros mejor será el rendimiento del servidor de aplicaciones, de igual forma el valor de la varianza es el que más tiende

a 0, es decir más concentrados están los valores recolectados, además se puede exponer que los datos no se encuentran dispersos uno de otro de acuerdo al valor de la desviación típica que no es muy alto. El análisis de la varianza y desviación típica es igual para el servidor de aplicaciones JOnAS y Gerónimo.

Tabla III.XXI. Estadística descriptiva - Uso de Disco

	N	Mínimo	Máximo	Media	Desv. típ.	Varianza
GlassFish	1308	0,00	1,36	0,0710	0,15213	0,023
JOnAS	1308	0,00	2,35	0,0828	0,20033	0,040
Gerónimo	1308	0,00	8,33	0,1002	0,31495	0,099

Utilizando los datos de la estadística descriptiva y el coeficiente de Pearson representados en la **Tabla III. XXII**, donde se observa la correlación entre cada pareja de variables y para aceptar o rechazar la hipótesis nula o alternativa se calcula el valor de t y se compara con el valor t obtenido de la tabla de Pearson en este caso es de 1,96 de acuerdo al **Anexo 1 - Sección B**, para un margen de error de α (0,03) y N-2 (1306) grados de libertad.

Tabla III. XXII. Estadística inferencial - Uso de Disco

		GlassFish	JOnAS	Geronimo
GlassFish	Correlación de Pearson	1	0,031	0,011
	T. (bilateral)	-	1,12	0,39
	Diferencia Significativa	-	0	0
	N	1308	1308	1308
JOnAS	Correlación de Pearson	0,031	1	0,025
	T. (bilateral)	1,12	-	0,903

	Diferencia Significativa	0	-	0
	N	1308	1308	1308
Gerónimo	Correlación de Pearson	0,011	0,025	1
	T. (bilateral)	0,39	0,903	-
	Diferencia Significativa	0	0	-
	N	1308	1308	13089

El valor de t calculado entre GlassFish y JOnAS es de 1,12. Entre GlassFish y Gerónimo el valor de t calculado es de 0,39 y entre JOnAS y Gerónimo es de 0,90. De los cuales los valores de t calculados son menores al valor t de la tabla de Pearson que es de 1,96. Entonces se acepta la hipótesis nula con un riesgo de 0.03% y se rechaza la hipótesis alternativa concluyendo que: el uso de disco del servidor de aplicaciones GlassFish es igual al uso de disco del servidor de aplicaciones JOnAS y Gerónimo.

Una vez analizado los datos con la estadística descriptiva e inferencial y al observar que no existe diferencia significativa entre los valores del uso de disco de los servidores de aplicación en estudio se procede a realizar la ponderación de acuerdo al porcentaje asignado como se detalla en la **Tabla III. XXIII**. En este caso los tres servidores de aplicación obtienen la misma ponderación al no existir diferencia significativa.

Tabla III. XXIII. Ponderación - Uso de Disco

	GlassFish	JOnAS	Gerónimo
Media	0,07%	0,08%	0,10%

Ponderación	10%	10%	10%
--------------------	-----	-----	-----

3.8.4. Tiempo de respuesta

Para determinar el servidor de aplicaciones con mejor tiempo de respuesta se plantea la hipótesis nula (H_0) y alternativa (H_1) como se detalla a continuación.

H_0 : El tiempo de respuesta del servidor de aplicaciones GlassFish es igual al tiempo de respuesta del servidor de aplicaciones JOnAS y Gerónimo.

$$H_0: \mu_1 = \mu_2 = \mu_3$$

H_1 : El tiempo de respuesta del servidor de aplicaciones GlassFish es diferente al tiempo de respuesta del servidor de aplicaciones JOnAS y Gerónimo.

$$H_a: \mu_1 \neq \mu_2 \neq \mu_3$$

En la **Tabla III. XXIV**, se detalla la estadística descriptiva del tiempo de respuesta de los servidores de aplicación, donde se puede observar que el servidor de aplicaciones con mejor media es el servidor de aplicaciones GlassFish con un valor de 853,56 del conjunto de datos, dado que en el punto 3.4 se definió que mientras menor sea el porcentaje obtenido de cada uno de los parámetros mejor será el rendimiento del servidor de aplicaciones, de igual forma el valor de la varianza es el que más tiende a 0, es decir más concentrados estas los valores estudiados alrededor de la media, además se puede exponer que los datos no se encuentran muy dispersos uno de otro de acuerdo al valor de la desviación típica que no es muy alto al igual que JOnAS y Gerónimo que sus valores de varianza y desviación típica tienden a 0.

Tabla III. XXIV. Estadística descriptiva - Tiempo de Respuesta

	N	Mínimo	Máximo	Media	Desv. típ.	Varianza
GlassFish	1308	830,25	957,36	853,56	0,26	0,068
JOnAS	1308	1223,21	1452,16	1377,42	0,29	0,084
Gerónimo	1308	1564,56	1705,12	1608,16	0,37	0,134

Utilizando los datos de la estadística descriptiva y el coeficiente de Pearson representados en la **Tabla III. XXV**, donde se observa la correlación entre cada pareja de variables y para aceptar o rechazar la hipótesis nula o alternativa se calculó el valor de t y se compara el valor t obtenido de la tabla de Pearson en este caso es de 1,96 de acuerdo al **Anexo 1 - Sección B**, para un margen de error α (0,03) y N-2 (1306) grados de libertad.

Tabla III. XXV. Estadística inferencial - Tiempo de Respuesta

		GlassFish	JOnAS	Gerónimo
GlassFish	Correlación de Pearson	1	0,4	0,58
	T. (bilateral)	-	15,77	25,73
	Diferencia Significativa	-	0,02	0,035
	N	1308	1308	1308
JOnAS	Correlación de Pearson	0,4	1	0,18
	T. (bilateral)	15,77	-	6,61
	Diferencia Significativa	0,042	-	0,02
	N	1308	1308	1308
Gerónimo	Correlación de Pearson	0,58	0,18	1
	T. (bilateral)	25,73	6,61	-
	Diferencia Significativa	0,58	0,58	-
	N	1308	1308	1308

El valor de t calculado entre GlassFish y JOnAS es de 15,77. Entre GlassFish y Gerónimo es de 25,73 y entre JOnAS y Gerónimo es de 6,61. De los cuales los valores de t calculados son mayores al valor t de la tabla de Pearson que es de 1,96. Entonces se rechaza la hipótesis nula con un riesgo (máximo) de 0.03% y se acepta la hipótesis alternativa concluyendo que el tiempo de respuesta del servidor de aplicaciones GlassFish es diferente al tiempo de respuesta del servidor de aplicaciones JOnAS y Gerónimo.

Una vez analizado los datos con la estadística descriptiva e inferencial y al observar que existe diferencia significativa entre los valores de tiempo de respuesta de los servidores de aplicación en estudio se procede a realizar la ponderación de acuerdo al porcentaje asignado como se detalla en la **Tabla III. XXVI**. Para lo cual se aplicó la regla de tres invertida detallada en el **Anexo 1 - Sección B**.

Tabla III. XXVI. Ponderación - Tiempo de Respuesta

	GlassFish	JOnAS	Gerónimo
Media	853,56	1377,42	1608,16
Ponderación	45%	27,89%	23,88%

3.9. Resultado del análisis y demostración de la hipótesis

En la **Tabla III. XXVII**, se puede observar el resultado obtenido, luego de realizar el análisis y ponderación de cada uno de los parámetros de comparación.

Tabla III. XXVII. Ponderación de los parámetros evaluados

	GlassFish	JOnAS	Gerónimo
Uso de RAM (20%)	20	14,98	11,81
Uso de CPU (25%)	25	10,05	10,05
Uso de Disco (10%)	10	10	10
Tiempo de respuesta (45%)	45	27,89	23,88
Total	100%	62,92%	55,74%

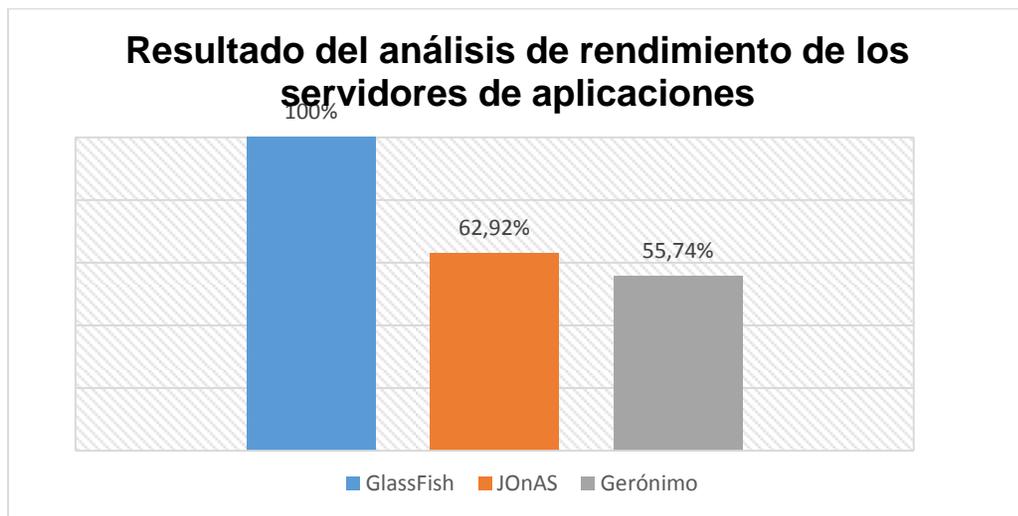


Figura III. 19. Resultado final del análisis

En la **Figura III. 19**, se observa claramente que el servidor de aplicaciones GlassFish supera con un porcentaje de 37,08% y 44,26% sobre JOnAS y Gerónimo respectivamente además utilizando la estadística descriptiva e inferencial se puede concluir que el servidor de aplicación JOnAS no obtuvo un mejor rendimiento en relación a GlassFish Server y Gerónimo al publicar servicios web, por lo tanto el servidor de aplicaciones con mejor

rendimiento es GlassFish, el mismo que ha sido utilizado para la publicación de la aplicación y servicio web de la unidad educativa.

CAPÍTULO IV

4. Implementación del sistema de integración y administración de aplicaciones

En este capítulo se muestra todo el proceso de desarrollo del sistema tomando como guía la metodología ágil de programación extrema (eXtreme Programming o XP), esta metodología se enfoca en el uso constante del dialogo entre los actores del proyecto (programadores, cliente). El proyecto consiste en el desarrollo de un sistema de integración y administración de aplicaciones.

4.1. Descripción del sistema

La aplicación web tiene como objetivo fundamental la integración del sistema de calificaciones para el docente (SCD) y el entorno virtual de aprendizaje, por medio del consumo de datos a través de servicios web.

El módulo de integración de sistemas del colegio Chiriboga, consumirá por medio de un servicio web los datos que se encuentren en el sistema de calificaciones para el docente,

esta información será migrada al entorno virtual de aprendizaje respetando la integridad referencial que tiene la base de datos de Moodle, con lo que se optimizará tiempos en la creación de la estructura académica (cursos, paralelos, materias, estudiantes, profesores) en Moodle.

Tras realizar evaluaciones a diferentes alternativas de lenguajes de programación, la aplicación se desarrolló bajo lo siguiente:

Lenguaje de programación java, framework jsf 2.0, componente primefaces 4.5 y como ide de desarrollo netbeans 7.3 por la sencillez que provee este lenguaje para el trabajo y debido a que los integrantes del grupo tienen experiencia con estas herramientas, de esta manera se minimizó los tiempos de desarrollo de la aplicación. Gracias a su alta gama de beneficios (multiplataforma, eficiente manejo de concurrencia, capacidad de almacenamiento, etc.) el motor de base de datos utilizado es postgresQL 9.1.

La **Figura IV. 20** se detalla la solución para integrar las aplicaciones dentro de la red local del colegio, donde podemos observar que se desplegará un servicio web que consumirá la información desde la base de datos Oracle del sistema académico y estará publicado en un servidor Glassfish, este servicio web podrá ser utilizado por cualquier sistema que requiera información de estudiantes, docentes, materias, cursos, etc, en el mismo servidor se implantará una aplicación web que permitirá al administrador utilizar el servicio web con la finalidad de construir la estructura académica en Moodle, a su vez Moodle esta publicado en un servidor apache tras un firewall permitiendo que estudiantes y docentes

utilicen el ambiente desde cualquier parte del mundo, esta aplicación está orientada para que docentes y estudiantes tengan un ambiente colaborativo de enseñanza y aprendizaje respectivamente.

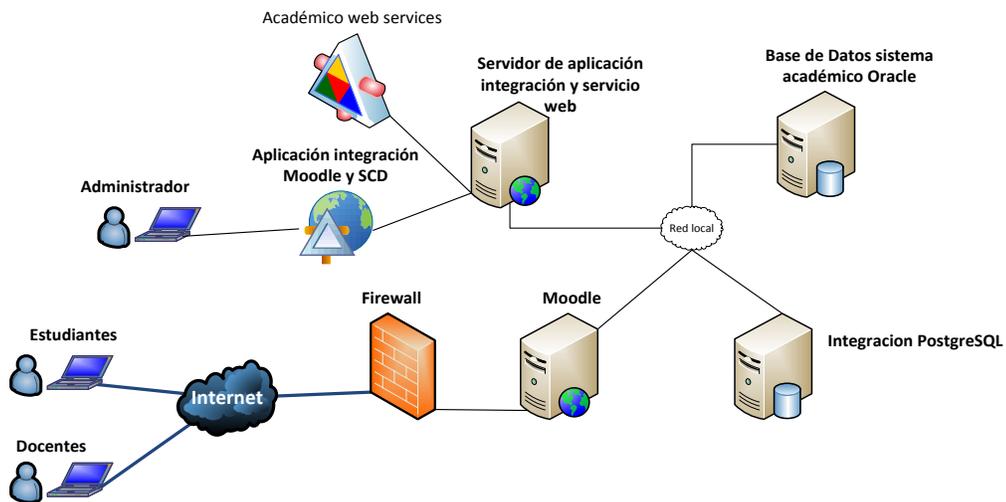


Figura IV. 20. Solución

La aplicación está desarrollada básicamente para que el administrador tenga una interfaz que facilite el trabajo al inicio de cada año lectivo en la integración de información desde sistema académico hacia Moodle, adicional a esto el administrador contará con una interfaz que le permitirá gestionar los permisos y accesos que tendrán los usuarios a la aplicación.

4.2. Historias de usuario

La planificación de requerimientos realizadas al inicio del proyecto fueron sencillas, tras estudiar y mantener conversaciones constantes con el cliente se realizaron depuraciones y posteriormente el requerimiento fue cambiando a lo largo del proyecto, algunas de estas

historias fueron eliminadas o modificadas en el transcurso del mismo a medida que cambiaban los requisitos del cliente o se tenía una concepción más clara del sistema.

Las historias de usuario fueron redactadas en cada reunión en conversaciones con el cliente, la terminología de estas historias eran realizadas a partir de la visión del usuario y no del desarrollador, ya que una vez redactadas estas historias el cliente tenía una idea clara de lo que se proponía realizar a partir de su visión.

4.3. Estimación de tiempo de historias usuario

En reuniones con el cliente se estableció la forma de entrega de cada requerimiento, se definió las fechas entre cada entrega, quedando de la siguiente forma:

Para la entrega de cada iteración al cliente se consideró el lapso de tiempo de una semana laborable de 5 días, trabajando 8 horas diarias. Para la aplicación de la metodología este tiempo se transformó en puntos de trabajo, cada punto equivale a 2 horas, es así que el día de trabajo era igual a 4 puntos y la semana laborable a 20 puntos, entonces al cliente se le entregaría un avance del proyecto por cada 20 puntos de trabajo.

La estimación del tiempo de trabajo que tomaría la realización de cada historia de usuario se definió en las reuniones del equipo de desarrollo con el cliente, estas historias se priorizaron y se realizó una proyección estimada para el cumplimiento de cada historia de usuario.

Para el desarrollo del proyecto se definieron nomenclaturas detalladas a continuación: la letra M equivale a metáfora, la letra H a historia de usuario y para identificar las tareas de

ingeniería se utilizó las letras del abecedario excluyendo la letra M y H, todo esto combinado por un numero de acuerdo al orden a desarrollarse, ejemplo A-1.

La **Tabla IV. XXVIII** contiene el detalle de las historias de usuarios, metáforas, puntos estimados y puntos reales que se definieron para el desarrollo del sistema.

Tabla IV. XXVIII. Historias de usuario y Metáforas

Iteración	Historia de Usuario - Metáfora (Número y Nombre)		Puntos Estimados	Puntos Reales
1	M1	Diseño de esquema de base de datos para control de sistemas	20	25
2	M2	Análisis de la base de datos del entorno de aprendizaje Moodle	12	12
	M3	Análisis de la base de datos del sistema académico del colegio	8	8
3	M4	Diseño de esquema de base de datos para Integración de sistemas	12	14
	M5	Definición de arquitectura de hardware y software	4	4
	M6	Definición de interfaces de usuario y estándar de codificación	4	4
4	M7	Creación de servicio web para publicar información desde la base de datos del sistema académico de la institución.	8	8
	H1	Autenticación de Usuario	8	8
	H2	Recuperación de Credenciales de usuario	4	4
5	H3	Gestión de Usuarios	8	10
	H4	Gestión de Grupos	12	14
6	H5	Gestión de Aplicaciones	10	12

	H6	Gestión de Semillas para aplicación	2	4
	H7	Gestión de Módulos	8	10
7	H8	Gestión de Recursos	8	10
	H9	Gestión de Cargos	4	6
	H10	Gestión de Áreas	8	10
8	H11	Gestión de Permisos	4	4
	H12	Diagrama Diurna	16	20
9	H13	Integración de datos de estudiantes a Moodle	10	10
	H14	Integración de datos de docentes a Moodle	10	10
10	H15	Integración de cursos, paralelos, Materias a Moodle	20	16
11	H16	Matricula de estudiantes en Moodle	10	10
	H17	Asignación de materias a profesores en Moodle	10	10
12	H18	Gestión de reportes	4	4
	M8	Implementación dela aplicación	4	4
	M9	Capacitación a usuarios	2	2

El proyecto tuvo un desfase de 23 puntos de trabajo, ya que en el inicio del proyecto el cliente solicito cambios en funcionalidades, esto conllevó a realizar modificaciones en métodos y flujos que ya estaban creados con anterioridad. En la **Figura IV. 21** se observa de mejor manera las historias de usuario que retrasaron el proyecto.

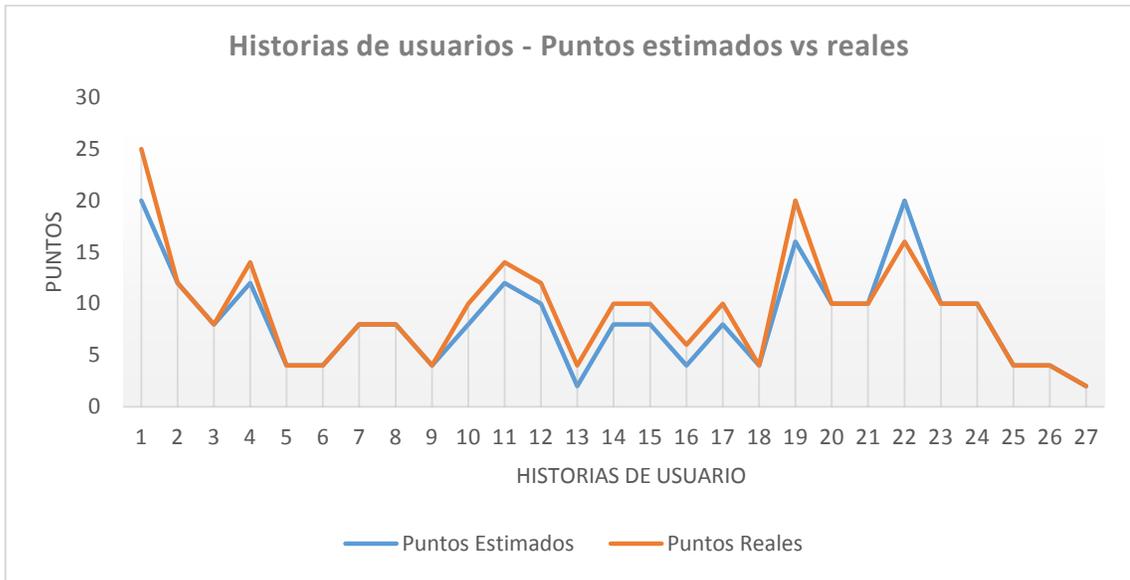


Figura IV. 21. Puntos estimados vs Puntos reales

4.4. Plan de entregas de cada iteración al cliente

Para el desarrollo de la aplicación se realizaron 9 metáforas y 18 historias de usuario, las cuales están distribuidas en 12 iteraciones, las fechas estimadas y fechas reales de entregas al cliente se definen en la **Tabla IV. XXIX**, iniciando el proyecto el 10 de marzo de 2014 y finalizando el 5 de junio de 2014, teniendo un desfase total máximo de 1 semana y 1 día en la entrega.

La duración estimada de cada entregable será máximo de una semana esto se definió en la estimación de tiempo.

Tabla IV. XXIX. Plan de entregables

Iteración	Estimación		Real		Historias de usuario	
	Fecha inicio	Fecha finalización	Fecha inicio	Fecha finalización		
1	10/03/2014	14/03/2014	10/03/2014	18/03/2014	M1	Diseño de esquema de base de datos para control de sistemas
2	17/03/2014	21/03/2014	18/03/2014	25/03/2014	M2	Análisis de la base de datos del entorno de aprendizaje Moodle
					M3	Análisis de la base de datos del sistema académico del colegio
3	24/03/2014	28/03/2014	25/03/2014	01/04/2014	M4	Diseño de esquema de base de datos para Integración de sistemas
					M5	Definición de arquitectura de hardware y software
					M6	Definición de interfaces de usuario y estándar de codificación
4	31/03/2014	04/04/2014	01/04/2014	08/04/2014	M7	Creación de servicio web para publicar información desde la base de datos del sistema académico de la institución.
					H1	Autenticación de Usuario
					H2	Recuperación de Credenciales de usuario
5	07/04/2014	11/04/2014	08/04/2014	16/04/2014	H3	Gestión de Usuarios

					H4	Gestión de Grupos
6	14/04/2014	18/04/2014	16/04/2014	25/04/2014	H5	Gestión de Aplicaciones
					H6	Gestión de Semillas para aplicación
					H7	Gestión de Módulos
7	21/04/2014	25/04/2014	25/04/2014	05/05/2014	H8	Gestión de Recursos
					H9	Gestión de Cargos
					H10	Gestión de Áreas
8	28/04/2014	02/05/2014	05/05/2014	13/05/2014	H11	Gestión de Permisos
					H12	Diagrama Diurna
9	05/05/2014	09/05/2014	13/05/2014	20/05/2014	H13	Integración de datos de estudiantes a Moodle
					H14	Integración de datos de docentes a Moodle
10	12/05/2014	16/05/2014	20/05/2014	26/05/2014	H15	Integración de cursos, paralelos, Materias a Moodle
11	19/05/2014	23/05/2014	26/05/2014	02/06/2014	H16	Matricula de estudiantes en Moodle
					H17	Asignación de materias a profesores en Moodle
12	26/05/2014	28/05/2014	02/06/2014	05/06/2014	H18	Gestión de reportes
					M8	Implementación dela aplicación
					M9	Capacitación a usuarios

4.5. Formato de historia usuario, tareas de ingeniería y pruebas de aceptación.

En la **Figura IV.22**, se observa el formato para las historias usuario desarrolladas en el presente proyecto.

Historia de usuario	
Número:	Nombre historia:
Usuario:	Iteración Asignada:
Prioridad en Negocio: (Alta/Media/Baja)	Puntos Estimados:
Riesgo en Desarrollo: (Alta/Media/Baja)	Puntos Reales:
Descripción:	
Observación:	

Figura IV. 22. Formato historia de usuario

En la **Tabla IV. XXX** se detalla la descripción del formato de historia de usuario

Tabla IV. XXX. Descripción de historia de usuario

Formato	Descripción
Número	Corresponde al número de historia de usuario a redactar.
Usuario	Es la persona que redacta la historia de usuario.
Iteración asignada	Corresponde al número de iteración en el cual se va a desarrollar la historia o requerimiento.
Prioridad en negocio	Es la prioridad que establece el cliente en el desarrollo de la historia.
Puntos estimados	Es la representación del tiempo que va tomar realizar una historia, el mismo que debe ser planificado antes del desarrollo de la historia de usuario.
Riesgo en desarrollo	Es el riesgo que corre el programador al no cumplir con la historia de usuario.
Puntos reales	Es la representación del tiempo que realmente tomo desarrollar la historia de usuario

Descripción	Es la narración del requerimiento que solicita el cliente, tomando el siguiente formato: Como (Rol) requiero (funcionalidad) para (beneficio)
Observaciones	Es la justificación que determina el cliente para realizar una determinada funcionalidad

Tareas de ingeniería

El formato establecido para el desarrollo de tareas de ingeniería se muestra en la **Figura**

IV.23.

Tarea de Ingeniería	
Número Tarea:	Historia de Usuario:
Nombre Tarea:	
Tipo Tarea: (Creación, Modificación, Mejora, Otra.)	Puntos Estimados:
Fecha Inicio:	Fecha Fin:
Analista Responsable:	
Descripción:	

Figura IV. 23. Formato tarea de ingeniería

En la **Tabla IV. XXXI** se detalla la descripción del formato de tarea de ingeniería.

Tabla IV. XXXI. Descripción de tarea de ingeniería

Formato	Descripción
Número Tarea	Corresponde a la combinación entre una letra y un número, representando el código de la tarea a desarrollar
Historia de Usuario	Corresponde al nombre de la historia a la que pertenece la tarea, una historia puede tener de 1 a varias tareas
Nombre Tarea	Es el nombre que se le da a la tarea a desarrollar
Tipo Tarea	Es la definición que se le da a la tarea, ejemplo: creación, modificación, mejora, otra

Puntos estimados	Es la representación de tiempo que se le da una tarea asignada a un programador
Fecha inicio	Es el fecha inicial que se le establece al programador para que cumpla con la tarea
Fecha fin	Es el fecha final que se le establece al programador para el cumplimiento de la tarea
Analista responsable	En esta parte se define el nombre del programador encargado del desarrollo de la tarea
Descripción	Es el detalle de lo que se debe hacer por parte del programador

Pruebas de aceptación

El formato establecido para el desarrollo de pruebas de aceptación se muestra en la

Figura IV.24.

Caso de Prueba de Aceptación	
Número caso de Prueba:	Historia de Usuario:
Nombre:	
Descripción:	
Condiciones de Ejecución:	
Entrada / Pasos de Ejecución:	
Resultados esperados 1:	
Resultados esperados 2:	
Resultados esperados 3:	

Figura IV. 24. Formato prueba de aceptación

En la **Tabla IV. XXXII** se detalla la descripción del formato de prueba de aceptación.

Tabla IV. XXXII. Descripción de pruebas de aceptación

Formato	Descripción
Número de caso de prueba	Es el número que le corresponde a la prueba
Historia de usuario	Corresponde al nombre de la historia a la que pertenece la prueba

Nombre	Es el alias que se le da a la prueba de aceptación
Descripción	Es el detalle de lo que se desea probar
Condiciones de ejecución	Es el escenario en el cual se deben realizar las pruebas
Pasos de ejecución	Es detalle de cada paso a seguir para la prueba de funcionalidad
Resultados esperados	Es detalle de los resultados esperados al realizar la prueba

4.6. Desarrollo de historias de usuario y metáforas

Las historias de usuario fueron redactadas con un lenguaje adecuado y se utilizaron frases propias del cliente, en el siguiente apartado se detallan los resultados de las historias de usuarios desarrolladas para la creación del sistema de integración y servicio web, la planificación completa de las iteraciones, historias de usuario, cronograma, tareas de ingeniería, y prueba de aceptación se encuentran detallados en el **Anexo 1 - Sección C**.

4.6.1. Iteración 1

En esta iteración se desarrolló la metáfora de diseño de base de datos para el control de sistemas, esta metáfora constituye un pilar fundamental para las historias de usuario que se realizaron en el transcurso del proyecto.

Para el cumplimiento de la metáfora se contempló un total de 5 tareas de ingeniería las cuales están definidas en un cronograma que el programador debe cumplir. Las tareas de ingeniería contemplaron lo siguiente: generación del diagrama DER de base de datos, normalización de tablas, diagrama lógico, diagrama físico, proyección de crecimiento de la

base de datos, script de despliegue y diccionario de datos, para un mayor detalle de lo establecido en esta metáfora seguir el **Anexo 1 - Sección C**.

Diagrama entidad relación (DER)

El desarrollo del diagrama entidad relación permitió representar las entidades, propiedades y relaciones relevantes del cual está conformado el sistema de información.

La **Figura IV. 25** detalla el diagrama entidad relación desarrollado para cumplir con los requerimientos establecidos por el cliente, este diagrama contiene 11 entidades representadas por rectángulos, 10 relaciones representados por rombos y las cardinalidades entre las relaciones y entidades.

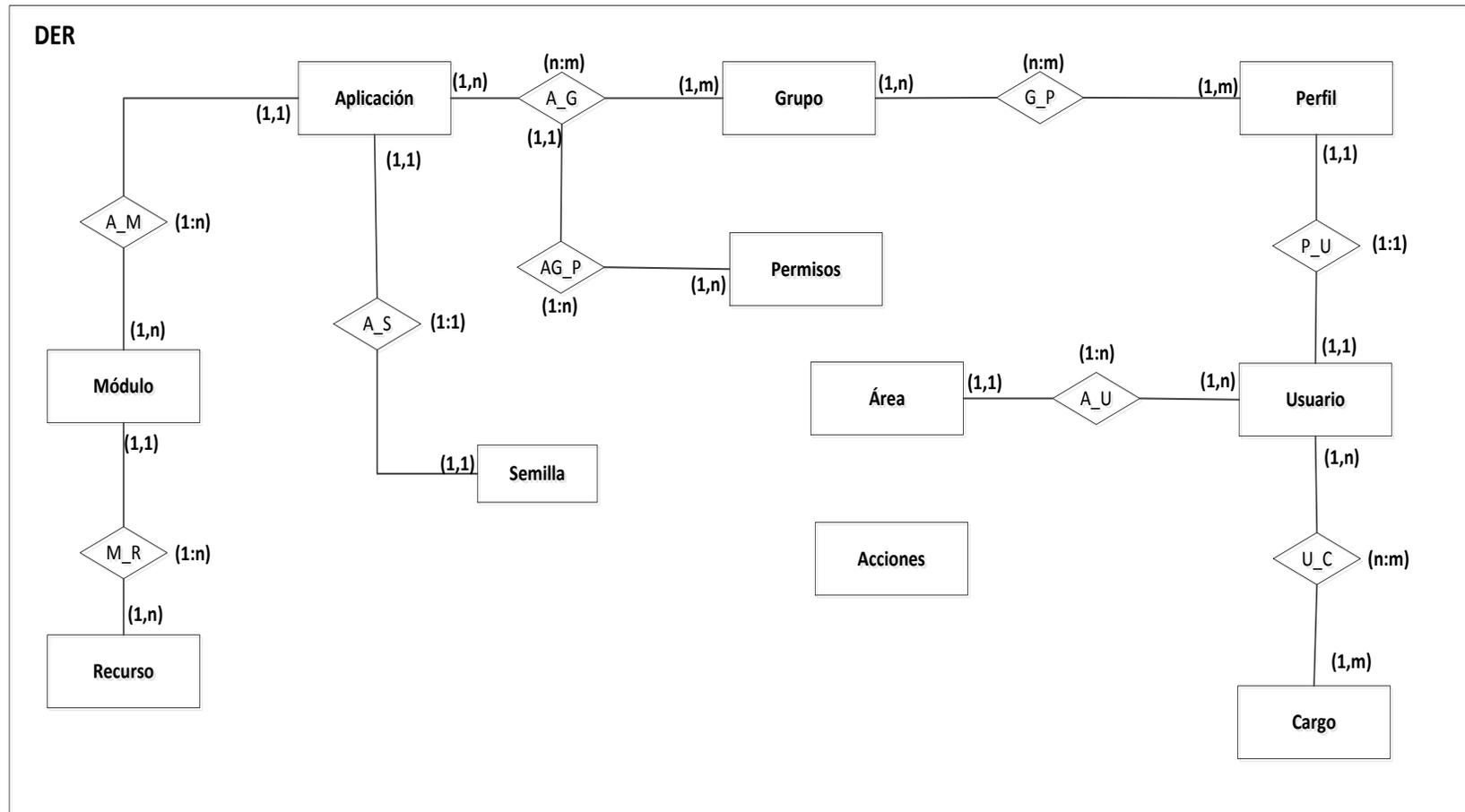


Figura IV. 25. Diagrama entidad relación DER

La normalización en las tablas se la realizo a partir del diseño del modelo conceptual, se aplicaron las 3 primeras formas normales para poder cumplir con integridad de datos, evitar redundancia de datos y permitir que la estructura de base de datos sea eficiente.

A partir de la normalización se diseñó el modelo lógico, donde se definieron tipos de datos y tamaño considerando la información que va a almacenar cada uno.

El modelo físico de base de datos, diccionario de datos y script de despliegue se encuentra detallado en el **Anexo 1 - Sección C.**

Proyección de crecimiento

Para el correcto mantenimiento de la base de datos se especificó una estimación de crecimiento, esta proyección podrá ayudar al administrador del colegio a tomar las precauciones adecuadas para evitar que el hardware donde se encuentra alojada la base de datos colapse.

Al ser una base de datos nueva y que no cuenta con estadísticas para establecer el índice de crecimiento se consideró el registro inicial. El esquema de base de datos básicamente está diseñado para controlar el accesos de usuarios a aplicaciones, el registro inicial de la base de datos contemplo guardar toda la información del módulo de integración de aplicaciones, el modulo contempló el registro de 20 recursos, divididos en 3 módulos, la creación de 1 grupo en el cual se asignó 1 usuario, se parametrizo las áreas, cargos, etc., con los que cuenta la institución educativa, todo esto equivale al peso que podemos observar en la **Tabla IV. XXXIII.**

Tabla IV. XXXIII. Tamaño inicial

Nombre Tabla	Size
acciones	40 kB
aplicación	40 kB
área	40 kB
cargo	40 kB
grupo	40 kB
grupo_aplicacion	24 kB
modulo	40 kB
perfil	12 kB
perfil_grupo	12 kB
permisos	24 kB
recursos	40 kB
semilla	40 kB
usuario	12 kB
usuario_cargo	12 kB
Total	416 KB

Entonces se considera que se va a registrar una nueva aplicación con todos sus módulos, recursos, etc., y que el personal que ahora se va a registrar y realizar acciones sobre la base de datos es la siguiente: alumnos registrados son alrededor de 2500 entre sección diurna y nocturna para el año lectivo actual, 84 docentes y personal administrativo 20, dando un total de 2604.

En la **Tabla IV. XXXIV** se encuentra una proyección del crecimiento que tendrá la base de datos cuando se realice un nuevo registro de aplicación, modulo, recurso, etc., y también cuando aumente el números de usuarios registrados y realizando acciones sobre la misma. El cálculo se lo realizo de la siguiente manera: para la segunda aplicación

registrada no varía los tamaños, pero para el registro de usuario se hizo multiplicando la población total * el tamaño actual.

Tabla IV. XXXIV. Proyección de crecimiento

Nombre Tabla	Tamaño actual	2da aplicación	Aumento de usuarios
acciones	40 kB	0 kB	104160 KB
aplicación	40 kB	40 kB	0 kB
área	40 kB	0 kB	0 kB
cargo	40 kB	0 kB	0 kB
grupo	40 kB	40 kB	0 kB
grupo_aplicacion	24 kB	24 kB	0 kB
modulo	40 kB	40 kB	0 kB
perfil	12 kB	0 kB	31248 kB
perfil_grupo	12 kB	0 kB	31248 kB
permisos	24 kB	24 kB	0 kB
recursos	40 kB	24 kB	0 kB
semilla	40 kB	40 kB	0 kB
usuario	12 kB	0 kB	31248 kB
usuario_cargo	12 kB	0 kB	31248 kB
Total	416 KB	232 KB	229152 KB
Total aproximado	229800 Kb equivale a 225 Mb		

El administrador tendrá que tener estimado un espacio de almacenamiento físico de por lo menos 225 Mb cuando se registre una nueva aplicación con un índice mayor de usuarios.

Pruebas de aceptación

Para el proceso de pruebas de aceptación se definió el flujo que sigue toda base de datos el cual es: datos – proceso - información. Para obtener información confiable los datos y el proceso realizado debe estar correcto. Es por esto que como primer punto se diseñó un

esquema conceptual, después de normalizar las tablas y campos se llegó a un esquema lógico esto permitió que las tablas guarden un esquema referencial confiable.

Validando que el proceso de almacenar datos y generar información esté correcto el diseño establecido para el colegio cumplió con todas las pruebas de satisfacción establecidas para la entrega de un producto de calidad al cliente.

La planificación total de pruebas definidas para la aceptación de la base de datos desarrollada, se describe en el **Anexo 1 - Sección C**.

4.6.2. Iteración 2

En esta iteración se desarrollaron 2 metáforas las cuales consistían en el análisis de la base de datos del entorno de aprendizaje Moodle y también análisis de la base de datos del sistema académico del colegio.

Este análisis de las bases de datos consistió en investigar las tablas y campos que servirían para la integración mediante el módulo de administración de sistemas.

Metáfora 2

Moodle cuenta con 314 tablas de las cuales se tuvieron que identificar las tablas que serían afectadas para la integración con el sistema académico del colegio.

Resultado del análisis

Después de las investigaciones realizadas en la base de datos de Moodle se determinó que en el proceso inicial de creación de usuario, cursos, paralelos, materias. Moodle realiza lo siguiente:

Para la creación de usuario moodle utiliza las tablas mdl_user, mdl_context, mdl_user_preferences, donde en la tabla mdl_user se almacena toda la información personal del usuario (docente o estudiante), la tabla mdl_context es una tabla genérica que almacena el código del usuario dependiendo de un tipo que previamente está definido en las reglas de moodle y también mdl_user_preferences, esta tabla hereda el código de la tabla mdl_user. En la **Figura IV. 26** se observan las entidades necesarias para la creación de usuario.

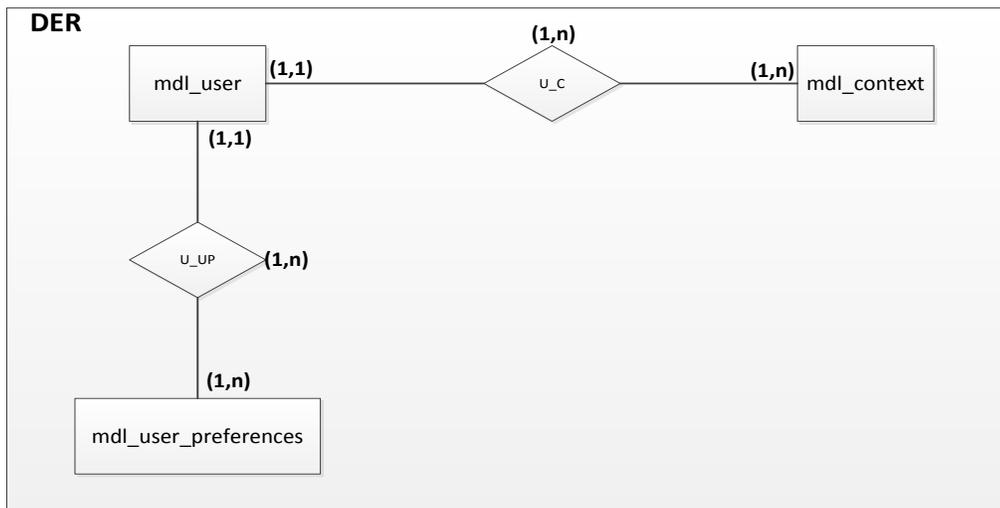


Figura IV. 26. Entidades usadas para la creación de usuarios en Moodle

En el **Anexo 1 - Sección C** se encuentra el detalle completo de las tablas y campos utilizados para la creación de usuarios en Moodle.

En Moodle los cursos, paralelos son denominados categorías, para la creación de categorías se utilizan las tablas: mdl_course_categories y mdl_context. La tabla mdl_course_categories es donde se almacena los datos de la categoría (cursos, paralelos), está relacionada con la tabla mdl_context, la tabla mdl_context es una tabla genérica que almacena el código del usuario dependiendo de un tipo que previamente está definido en las reglas de Moodle. En la **Figura IV. 27** se observa las entidades necesarias para la creación de categorías.

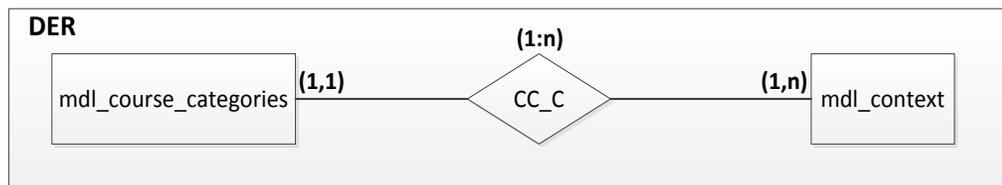


Figura IV. 27. Entidades usadas para la creación de categorías en Moodle

En el **Anexo 1 - Sección C** se encuentra el detalle completo de las tablas y campos utilizados para la creación de categorías en Moodle.

En Moodle no existe el término materia, a su vez este término es reemplazado por curso, las tablas utilizadas para la creación de cursos son: mdl_course, mdl_course_format_options, mdl_context, mdl_block_instances, mdl_course_sections, mdl_course_categories. En la **Figura IV. 28** se observa las entidades necesarias para la creación de cursos.

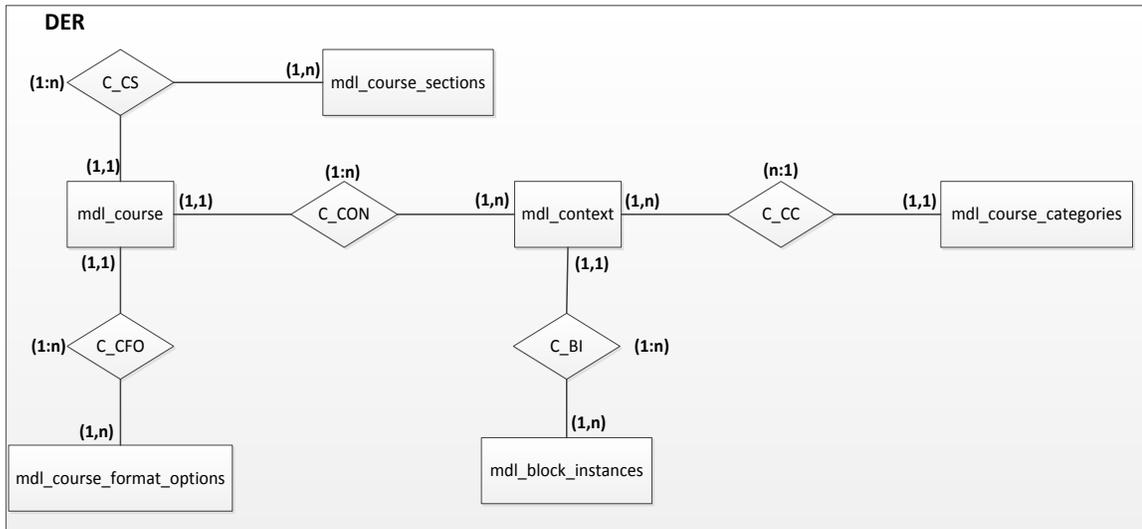


Figura IV. 28. Entidades usadas para la creación de cursos en Moodle

En el **Anexo 1 - Sección C** se encuentra el detalle completo de las tablas y campos utilizados para la creación de cursos en Moodle.

Para la matriculación de estudiantes en los cursos y asignación de profesores se utilizan las siguientes tablas: md_course, mdl_enrol, mdl_user_enrolments, mdl_role_assignments, mdl_context. En la **Figura IV. 29** se observa las tablas y campos utilizados para la matriculación de estudiantes y asignación de profesores a los cursos.

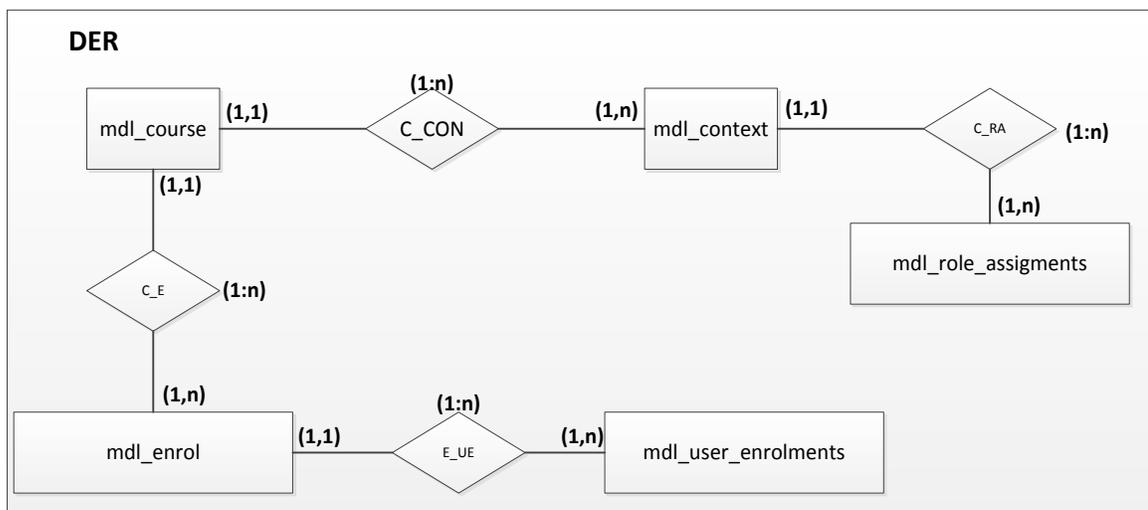


Figura IV. 29. Entidades usadas para matriculación de estudiantes y docentes en Moodle

En el **Anexo 1 - Sección C** se encuentra el detalle completo de las tablas y campos utilizados para la asignación de roles en Moodle.

Metáfora 3

La base de datos del sistema académico consta de un total de 50 tablas, las mismas que fueron analizadas para detectar los campos que se utilizarían para la integración con Moodle.

Resultados de análisis

Después del análisis y reuniones con las personas encargadas del mantenimiento de la base de datos se determinó que las tablas utilizadas para el almacenamiento de información de estudiantes, docentes, cursos, paralelos, materias, año lectivo y matriculas son las siguientes:

La tabla donde se almacena la información personal del estudiante se denomina T_ESTUDIANTE_R, la tabla utilizada para el almacenamiento de la información del docente se denomina T_DOCENTE_S.

La tabla de almacenamiento de la información de cursos, paralelos se denomina T_CURSOS_K, la tabla donde se almacena la información de las materias T_MATERIA_K.

Para determinar el año lectivo vigente se revisó que la tabla que almacena esta información es T_ANIOLECTIVO_K, la tabla que almacena los estudiantes matriculados se denomina T_MATRICULA_R.

En la **Figura IV. 30**, se detalla las tablas del sistema académico analizadas, que servirán para la extracción de datos.

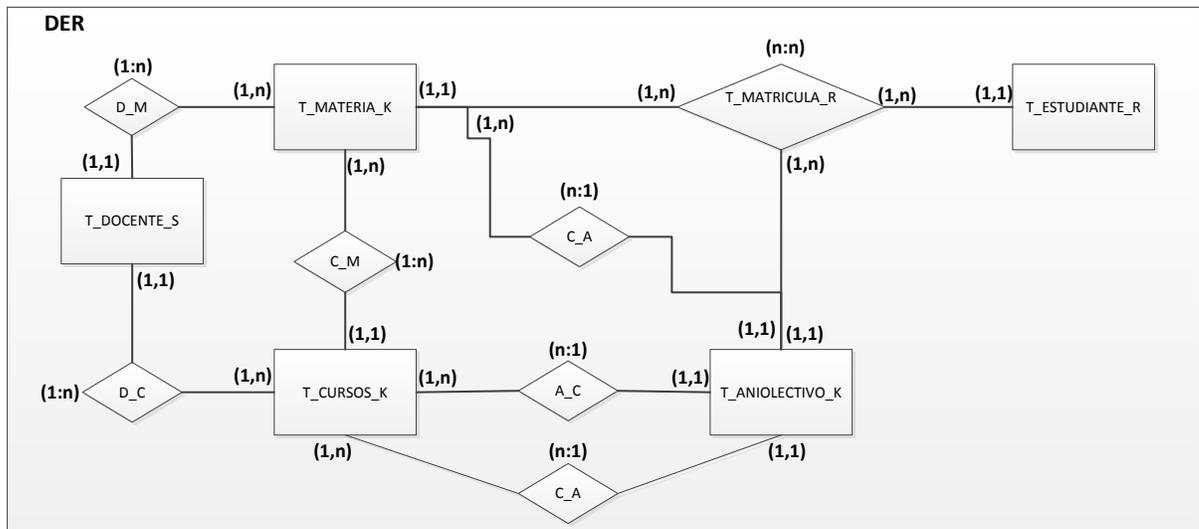


Figura IV. 30. Tablas sistema académico

En el **Anexo 1 - Sección C** se encuentra el detalle completo de las tablas y campos utilizados para la extracción de datos desde el sistema académico.

4.6.3. Iteración 3

En la presente iteración se desarrollaron 3 metáforas, la primera en desarrollarse es el diseño de base de datos para la integración de sistemas, esta base de datos se desarrolló basándose en la investigación y análisis realizados anteriormente, también se definió la arquitectura de software que contemplo la aplicación y servicio web, adicional a esto se planteó la arquitectura de hardware en el cual se iba a implementar el aplicativo. Con las directrices que sigue el colegio en cuanto a diseño de interfaces se estableció el estándar de interfaces que seguiría la aplicación.

Metáfora 4

El modelo de base datos desarrollado en esta metáfora es de vital importancia, ya que en esta base se almacenaran temporalmente los datos que se extraigan a través del servicio web que publicará información concerniente a cursos, estudiantes, materias, docentes, etc. Este diseño se estableció tomando el análisis de las bases de datos implicadas en la integración.

Para el cumplimiento de la metáfora se contempló un total de 5 tareas de ingeniería las cuales están definidas en un cronograma que el programador debe cumplir. Las tareas de ingeniería contemplaron lo siguiente: generación el diagrama DER de base de datos, normalización de tablas, diagrama lógico, diagrama físico, proyección de crecimiento de la base de datos, script de despliegue y diccionario de datos, para un mayor detalle de lo establecido en esta metáfora seguir el **Anexo 1 - Sección C**.

Diagrama entidad relación (DER)

Después del análisis de la base de datos del sistema académico y de la base de datos de Moodle, se desarrolló una base intermedia la cual permitirá mejorar la manejabilidad de la información, ya que los datos obtenidos están incompletos, además se diseñó con el objetivo de tener campos que puedan servir para manejar los estados de los datos sincronizados y no sincronizados a Moodle. La Figura IV. 31 hace referencia a la base de datos desarrollada para la integración de sistema

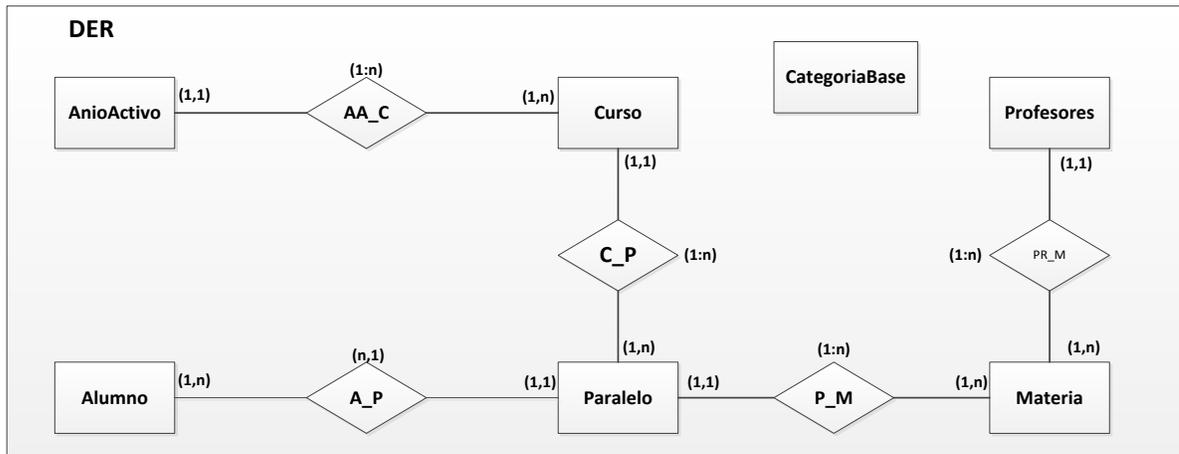


Figura IV. 31. Diagrama entidad relación

La normalización en las tablas se la realizó a partir del diseño conceptual, se aplicaron las tres primeras formas normales para poder cumplir con integridad de datos, evitar redundancia de datos y permitir que la estructura de base de datos sea eficiente.

A partir de la normalización se diseñó el modelo lógico, se definieron tipos de datos y tamaño considerando la información que va a almacenar cada uno. La definición de cada

tabla del modelo lógico se encuentra explicado con más detalle en la generación del diccionario de datos que se encuentra en el **Anexo 1 - Sección C**.

Proyección de crecimiento

La base de datos desarrollada permite almacenar temporalmente los datos extraídos a través del servicio web que consume datos desde la base de datos del sistema académico, es por esto que esta base de datos no va obtener un crecimiento secuencial por que al iniciar el nuevo año lectivo la data que se encuentra en este año lectivo se borrara automáticamente y dará paso a la nueva información.

El escenario inicial del peso de la base de datos se obtuvo de los datos registrados para el año lectivo en curso donde el peso total de la base de datos es el que se describe en la **Tabla IV. XXXV**.

Tabla IV. XXXV. Tamaño Actual

Table Name	Size
alumnos	408 kB
anioactivo	24 kB
categoriabase	24 kB
cursos	24 kB
materias	96 kB
paralelos	24 kB
profesores	32 kB
Tamaño total	632 Kb

Entonces esto nos permite decir que el peso de esta base de datos dependerá del número de registro extraídos desde el servicio web, si el colegio tiene un incremento del doble de

registro el tamaño para el nuevo año lectivo será máximo de 1264 kB, este valor se obtiene de especificar que para el nuevo año lectivo el colegio tendrá el doble de estudiantes, docentes, cursos, materias, etc.

Pruebas de aceptación

Validando que el proceso de almacenar datos y generar información esté correcto, el diseño de base de datos establecido para el colegio que realiza el registro temporal de información cumplió con todas las pruebas de satisfacción establecidas por el cliente. La planificación total de pruebas definidas para la aceptación de la base de datos desarrollada, se describe en el **Anexo 1 - Sección C**.

Metáfora 5

En el desarrollo de esta metáfora se definieron los estándares de arquitectura de despliegue y software que se utilizó para el desarrollo e instalación de la aplicación

Definición de arquitectura de software

El modelo definido para el desarrollo de las aplicaciones es MVC (Modelo Vista Controlador), la utilización de este modelo arquitectónico permite que la aplicación mantenga un orden al momento de extraer, procesar y presentar información.

El modelo representa la capa en donde se procesan todas las peticiones hacia la base de datos, esta capa también recibe todos los movimientos que realice el controlador.

El controlador o también denominado capa de negocio es en donde se procesan todas las peticiones que vienen desde la vista o capa de presentación, en esta capa también se definen todas las reglas y flujos que deben cumplirse para el correcto funcionamiento del aplicativo.

La vista o capa de presentación, es la encargada de interactuar con el usuario permitiendo él envié de solicitudes y transacciones a las capas de más bajo nivel como son el controlador y el modelo.

En la **Figura IV.32** se muestra el diseño arquitectónico de la aplicación y servicio web desarrollados para el colegio.

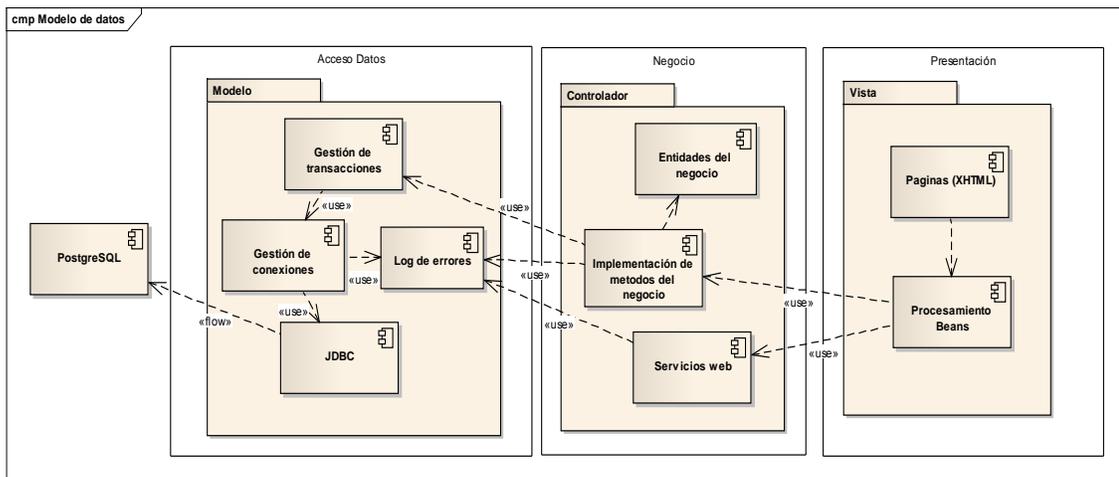


Figura IV. 32. Modelo de base de datos para la integración

Definición del despliegue de la aplicación

La solución informática que se elaboró para el colegio tiene el siguiente diseño arquitectónico, se desplegará en un servidor Glassfish tanto el servicio web que consumirá

información desde la base de datos del sistema académico del colegio como aplicación web de integración, la base de datos PostgreSQL que utiliza el sistema de integración estará desplegada en el servidor donde se aloja la base de datos de Moodle. En la **Figura IV. 33** se detalla el despliegue de la aplicación, servicio web y base de datos.

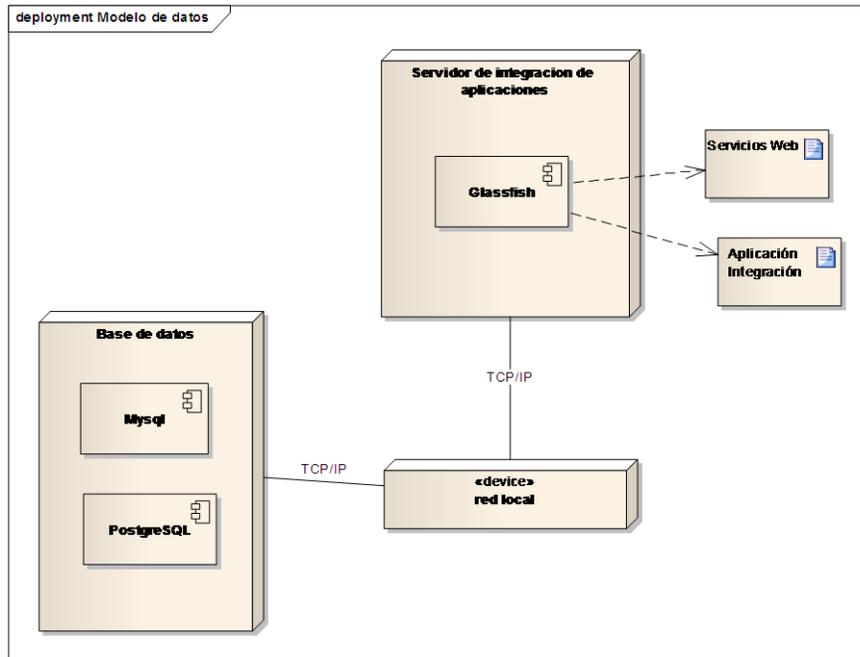


Figura IV. 33. Arquitectura de despliegue

Metáfora 6

En el desarrollo de esta metáfora se definieron los estándares de interfaces de usuario y estándares de codificación que se utilizaron en la aplicación

Definición de interfaces

La interfaz gráfica está definida por colores rojos y blanco, se desarrolló de forma sencilla y amigable, esto con el objetivo de disminuir la curva de aprendizaje del usuario, y que

permita que usuarios con poca experiencia puedan manipular el aplicativo. En la **Figura IV. 34** se detalla un bosquejo de la plantilla de usuario del menú principal.

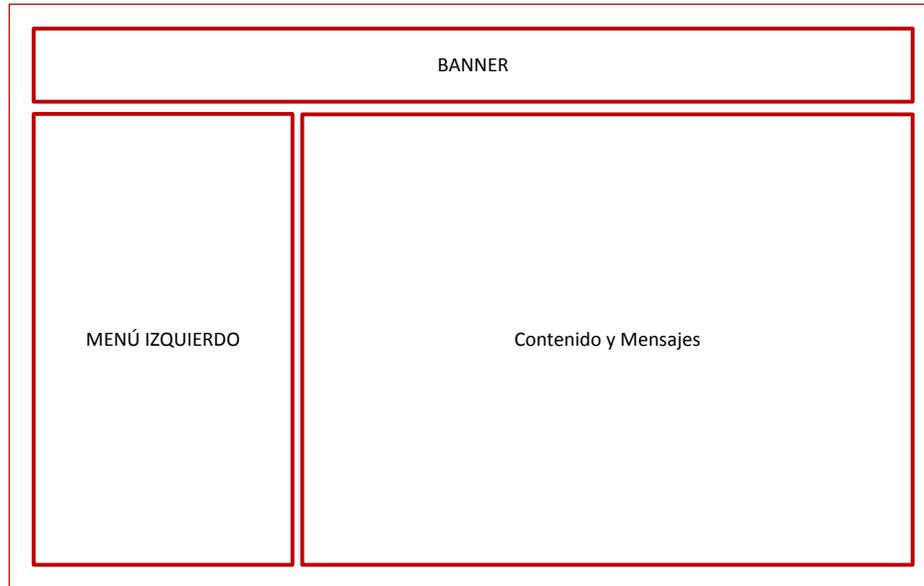


Figura IV. 34. Plantilla de menú principal

Banner.- En la parte superior de la pantalla se ubicará el logo y nombres referente al colegio.

Menú Izquierdo.- En esta sección se ubicarán las opciones del sistema.

Contenido y Mensajes.- En esta sección se presentarán las páginas y mensajes de la aplicación.

En el desarrollo de las historias de usuario se definirán las interfaces graficas de cada una de las historias establecidas por el cliente.

Definición de estándar de codificación

El estándar de codificación definido para el desarrollo de la aplicación debe permitir entender de manera rápida y sencilla el código empleado en el desarrollo, todo esto para el correcto mantenimiento y corrección de errores de la aplicación a futuro. La aplicación en su totalidad está desarrollada en java es por esto que los programadores deben seguir el documento “Code Conventions for the Java™ Programming Language” (27) publicado por Oracle corporation.

4.6.4. Iteración 9

En la presente iteración se desarrollan las historias de usuario para la integración de información de estudiantes y docentes en Moodle.

Historia de usuario 14

Para el cumplimiento de la historia de usuario 14 la cual consiste en la integración de la información del estudiante desde el sistema académico a Moodle se establecieron 2 tareas de ingeniería, cronograma de cumplimiento de tareas y pruebas de aceptación. Estas tarjetas se encuentran detalladas en el **Anexo 1 - Sección C**.

El desarrollo de este requerimiento tomó un total de 754 líneas de código en java, 5 funciones en postgresSQL, con lo que se cumplió con el desarrollo total del requerimiento.

La interfaz de usuario contará con 2 botones los cuales permitirán mostrar la información de estudiante y sincronizar la información hacia Moodle. El detalle completo de la interfaz gráfica se encuentra en el **Anexo 1 - Sección C**.

La historia de usuario realizada cumplió con todas las pruebas de satisfacción establecidas, permitiendo determinar que el requerimiento cumple con todas las condiciones definidas por el cliente.

Historia de usuario 15

Para el cumplimiento de la historia de usuario 15 la cual consiste en la integración de la información del docente desde el sistema académico a Moodle se establecieron 2 tareas de ingeniería, cronograma de cumplimiento de tareas y pruebas de aceptación. Estas tarjetas se encuentran detalladas en el **Anexo 1 - Sección C**.

El desarrollo de este requerimiento tomo un total de 815 líneas de código en java y 5 funciones realizadas en postgresQL, con lo que se cumplió con el desarrollo total del requerimiento.

La interfaz de usuario contará con 2 botones los cuales permitirán mostrar la información del docente y sincronizar la información hacia Moodle. El detalle completo de la interfaz gráfica se encuentra en el **Anexo 1 - Sección C**.

La historia de usuario realizada cumplió con todas las pruebas de satisfacción establecidas, permitiendo determinar que el requerimiento cumple con todas las condiciones definidas por el cliente.

CONCLUSIONES

- Los servidores de aplicaciones y servicios web en la actualidad conforman un conjunto de herramientas que ayudan a las empresas a compartir información entre sistemas, permitiendo de esta manera ahorrar tiempos, disminuir duplicidad de información y mejorar sustancialmente los procesos que realiza la institución.
- Después del análisis estadístico inferencial aplicado al parámetro uso de RAM se observa que existen diferencias significativas sobre los resultados obtenidos de los servidores de aplicación en estudio, concluyendo que GlassFish con una media de 57,76% es mejor que JOnAS y Geronimo con medias de 77,10% y 97,83% respectivamente.
- En base al análisis estadístico inferencial aplicado al parámetro uso de CPU se observa que existen diferencias significativas sobre los resultados obtenidos de los servidores de aplicación en estudio, concluyendo que GlassFish con una media de 13,61% es mejor que JOnAS y Geronimo con medias de 33,87% y 33,15% respectivamente.
- Después del análisis estadístico inferencial aplicado al parámetro uso de Disco duro se observa que no existen diferencias significativas sobre los resultados obtenidos de los servidores de aplicación en estudio, concluyendo que GlassFish

con una media de 0,07% es igual a JOnAS y Gerónimo con medias de 0,08% y 0,10% respectivamente.

- En base al análisis estadístico inferencial aplicado al parámetro Tiempo de respuesta se observa que existen diferencias significativas sobre los resultados obtenidos de los servidores de aplicación en estudio, concluyendo que GlassFish con una media de 853,56s optimiza el tiempo de respuesta en comparación a JOnAS y Gerónimo con medias de 1377,42s y 1608,16s respectivamente.
- Como resultado del análisis estadístico realizado a los parámetros de comparación se determinó que el servidor de aplicaciones GlassFish supera con un porcentaje de 37.08% y 44,26% sobre JOnAS y Gerónimo respectivamente, siendo GlassFish el servidor de aplicaciones con mejores características para manejar los recursos hardware del servidor, ya que optimiza el consumo de memoria RAM, procesador, disco duro y es más eficiente en la respuesta que ofrece a peticiones realizadas por usuarios o aplicaciones que requieran de información.
- Para el Colegio Capitán Edmundo Chiriboga se realizó una aplicación que consume servicios web, este aplicativo realiza el proceso de integrar el sistema académico de la institución con el entorno virtual de aprendizaje (Moodle) permitiendo que estos sistemas compartan información y reduzcan el tiempo que utilizaban los

miembros de la institución al ingresar información de cursos, paralelos, materias, estudiantes, docentes, etc., de forma individual en ambos sistemas, tanto el sistema como los servicios web están alojados en el servidor de aplicaciones GlassFish.

RECOMENDACIONES

- En la actualidad en una organización existe software de distintas tecnologías, lenguajes y base de datos, se recomienda el uso de servicios web, ya que es un mecanismo que permite integrar tecnologías logrando compartir información, restando tiempos, minimizando el esfuerzo que deben hacer las personas de la organización al realizar operaciones redundantes en aplicaciones que comparten información.
- Para aumentar la productividad de los programadores de sistemas se recomienda la utilización del IDE NetBeans integrado con el servidor de aplicaciones GlassFish.
- Se recomienda realizar una nueva investigación de servidores de aplicación donde se compare el servidor GlassFish ganador de esta investigación con un servidor de aplicaciones propietario.
- Es recomendable para la realización de proyectos de software utilizar una metodología de desarrollo, esto permitirá optimizar tiempos, organizar al equipo de desarrollo, aumentar la calidad del software y disminuir los costos de desarrollo en tanto y cuanto sea posible.
- Se recomienda construir nuevos servicios web o consumir los implementados para que las aplicaciones existentes en el Colegio Capitán Edmundo Chiriboga, interactúen entre si y compartan información.

RESUMEN

Análisis comparativo de servidores de aplicación para la integración de sistemas mediante servicios web XML. Caso práctico: Colegio Capitán Edmundo Chiriboga de la ciudad de Riobamba. Se realizó una investigación descriptiva e inferencial, en la cual se establecieron prototipos con los servidores de aplicaciones: GlassFish, JOnAS y Gerónimo, a fin de determinar el rendimiento de los mismos.

Para esto se ha diseñado una aplicación que consume servicios web (Módulo de Integración y Administración de Sistemas), este aplicativo realiza el proceso de integrar el sistema académico de la institución con el entorno virtual de aprendizaje (Moodle) permitiendo que estos sistemas compartan información y reduzcan el tiempo que utilizaban los miembros de la institución al ingresar información de cursos, paralelos, materias, estudiantes y docentes, de forma individual en ambos sistemas.

De acuerdo al análisis comparativo de estos servidores de aplicación y el establecimiento de parámetros de comparación para determinar el rendimiento: uso de RAM, uso de CPU, uso de disco, tiempo de respuesta y mediante el uso del coeficiente de Pearson, se determina que GlassFish supera con un porcentaje de 37.08% y 44,26% sobre JOnAS y Gerónimo respectivamente, se concluye que GlassFish es el servidor de aplicaciones con mejor rendimiento y características para manejar los recursos hardware del servidor. Se recomienda realizar una nueva investigación de servidores de aplicación donde se

compare GlassFish como servidor de aplicaciones ganador de esta investigación con un servidor de aplicaciones propietario.

SUMMARY

Comparative Analysis of application servers for the integration system through XML Web services. Case study: Captain Edmundo Chiriboga High School in Riobamba. GlassFish, JOnAS and Geronimo, to determine the performance of the same: a descriptive and inferential research, in which prototypes with application servers were set was performed. A describe and inferential research was performed, in which prototypes were set with application server GlassFish, JOnAS and Geronimo, aiming to determine their performance.

In order to do this, it was designe dan application which consumes web services (Module Integration and Systems Administration), this application integrates the academic system of the institution to the virtual learning environment (Moodle) allowing so this systems to share information and reduce the time the personnel of the institutions used to load data from course, levels, subject, students and teachers individually in both systems.

According to the comparative analysis of these application servers and the set of comparison parameters to determine performance: RAM use, CPU use, disc use, time of response and by means of the use Pearson coefficient; it was determined that GlassFish exceeds a percentage of 37.08% and 44.26% to JOnAS and Geronimo respectively, in short GlassFish is the application server with the best performance and features to manage the hardware resources of the server. It is recommended to develop a new research work of

application server where GlassFish as the winner applications server is compared to an owner applications server.

GLOSARIO

CPU

La unidad central de procesamiento, es el hardware dentro de un computador u otros dispositivos programables, que interpreta las instrucciones de un programa de ordenador mediante la realización de las operaciones básicas aritméticas, lógicas y de entrada/salida del sistema.

Contenedor Web

Es una interfaz entre los componentes Web (servlets, páginas JavaServer Faces o Páginas JSP) y el servidor web. En tiempo de ejecución ofrece los servicios requeridos por los componentes web.

Desviación estándar

Es una medida de dispersión, que nos indica cuánto pueden alejarse los valores respecto al promedio (media), por lo tanto es útil para buscar probabilidades de que un evento ocurra.

Escalabilidad

Es un término usado en tecnología para referirse a la propiedad de aumentar la capacidad de trabajo o de tamaño de un sistema sin comprometer su funcionamiento y calidad

normales. Cuando un sistema tiene esta propiedad, se le refiere comúnmente como "sistema escalable".

Estadística Inferencial

Utiliza los datos de la estadística descriptiva y se apoya en el cálculo de probabilidades para la obtención de conclusiones sobre una población a partir de los resultados obtenidos de una muestra.

Integridad

La integridad de datos se refiere a los valores reales que se almacenan y se utilizan en las estructuras de datos de la aplicación. La aplicación debe ejercer un control deliberado sobre todos los procesos que utilicen los datos para garantizar la corrección permanente de la información.

Servicios Web

Un Servicio Web es una aplicación software identificada por un URI (Uniform Resource Identifier), cuyas interfaces se pueden definir, describir y descubrir mediante documentos XML. Los servicios web hacen posible la interacción entre "agentes" software (aplicaciones) utilizando mensajes XML intercambiados mediante protocolos de Internet.

URL

Son las siglas en inglés de uniform resource locator, que sirve para nombrar recursos en Internet. Este nombre tiene un formato estándar y tiene como propósito asignar una dirección única a cada uno de los recursos disponibles en Internet, como por ejemplo textos, imágenes, vídeos, etc.

ANEXO 1

Sección A

En la sección A se detalla los resultados obtenidos al realizar el monitoreo de los servidores de aplicación GlassFish, JOnAS y Gerónimo.

A continuación visualizamos los datos del uso de RAM al enviar 20000 solicitudes, en el servidor de aplicaciones Glassfish.

21:10:21	438908	592464	57,44	32228	262468	640484	30,79
21:10:22	438908	592464	57,44	32228	262468	640484	30,79
21:10:23	438908	592464	57,44	32236	262468	640484	30,79
21:10:24	438908	592464	57,44	32236	262468	640484	30,79
21:10:25	438908	592464	57,44	32236	262468	640484	30,79
21:10:26	438908	592464	57,44	32236	262468	640484	30,79
21:10:27	438900	592472	57,45	32236	262468	640484	30,79
21:10:28	438900	592472	57,45	32236	262468	640484	30,79
21:10:29	438900	592472	57,45	32244	262468	640484	30,79
21:10:30	438900	592472	57,45	32244	262468	640484	30,79
21:10:31	438900	592472	57,45	32244	262468	640484	30,79
21:10:32	438900	592472	57,45	32244	262468	640484	30,79
21:10:33	438900	592472	57,45	32244	262468	640484	30,79
21:10:34	438908	592464	57,44	32252	262468	640484	30,79
21:10:35	438908	592464	57,44	32252	262472	640484	30,79
21:10:36	438908	592464	57,44	32252	262472	640484	30,79
21:10:37	438908	592464	57,44	32252	262476	640484	30,79
21:10:38	438908	592464	57,44	32252	262476	640484	30,79
21:10:39	438908	592464	57,44	32252	262476	640484	30,79
21:10:40	438908	592464	57,44	32260	262476	640484	30,79
21:10:41	438908	592464	57,44	32260	262476	640484	30,79
21:10:42	438908	592464	57,44	32260	262476	640484	30,79
21:10:43	438876	592496	57,45	32260	262476	650740	31,29
21:10:44	438900	592472	57,45	32260	262476	650740	31,29
21:10:45	438900	592472	57,45	32268	262476	650740	31,29
Media:	440983	590389	57,24	31276	262041	643320	30,93

A continuación visualizamos los datos del uso de RAM al enviar 40000 solicitudes, en el servidor de aplicaciones Glassfish.

```
Linux 2.6.32-71.el6.i686 (localhost.localdomain)      04/07/14      _i686_ (1 CPU)
21:15:16      kbmemfree kbmemused  %memused kbbuffers  kbcached  kbcommit  %commit
21:15:17      440924    590448    57,25    32428     262644    649016    31,20
21:15:18      440768    590604    57,26    32428     262644    659260    31,70
21:15:19      440792    590580    57,26    32428     262644    659260    31,70
21:15:20      440792    590580    57,26    32428     262644    659260    31,70
21:36:58      435192    596180    57,80    34344     262792    652016    31,35
21:36:59      435192    596180    57,80    34344     262800    652016    31,35
21:37:00      435192    596180    57,80    34356     262800    652016    31,35
21:37:01      435192    596180    57,80    34356     262800    652016    31,35
21:37:02      435192    596180    57,80    34356     262800    652016    31,35
21:37:03      435192    596180    57,80    34356     262800    652016    31,35
21:37:04      435192    596180    57,80    34356     262800    652016    31,35
21:37:05      435192    596180    57,80    34364     262800    652016    31,35
Media:        435656    595716    57,76    33396     262724    656200    31,55
```

A continuación visualizamos los datos del uso de RAM al enviar 60000 solicitudes, en el servidor de aplicaciones Glassfish.

```
Linux 2.6.32-71.el6.i686 (localhost.localdomain)      04/07/14      _i686_ (1 CPU)
21:40:15      kbmemfree kbmemused  %memused kbbuffers  kbcached  kbcommit  %commit
21:40:16      438444    592928    57,49    34476     262876    648928    31,20
21:40:17      438436    592936    57,49    34476     262876    659172    31,69
21:40:18      438436    592936    57,49    34476     262876    659172    31,69
21:40:19      438436    592936    57,49    34476     262876    659172    31,69
21:40:20      438436    592936    57,49    34476     262876    659172    31,69
21:40:21      438436    592936    57,49    34484     262876    659172    31,69
21:40:22      438436    592936    57,49    34484     262876    659172    31,69
21:40:23      438436    592936    57,49    34484     262876    659172    31,69
21:40:24      438436    592936    57,49    34484     262876    659172    31,69
22:01:56      431048    600324    58,21    36564     265908    659932    31,73
22:01:57      431048    600324    58,21    36572     265908    659932    31,73
22:01:58      431048    600324    58,21    36572     265908    659932    31,73
22:01:59      431048    600324    58,21    36572     265908    659932    31,73
22:02:00      431048    600324    58,21    36572     265908    659932    31,73
22:02:01      431056    600316    58,21    36572     265908    659932    31,73
22:02:02      431056    600316    58,21    36580     265904    659932    31,73
22:02:03      431056    600316    58,21    36580     265908    659932    31,73
Media:        427372    604000    58,56    35611     265670    668961    32,16
```

A continuación visualizamos los datos del uso de CPU al enviar 20000 solicitudes, en el servidor de aplicaciones Glassfish.

```
Linux 2.6.32-71.el6.i686 (localhost.localdomain) 03/07/14 _i686_ (1 CPU)
22:00:37 CPU %user %nice %system %iowait %steal %idle
22:00:38 all 1,01 0,00 3,03 2,02 0,00 93,94
22:00:39 all 7,00 0,00 11,00 0,00 0,00 82,00
22:00:40 all 7,00 0,00 9,00 0,00 0,00 84,00
22:00:41 all 4,04 0,00 5,05 0,00 0,00 90,91
22:00:42 all 2,00 0,00 5,00 0,00 0,00 93,00
22:00:43 all 1,00 0,00 4,00 0,00 0,00 95,00
22:00:44 all 2,00 0,00 4,00 0,00 0,00 94,00
22:00:45 all 1,00 0,00 4,00 0,00 0,00 95,00
22:00:46 all 2,00 0,00 2,00 0,00 0,00 96,00
!2:22:20 all 3,03 0,00 3,03 0,00 0,00 93,94
!2:22:21 all 1,00 0,00 2,00 0,00 0,00 97,00
!2:22:22 all 1,01 0,00 2,02 0,00 0,00 96,97
!2:22:23 all 0,99 0,00 2,97 0,00 0,00 96,04
!2:22:24 all 1,00 0,00 3,00 0,00 0,00 96,00
!2:22:25 all 0,99 0,00 1,98 0,00 0,00 97,03
Media: all 8,25 0,00 5,99 0,22 0,00 85,54
```

A continuación visualizamos los datos del uso de CPU al enviar 40000 solicitudes, en el servidor de aplicaciones Glassfish.

```
Linux 2.6.32-71.el6.i686 (localhost.localdomain) 03/07/14 _i686_ (1 CPU)
22:27:16 CPU %user %nice %system %iowait %steal %idle
22:27:17 all 1,00 0,00 2,00 0,00 0,00 97,00
22:27:18 all 5,00 0,00 10,00 0,00 0,00 85,00
22:27:19 all 2,00 0,00 2,00 0,00 0,00 96,00
22:27:20 all 0,00 0,00 3,00 0,00 0,00 97,00
22:27:21 all 2,00 0,00 2,00 0,00 0,00 96,00
22:27:22 all 0,00 0,00 2,02 0,00 0,00 97,98
22:27:23 all 2,02 0,00 2,02 0,00 0,00 95,96
22:48:58 all 2,97 0,00 3,96 0,00 0,00 93,07
22:48:59 all 2,02 0,00 3,03 0,00 0,00 94,95
22:49:00 all 4,04 0,00 3,03 0,00 0,00 92,93
22:49:01 all 2,97 0,00 3,96 0,00 0,00 93,07
22:49:02 all 2,00 0,00 4,00 0,00 0,00 94,00
22:49:03 all 4,00 0,00 3,00 0,00 0,00 93,00
22:49:04 all 2,00 0,00 4,00 0,00 0,00 94,00
Media: all 13,60 0,00 8,62 0,01 0,00 77,77
```

A continuación visualizamos los datos del uso de CPU al enviar 60000 solicitudes, en el servidor de aplicaciones Glassfish.

```
Linux 2.6.32-71.el6.i686 (localhost.localdomain) 03/07/14 _i686_ (1 CPU)
22:51:19 CPU %user %nice %system %iowait %steal %idle
22:51:20 all 1,01 0,00 1,01 0,00 0,00 97,98
22:51:21 all 8,00 0,00 8,00 0,00 0,00 84,00
22:51:22 all 0,00 0,00 3,00 0,00 0,00 97,00
22:51:23 all 1,98 0,00 1,98 0,00 0,00 96,04
22:51:24 all 2,02 0,00 3,03 0,00 0,00 94,95
22:51:25 all 1,01 0,00 3,03 0,00 0,00 95,96
22:51:26 all 1,00 0,00 3,00 0,00 0,00 96,00
22:51:27 all 1,01 0,00 3,03 0,00 0,00 95,96
22:51:28 all 55,34 0,00 33,01 0,00 0,00 11,65
22:51:29 all 58,16 0,00 33,67 0,00 0,00 8,16
22:51:30 all 61,00 0,00 32,00 0,00 0,00 7,00
22:51:31 all 60,61 0,00 31,31 0,00 0,00 8,08
22:51:32 all 58,42 0,00 35,64 0,00 0,00 5,94
23:13:01 all 5,00 0,00 5,00 0,00 0,00 90,00
23:13:02 all 2,02 0,00 6,06 0,00 0,00 91,92
23:13:03 all 4,00 0,00 5,00 0,00 0,00 91,00
23:13:04 all 2,97 0,00 4,95 0,00 0,00 92,08
23:13:05 all 4,00 0,00 5,00 0,00 0,00 91,00
23:13:06 all 4,00 0,00 4,00 0,00 0,00 92,00
23:13:07 all 5,00 0,00 5,00 0,00 0,00 90,00
Media: all 20,10 0,00 12,09 0,03 0,00 67,77
```

A continuación visualizamos los datos del uso de disco al enviar 20000 solicitudes, en el servidor de aplicaciones Glassfish.

```
Linux 2.6.32-71.el6.i686 (localhost.localdomain) 04/07/14 _i686_ (1 CPU)
22:05:43 DEV tps rd_sec/s wr_sec/s avgrq-sz avgqu-sz await svctm %util
22:05:44 dev11-0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
22:05:44 dev8-0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
22:05:44 dev253-0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
22:05:44 dev253-1 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
22:05:44 DEV tps rd_sec/s wr_sec/s avgrq-sz avgqu-sz await svctm %util
22:05:45 dev11-0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
22:05:45 dev8-0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
22:05:45 dev253-0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
22:05:45 dev253-1 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
22:05:45 DEV tps rd_sec/s wr_sec/s avgrq-sz avgqu-sz await svctm %util
22:05:46 dev11-0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
22:05:46 dev8-0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
22:05:46 dev253-0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
22:05:46 dev253-1 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
22:05:46 DEV tps rd_sec/s wr_sec/s avgrq-sz avgqu-sz await svctm %util
22:05:47 dev11-0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
22:05:47 dev8-0 14.43 0.00 98.97 6.86 0.00 0.29 0.14 0.21
22:05:47 dev253-0 12.37 0.00 98.97 8.00 0.00 0.25 0.25 0.31
22:05:47 dev253-1 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
22:27:31 DEV tps rd_sec/s wr_sec/s avgrq-sz avgqu-sz await svctm %util
22:27:32 dev11-0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
22:27:32 dev8-0 2.02 0.00 24.24 12.00 0.01 3.50 2.00 0.40
22:27:32 dev253-0 3.03 0.00 24.24 8.00 0.01 3.33 1.33 0.40
22:27:32 dev253-1 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
Media: DEV tps rd_sec/s wr_sec/s avgrq-sz avgqu-sz await svctm %util
Media: dev11-0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
Media: dev8-0 3.29 108.35 19.51 38.91 0.00 1.54 1.05 0.34
Media: dev253-0 3.62 108.35 19.51 35.32 0.01 2.12 0.97 0.35
Media: dev253-1 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
```

A continuación visualizamos los datos del uso de disco al enviar 40000 solicitudes, en el servidor de aplicaciones Glassfish.

```
Linux 2.6.32-71.el6.i686 (localhost.localdomain) 04/07/14 _i686_ (1 CPU)
22:30:08 DEV tps rd_sec/s wr_sec/s avgrq-sz avgqu-sz await svctm %util
22:30:09 dev11-0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
22:30:09 dev8-0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
22:30:09 dev253-0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
22:30:09 dev253-1 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
22:30:09 DEV tps rd_sec/s wr_sec/s avgrq-sz avgqu-sz await svctm %util
22:30:10 dev11-0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
22:30:10 dev8-0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
22:30:10 dev253-0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
22:30:10 dev253-1 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
22:30:10 DEV tps rd_sec/s wr_sec/s avgrq-sz avgqu-sz await svctm %util
22:30:11 dev11-0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
22:30:11 dev8-0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
22:30:11 dev253-0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
22:30:11 dev253-1 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
22:51:55 DEV tps rd_sec/s wr_sec/s avgrq-sz avgqu-sz await svctm %util
22:51:56 dev11-0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
22:51:56 dev8-0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
22:51:56 dev253-0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
22:51:56 dev253-1 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
Media: DEV tps rd_sec/s wr_sec/s avgrq-sz avgqu-sz await svctm %util
Media: dev11-0 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
Media: dev8-0 2.02 0.00 15.28 7.58 0.00 0.65 0.35 0.07
Media: dev253-0 1.91 0.00 15.28 7.99 0.00 1.12 0.38 0.07
Media: dev253-1 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
```

A continuación visualizamos los datos del uso de disco al enviar 60000 solicitudes, en el servidor de aplicaciones Glassfish.

```

Linux 2.6.32-71.el6.i686 (localhost.localdomain) 04/07/14 _i686_ (1 CPU)
22:58:06 DEV tps rd_sec/s wr_sec/s avgrq-sz avgqu-sz await svctm %util
22:58:07 dev11-0 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
22:58:07 dev8-0 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
22:58:07 dev253-0 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
22:58:07 dev253-1 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00

22:58:07 DEV tps rd_sec/s wr_sec/s avgrq-sz avgqu-sz await svctm %util
22:58:08 dev11-0 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
22:58:08 dev8-0 13,13 0,00 88,89 6,77 0,00 0,23 0,08 0,10
22:58:08 dev253-0 11,11 0,00 88,89 8,00 0,00 0,27 0,09 0,10
22:58:08 dev253-1 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00

22:58:08 DEV tps rd_sec/s wr_sec/s avgrq-sz avgqu-sz await svctm %util
22:58:09 dev11-0 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
22:58:09 dev8-0 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
22:58:09 dev253-0 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
22:58:09 dev253-1 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00

23:19:53 DEV tps rd_sec/s wr_sec/s avgrq-sz avgqu-sz await svctm %util
23:19:54 dev11-0 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
23:19:54 dev8-0 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
23:19:54 dev253-0 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
23:19:54 dev253-1 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00

Media: DEV tps rd_sec/s wr_sec/s avgrq-sz avgqu-sz await svctm %util
Media: dev11-0 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
Media: dev8-0 2,02 0,00 15,72 7,78 0,00 0,73 0,39 0,08
Media: dev253-0 1,97 0,00 15,72 7,99 0,00 1,28 0,42 0,08
Media: dev253-1 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00

```

A continuación visualizamos los resultados del tiempo de respuesta al reponder 20000 solicitudes, en el servidor de aplicaciones Glassfish.

Al ejecutar el comando:

```

JGPB@localhost:~/Escritorio
Archivo Editar Ver Buscar Terminal Ayuda
[JGPB@localhost Escritorio]$ time java -jar ClienteGlassFish.jar

```

El resultado es el siguiente:

```

real    8m2.682s
user    0m7.620s
sys     0m34.568s
[JGPB@dhcpc2 Escritorio]$

```

Para llevar a una misma unidad de tiempo los minutos se transforman a segundos, obtenido el siguiente resultado

$$TR_{glassFish} 20000 = 482,68seg$$

A continuación visualizamos los resultados del tiempo de respuesta al reponder 40000 solicitudes, en el servidor de aplicaciones Glassfish.

Al ejecutar el comando:



```
JGPB@localhost:~/Escritorio
Archivo Editar Ver Buscar Terminal Ayuda
[JGPB@localhost Escritorio]$ time java -jar ClienteGlassFish.jar
```

El resultado es el siguiente:



```
real    14m13.562s
user    0m14.978s
sys     1m9.522s
[JGPB@dhcpc2 Escritorio]$
```

Para llevar a una misma unidad de tiempo los minutos se transforman a segundos, obtenido el siguiente resultado

$$TR_{glassFish} 40000 = 853,56seg$$

A continuación visualizamos los resultados del tiempo de respuesta al reponder 60000 solicitudes, en el servidor de aplicaciones Glassfish.

Al ejecutar el comando:



```
JGPB@localhost:~/Escritorio
Archivo Editar Ver Buscar Terminal Ayuda
[JGPB@localhost Escritorio]$ time java -jar ClienteGlassFish.jar
```

El resultado es el siguiente:



```
real    20m1.931s
user    0m20.478s
sys     1m29.939s
[JGPB@dhcpc2 Escritorio]$
```

Para llevar a una misma unidad de tiempo los minutos se transforman a segundos, obtenido el siguiente resultado

$$TR_{glassFish} 60000 = 1201,93seg$$

SERVIDOR DE APLICACIÓN JONAS

A continuación visualizamos los datos del uso de RAM al enviar 20000 solicitudes, en el servidor de aplicaciones JONAS.

```
Linux 2.6.32-71.el6.i686 (localhost.localdomain) 02/07/14 _i686_ (1 CPU)

23:59:58      kmemfree kmemused  %memused kbbuffers  kbcached  kbcommit  %commit
23:59:59      251380   779992    75,63    38668     299336   776744    25,09
00:00:00      251372   780000    75,63    38668     299336   786988    25,42
00:00:01      251372   780000    75,63    38668     299336   786988    25,42
00:00:02      251100   780272    75,65    38676     299328   786988    25,42
00:00:03      251116   780256    75,65    38676     299340   786988    25,42
00:00:04      251116   780256    75,65    38676     299340   786988    25,42
00:00:05      251116   780256    75,65    38676     299340   786988    25,42
00:00:06      251100   780272    75,65    38676     299340   786988    25,42
00:00:07      250976   780396    75,67    38676     299340   786988    25,42
00:00:08      250984   780388    75,67    38684     299340   786988    25,42
00:00:09      250896   780536    75,68    38684     299340   786988    25,42
00:21:40      243620   787752    76,38    40620    301884   789936    25,52
00:21:41      243620   787752    76,38    40620    301884   789936    25,52
00:21:42      243620   787752    76,38    40620    301884   789936    25,52
00:21:43      243620   787752    76,38    40620    301884   789936    25,52
00:21:44      243620   787752    76,38    40620    301884   789936    25,52
00:21:45      243620   787752    76,38    40620    301884   789936    25,52
00:21:46      243620   787752    76,38    40628    301884   789936    25,52
Media:        243412   787960    76,40    39648    301434   789955    25,32
```

A continuación s visualizamos los datos del uso de RAM al enviar 40000 solicitudes, en el servidor de aplicaciones JONAS.

```
Linux 2.6.32-71.el6.i686 (localhost.localdomain) 03/07/14 _i686_ (1 CPU)

00:23:08      kmemfree kmemused  %memused kbbuffers  kbcached  kbcommit  %commit
00:23:09      244808   786564    76,26    40676    301892   778620    25,15
00:23:10      244676   786696    76,28    40676    301892   788864    25,48
00:23:11      244676   786696    76,28    40676    301892   788864    25,48
00:23:12      244676   786696    76,28    40676    301892   788864    25,48
00:23:13      244676   786696    76,28    40684    301888   788864    25,48
00:23:14      244676   786696    76,28    40684    301892   788864    25,48
00:23:15      244676   786696    76,28    40684    301892   788864    25,48
00:23:16      244676   786696    76,28    40684    301892   788864    25,48
00:23:17      244676   786696    76,28    40684    301892   788864    25,48
00:23:18      244676   786696    76,28    40684    301892   788864    25,48
00:44:52      236296   795076    77,09    42684    306672   779972    25,19
00:44:53      236296   795076    77,09    42684    306672   779972    25,19
00:44:54      236296   795076    77,09    42684    306672   779972    25,19
00:44:55      236296   795076    77,09    42684    306672   779972    25,19
00:44:56      236296   795076    77,09    42684    306672   779972    25,19
Media:        236235   795137    77,10    41691    305163   784958    25,36
```

A continuación visualizamos los datos del uso de RAM al enviar 60000 solicitudes, en el servidor de aplicaciones JONAS.

```

Linux 2.6.32-71.el6.i686 (localhost.localdomain) 03/07/14 _i686_ (1 CPU)

00:48:31 kbmemfree kbmemused %memused kbbuffers kbcached kbcommmit %commmit
00:48:32 239228 792144 76,80 42852 306676 777012 25,10
00:48:33 239196 792176 76,81 42852 306676 787256 25,43
00:48:34 239220 792152 76,81 42852 306676 787256 25,43
00:48:35 239220 792152 76,81 42852 306676 787256 25,43
00:48:36 239220 792152 76,81 42852 306676 787256 25,43
00:48:37 239220 792152 76,81 42860 306676 787256 25,43
00:48:38 239220 792152 76,81 42860 306676 787256 25,43
00:48:39 239220 792152 76,81 42860 306676 787256 25,43
00:48:40 239220 792152 76,81 42860 306676 787256 25,43
00:48:41 239220 792152 76,81 42860 306676 787256 25,43
00:48:42 239212 792160 76,81 42860 306676 787256 25,43
01:10:16 232724 798648 77,44 44784 309004 778888 25,16
01:10:17 232724 798648 77,44 44784 309004 778888 25,16
01:10:18 232724 798648 77,44 44792 309008 778888 25,16
01:10:19 232724 798648 77,44 44792 309008 778888 25,16
01:10:20 232724 798648 77,44 44792 309008 778888 25,16
Media: 232608 798764 77,45 43824 308604 782930 25,29

```

A continuación visualizamos los datos del uso de CPU al enviar 20000 solicitudes, en el servidor de aplicaciones JOnAS.

```

Linux 2.6.32-71.el6.i686 (localhost.localdomain) 02/07/14 _i686_ (1 CPU)

22:47:13 CPU %user %nice %system %iowait %steal %idle
22:47:14 all 0,99 0,00 1,98 0,00 0,00 97,03
22:47:15 all 4,04 0,00 4,04 0,00 0,00 91,92
22:47:16 all 0,99 0,00 2,97 0,00 0,00 96,04
22:47:17 all 0,00 0,00 3,03 0,00 0,00 96,97
22:47:18 all 0,99 0,00 1,98 0,00 0,00 97,03
22:47:19 all 1,01 0,00 2,02 0,00 0,00 96,97
22:47:20 all 3,00 0,00 4,00 0,00 0,00 93,00
22:47:21 all 0,99 0,00 2,97 0,00 0,00 96,04
22:47:22 all 44,44 0,00 31,31 0,00 0,00 24,24
22:47:23 all 55,00 0,00 42,00 0,00 0,00 3,00
22:47:24 all 59,60 0,00 39,39 0,00 0,00 1,01
22:47:25 all 63,00 0,00 35,00 0,00 0,00 2,00
23:08:56 all 1,00 0,00 2,00 0,00 0,00 97,00
23:08:57 all 1,01 0,00 1,01 0,00 0,00 97,98
23:08:58 all 0,00 0,00 2,02 0,00 0,00 97,98
23:08:59 all 0,97 0,00 4,85 0,00 0,00 94,17
23:09:00 all 3,00 0,00 3,00 0,00 0,00 94,00
23:09:01 all 1,01 0,00 2,02 0,00 0,00 96,97
23:09:02 all 1,98 0,00 1,98 0,00 0,00 96,04
Media: all 18,14 0,00 10,92 0,01 0,00 70,93

```

A continuación visualizamos los datos del uso de CPU al enviar 40000 solicitudes, en el servidor de aplicaciones JOnAS.

```

Linux 2.6.32-71.el6.i686 (localhost.localdomain) 02/07/14 _i686_ (1 CPU)

23:11:03 CPU %user %nice %system %iowait %steal %idle
23:11:04 all 2,00 0,00 4,00 0,00 0,00 94,00
23:11:05 all 6,86 0,00 10,78 0,00 0,00 82,35
23:11:06 all 1,03 0,00 3,09 0,00 0,00 95,88
23:11:07 all 7,00 0,00 9,00 0,00 0,00 84,00
23:11:08 all 1,98 0,00 4,95 0,00 0,00 93,07
23:11:09 all 1,01 0,00 4,04 0,00 0,00 94,95
23:11:10 all 2,00 0,00 3,00 0,00 0,00 95,00
23:11:11 all 1,01 0,00 3,03 0,00 0,00 95,96
23:11:12 all 5,94 0,00 7,92 0,00 0,00 86,14
23:11:13 all 3,03 0,00 5,05 0,00 0,00 91,92
23:11:14 all 1,00 0,00 3,00 0,00 0,00 96,00
23:11:15 all 1,00 0,00 4,00 0,00 0,00 95,00
23:11:16 all 1,02 0,00 4,08 0,00 0,00 94,90
23:11:17 all 3,96 0,00 6,93 0,00 0,00 89,11
23:11:18 all 62,00 0,00 37,00 0,00 0,00 1,00
23:11:19 all 59,60 0,00 38,38 0,00 0,00 2,02
23:11:20 all 59,00 0,00 39,00 0,00 0,00 2,00
23:11:21 all 56,00 0,00 43,00 0,00 0,00 1,00
23:11:22 all 60,00 0,00 38,00 0,00 0,00 2,00
23:11:23 all 62,63 0,00 37,37 0,00 0,00 0,00
23:11:24 all 57,43 0,00 41,58 0,00 0,00 0,99
23:11:25 all 61,00 0,00 38,00 0,00 0,00 1,00
23:32:48 all 4,00 0,00 3,00 0,00 0,00 93,00
23:32:49 all 3,00 0,00 3,00 0,00 0,00 94,00
23:32:50 all 3,03 0,00 3,03 0,00 0,00 93,94
23:32:51 all 6,06 0,00 3,03 0,00 0,00 90,91
Media: all 33,87 0,00 19,27 0,00 0,00 46,86

```

A continuación visualizamos los datos del uso de CPU al enviar 60000 solicitudes, en el servidor de aplicaciones JOnAS.

```
Linux 2.6.32-71.el6.i686 (localhost.localdomain) 02/07/14 _i686_ (1 CPU)

23:35:35      CPU      %user      %nice      %system      %iowait      %steal      %idle
23:35:36      all        1,01        0,00        1,01        0,00        0,00        97,98
23:35:37      all        4,95        0,00        7,92        0,00        0,00        87,13
23:35:38      all        1,00        0,00        2,00        0,00        0,00        97,00
23:35:39      all        1,00        0,00        4,00        0,00        0,00        95,00
23:35:40      all        2,00        0,00        3,00        0,00        0,00        95,00
23:35:41      all        2,02        0,00        3,03        0,00        0,00        94,95
23:35:42      all        1,00        0,00        3,00        0,00        0,00        96,00
23:35:43      all        2,02        0,00        4,04        0,00        0,00        93,94
23:35:44      all        1,01        0,00        3,03        0,00        0,00        95,96
23:35:45      all        2,94        0,00        3,92        0,00        0,00        93,14
23:35:46      all        1,01        0,00        3,03        0,00        0,00        95,96
23:35:47      all        1,00        0,00        2,00        0,00        0,00        97,00
23:35:48      all        1,00        0,00        2,00        0,00        0,00        97,00
23:35:49      all        0,00        0,00        2,04        0,00        0,00        97,96
23:35:50      all        1,00        0,00        2,00        0,00        0,00        97,00
23:35:51      all        1,96        0,00        3,92        0,00        0,00        94,12
23:35:52      all        0,00        0,00        2,02        0,00        0,00        97,98
23:35:53      all        1,01        0,00        2,02        0,00        0,00        96,97
23:35:54      all        1,00        0,00        3,00        0,00        0,00        96,00
23:35:55      all        0,00        0,00        3,00        0,00        0,00        97,00
23:35:56      all        48,48        0,00        32,32        0,00        0,00        19,19
23:35:57      all        65,35        0,00        34,65        0,00        0,00        0,00
23:35:58      all        64,00        0,00        35,00        0,00        0,00        1,00
23:35:59      all        63,00        0,00        36,00        0,00        0,00        1,00
23:57:22      all        2,00        0,00        5,00        0,00        0,00        93,00
23:57:23      all        0,00        0,00        3,06        0,00        0,00        96,94
23:57:24      all        0,99        0,00        1,98        0,00        0,00        97,03
Media:         all        49,75        0,00        26,95        0,00        0,00        23,29
```

A continuación visualizamos los datos del uso de disco al enviar 20000 solicitudes, en el servidor de aplicaciones JOnAS.

```
Linux 2.6.32-71.el6.i686 (localhost.localdomain) 03/07/14 _i686_ (1 CPU)

20:11:35      DEV      tps  rd_sec/s  wr_sec/s  avgrq-sz  avgqu-sz  await  svctm  %util
20:11:36      dev8-0   0,00    0,00    0,00    0,00    0,00    0,00    0,00    0,00
20:11:36      dev253-0 0,00    0,00    0,00    0,00    0,00    0,00    0,00    0,00
20:11:36      dev253-1 0,00    0,00    0,00    0,00    0,00    0,00    0,00    0,00

20:13:59      DEV      tps  rd_sec/s  wr_sec/s  avgrq-sz  avgqu-sz  await  svctm  %util
20:14:00      dev8-0   21,78    0,00   190,10    8,73    0,01    0,55    0,23    0,50
20:14:00      dev253-0 23,76    0,00   190,10    8,00    0,02    0,79    0,21    0,50
20:14:00      dev253-1 0,00    0,00    0,00    0,00    0,00    0,00    0,00    0,00

Media:         DEV      tps  rd_sec/s  wr_sec/s  avgrq-sz  avgqu-sz  await  svctm  %util
Media:         dev8-0   2,14    0,25   19,70    9,34    0,00    0,84    0,44    0,09
Media:         dev253-0 2,47    0,25   19,70    8,07    0,00    1,84    0,39    0,10
Media:         dev253-1 0,00    0,00    0,00    0,00    0,00    0,00    0,00    0,00
```

A continuación visualizamos los datos del uso de disco al enviar 40000 solicitudes, en el servidor de aplicaciones JOnAS.

```
Linux 2.6.32-71.el6.i686 (localhost.localdomain) 03/07/14 _i686_ (1 CPU)

20:35:12 DEV tps rd_sec/s wr_sec/s avgrq-sz avgqu-sz await svctm %util
20:35:13 dev8-0 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
20:35:13 dev253-0 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
20:35:13 dev253-1 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
20:55:48 DEV tps rd_sec/s wr_sec/s avgrq-sz avgqu-sz await svctm %util
20:55:49 dev8-0 11,88 0,00 79,21 6,67 0,01 0,58 0,25 0,30
20:55:49 dev253-0 9,90 0,00 79,21 8,00 0,01 0,70 0,30 0,30
20:55:49 dev253-1 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
Media: DEV tps rd_sec/s wr_sec/s avgrq-sz avgqu-sz await svctm %util
Media: dev8-0 2,12 0,02 22,84 10,77 0,00 0,62 0,37 0,08
Media: dev253-0 2,86 0,02 22,84 8,00 0,01 2,41 0,28 0,08
Media: dev253-1 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
```

A continuación visualizamos los datos del uso de disco al enviar 60000 solicitudes, en el servidor de aplicaciones JOnAS.

```
Linux 2.6.32-71.el6.i686 (localhost.localdomain) 03/07/14 _i686_ (1 CPU)

21:09:52 DEV tps rd_sec/s wr_sec/s avgrq-sz avgqu-sz await svctm %util
21:09:53 dev8-0 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
21:09:53 dev253-0 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
21:09:53 dev253-1 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
21:15:13 DEV tps rd_sec/s wr_sec/s avgrq-sz avgqu-sz await svctm %util
21:15:14 dev8-0 15,84 0,00 110,89 7,00 0,01 0,88 0,31 0,50
21:15:14 dev253-0 13,86 0,00 110,89 8,00 0,01 1,00 0,36 0,50
21:15:14 dev253-1 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
Media: DEV tps rd_sec/s wr_sec/s avgrq-sz avgqu-sz await svctm %util
Media: dev8-0 2,17 0,00 26,81 12,34 0,00 0,60 0,35 0,08
Media: dev253-0 3,35 0,00 26,81 7,99 0,01 2,56 0,23 0,08
Media: dev253-1 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
```

A continuación visualizamos los resultados del tiempo de respuesta al responder 20000 solicitudes, en el servidor de aplicaciones JOnAS.

Al ejecutar el comando:

```
JGPB@localhost:~/Escritorio
Archivo Editar Ver Buscar Terminal Ayuda
[JGPB@localhost Escritorio]$ time java -jar WSJonasClient.jar
```

El resultado es el siguiente:

```
[JGPB@localhost Escritorio]$
real    12m12.974s
user    0m6.849s
sys     0m47.710s
[JGPB@localhost Escritorio]$
```

Para llevar a una misma unidad de tiempo los minutos se transforman a segundos, obtenido el siguiente resultado

$$TR_{jonas} 20000 = 732.97seg$$

A continuación visualizamos los resultados del tiempo de respuesta al reponder 40000 solicitudes, en el servidor de aplicaciones JOnAS.

Al ejecutar el comando:

A terminal window titled 'JGPB@localhost:~/Escritorio' with a menu bar containing 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The command prompt shows '[JGPB@localhost Escritorio]\$ time java -jar WSJonasClient.jar' followed by a cursor.

```
JGPB@localhost:~/Escritorio
Archivo Editar Ver Buscar Terminal Ayuda
[JGPB@localhost Escritorio]$ time java -jar WSJonasClient.jar
```

El resultado es el siguiente:

A terminal window showing the output of the 'time' command. The output lists 'real' time as 22m57.425s, 'user' time as 0m12.019s, and 'sys' time as 1m32.057s. The prompt is '[JGPB@dhcpc2 Escritorio]\$' followed by a cursor.

```
real    22m57.425s
user    0m12.019s
sys     1m32.057s
[JGPB@dhcpc2 Escritorio]$
```

Para llevar a una misma unidad de tiempo los minutos se transforman a segundos, obtenido el siguiente resultado

$$TR_{jonas} 40000 = 1377,42seg$$

A continuación visualizamos los resultados del tiempo de respuesta al reponder 60000 solicitudes, en el servidor de aplicaciones JOnAS.

Al ejecutar el comando:

```
JGPB@localhost:~/Escritorio
Archivo Editar Ver Buscar Terminal Ayuda
[JGPB@localhost Escritorio]$ time java -jar WSJonasClient.jar
```

El resultado es el siguiente:

```
real    30m54.891s
user    0m14.729s
sys     1m58.327s
[JGPB@dhcp2 Escritorio]$
```

Para llevar a una misma unidad de tiempo los minutos se transforman a segundos, obtenido el siguiente resultado

$$TR_{jonas} 60000 = 1854,89seg$$

SERVIDOR DE APLICACIÓN GERÓNIMO

A continuación visualizamos los datos del uso de RAM al enviar 20000 solicitudes, en el servidor de aplicaciones Gerónimo.

```
Linux 2.6.32-71.el6.i686 (localhost.localdomain) 06/07/14 _i686_ (1 CPU)
```

Time	kbmemfree	kbmemused	%memused	kbbuffers	kbcached	kbcommit	%commit
20:14:21	40976	990396	96,03	37980	319060	1080528	34,90
20:14:22	40944	990428	96,03	37980	319060	1090772	35,23
20:14:23	40944	990428	96,03	37980	319060	1090772	35,23
20:14:24	40992	990380	96,03	37980	319060	1090772	35,23
20:14:25	40992	990380	96,03	37980	319060	1090772	35,23
20:14:26	40992	990380	96,03	37980	319060	1090772	35,23
20:14:27	40992	990380	96,03	37988	319060	1090772	35,23
20:14:28	40992	990380	96,03	37988	319060	1090772	35,23
20:14:29	40992	990380	96,03	37988	319060	1090772	35,23
20:14:30	40992	990380	96,03	37988	319060	1090772	35,23
20:14:31	40992	990380	96,03	37988	319060	1090772	35,23
20:36:06	33884	997488	96,71	39904	321540	1083032	34,98
20:36:07	33884	997488	96,71	39904	321540	1083032	34,98
20:36:08	33884	997488	96,71	39904	321540	1083032	34,98
20:36:09	33884	997488	96,71	39904	321540	1083032	34,98
20:36:10	33884	997488	96,71	39912	321536	1083032	34,98
Media:	35118	996254	96,60	38941	321307	1087116	35,12

A continuación visualizamos los datos del uso de RAM al enviar 40000 solicitudes, en el servidor de aplicaciones Gerónimo.

```
Linux 2.6.32-71.el6.i686 (localhost.localdomain) 06/07/14 _i686_ (1 CPU)

20:41:09 ktmemfree ktmemused %memused kbbuffers kbcached kbccommit %ccommit
20:41:10 34816 996556 96,62 40080 321636 1082104 34,95
20:41:11 34536 996896 96,65 40080 321636 1092348 35,29
20:41:12 34560 996812 96,65 40080 321636 1092348 35,29
20:41:13 34560 996812 96,65 40080 321636 1092348 35,29
20:41:14 34568 996804 96,65 40080 321636 1092348 35,29
20:41:15 34568 996804 96,65 40088 321636 1092348 35,29
20:41:16 34568 996804 96,65 40088 321636 1092348 35,29
20:41:17 34568 996804 96,65 40088 321636 1092348 35,29
20:41:18 33808 997564 96,72 40096 321628 1093264 35,32
21:02:52 25296 1006076 97,55 55184 295080 1081892 34,95
21:02:53 25332 1006040 97,54 55184 295080 1081892 34,95
21:02:54 25304 1006068 97,55 55184 295084 1081880 34,95
21:02:55 25304 1006068 97,55 55184 295084 1081880 34,95
21:02:56 25304 1006068 97,55 55184 295084 1081880 34,95
21:02:57 25304 1006068 97,55 55192 295076 1081880 34,95
Media: 22390 1008982 97,83 53784 299224 1085857 35,08
```

A continuación visualizamos los datos del uso de RAM al enviar 60000 solicitudes, en el servidor de aplicaciones Gerónimo.

```
Linux 2.6.32-71.el6.i686 (localhost.localdomain) 06/07/14 _i686_ (1 CPU)

21:28:30 ktmemfree ktmemused %memused kbbuffers kbcached kbccommit %ccommit
21:28:31 38116 993256 96,30 56540 283356 1079760 34,88
21:28:32 38092 993280 96,31 56548 283356 1090004 35,21
21:28:33 38116 993256 96,30 56548 283356 1090004 35,21
21:28:34 38124 993248 96,30 56548 283356 1090004 35,21
21:28:35 38124 993248 96,30 56548 283356 1090004 35,21
21:28:36 38124 993248 96,30 56548 283356 1090004 35,21
21:28:37 38148 993224 96,30 56548 283356 1090004 35,21
21:28:38 38148 993224 96,30 56556 283356 1090004 35,21
21:28:39 38148 993224 96,30 56556 283356 1090004 35,21
21:28:40 38148 993224 96,30 56556 283356 1090004 35,21
21:28:41 38148 993224 96,30 56556 283356 1090004 35,21
21:28:42 38148 993224 96,30 56556 283356 1090004 35,21
21:50:13 33920 997452 96,71 57908 283416 1092828 35,30
21:50:14 33920 997452 96,71 57908 283416 1092828 35,30
21:50:15 33920 997452 96,71 57908 283416 1092828 35,30
21:50:16 33920 997452 96,71 57908 283416 1092828 35,30
21:50:17 33920 997452 96,71 57908 283416 1092828 35,30
21:50:18 33920 997452 96,71 57908 283416 1092828 35,30
Media: 29463 1001909 97,14 57073 283681 1087553 35,13
```

A continuación visualizamos los datos del uso de CPU al enviar 20000 solicitudes, en el servidor de aplicaciones Gerónimo.

```
Linux 2.6.32-71.el6.i686 (localhost.localdomain) 04/07/14 _i686_ (1 CPU)

23:35:41 CPU %user %nice %system %iowait %steal %idle
23:35:42 all 36,63 0,00 4,95 0,00 0,00 58,42
23:35:43 all 21,00 0,00 5,00 0,00 0,00 74,00
23:35:44 all 34,65 0,00 8,91 0,00 0,00 56,44
23:35:45 all 47,47 0,00 12,12 0,00 0,00 40,40
23:35:46 all 55,56 0,00 14,14 0,00 0,00 30,30
23:35:47 all 25,00 0,00 8,00 0,00 0,00 67,00
23:35:48 all 42,00 0,00 15,00 0,00 0,00 43,00
23:35:49 all 29,29 0,00 9,09 0,00 0,00 61,62
23:35:50 all 36,36 0,00 4,04 0,00 0,00 59,60
23:35:51 all 37,62 0,00 5,94 0,00 0,00 56,44
23:35:52 all 34,00 0,00 4,00 0,00 0,00 62,00
23:35:53 all 23,00 0,00 4,00 0,00 0,00 73,00
23:35:54 all 30,00 0,00 2,00 0,00 0,00 68,00
23:57:24 all 15,31 0,00 2,04 0,00 0,00 82,65
23:57:25 all 28,00 0,00 4,00 0,00 0,00 68,00
23:57:26 all 36,00 0,00 5,00 0,00 0,00 59,00
23:57:27 all 37,00 0,00 4,00 0,00 0,00 59,00
23:57:28 all 32,00 0,00 4,00 0,00 0,00 64,00
23:57:29 all 16,00 0,00 4,00 0,00 0,00 80,00
23:57:30 all 38,38 0,00 3,03 0,00 0,00 58,59
Media: all 32,98 0,00 7,18 0,02 0,00 59,82
```

A continuación visualizamos los datos del uso de CPU al enviar 40000 solicitudes, en el servidor de aplicaciones Gerónimo.

```
Linux 2.6.32-71.el6.i686 (localhost.localdomain) 04/07/14 _i686_ (1
CPU)

23:59:26 CPU %user %nice %system %iowait %steal %idle
23:59:27 all 16,16 0,00 4,04 0,00 0,00 79,80
23:59:28 all 27,00 0,00 11,00 0,00 0,00 62,00
23:59:29 all 38,00 0,00 4,00 0,00 0,00 58,00
23:59:30 all 36,36 0,00 5,05 0,00 0,00 58,59
23:59:31 all 36,63 0,00 5,94 0,00 0,00 57,43
23:59:32 all 16,16 0,00 5,05 0,00 0,00 78,79
23:59:33 all 35,00 0,00 4,00 0,00 0,00 61,00
23:59:34 all 40,40 0,00 4,04 0,00 0,00 55,56
00:21:08 all 31,68 0,00 4,95 0,00 0,00 63,37
00:21:09 all 33,33 0,00 3,03 0,00 0,00 63,64
00:21:10 all 14,00 0,00 3,00 0,00 0,00 83,00
00:21:11 all 24,00 0,00 4,00 0,00 0,00 72,00
00:21:12 all 32,00 0,00 2,00 0,00 0,00 66,00
00:21:13 all 39,39 0,00 5,05 0,00 0,00 55,56
00:21:14 all 7,92 0,00 2,97 0,00 0,00 89,11
00:21:15 all 30,00 0,00 3,00 0,00 0,00 67,00
Media: all 33,15 0,00 9,83 0,01 0,00 57,01
```

A continuación visualizamos los datos del uso de CPU al enviar 60000 solicitudes, en el servidor de aplicaciones Gerónimo.

```
Linux 2.6.32-71.el6.i686 (localhost.localdomain) 06/07/14 _i686_ (1
CPU)

21:55:27 CPU %user %nice %system %iowait %steal %idle
21:55:28 all 19,39 0,00 4,08 0,00 0,00 76,53
21:55:29 all 43,56 0,00 8,91 0,00 0,00 47,52
21:55:30 all 39,39 0,00 5,05 0,00 0,00 55,56
21:55:31 all 17,82 0,00 5,94 0,00 0,00 76,24
21:55:32 all 34,34 0,00 4,04 0,00 0,00 61,62
21:55:33 all 36,36 0,00 4,04 0,00 0,00 59,60
21:55:34 all 20,00 0,00 4,00 0,00 0,00 76,00
21:55:35 all 34,00 0,00 3,00 0,00 0,00 63,00
21:55:36 all 37,37 0,00 5,05 0,00 0,00 57,58
21:55:37 all 24,51 0,00 5,88 0,00 0,00 69,61
21:55:38 all 23,23 0,00 3,03 0,00 0,00 73,74
22:17:09 all 41,00 0,00 5,00 0,00 0,00 54,00
22:17:10 all 23,47 0,00 5,10 0,00 0,00 71,43
22:17:11 all 38,61 0,00 4,95 0,00 0,00 56,44
22:17:12 all 44,55 0,00 6,93 0,00 0,00 48,51
22:17:13 all 38,38 0,00 6,06 0,00 0,00 55,56
22:17:14 all 26,73 0,00 5,94 0,00 0,00 67,33
22:17:15 all 37,37 0,00 4,04 0,00 0,00 58,59
Media: all 36,52 0,00 14,49 0,19 0,00 48,80
```

A continuación visualizamos los datos del uso de disco al enviar 20000 solicitudes, en el servidor de aplicaciones Gerónimo.

```
Linux 2.6.32-71.el6.i686 (localhost.localdomain) 06/07/14 _i686_ (1 CPU)

22:20:58 DEV tps rd_sec/s wr_sec/s avgrq-sz avgqu-sz await svctm %util
22:20:59 dev8-0 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
22:20:59 dev253-0 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
22:20:59 dev253-1 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00

22:20:59 DEV tps rd_sec/s wr_sec/s avgrq-sz avgqu-sz await svctm %util
22:21:00 dev8-0 10,89 0,00 71,29 6,55 0,00 0,09 0,09 0,10
22:21:00 dev253-0 8,91 0,00 71,29 8,00 0,00 0,11 0,11 0,10
22:21:00 dev253-1 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00

Media: DEV tps rd_sec/s wr_sec/s avgrq-sz avgqu-sz await svctm %util
Media: dev8-0 2,07 0,59 19,14 9,51 0,00 0,82 0,43 0,09
Media: dev253-0 2,41 0,59 19,14 8,19 0,01 2,87 0,39 0,09
Media: dev253-1 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
```

A continuación visualizamos los datos del uso de disco al enviar 40000 solicitudes, en el servidor de aplicaciones Gerónimo.

```
Linux 2.6.32-71.el6.i686 (localhost.localdomain) 06/07/14 _i686_ (1 CPU)

22:44:14 DEV tps rd_sec/s wr_sec/s avgrq-sz avgqu-sz await svctm %util
22:44:15 dev8-0 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
22:44:15 dev253-0 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
22:44:15 dev253-1 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00

22:50:56 DEV tps rd_sec/s wr_sec/s avgrq-sz avgqu-sz await svctm %util
22:50:57 dev8-0 23,00 0,00 208,00 9,04 0,02 0,87 0,35 0,80
22:50:57 dev253-0 26,00 0,00 208,00 8,00 0,03 1,08 0,27 0,70
22:50:57 dev253-1 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00

Media: DEV tps rd_sec/s wr_sec/s avgrq-sz avgqu-sz await svctm %util
Media: dev8-0 2,13 0,00 23,03 10,79 0,00 0,83 0,44 0,09
Media: dev253-0 2,88 0,00 23,03 7,99 0,01 4,44 0,34 0,10
Media: dev253-1 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
```

A continuación visualizamos los datos del uso de disco al enviar 60000 solicitudes, en el servidor de aplicaciones Gerónimo.

```
Linux 2.6.32-71.el6.i686 (localhost.localdomain) 06/07/14 _i686_ (1 CPU)

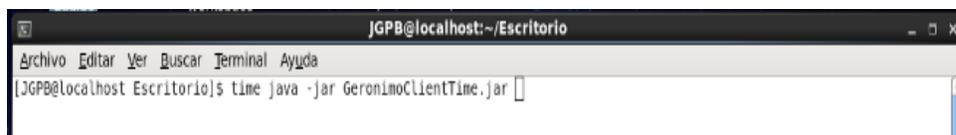
23:07:48 DEV tps rd_sec/s wr_sec/s avgrq-sz avgqu-sz await svctm %util
23:07:49 dev8-0 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
23:07:49 dev253-0 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
23:07:49 dev253-1 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00

23:19:26 DEV tps rd_sec/s wr_sec/s avgrq-sz avgqu-sz await svctm %util
23:19:27 dev8-0 10,00 0,00 64,00 6,40 0,00 0,20 0,10 0,10
23:19:27 dev253-0 8,00 0,00 64,00 8,00 0,00 0,25 0,25 0,20
23:19:27 dev253-1 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00

Media: DEV tps rd_sec/s wr_sec/s avgrq-sz avgqu-sz await svctm %util
Media: dev8-0 2,33 6,15 29,29 15,24 0,00 0,88 0,47 0,11
Media: dev253-0 3,78 6,15 29,29 9,39 0,02 4,50 0,31 0,12
Media: dev253-1 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00 0,00
```

A continuación visualizamos los resultados del tiempo de respuesta al responder 20000 solicitudes, en el servidor de aplicaciones Gerónimo.

Al ejecutar el comando:



```
JGPB@localhost:~/Escritorio
Archivo Editar Ver Buscar Terminal Ayuda
[JGPB@localhost Escritorio] time java -jar GeronimoClientTime.jar
```

El resultado es el siguiente:

```
real 14m45.459s
user 0m49.772s
sys 3m55.738s
[JGPB@hcppc2 Escritorio]
```

Para llevar a una misma unidad de tiempo los minutos se transforman a segundos, obtenido el siguiente resultado

$$TR_{\text{gerónimo}} 20000 = 885,45\text{seg}$$

A continuación visualizamos los resultados del tiempo de respuesta al reponder 40000 solicitudes, en el servidor de aplicaciones Gerónimo.

Al ejecutar el comando:



```
JGPB@localhost:~/Escritorio
Archivo Editar Ver Buscar Terminal Ayuda
[JGPB@localhost Escritorio]$ time java -jar GeronimoClientTime.jar
```

El resultado es el siguiente:

```
real 26m48.163s
user 3m30.420s
sys 4m33.470s
[JGPB@dhcpc2 dist]$
```

Para llevar a una misma unidad de tiempo los minutos se transforman a segundos, obtenido el siguiente resultado

$$TR_{\text{gerónimo}} 40000 = 1608,16\text{seg}$$

A continuación visualizamos los resultados del tiempo de respuesta al reponder 60000 solicitudes, en el servidor de aplicaciones Gerónimo.

Al ejecutar el comando:



```
JGPB@localhost:~/Escritorio
Archivo Editar Ver Buscar Terminal Ayuda
[JGPB@localhost Escritorio]$ time java -jar GeronimoClientTime.jar
```

El resultado es el siguiente:



```
real    34m30.127s
user    4m27.629s
sys     6m44.541s
[JGPB@dhcpc2 dist]$
```

Para llevar a una misma unidad de tiempo los minutos se transforman a segundos, obtenido el siguiente resultado

$$TR_{gerónimo} 60000 = 2074,12seg$$

Sección B

Desarrollo de Cálculos Probabilísticos

En el sección B se muestra las fórmulas aplicadas y los valores que se establecieron para tomar la decisiones.

Los valores que se van a encontrar son: media, varianza, desviación estándar e intervalo de confianza para cada una de las pruebas con una cantidad de carga de 40000 solicitudes, de los datos conocidos tenemos:

Nivel de confianza = 97%

α =Margen de error = 0.03%

z = para el nivel de confianza (97%)=2.17

p= frecuencia esperada

q=1-p

Tabla del valor de Z

Nivel de Confianza	Z
99%	2.58
98%	2.33
97%	2.17
96%	2.05
95%	1.96
90%	1.65
80%	1.28
50%	0.67

Cálculo de la muestra para cuando no se conoce el tamaño de la población

$$n = \frac{z^2 pq}{\alpha^2}$$

$$n = \frac{2,17^2 \cdot 0,5 \cdot 0,5}{(0,03)^2}$$

$$n = \frac{1,177225}{0,0009}$$

$$n = 1308$$

Fórmula de la Media

$$x = \frac{\sum_{i=1}^n x_i}{n}$$

Fórmula de la Varianza

$$s^2 = \frac{\sum_{i=1}^n (x_i - x)^2}{n}$$

Fórmula de la Desviación Estándar

$$s = \sqrt{s^2}$$

Coefficiente de Pearson – Fórmula para calcular t

$$t = \frac{r - 0}{\frac{1 - r^2}{N - 2}}$$

Donde:

r = coeficiente de Pearson

N = tamaño de la muestra

Tabla de Coeficiente de Pearson

α	0,10	0,05	0,025	0,01	0,005	0,0025	0,001
1 - α	0,90	0,95	0,975	0,99	0,995	0,9975	0,999
v	$t_{0.90}$	$t_{0.95}$	$t_{0.975}$	$t_{0.99}$	$t_{0.995}$	$t_{0.9975}$	$t_{0.999}$
1	3,078	6,314	12,71	31,82	63,660	127,3	318,3
2	1,886	2,920	4,303	6,965	9,925	14,09	22,33
3	1,638	2,353	3,182	4,541	5,541	7,453	10,21
4	1,533	2,132	2,776	3,747	4,604	5,598	7,173
5	1,476	2,015	2,571	3,365	4,032	4,773	5,893
6	1,440	1,943	2,447	3,143	3,707	4,317	5,208
7	1,415	1,895	2,365	2,998	3,499	4,029	4,785
8	1,397	1,860	2,306	2,896	3,335	3,833	4,501
9	1,383	1,833	2,262	2,821	3,250	3,690	4,297
10	1,372	1,812	2,228	2,764	3,169	3,581	4,144
11	1,363	1,796	2,201	2,718	3,106	3,497	4,025
12	1,356	1,782	2,179	2,681	3,055	3,428	3,930
13	1,350	1,771	2,160	2,650	3,012	3,372	3,852
14	1,345	1,761	2,145	2,624	2,977	3,326	3,787
15	1,341	1,753	2,131	2,602	2,947	3,286	3,733
16	1,337	1,746	2,120	2,583	2,921	3,252	3,686
17	1,333	1,740	2,110	2,567	2,898	3,222	3,646
18	1,330	1,734	2,101	2,552	2,878	3,197	3,610
19	1,328	1,729	2,093	2,539	2,861	3,174	3,579
20	1,325	1,725	2,086	2,528	2,845	3,153	3,552
21	1,323	1,721	2,080	2,518	2,831	3,135	3,527
22	1,321	1,717	2,074	2,508	2,819	3,119	3,505
23	1,319	1,714	2,069	2,500	2,807	3,104	3,485
24	1,318	1,711	2,064	2,492	2,797	3,091	3,467
25	1,316	1,708	2,060	2,485	2,787	3,078	3,450
26	1,315	1,706	2,056	2,479	2,779	3,067	3,435
27	1,314	1,703	2,052	2,473	2,771	3,057	3,421
28	1,313	1,701	2,048	2,467	2,763	3,047	3,408
29	1,311	1,699	2,045	2,462	2,756	3,038	3,396
30	1,310	1,697	2,042	2,457	2,750	3,030	3,385
40	1,303	1,684	2,021	2,423	2,704	2,971	3,307
50	1,299	1,676	2,009	2,403	2,678	2,937	3,261
60	1,296	1,671	2,000	2,390	2,660	2,915	3,232
70	1,29	1,667	1,994	2,381	2,648	2,899	3,211
80	1,292	1,664	1,990	2,374	2,639	2,887	3,195
90	1,291	1,662	1,987	2,368	2,632	2,878	3,183
100	1,290	1,660	1,984	2,364	2,626	2,871	3,174
200	1,286	1,653	1,972	2,345	2,601	2,838	3,131
500	1,283	1,648	1,965	2,334	2,586	2,820	3,107
∞	1,282	1,645	1,960	2,326	2,576	2,807	3,090

PONDERACIONES

En la tabla siguiente se detalla los cálculos para la asignación de los porcentajes reales de acuerdo al peso de cada parámetro de comparación. Donde los valores más bajos obtendrán el peso máximo de la ponderación.

	GlassFish	JOnAS	Gerónimo
% Uso de RAM	57,76	77,10	97,83
% Uso de CPU	13,62	33,87	33,15
% Uso de Disco	0,07	0,08	0,10
Tiempo de respuesta	853,56seg	1377,42seg	1608.16seg

Uso de RAM

El menor porcentaje alcanzado es de 57,76% correspondiente al servidor de aplicaciones GlassFish por lo tanto obtendrá el valor total del peso asignado al parámetro uso de RAM, en este caso 20% y aplicamos una regla de tres invertida para la obtener los porcentajes de los servidores JOnAS y Gerónimo.

JOnAS

Gerónimo

57,76 → 20%

57,76 → 20%

$$77,10 \rightarrow x$$

$$97,83 \rightarrow x$$

$$\frac{77,10}{57,76} = \frac{20}{x}$$

$$\frac{97,83}{57,76} = \frac{20}{x}$$

$$x = \frac{57,76 * 20}{77,10}$$

$$x = \frac{97,83 * 20}{57,76}$$

$$x = 14,93$$

$$x = 11,81$$

Uso de CPU

El menor porcentaje alcanzado es 13,62% correspondiente al servidor de aplicaciones GlassFish por lo tanto obtendrá el valor total del peso asignado al parámetro uso de CPU, en este caso 25% y aplicamos una regla de tres invertida para la obtener los porcentajes de los servidores JOnAS y Gerónimo.

JOnAS

Gerónimo

$$13,62 \rightarrow 25\%$$

$$13,62 \rightarrow 25\%$$

$$33,87 \rightarrow x$$

$$33,15 \rightarrow x$$

$$\frac{33,87}{13,62} = \frac{25}{x}$$

$$\frac{33,15}{13,62} = \frac{25}{x}$$

$$x = \frac{13,62 * 25}{33,87}$$

$$x = \frac{13,62 * 25}{33,15}$$

$$x = 10,05$$

$$x = 10,27$$

Uso de Disco

El menor porcentaje alcanzo el 0,07% correspondiente al servidor de aplicaciones GlassFish, pero al realizar el análisis estadístico de observo que no existe una diferencia significativa entre los tres servidores de aplicación por lo que el total del peso asignado al parámetro uso de disco en este caso del 10% será el mismo para GlassFih, JOnAS y Gerónimo.

Tiempo de Respuesta

El menor valor alcanzó es 853,56seg correspondiente al servidor de aplicaciones GlassFis, por lo tanto obtendrá el valor total del peso asignado al parámetro uso de disco, en este caso 45% y aplicamos una regla de tres invertida para la obtención de los porcentajes de los servidores JOnAS y Gerónimo.

JOnAS

Gerónimo

$$853,56 \rightarrow 45\%$$

$$853,56 \rightarrow 45\%$$

$$1377,42 \rightarrow x$$

$$1608,16 \rightarrow x$$

$$\frac{1377,42}{853,56} = \frac{45}{x}$$

$$\frac{1608,16}{853,56} = \frac{45}{x}$$

$$x = \frac{853,56 * 45}{1377,42}$$

$$x = \frac{853,56 * 45}{1608,16}$$

$$x = 27,88$$

$$x = 23,88$$

Sección C

Historias de usuario

A continuación se detallan las historias de usuario, tarea de ingeniería, interfaz grafica y pruebas realizadas en el desarrollo del sistema:

Iteración 1

Metáfora 1

Como consecuencia de las historias escritas por el usuario se redactó la metáfora de diseño de base datos la cual se detalla en la tabla adjunta.

Diseño de esquema de base de datos para el control de sistemas

Historia de usuario	
Número: M1	Nombre historia: Diseño de esquema de base de datos para control de sistemas
Usuario: Programador	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimado: 20
Riesgo en Desarrollo: Medio	Puntos Reales: 25
Descripción: Como programador quiero generar el diagrama DER de base de datos, normalización de tablas, diagrama lógico, diagrama físico, proyección de crecimiento de la base de datos, script de despliegue y diccionario de datos, para obtener un producto adecuado a las especificaciones del cliente.	

Cronograma de tareas de ingeniería

Para cumplir los tiempos para el entregable en la fecha establecida, se definió con los programadores el cronograma de entregas de cada tarea de ingeniería como se detalla en la tabla adjunta.

Cronograma de tareas de ingeniería metáfora 1

Tarea de ingeniería (código y nombre)		Puntos estimados	Puntos reales	Fecha inicio	Fecha fin
A-1	Diagrama entidad relación DER	6	8	10/03/2014	11/03/2014
A-2	Normalización del diagrama entidad relación	4	6	12/03/2014	13/03/2014
A-3	Diseño lógico	2	2	13/03/2014	13/03/2014
A-4	Diseño físico, diccionario de datos, script de despliegue	4	4	14/03/2014	14/03/2014
A-5	Proyección de crecimiento de base de datos.	4	5	17/03/2014	18/03/2014

Tarea de ingeniería A-1

Tarea de Ingeniería	
Número Tarea: A-1	Historia de Usuario: Diseño de esquema de base de datos para control de sistemas.
Nombre Tarea: Diagrama entidad relación DER	
Tipo Tarea: Creación	Puntos estimados: 6
Fecha Inicio: 10/03/2014	Fecha Fin: 11/03/2014
Analista Responsable: José Pinto	
Descripción: se realizó el esquema entidad relación DER	

Tarea de ingeniería A-2

Tarea de Ingeniería

Número Tarea: A-2	Historia de Usuario: Diseño de esquema de base de datos para control de sistemas.	
Nombre Tarea: Normalización del diagrama entidad relación		
Tipo Tarea: Creación		Puntos estimados: 4
Fecha Inicio: 12/03/2014		Fecha Fin: 13/03/2014
Analista Responsable: José Pinto		
Descripción: Realizar la normalización sobre el diagrama entidad relación hasta la 3FN		

Tarea de ingeniería A-3

Tarea de Ingeniería		
Número Tarea: A-3	Historia de Usuario: Diseño de esquema de base de datos para control de sistemas.	
Nombre Tarea: Diseño lógico		
Tipo Tarea: Creación		Puntos estimados: 2
Fecha Inicio: 13/03/2014		Fecha Fin: 13/03/2014
Analista Responsable: José Pinto		
Descripción: Generación del diagrama lógico a partir de la normalización del esquema conceptual		

Tarea de ingeniería A-4

Tarea de Ingeniería		
Número Tarea: A-4	Historia de Usuario: Diseño de esquema de base de datos para control de sistemas.	
Nombre Tarea: Diseño físico, diccionario de datos, script de despliegue		
Tipo Tarea: Creación		Puntos estimados: 4
Fecha Inicio: 14/03/2014		Fecha Fin: 14/03/2014
Analista Responsable: José Pinto		

Descripción: Realizar diagrama físico, diccionario de datos y script de despliegue

Tarea de ingeniería A-5

Tarea de Ingeniería	
Número Tarea: A-5	Historia de Usuario: Diseño de esquema de base de datos para control de sistemas.
Nombre Tarea: Proyección de crecimiento de base de datos.	
Tipo Tarea: Creación	Puntos estimados: 4
Fecha Inicio: 17/03/2014	Fecha Fin: 18/03/2014
Analista Responsable: José Pinto	
Descripción: Realizar proyección de crecimiento de la base de datos.	

Pruebas de aceptación

Caso de Prueba de Aceptación	
Número caso de Prueba: 1	Historia de Usuario: Diseño de esquema de base de datos para control de sistemas.
Nombre: Prueba de integridad y normalización de base de datos	
Descripción: Se realizara el análisis del esquema conceptual para comprobar integridad referencial, revisar normalización, etc.	
Condiciones de Ejecución: Ninguna	
Entrada / Pasos de Ejecución: Ninguna	
Resultados esperados 1: Después de realizar el análisis sobre la base de datos se comprobó que cumple satisfactoriamente con integridad referencial, tercera forma normal.	

Modelo lógico y físico de base de datos

Después de realizar todas las etapas en el desarrollo de base de datos, a continuación se adjunta el diagrama lógico y físico de base de datos.

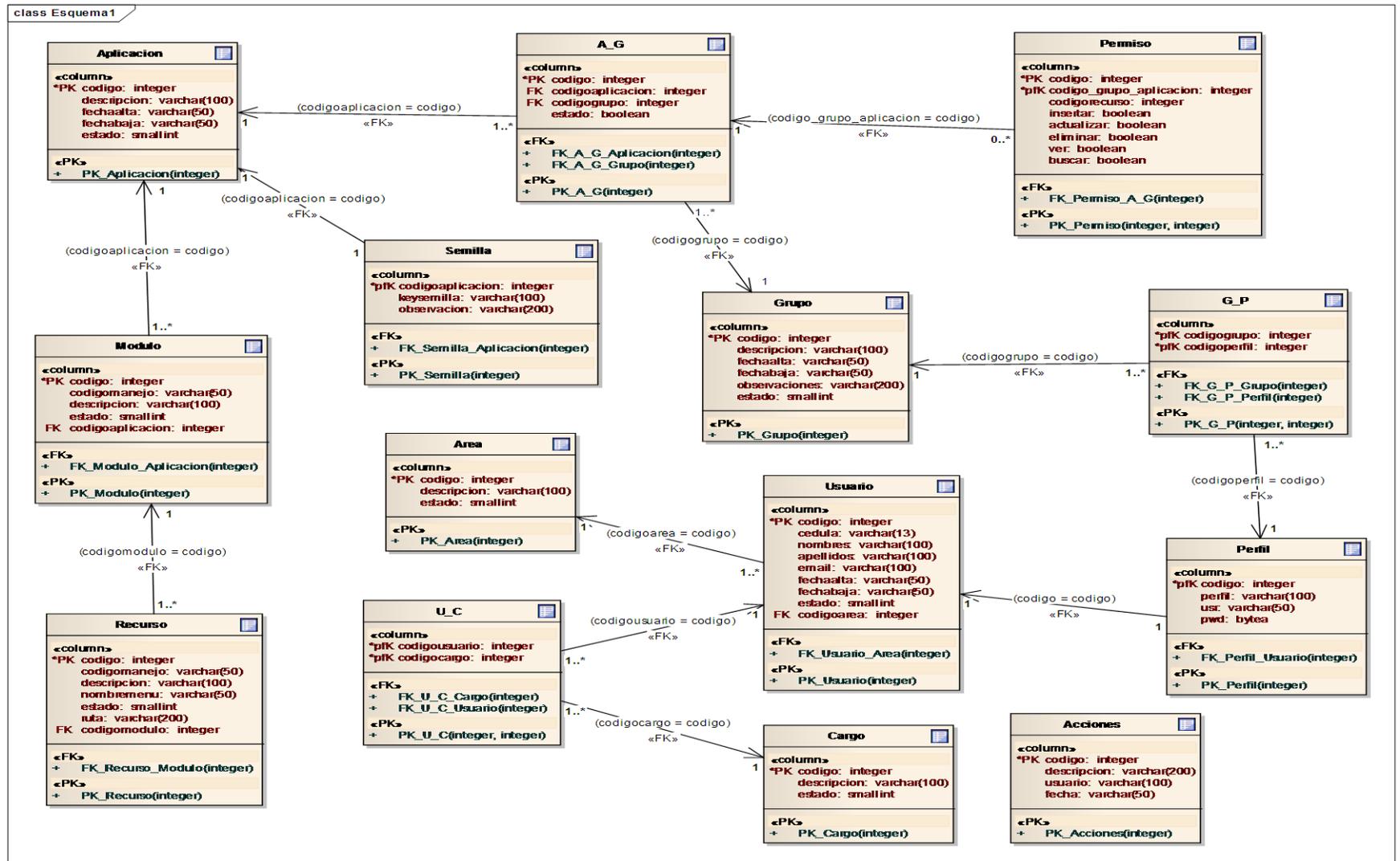


Diagrama lógico

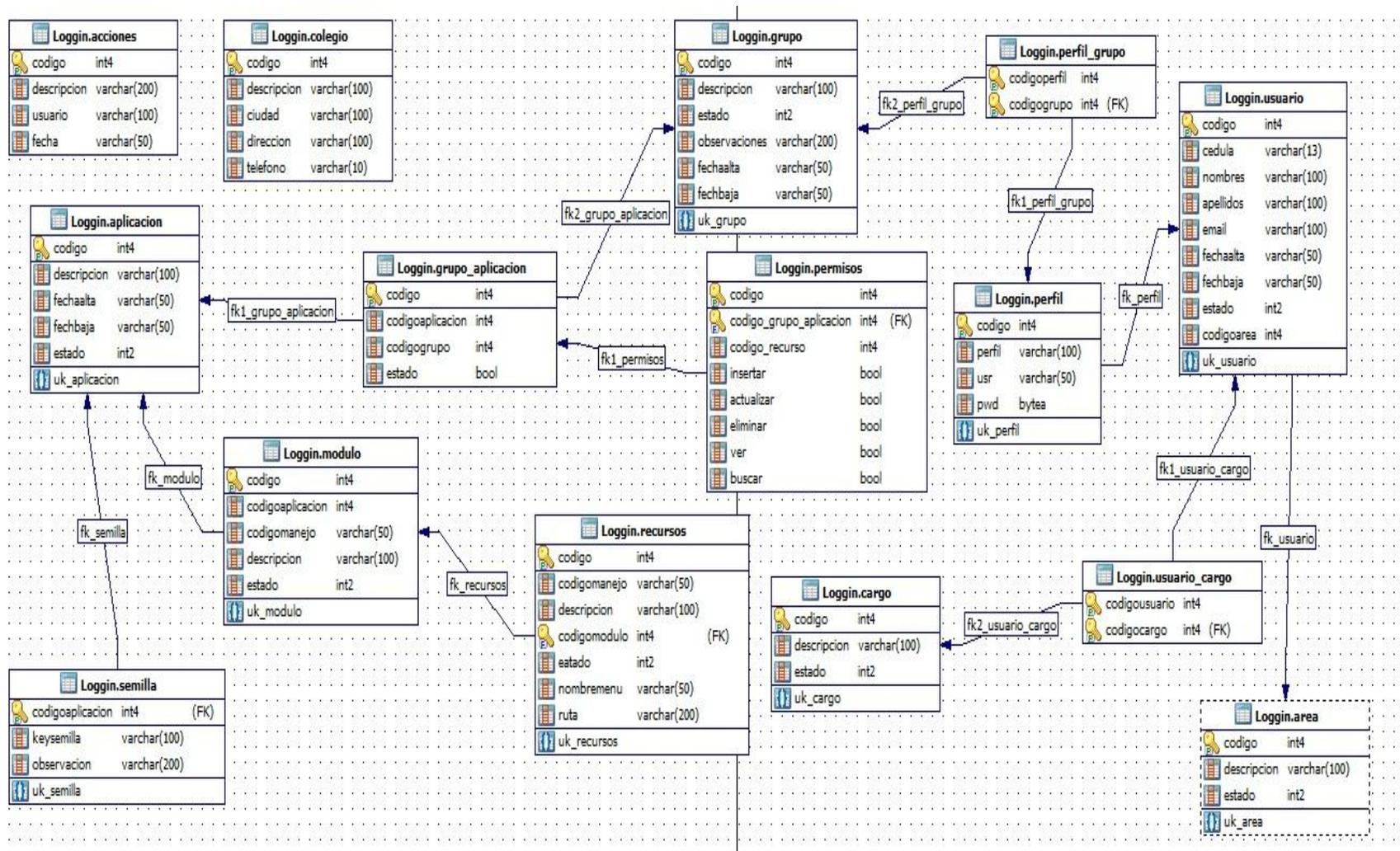


Diagrama Físico

Diccionario de datos

Tabla acciones:

En la tabla adjunta se almacena la información de las acciones que realizan los usuarios en el Módulo de Integración de sistemas.

Descripción de la tabla acciones

Columns	Data Type	Primary Key	Not Null	AutoInc
Código	int4	si	Si	si
Descripción	varchar(200)	-	Si	-
Usuario	varchar(100)	-	Si	-
Fecha	varchar(50)	-	Si	-
Indexes	Unique		Columns	
pk_acciones	si		codigo	
Constraints	Kind		Columns	
pk_acciones	PRIMARY KEY		codigo	

Tabla aplicación:

En la tabla adjunta, se almacena toda la información de las aplicaciones que pueden ser administradas desde el módulo de integración de sistemas.

Descripción de la tabla aplicación

Columns	Data Type	Primary Key	Not Null	AutoInc
Código	int4	Si	Si	si
Descripción	varchar(100)	-	-	-
Fechaalta	varchar(50)	-	Si	-
Fechbaja	varchar(50)	-	-	-
Estado	int2	-	Si	-
Indexes	Unique		Columns	
pk_aplicacion	si		codigo	
uk_aplicacion	si		descripcion	
Constraints	Kind		Columns	

pk_aplicacion	PRIMARY KEY	codigo
uk_aplicacion	UNIQUE	descripcion

Tabla área:

En la tabla adjunta se almacena toda la información de las áreas en donde están asignados los usuarios en el módulo de Integración de sistemas.

Descripción de la tabla área

Columns	Data Type	Primary Key	Not Null	AutoInc
Código	int4	si	Si	si
Descripción	varchar(100)	-	-	-
Estado	int2	-	Si	-
Indexes	Unique		Columns	
pk_area	Si		codigo	
uk_area	Si		descripcion	
Constraints	Kind		Columns	
pk_area	PRIMARY KEY		codigo	
uk_area	UNIQUE		descripcion	

Tabla cargo:

En la tabla adjunta se almacena la información de los cargos que ocupan los usuarios dentro de la institución.

Descripción de la tabla cargo

Columns	Data Type	Primary Key	Not Null	AutoInc
Código	int4	Si	Si	si
Descripción	varchar(100)	-	-	-
Estado	int2	-	Si	-
Indexes	Unique		Columns	
pk_cargo	Si		codigo	

uk_cargo	Si	descripcion
Constraints	Kind	Columns
pk_cargo	PRIMARY KEY	codigo
uk_cargo	UNIQUE	descripcion

Tabla colegio:

En la tabla adjunta se almacena información básica del colegio Capitán Edmundo Chiriboga.

Descripción de la tabla colegio

Columns	Data Type	Primary Key	Not Null	AutoInc
Código	int4	si	Si	si
Descripción	varchar(100)	-	Si	-
Ciudad	varchar(100)	-	Si	-
Dirección	varchar(100)	-	Si	-
Teléfono	varchar(10)	-	Si	-
Indexes	Unique	Columns		
pk_colegio	Si	codigo		
Constraints	Kind	Columns		
pk_colegio	PRIMARY KEY	codigo		

Tabla grupo:

En la tabla adjunta, se almacena información de los grupos que tendrán acceso a las aplicaciones registradas en el Módulo de Integración de sistemas.

Descripción de la tabla grupo

Columns	Data Type	Primary Key	Not Null	AutoInc
Código	int4	si	Si	si
Descripción	varchar(100)	-	Si	-
Estado	int2	-	Si	-

Observaciones	varchar(200)	-	-	-
Fechaalta	varchar(50)	-	Si	-
Fechbaja	varchar(50)	-	-	-
Indexes	Unique	Columns		
pk_grupo	si	codigo		
uk_grupo	si	descripcion		
Constraints	Kind	Columns		
pk_grupo	PRIMARY KEY	codigo		
uk_grupo	UNIQUE	descripcion		

Tabla grupo_aplicacion:

En la tabla adjunta, se almacena la información intermedia entre los grupos asignados a los usuarios y las aplicaciones que pueden administrar los usuarios.

Descripción de la tabla grupo_aplicacion

Columns	Data Type	Primary Key	Not Null	AutoInc
Código	int4	si	Si	si
Codigoaplicacion	int4	-	Si	-
Codigogrupo	int4	-	Si	-
Estado	bool	-	Si	-
Indexes	Unique	Columns		
pk_grupo_aplicacion	Si	codigo		
Constraints	Kind	Columns		
pk_grupo_aplicacion	PRIMARY KEY	codigo		

Tabla modulo:

En la tabla adjunta, almacena la información de los módulos asignados a las aplicaciones.

Descripción de la tabla modulo

Columns	Data Type	Primary Key	Not Null	AutoInc
Código	int4	Si	Si	si
codigoaplicacion	int4	-	Si	-

codigomanejo	varchar(50)	-	Si	-
descripcion	varchar(100)	-	Si	-
Estado	int2	-	Si	-
Indexes	Unique	Columns		
pk_modulo	Si	Código		
uk_modulo	Si	codigomanejo		
Constraints	Kind	Columns		
pk_modulo	PRIMARY KEY	Código		
uk_modulo	UNIQUE	codigomanejo		

Tabla perfil:

En la tabla adjunta, se almacena la información de las credenciales de los usuarios.

Descripción de la tabla perfil

Columns	Data Type	Primary Key	Not Null	AutoInc
Código	int4	Si	Si	-
Perfil	varchar(100)	-	Si	-
Usr	varchar(50)	-	Si	-
Pwd	bytea	-	Si	-
Indexes	Unique	Columns		
pk_perfil	Si	codigo		
uk_perfil	Si	usr		
Constraints	Kind	Columns		
pk_perfil	PRIMARY KEY	codigo		
uk_perfil	UNIQUE	usr		

Tabla perfil_grupo:

En la tabla adjunta, se almacena la información de la relación entre la tabla perfil y tabla grupo.

Descripción de la tabla perfil_grupo

Columns	Data Type	Primary Key	Not Null	AutoInc
Codigoperfil	int4	si	Si	-
Codigogrupo	int4	si	Si	-
Indexes	Unique		Columns	
pk_perfil_grupo	Si		codigoperfil,codigogrupo	
Constraints	Kind		Columns	
pk_perfil_grupo	PRIMARY KEY		codigoperfil,codigogrupo	

Tabla permisos:

En la tabla adjunta, se almacena la información de los permisos que son asignados a los grupos.

Descripción de la tabla permisos

Columns	Data Type	Primary Key	Not Null	AutoInc
Código	int4	Si	Si	si
codigo_grupo_aplicacion	int4	-	Si	-
codigo_recurso	int4	-	Si	-
Insertar	bool	-	Si	-
Actualizar	bool	-	Si	-
Eliminar	bool	-	Si	-
Ver	bool	-	Si	-
Buscar	bool	-	Si	-
Indexes	Unique		Columns	
pk_permisos	Si		codigo	
Constraints	Kind		Columns	
pk_permisos	PRIMARY KEY		codigo	

Tabla recursos:

En la tabla adjunta, se almacena la información de los recursos y/o opciones que tendrá la aplicación.

Descripción de la tabla recursos

Columns	Data Type	Primary Key	Not Null	AutoInc
Código	int4	Si	Si	si
codigomanejo	varchar(50)	-	Si	-
descripcion	varchar(100)	-	Si	-
codigomodulo	int4	-	Si	-
Eatado	int2	-	Si	-
nombremenu	varchar(50)	-	-	-
Ruta	varchar(200)	-	-	-
Indexes	Unique	Columns		
pk_recursos	Si	codigo		
uk_recursos	Si	codigomanejo		
Constraints	Kind	Columns		
pk_recursos	PRIMARY KEY	codigo		
uk_recursos	UNIQUE	codigomanejo		

Tabla semilla:

En la tabla adjunta, se almacena la información de la clave que tendrá asignada una aplicación, esta semilla permite encriptar las claves de los usuarios que tendrán acceso a la aplicación.

Descripción de la tabla semilla

Columns	Data Type	Primary Key	Not Null	AutoInc
Codigoaplicacion	int4	si	Si	-
Keysemilla	varchar(100)	-	Si	-

observacion	varchar(200)	-	-	-
Indexes	Unique		Columns	
pk_semilla	Si		codigoaplicacion	
uk_semilla	si		keysemilla	
Constraints	Kind		Columns	
pk_semilla	PRIMARY KEY		codigoaplicacion	
uk_semilla	UNIQUE		keysemilla	

Tabla usuario:

En la tabla adjunta, se almacena la información personal de los usuarios que tendrán acceso a las distintas aplicaciones que se encuentren registradas en el sistema.

Descripción de la tabla usuario

Columns	Data Type	Primary Key	Not Null	AutoInc
Código	int4	si	Si	Si
Cedula	varchar(13)	-	Si	-
nombres	varchar(100)	-	Si	-
apellidos	varchar(100)	-	Si	-
Email	varchar(100)	-	Si	-
fechaalta	varchar(50)	-	Si	-
Fechbaja	varchar(50)	-	-	-
Estado	int2	-	Si	-
codigoarea	int4	-	-	-
Indexes	Unique		Columns	
pk_usuario	si		Código	
uk_usuario	si		Cedula	
Constraints	Kind		Columns	
pk_usuario	PRIMARY KEY		Código	
uk_usuario	UNIQUE		Cedula	

Tabla usuario_carga:

En la tabla adjunta, se almacena la información de la relación entre la tabla usuario y la tabla carga, ya que un usuario puede desempeñar varios cargos en la institución.

Descripción de la tabla usuario_carga

Columns	Data Type	Primary Key	Not Null	AutoInc
Codigousuario	int4	si	Si	-
Codigocargo	int4	si	Si	-
Indexes	Unique	Columns		
pk_usuario_cargo	si	codigousuario,codigocargo		
Constraints	Kind	Columns		
pk_usuario_cargo	PRIMARY KEY	codigousuario,codigocargo		

Script de despliegue

En la tabla adjunta se detalla el script generado para el despliegue de la base de datos.

Tabla	Script
acciones	<pre>CREATE TABLE "Loggin".acciones (codigo integer NOT NULL DEFAULT nextval("Loggin".acciones_codigo_seq::regclass), descripcion character varying(200) NOT NULL, usuario character varying(100) NOT NULL, fecha character varying(50) NOT NULL, CONSTRAINT pk_acciones PRIMARY KEY (codigo))</pre>
aplicación	<pre>CREATE TABLE "Loggin".aplicacion (codigo integer NOT NULL DEFAULT nextval("Loggin".aplicacion_codigo_seq::regclass), descripcion character varying(100), fechaalta character varying(50) NOT NULL, fehbaja character varying(50), estado smallint NOT NULL DEFAULT 0, CONSTRAINT pk_aplicacion PRIMARY KEY (codigo), CONSTRAINT uk_aplicacion UNIQUE (descripcion))</pre>
area	<pre>CREATE TABLE "Loggin".area (</pre>

	<pre> codigo integer NOT NULL DEFAULT nextval('"Loggin".area_codigo_seq'::regclass), descripcion character varying(100), estado smallint NOT NULL DEFAULT 0, CONSTRAINT pk_area PRIMARY KEY (codigo), CONSTRAINT uk_area UNIQUE (descripcion)) </pre>
cargo	<pre> CREATE TABLE "Loggin".cargo (codigo integer NOT NULL DEFAULT nextval('"Loggin".cargo_codigo_seq'::regclass), descripcion character varying(100), estado smallint NOT NULL DEFAULT 0, CONSTRAINT pk_cargo PRIMARY KEY (codigo), CONSTRAINT uk_cargo UNIQUE (descripcion)) </pre>
grupo	<pre> CREATE TABLE "Loggin".grupo (codigo integer NOT NULL DEFAULT nextval('"Loggin".grupo_codigo_seq'::regclass), descripcion character varying(100) NOT NULL, estado smallint NOT NULL DEFAULT 0, observaciones character varying(200), fechaalta character varying(50) NOT NULL, fechbaja character varying(50), CONSTRAINT pk_grupo PRIMARY KEY (codigo), CONSTRAINT uk_grupo UNIQUE (descripcion)) </pre>
grupo_aplicacion	<pre> CREATE TABLE "Loggin".grupo_aplicacion (codigo integer NOT NULL DEFAULT nextval('"Loggin".grupo_aplicacion_codigo_seq'::regclass), codigoaplicacion integer NOT NULL, codigogrupo integer NOT NULL, estado boolean NOT NULL, CONSTRAINT pk_grupo_aplicacion PRIMARY KEY (codigo), CONSTRAINT fk1_grupo_aplicacion FOREIGN KEY (codigoaplicacion) REFERENCES "Loggin".aplicacion (codigo) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION, CONSTRAINT fk2_grupo_aplicacion FOREIGN KEY (codigogrupo) REFERENCES "Loggin".grupo (codigo) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION) </pre>
modulo	<pre> CREATE TABLE "Loggin".modulo (</pre>

	<pre> codigo integer NOT NULL DEFAULT nextval('"Loggin".modulo_codigo_seq'::regclass), codigoaplicacion integer NOT NULL, codigomanejo character varying(50) NOT NULL, descripcion character varying(100) NOT NULL, estado smallint NOT NULL DEFAULT 0, CONSTRAINT pk_modulo PRIMARY KEY (codigo), CONSTRAINT fk_modulo FOREIGN KEY (codigoaplicacion) REFERENCES "Loggin".aplicacion (codigo) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION, CONSTRAINT uk_modulo UNIQUE (codigomanejo)) </pre>
perfil	<pre> CREATE TABLE "Loggin".perfil (codigo integer NOT NULL, perfil character varying(100) NOT NULL, usr character varying(50) NOT NULL, pwd bytea NOT NULL, CONSTRAINT pk_perfil PRIMARY KEY (codigo), CONSTRAINT fk_perfil FOREIGN KEY (codigo) REFERENCES "Loggin".usuario (codigo) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION, CONSTRAINT uk_perfil UNIQUE (usr)) </pre>
perfil_grupo	<pre> CREATE TABLE "Loggin".perfil_grupo (codigoperfil integer NOT NULL, codigogrupo integer NOT NULL, CONSTRAINT pk_perfil_grupo PRIMARY KEY (codigoperfil , codigogrupo), CONSTRAINT fk1_perfil_grupo FOREIGN KEY (codigoperfil) REFERENCES "Loggin".perfil (codigo) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION, CONSTRAINT fk2_perfil_grupo FOREIGN KEY (codigogrupo) REFERENCES "Loggin".grupo (codigo) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION) </pre>
permisos	<pre> CREATE TABLE "Loggin".permisos (codigo integer NOT NULL DEFAULT nextval('"Loggin".permisos_codigo_seq'::regclass), codigo_grupo_aplicacion integer NOT NULL, codigo_recurso integer NOT NULL, insertar boolean NOT NULL DEFAULT false, actualizar boolean NOT NULL DEFAULT false, </pre>

	<pre> eliminar boolean NOT NULL DEFAULT false, ver boolean NOT NULL DEFAULT false, buscar boolean NOT NULL DEFAULT false, CONSTRAINT pk_permisos PRIMARY KEY (codigo), CONSTRAINT fk1_permisos FOREIGN KEY (codigo_grupo_aplicacion) REFERENCES "Loggin".grupo_aplicacion (codigo) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION) </pre>
recursos	<pre> CREATE TABLE "Loggin".recursos (codigo integer NOT NULL DEFAULT nextval('"Loggin".recursos_codigo_seq'::regclass), codigomanejo character varying(50) NOT NULL, descripcion character varying(100) NOT NULL, codigomodulo integer NOT NULL, eatado smallint NOT NULL DEFAULT 0, nombremenu character varying(50), ruta character varying(200), CONSTRAINT pk_recursos PRIMARY KEY (codigo), CONSTRAINT fk_recursos FOREIGN KEY (codigomodulo) REFERENCES "Loggin".modulo (codigo) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION, CONSTRAINT uk_recursos UNIQUE (codigomanejo)) </pre>
Semilla	<pre> CREATE TABLE "Loggin".semilla (codigoaplicacion integer NOT NULL, keysemilla character varying(100) NOT NULL, observacion character varying(200), CONSTRAINT pk_semilla PRIMARY KEY (codigoaplicacion), CONSTRAINT fk_semilla FOREIGN KEY (codigoaplicacion) REFERENCES "Loggin".aplicacion (codigo) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION, CONSTRAINT uk_semilla UNIQUE (keysemilla)) </pre>
usuario	<pre> CREATE TABLE "Loggin".usuario (codigo integer NOT NULL DEFAULT nextval('"Loggin".usuario_codigo_seq'::regclass), cedula character varying(13) NOT NULL, nombres character varying(100) NOT NULL, apellidos character varying(100) NOT NULL, email character varying(100) NOT NULL, fechaalta character varying(50) NOT NULL, fechbaja character varying(50), estado smallint NOT NULL DEFAULT 0, </pre>

	<pre> codigoarea integer, CONSTRAINT pk_usuario PRIMARY KEY (codigo), CONSTRAINT fk_usuario FOREIGN KEY (codigoarea) REFERENCES "Loggin".area (codigo) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION, CONSTRAINT uk_usuario UNIQUE (cedula)) </pre>
usuario_cargo	<pre> CREATE TABLE "Loggin".usuario_cargo (codigousuario integer NOT NULL, codigocargo integer NOT NULL, CONSTRAINT pk_usuario_cargo PRIMARY KEY (codigousuario , codigocargo), CONSTRAINT fk1_usuario_cargo FOREIGN KEY (codigousuario) REFERENCES "Loggin".usuario (codigo) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION, CONSTRAINT fk2_usuario_cargo FOREIGN KEY (codigocargo) REFERENCES "Loggin".cargo (codigo) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION) </pre>

Iteración 2

Metáfora 2

La adjunta detalla la metáfora de análisis de la base de datos del entorno de aprendizaje Moodle.

Análisis de la base de datos del entorno de aprendizaje Moodle

Historia de usuario	
Número: M2	Nombre historia: Análisis de la base de datos del entorno de aprendizaje Moodle
Usuario: Analista	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimado: 12
Riesgo en Desarrollo: Medio	Puntos Reales: 12

Descripción: Como analista deseo que la investigación de la estructura de la base de datos de Moodle consista en un detallado informe de las tablas, campos, orden que se tienen que registrar los datos en Moodle, para poder integrar este sistema con el sistema académico.

Cronograma de tareas de ingeniería

Para cumplir con la fecha establecida por la iteración. En la tabla adjunta se establece el cronograma y tarea de ingeniería a desarrollarse.

Cronograma de tareas de ingeniería metáfora 2

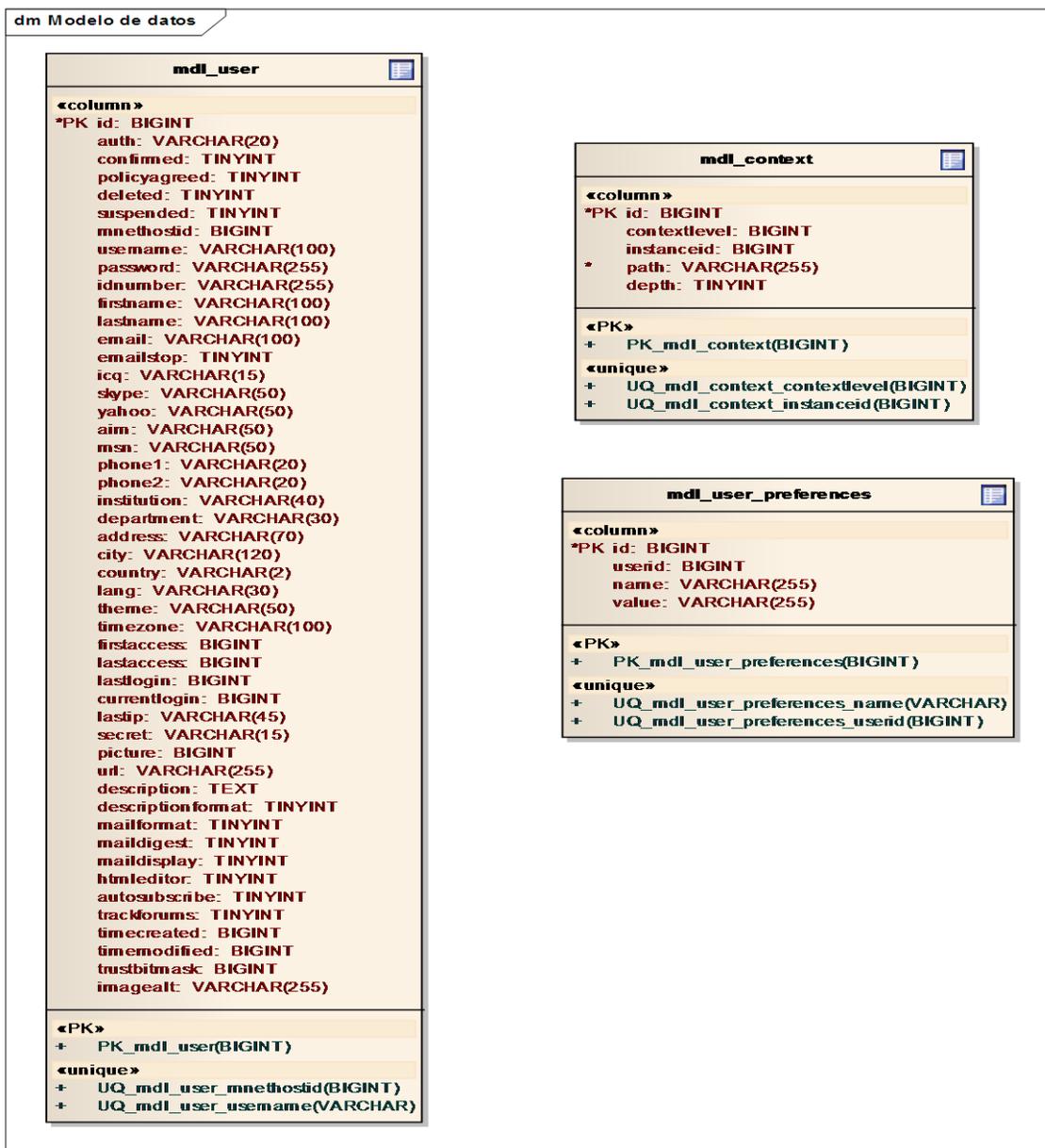
Tarea de ingeniería (código y nombre)		Puntos estimados	Puntos reales	Fecha inicio	Fecha fin
B-1	Investigación y análisis de la estructura de base de datos de Moodle	12	12	18/03/2014	21/03/2014

Tarea de ingeniería B-1

Tarea de Ingeniería	
Número Tarea: B-1	Historia de Usuario: Análisis de la base de datos del entorno de aprendizaje Moodle
Nombre Tarea: Investigación y análisis de la estructura de base de datos de Moodle	
Tipo Tarea: Investigación y análisis	Puntos estimados: 8
Fecha Inicio: 18/03/2014	Fecha Fin: 21/03/2014
Analista Responsable: José Pinto	
Descripción: El análisis se lo realizara buscando documentación de Moodle, acudiendo a foros, visitando usuarios expertos en Moodle, etc.	

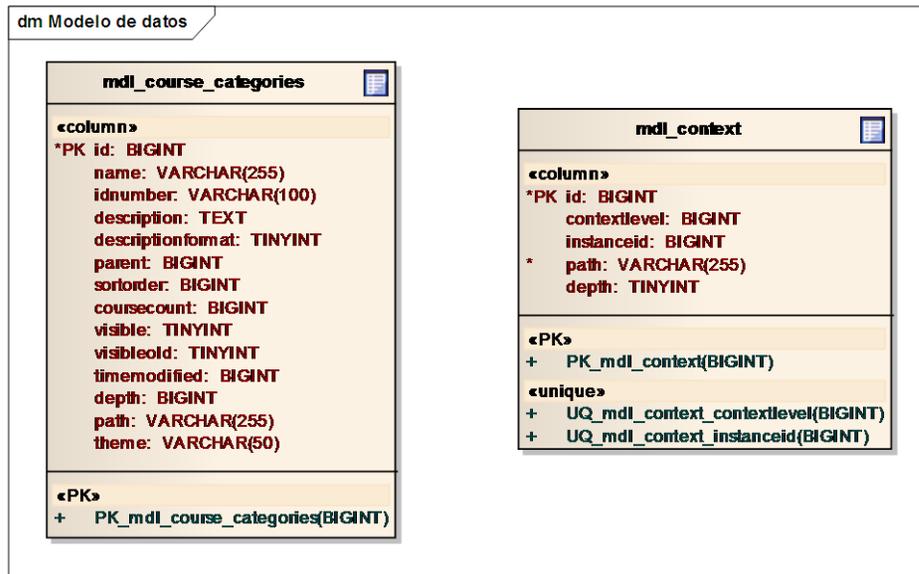
Tablas analizadas de Moodle

Para la creación de usuario utiliza las tablas mdl_user, mdl_context, mdl_user_preferences. En la Figura adjunta se observan las tablas y campos que son necesarios para la creación de usuario.



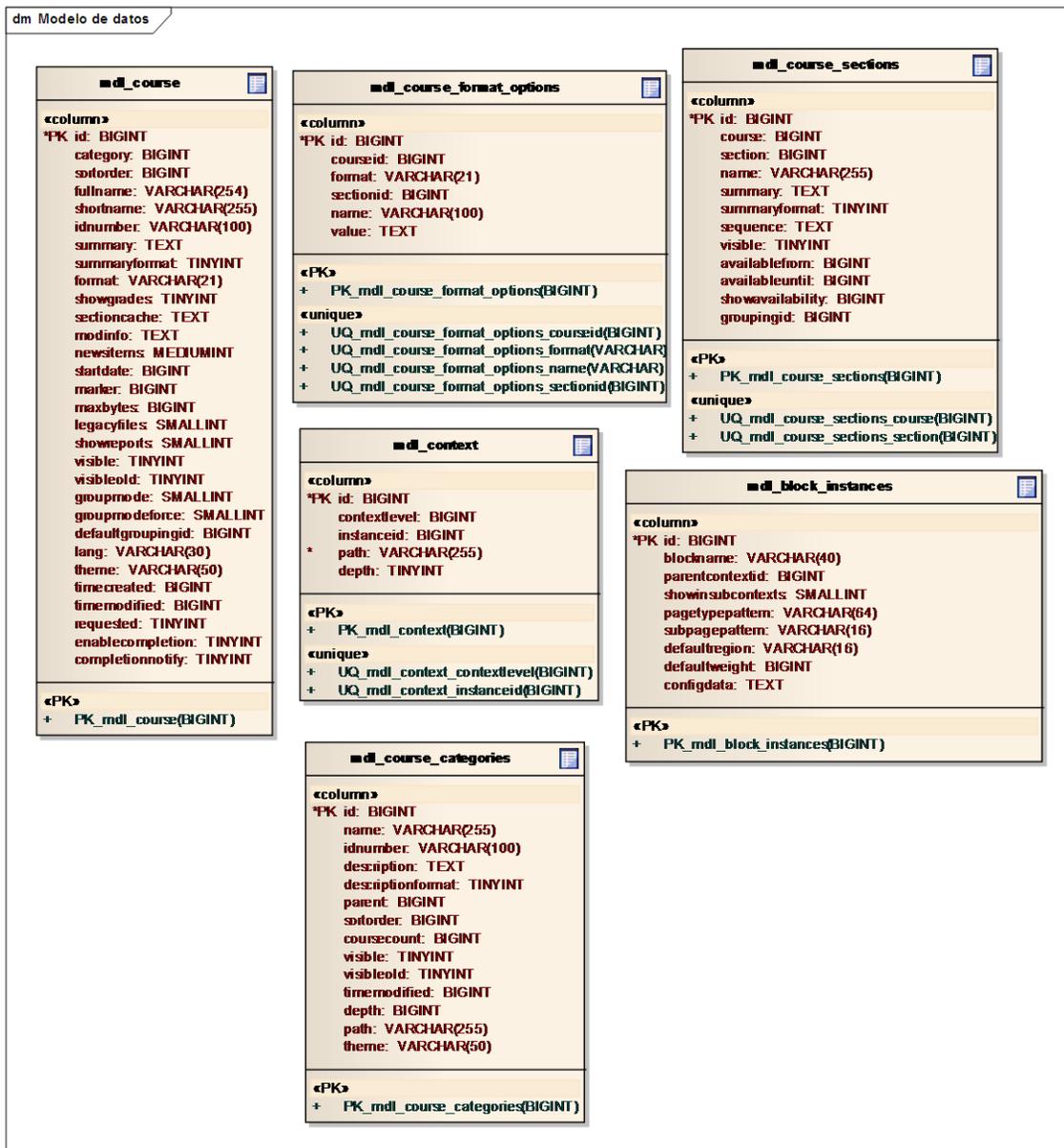
Tablas usadas para la creación de usuarios en Moodle

En Moodle los cursos, paralelos son denominados como categorías, para la creación de categorías se utilizan las tablas, mdl_course_categories, mdl_context. En la figura adjunta se observa las tablas y campos necesarios para la creación de categorías.



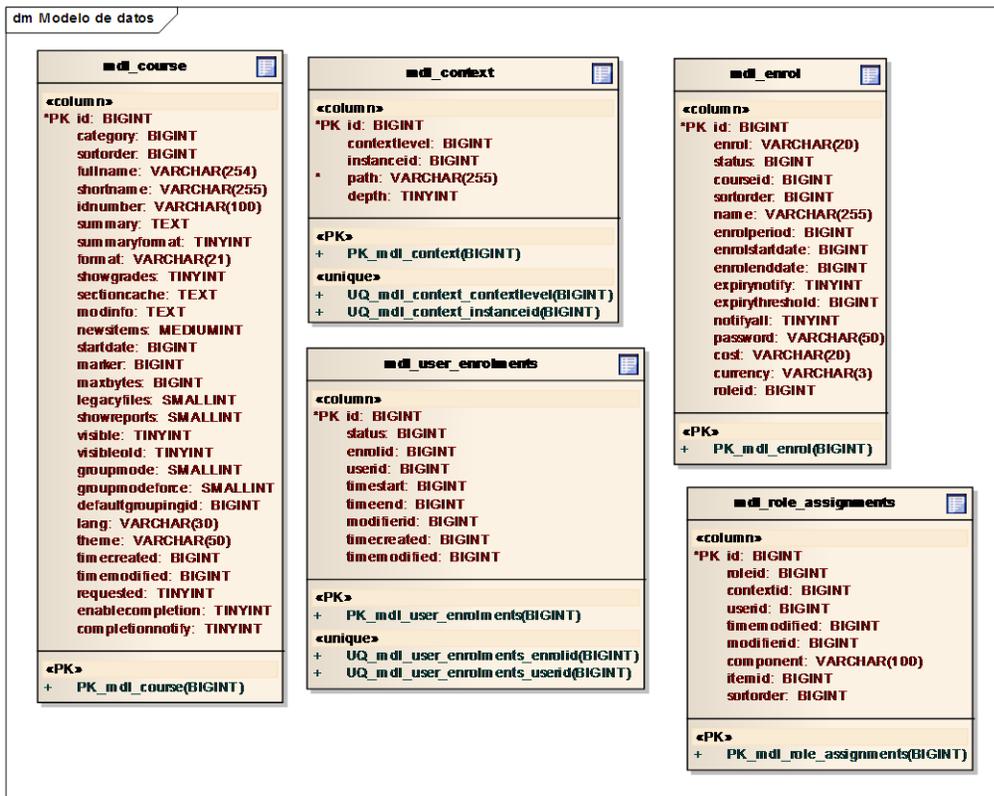
Tablas usadas para la creación de categorías en Moodle

En Moodle no existe el término materia, a su vez este término es reemplazado por el de curso las tablas utilizadas para la creación de cursos son: mdl_course, mdl_course_format_options, mdl_context, mdl_block_instances, mdl_course_sections, mdl_course_categories. En la figura adjunta se observa las tablas y campos necesarios para la creación de cursos.



Tablas usadas para la creación de cursos en Moodle

Para la matriculación de estudiantes en los cursos y asignación de profesores se utilizan las siguientes tablas: mdl_course, mdl_enrol, mdl_user_enrolments, mdl_role_assignments, mdl_context. En la figura adjunta se observa las tablas y campos utilizados para la matriculación de estudiantes y asignación de profesores a los cursos.



Tablas usadas para matriculación de estudiantes y docentes en Moodle

Metáfora 3

La base de datos del sistema académico consta de un total de 50 tablas, las mismas que fueron analizadas para detectar los campos que servirán para la integración con Moodle. En la tabla adjunta se observa la definición de la metáfora 3.

Análisis de la base de datos del sistema académico del colegio

Historia de usuario	
Número: M3	Nombre historia: Análisis de la base de datos del sistema académico del colegio
Usuario: Analista	Iteración Asignada: 2

Prioridad en Negocio: Alta	Puntos Estimado: 8
Riesgo en Desarrollo: Medio	Puntos Reales: 8
Descripción: Como analista deseo que la investigación de las tablas del sistema académico consten de un completo informe, esto servirá para la creación de los servicios web que extraerán datos del sistema académico que serán modelados en una aplicación intermedia y después serán sincronizados hacia Moodle.	

Cronograma de tareas de ingeniería

El cronograma definido para el desarrollo de la tarea de ingeniería es el que detalla en la tabla adjunta en la cual constan los tiempos para la entrega de la información.

Cronograma de tareas de ingeniería metáfora 3

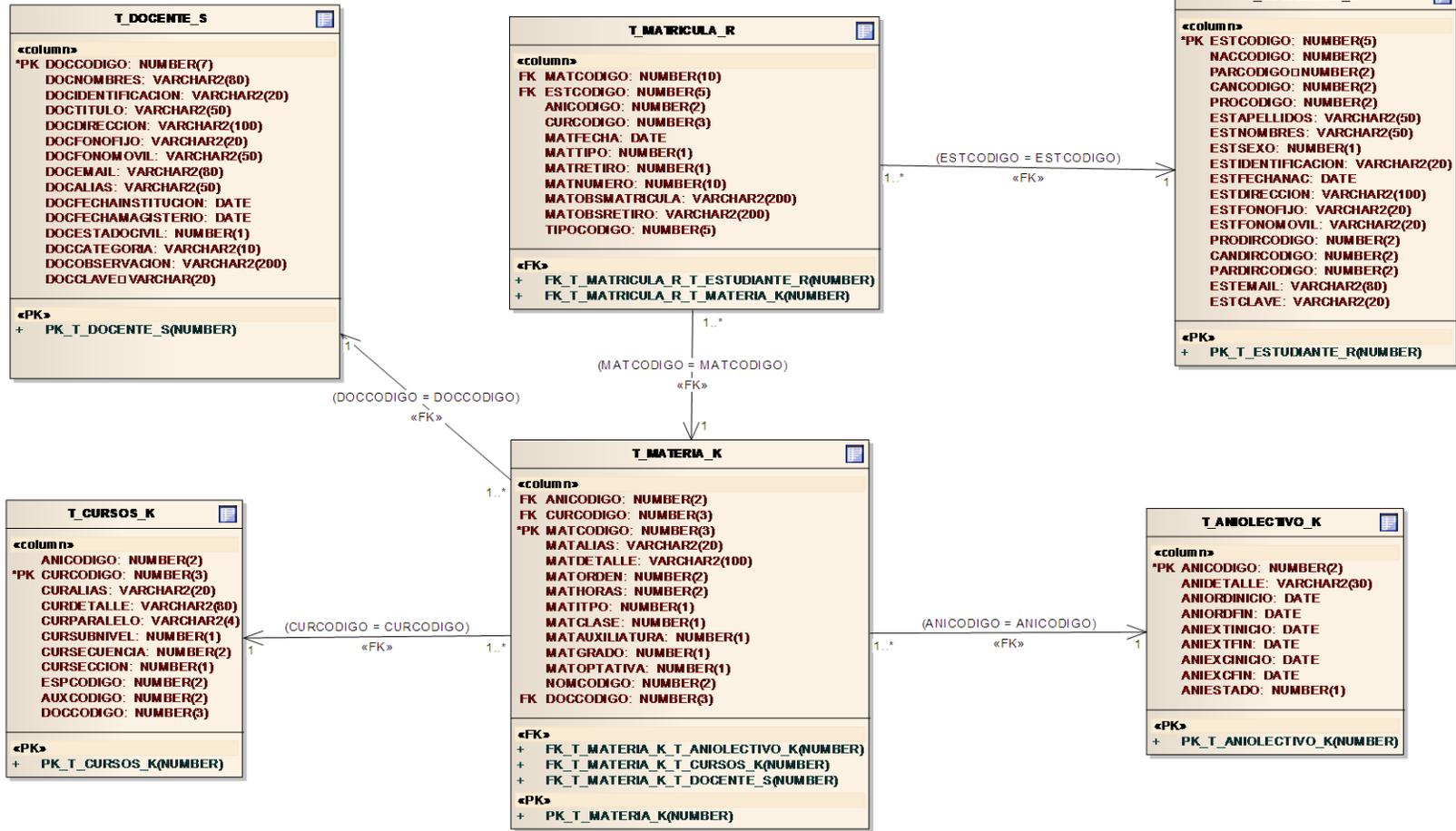
Tarea de ingeniería (código y nombre)		Puntos estimados	Puntos reales	Fecha inicio	Fecha fin
C-1	Investigación y análisis de la estructura de base de datos de sistema académico	8	8	21/03/2014	25/03/2014

Tarea de ingeniería C-1

Tarea de Ingeniería	
Número Tarea: C-1	Historia de Usuario: Análisis de la base de datos del sistema académico del colegio
Nombre Tarea: Investigación y análisis de la estructura de base de datos de sistema académico.	
Tipo Tarea: Investigación y análisis	Puntos estimados: 8
Fecha Inicio: 21/03/2014	Fecha Fin: 25/03/2014
Analista Responsable: José Pinto	

Descripción: El análisis se lo realizara manteniendo conversación con el encargado del mantenimiento de la base de datos del colegio, y también revisando el flujo que sigue el colegio para la matriculación de sus estudiantes.

En la figura adjunta se detalla las tablas del sistema académico analizadas, que servirán para la extracción de datos.



Tablas sistema académico

Iteración 3

Metáfora 4

Diseño de esquema de base de datos para integración de sistemas

Historia de usuario	
Número: M4	Nombre historia: Diseño de esquema de base de datos para Integración de sistemas
Usuario: Programador	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimado: 12
Riesgo en Desarrollo: Medio	Puntos Reales: 14
Descripción: Como programador quiero generar el diagrama DER de base de datos, normalización de tablas, diagrama lógico, diagrama físico, proyección de crecimiento de la base de datos, script de despliegue y diccionario de datos, para obtener un producto adecuado a las especificaciones del cliente.	

Cronograma de tareas de ingeniería

El cronograma utilizado para que los desarrolladores tengan los tiempos claros y también lo que se desea hacer para esta iteración se detalla en la tabla adjunta, en este apartado se definen las fechas de inicio y finalización así como los puntos asignados para las tareas:

Cronograma de tareas de ingeniería metáfora 4

Tarea de ingeniería (código y nombre)	Puntos estimados	Puntos reales	Fecha inicio	Fecha fin
D-1 Diagrama entidad relación DER	6	6	25/03/2014	26/03/2014
D-2 Normalización del diagrama entidad relación	2	4	26/03/2014	27/03/2014
D-3 Diseño lógico	2	2	27/03/2014	28/03/2014

D-4	Diseño físico, diccionario de datos, script de despliegue	1	1	28/03/2014	28/03/2014
D-5	Proyección de crecimiento de base de datos.	1	1	28/03/2014	28/03/2014

Tarea de ingeniería D-1

Tarea de Ingeniería	
Número Tarea: D-1	Historia de Usuario: Diseño de esquema de base de datos para Integración de sistemas
Nombre Tarea: Diagrama entidad relación DER	
Tipo Tarea: Creación	Puntos estimados: 6
Fecha Inicio: 25/03/2014	Fecha Fin: 26/03/2014
Analista Responsable: José Pinto	
Descripción: se realizó el esquema entidad relación DER	

Tarea de ingeniería D-2

Tarea de Ingeniería	
Número Tarea: D-2	Historia de Usuario: Diseño de esquema de base de datos para Integración de sistemas
Nombre Tarea: Normalización del diagrama entidad relación	
Tipo Tarea: Creación	Puntos estimados: 4
Fecha Inicio: 26/03/2014	Fecha Fin: 27/03/2014
Analista Responsable: José Pinto	
Descripción: Realizar la normalización sobre el diagrama entidad relación hasta la 3FN	

Tarea de ingeniería D-3

Tarea de Ingeniería	
Número Tarea: D-3	Historia de Usuario: Diseño de esquema de base de datos para Integración de sistemas
Nombre Tarea: Diseño lógico	
Tipo Tarea: Creación	Puntos estimados: 2
Fecha Inicio: 27/03/2014	Fecha Fin: 28/03/2014
Analista Responsable: José Pinto	
Descripción: Generación del diagrama lógico a partir de la normalización del esquema conceptual	

Tarea de ingeniería D-4

Tarea de Ingeniería	
Número Tarea: D-4	Historia de Usuario: Diseño de esquema de base de datos para Integración de sistemas
Nombre Tarea: Diseño físico, diccionario de datos, script de despliegue	
Tipo Tarea: Creación	Puntos estimados: 1
Fecha Inicio: 28/03/2014	Fecha Fin: 28/03/2014
Analista Responsable: José Pinto	
Descripción: Realizar diagrama físico, diccionario de datos y script de despliegue	

Tarea de ingeniería D-5

Tarea de Ingeniería	
Número Tarea: D-5	Historia de Usuario: Diseño de esquema de base de datos para Integración de sistemas
Nombre Tarea: Proyección de crecimiento de base de datos.	
Tipo Tarea: Creación	Puntos estimados: 1

Fecha Inicio: 28/03/2014	Fecha Fin: 28/03/2014
Analista Responsable: José Pinto	
Descripción: Realizar proyección de crecimiento de la base de datos.	

Pruebas de aceptación

Caso de Prueba de Aceptación	
Número caso de Prueba: 2	Historia de Usuario: Diseño de esquema de base de datos para Integración de sistemas
Nombre: Prueba de integridad y normalización de base de datos	
Descripción: Se realizara el análisis del esquema conceptual para comprobar integridad referencial, revisar normalización, etc.	
Condiciones de Ejecución: Ninguna	
Entrada / Pasos de Ejecución: Ninguna	
Resultados esperados 1: Después de realizar el análisis sobre la base de datos se comprobó que cumple satisfactoriamente con integridad referencial, tercera forma normal.	

Modelo lógico y físico de base de datos

La figura adjunta hace referencia al modelo físico de la base de datos desarrollada para la integración de sistemas.

class Modelo de datos

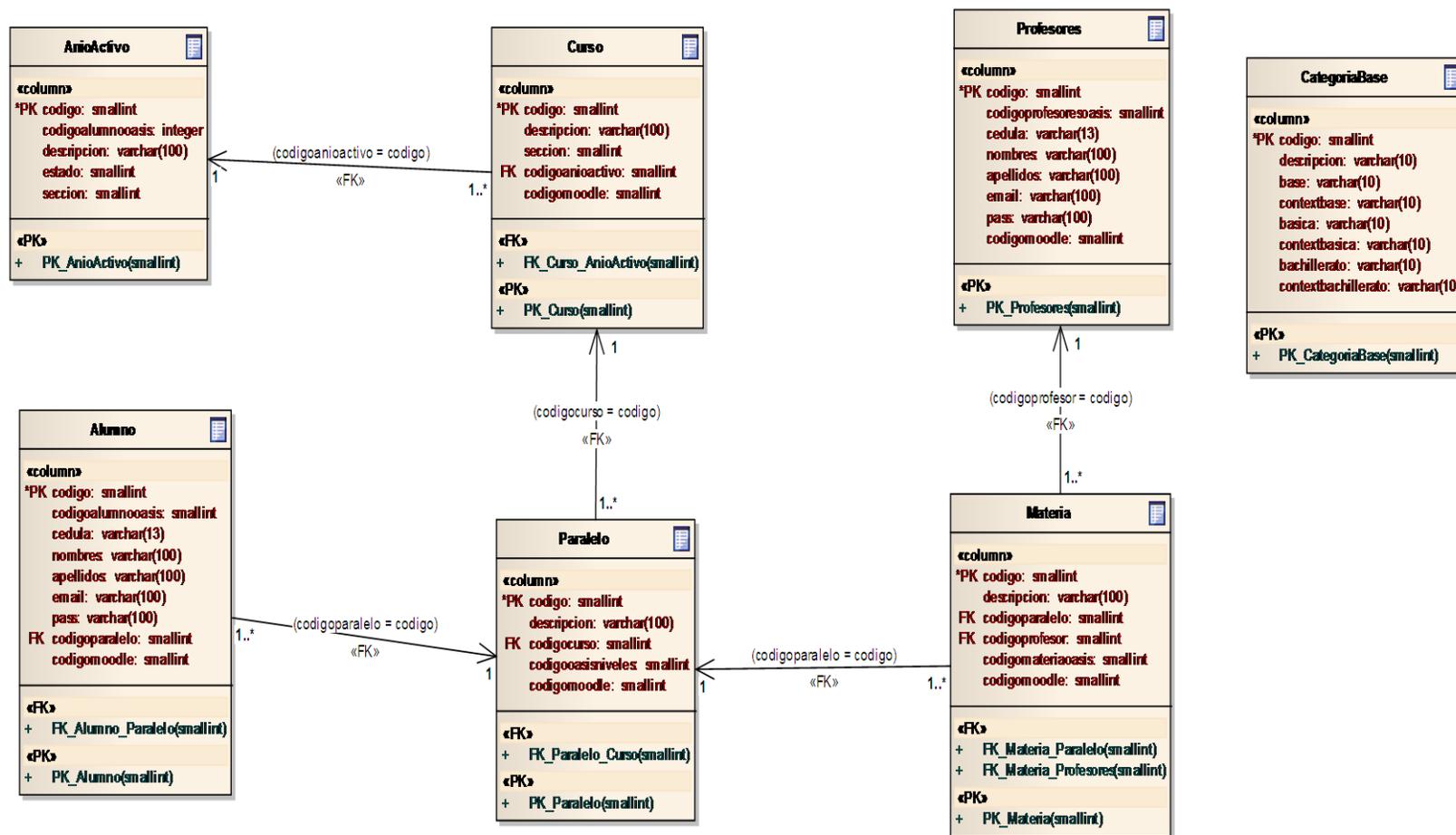


Diagrama Lógico

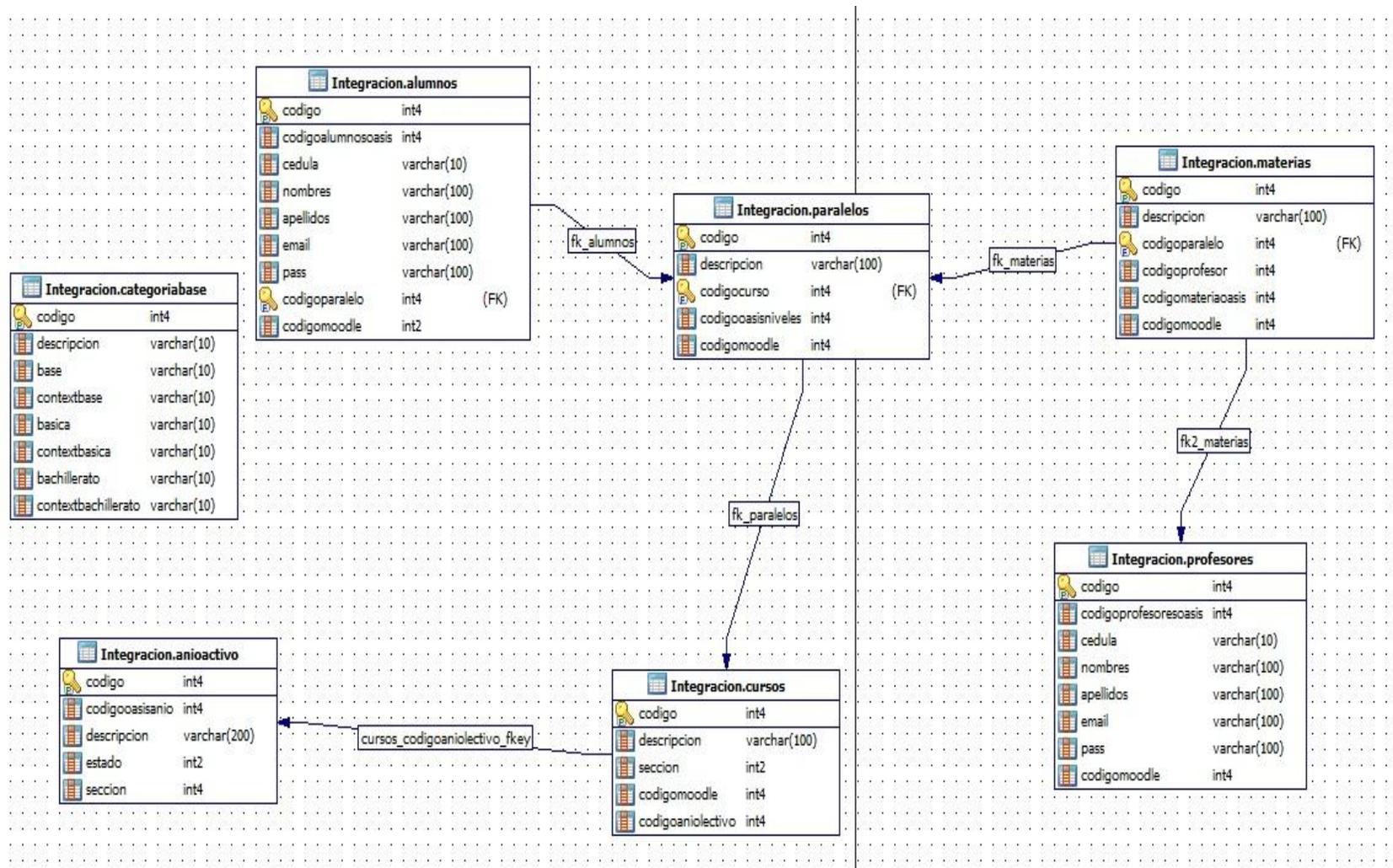


Diagrama Físico

Diccionario de datos del esquema de Integración

Tabla alumnos:

La tabla adjunta se almacena la información de los alumnos que se extrajeron desde el sistema académico

Descripción de la tabla alumnos

Columns	Data Type	Primary Key	Not Null	AutoInc
Código	int4	Si	Si	Si
codigoalumnooasis	int4	-	Si	-
Cedula	varchar(10)	-	Si	-
Nombres	varchar(100)	-	Si	-
Apellidos	varchar(100)	-	Si	-
Email	varchar(100)	-	Si	-
Pass	varchar(100)	-	Si	-
codigoparalelo	int4	-	Si	-
codigomoodle	int2	-	Si	-
Indexes	Unique	Columns		
pk_alumnos	si	Código		
Constraints	Kind	Columns		
pk_alumnos	PRIMARY KEY	Código		

Tabla anioactivo:

En la tabla adjunta se almacena la información del año activo vigente el cual fue migrado desde el sistema académico del colegio.

Descripción de la tabla anioactivo

Columns	Data Type	Primary Key	Not Null	AutoInc
Código	int4	si	Si	Si
codigooasisanio	int4	-	Si	-
descripcion	varchar(200)	-	Si	-
Estado	int2	-	Si	-
Sección	int4	-	-	-

Indexes	Unique	Columns
pk_anioactivo	Si	codigo
Constraints	Kind	Columns
pk_anioactivo	PRIMARY KEY	codigo

Tabla categoriabase:

La tabla adjunta se observa información de las categorías sea básicos o bachillerato el cual será migrado desde el Módulo de Integración de Sistemas a Moodle.

Descripción de la tabla categoriabase

Columns	Data Type	Primary Key	Not Null	AutoInc
Código	int4	Si	Si	Si
Descripción	varchar(10)	-	Si	-
Base	varchar(10)	-	Si	-
Contextbase	varchar(10)	-	Si	-
Básica	varchar(10)	-	Si	-
contextbasica	varchar(10)	-	Si	-
bachillerato	varchar(10)	-	Si	-
contextbachillerato	varchar(10)	-	Si	-
Indexes	Unique	Columns		
pk_categoriabase	si	codigo		
Constraints	Kind	Columns		
pk_categoriabase	PRIMARY KEY	codigo		

Tabla cursos:

La tabla adjunta, se almacena la información de los cursos que se extrajeron desde el sistema académico y que serán migrados desde el Módulo de Integración de Sistemas a Moodle.

Descripción de la tabla cursos

Columns	Data Type	Primary Key	Not Null	AutoInc
---------	-----------	-------------	----------	---------

Código	int4	Si	Si	si
descripcion	varchar(100)	-	-	-
Sección	int2	-	Si	-
codigomoodle	int4	-	-	-
codigoaniolectivo	int4	-	-	-
Indexes	Unique		Columns	
pk_area	si		codigo	
Constraints	Kind		Columns	
pk_area	PRIMARY KEY		codigo	

Tabla materias:

La tabla adjunta se almacena la información de las materias que se extrajeron desde el sistema académico y que serán migradas desde el Módulo de Integración de Sistemas a Moodle.

Descripción de la tabla materias

Columns	Data Type	Primary Key	Not Null	AutoInc
Código	int4	si	Si	si
Descripción	varchar(100)	-	Si	-
codigoparalelo	int4	-	Si	-
codigoprofesor	int4	-	Si	-
codigomateriaoasis	int4	-	Si	-
codigomoodle	int4	-	-	-
Indexes	Unique		Columns	
pk_materias	si		codigo	
Constraints	Kind		Columns	
pk_materias	PRIMARY KEY		codigo	

Tabla paralelos:

La tabla adjunta se observa la información de los paralelos que se extrajeron desde el sistema académico y que serán migradas desde el Módulo de Integración de Sistemas a Moodle.

Descripción de la tabla paralelos

Columns	Data Type	Primary Key	Not Null	AutoInc
Código	int4	si	Si	Si
Descripción	varchar(100)	-	-	-
Codigocurso	int4	-	-	-
codigooasisniveles	int4	-	-	-
codigomoodle	int4	-	-	-
Indexes	Unique		Columns	
pk_paralelos	Si		codigo	
Constraints	Kind		Columns	
pk_paralelos	PRIMARY KEY		codigo	

Tabla profesores:

La tabla adjunta se almacena la información de los profesores que se extrajeron desde el sistema académico y que serán migradas desde el Módulo de Integración de Sistemas a Moodle.

Descripción de la tabla profesores

Columns	Data Type	Primary Key	Not Null	AutoInc
Código	int4	si	Si	si
codigoprofesoresoasis	int4	-	Si	-
Cedula	varchar(10)	-	Si	-
Nombres	varchar(100)	-	Si	-
Apellidos	varchar(100)	-	Si	-
Email	varchar(100)	-	Si	-
Pass	varchar(100)	-	Si	-

codigomoodle	int4	-	-	-
Indexes	Unique		Columns	
pk_profesores	si		codigo	
Constraints	Kind		Columns	
pk_profesores	PRIMARY KEY		codigo	

Script de despliegue

En la tabla adjunta se detalla el script de despliegue de la base de datos de integración.

Tabla	Script
alumnos	<pre>CREATE TABLE "Integracion".alumnos (codigo integer NOT NULL DEFAULT nextval('"Integracion".alumnos_codigo_seq'::regclass), codigoalumnosoasis integer NOT NULL, cedula character varying(10) NOT NULL, nombres character varying(100) NOT NULL, apellidos character varying(100) NOT NULL, email character varying(100) NOT NULL, pass character varying(100) NOT NULL, codigoparalelo integer NOT NULL, codigomoodle smallint NOT NULL DEFAULT 0, CONSTRAINT pk_alumnos PRIMARY KEY (codigo), CONSTRAINT fk_alumnos FOREIGN KEY (codigoparalelo) REFERENCES "Integracion".paralelos (codigo) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION)</pre>
anioactivo	<pre>CREATE TABLE "Integracion".anioactivo (codigo integer NOT NULL DEFAULT nextval('"Integracion".anioactivo_codigo_seq'::regclass), codigooasisanio integer NOT NULL, descripcion character varying(200) NOT NULL, estado smallint NOT NULL DEFAULT 0, seccion integer DEFAULT 0, inicio character varying(100), fin character varying(100), CONSTRAINT pk_anioactivo PRIMARY KEY (codigo))</pre>
categoriabase	<pre>CREATE TABLE "Integracion".categoriabase (</pre>

	<pre> codigo integer NOT NULL, descripcion character varying(10), base character varying(10), contextbase character varying(10), basica character varying(10), contextbasica character varying(10), bachillerato character varying(10), contextbachillerato character varying(10), CONSTRAINT pk_categoriabase PRIMARY KEY (codigo)) </pre>
curso	<pre> CREATE TABLE "Integracion".cursos (codigo integer NOT NULL DEFAULT nextval("Integracion".cursos_codigo_seq'::regclass), descripcion character varying(100), seccion smallint NOT NULL DEFAULT 0, codigomoodle integer DEFAULT 0, codigoaniolectivo integer DEFAULT 0, CONSTRAINT pk_area PRIMARY KEY (codigo), CONSTRAINT cursos_codigoaniolectivo_fkey FOREIGN KEY (codigoaniolectivo) REFERENCES "Integracion".anioactivo (codigo) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION) </pre>
materias	<pre> CREATE TABLE "Integracion".materias (codigo integer NOT NULL DEFAULT nextval("Integracion".materias_codigo_seq'::regclass), descripcion character varying(100) NOT NULL, codigoparalelo integer NOT NULL, codigoprofesor integer NOT NULL, codigomateriaoasis integer NOT NULL, codigomoodle integer DEFAULT 0, CONSTRAINT pk_materias PRIMARY KEY (codigo), CONSTRAINT fk2_materias FOREIGN KEY (codigoprofesor) REFERENCES "Integracion".profesores (codigo) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION, CONSTRAINT fk_materias FOREIGN KEY (codigoparalelo) REFERENCES "Integracion".paralelos (codigo) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION) </pre>
paralelos	<pre> CREATE TABLE "Integracion".paralelos </pre>

	<pre>(codigo integer NOT NULL DEFAULT nextval("Integracion".paralelos_codigo_seq::regclass), descripcion character varying(100), codigocurso integer, codigooasisniveles integer, codigomoodle integer DEFAULT 0, CONSTRAINT pk_paralelos PRIMARY KEY (codigo), CONSTRAINT fk_paralelos FOREIGN KEY (codigocurso) REFERENCES "Integracion".cursos (codigo) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION)</pre>
profesores	<pre>CREATE TABLE "Integracion".profesores (codigo integer NOT NULL DEFAULT nextval("Integracion".profesores_codigo_seq::regclass), codigoprofesoresoasis integer NOT NULL, cedula character varying(10) NOT NULL, nombres character varying(100) NOT NULL, apellidos character varying(100) NOT NULL, email character varying(100) NOT NULL, pass character varying(100) NOT NULL, codigomoodle integer DEFAULT 0, CONSTRAINT pk_profesores PRIMARY KEY (codigo))</pre>

Metáfora 5

En el desarrollo de esta metáfora se definieron los estándares de arquitectura de hardware y software que se utilizó para el desarrollo e instalación de la aplicación. En la tabla adjunta se detalla el desarrollo de la metáfora

Definición de arquitectura de hardware y software

Historia de usuario	
Número: M5	Nombre historia: Definición de arquitectura de hardware y software
Usuario: Arquitecto	Iteración Asignada: 3

Prioridad en Negocio: Alta	Puntos Estimado: 4
Riesgo en Desarrollo: Medio	Puntos Reales: 4
Descripción: Como arquitecto deseo que el diseño arquitectónico de la aplicación se base en estándares ordenados, para que todo el procesamiento y extracción de datos mantenga el flujo correcto y se pueda optimizar código y reutilizar funciones, etc.	

Cronograma de tareas de ingeniería

Para cumplir con la fecha establecida por la iteración, en la tabla adjunta se establece el cronograma y tarea de ingeniería a desarrollarse.

Cronograma de tareas de ingeniería metáfora 5

Tarea de ingeniería (código y nombre)		Puntos estimados	Puntos reales	Fecha inicio	Fecha fin
E-1	Definición de arquitectura de hardware y software	4	4	28/03/2014	31/03/2014

Tarea de ingeniería E-1

Tarea de Ingeniería	
Número Tarea: E-1	Historia de Usuario: : Definición de arquitectura de hardware y software
Nombre Tarea: : Definición de arquitectura de hardware y software	
Tipo Tarea: Creación	Puntos estimados: 4
Fecha Inicio: 28/03/2014	Fecha Fin: 31/03/2014
Analista Responsable: José Pinto	
Descripción: Definición de arquitectura de hardware y software para el desarrollo de la aplicación.	

Metáfora 6

En el desarrollo de esta metáfora se definieron los estándares de interfaces de usuario y estándares de codificación que se utilizaron en la aplicación. En la se detalla el desarrollo de la metáfora.

Definición de arquitectura de despliegue y software

Historia de usuario	
Número: M6	Nombre historia: Definición de interfaces de usuario y estándar de codificación
Usuario: Programador	Iteración Asignada: 3
Prioridad en Negocio: Alta	Puntos Estimado: 4
Riesgo en Desarrollo: Medio	Puntos Reales: 4
Descripción: Como programador deseo definir el diseño de interfaces de usuario y el estándar de codificación, para que la aplicación tenga vistosidad en la presentación y correcto funcionamiento.	

Cronograma de tareas de ingeniería

Para cumplir con la fecha establecida por la iteración, en la tabla adjunta se establece el cronograma y tarea de ingeniería a desarrollarse.

Cronograma de tareas de ingeniería metáfora 6

Tarea de ingeniería (código y nombre)		Puntos estimados	Puntos reales	Fecha inicio	Fecha fin
F-1	Definición de interfaces de usuario y estándar de codificación	4	4	31/03/2014	01/04/2014

Tarea de ingeniería F-1

Tarea de Ingeniería	
Número Tarea: F-1	Historia de Usuario: : Definición de interfaces de usuario y estándar de codificación
Nombre Tarea: : Definición de interfaces de usuario y estándar de codificación	
Tipo Tarea: Creación	Puntos estimados: 4
Fecha Inicio: 31/03/2014	Fecha Fin: 01/04/2014
Analista Responsable: José Pinto	
Descripción: Definición de interfaces de usuario y estándar de codificación	

Iteración 4

Metáfora 7

La tabla adjunta se refiere a la creación del servicio web que funciona como un intermediario entre la base de datos del sistema de calificación del docente y las aplicaciones que requieran información.

Creación servicio web

Historia de Usuario	
Número: M7	Nombre historia: Creación de servicio web para publicar información desde la base de datos del sistema académico de la institución
Usuario: Programador	Interacción Asignada: 4
Prioridad en Negocio: Alta	Puntos Estimado: 8
Riesgo en Desarrollo: Media	Puntos Reales: 8
Descripción: Como programador deseo publicar información de estudiantes, docentes, cursos, materias, cursos y paralelos, para poder utilizar esta información en otras aplicaciones	

Cronograma de tareas de ingeniería

Para cumplir con la fecha establecida por la iteración, en la tabla adjunta se establece el cronograma y tarea de ingeniería a desarrollarse.

Cronograma de tareas de ingeniería metáfora 7

Tarea de ingeniería (código y nombre)		Puntos estimados	Puntos reales	Fecha inicio	Fecha fin
G-1	Creación de métodos para extracción masiva de información	4	4	01/04/2014	02/04/2014
G-2	Creación de métodos para extracción de información mediante un parámetro	4	4	02/04/2014	03/04/2014

Tarea de ingeniería G-1

Tarea de Ingeniería	
Número Tarea: G-1	Historia de Usuario: Creación de servicio web para publicar información desde la base de datos del sistema académico de la institución
Nombre Tarea: Creación de métodos para extracción masiva de información	
Tipo Tarea: Creación	Puntos Estimado: 4
Fecha Inicio: 01/04/2014	Fecha Fin: 02/04/2014
Analista Responsable: José Pinto	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en Oracle, Crear paquetes en Oracle, Crear métodos en Java, Crear clase en Java, registrar excepciones.	

Tarea de ingeniería G-2

Tarea de Ingeniería	
Número Tarea: G-2	Historia de Usuario: Creación de servicio web para publicar información desde la base de datos del sistema académico de la institución
Nombre Tarea: Creación de métodos para extracción de información mediante un parámetro	
Tipo Tarea: Creación	Puntos Estimado: 4
Fecha Inicio: 02/04/2014	Fecha Fin: 03/04/2014
Analista Responsable: José Pinto	
<p>Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente:</p> <p>Crear funciones en Oracle, Crear paquetes en Oracle, Crear métodos en Java, Crear clase en Java, registrar excepciones.</p>	

Pruebas de aceptación

Prueba de Aceptación metáfora 7

Caso de Prueba de Aceptación	
Número caso de Prueba: 3	Historia de Usuario: Creación de servicio web para publicar información desde la base de datos del sistema académico de la institución
Nombre: Prueba de extracción de datos mediante aplicación consola	
Descripción: Se probara la extracción de informacion desde el sistema academico del colegio a través del servicio web que publica informacion de estudiantes, docentes, etc.	
Condiciones de Ejecución: No es necesario tener condiciones especiales para ejecutar el programa consola.	
<p>Entrada / Pasos de Ejecución:</p> <ol style="list-style-type: none"> 1. Ingresar la cedula de identidad de un estudiante 	
Resultados esperados 1: El programa consola muestra la informacion del estudiante que se extrajo a partir del servicio web publicado.	

Resultados esperados 2: El programa consola no muestra la información del estudiante ya que no existe en la base de datos del sistema académico.

Resultados esperados 2: El programa consola no muestra la información solicitada ya que el servicio de base de datos no estaba disponible en el momento de la prueba.

Historia de usuario 1

La historia de usuario 1, se refiere a la autenticación de usuarios de sistema de integración de sistemas.

Autenticación de usuarios

Historia de Usuario	
Número: H1	Nombre Historia Usuario: Autenticación de Usuario
Usuario: Administrador	Interacción Asignada: 4
Prioridad en Negocio: Alta	Puntos Estimado: 8
Riesgo en Desarrollo: Medio	Puntos Reales: 8
Descripción: Como administrador deseo que la aplicación permita la autenticación de usuarios previo al ingreso al sistema para evitar el ingreso no permitido.	

Cronograma de tareas de ingeniería

Para cumplir con la fecha establecida por la iteración, en la tabla adjunta se establece el cronograma y tarea de ingeniería a desarrollarse.

Cronograma de tareas de ingeniería historia 1

Tarea de ingeniería (código y nombre)	Puntos	Puntos	Fecha inicio	Fecha fin
---------------------------------------	--------	--------	--------------	-----------

		estimados	reales		
I-1	Generar la validación de carga de perfiles	8	8	03/04/2014	07/04/2014

Tarea de Ingeniería I-1

Tarea de Ingeniería	
Número Tarea: I-1	Historia de Usuario: Autenticación de Usuario
Nombre Tarea: Generar la validación de carga de perfiles	
Tipo Tarea: Creación	Puntos Estimado: 8
Fecha Inicio: 03/04/2014	Fecha Fin: 07/04/2014
Analista Responsable: José Pinto	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones.	

Interfaz de usuario

Usuario:

Contraseña:

[Olvido su contraseña?](#)

Inicio de sesión

Pruebas de aceptación

Prueba de Aceptación historia 1

Caso de Prueba de Aceptación	
Número caso de Prueba: 4	Historia de Usuario: Autenticación de Usuario
Nombre: Prueba de ingreso de usuario a la aplicación web	
Descripción: Se probará el acceso al menú principal de la aplicación	
Condiciones de Ejecución: El usuario tiene que estar asignado a un grupo con permisos para el acceso a la aplicación.	
Entrada / Pasos de Ejecución:	
<ol style="list-style-type: none"> 1. El administrador ingresa a la página principal del aplicativo que se encuentra en el siguiente link (http://186.42.198.158:8080/Integracion/login.jsf) 2. El administrador ingresa su nombre de usuario, contraseña y luego selecciona el botón Autenticar. 	
Resultados esperados 1:	
<ol style="list-style-type: none"> 1. El sistema no permite el ingreso a la pagina principal del sitio despliega error (Se produjo un error comuníquese con el administrador del sistema!!) 2. El error se produce ya que la aplicación no estableció conexión con la base de datos. 	
Resultados esperados 2:	
<ol style="list-style-type: none"> 1. El sistema no permite el ingreso a la pagina principal del sitio despliega error (No existe el usuario y contraseña ingresados!!). 2. El error se produce ya que el usuario ingreso incorrectamente las credenciales. 	
Resultados esperados 3:	
<ol style="list-style-type: none"> 1. El sistema no permite el ingreso a la pagina principal del sitio, despliega el error (La aplicación se encuentra deshabilitada para este usuario!!) 2. El error se produce ya que el usuario no está asignado a un grupo con privilegios para esta aplicación. 	
Resultados esperados 4:	
<ol style="list-style-type: none"> 1. El sistema permite el ingreso a la página principal del sitio, pero no despliega el menú con opciones. 2. El error se da ya que el grupo al que pertenece el usuario no tiene asignado los permisos. 	

Resultados esperados 5:

1. El sistema permite el ingreso al usuario y despliega correctamente las opciones de la página principal del sitio.

Historia de usuario 2

Recuperación de credenciales de usuario.

Historia de Usuario	
Número: H2	Nombre Historia Usuario: Recuperación de Credenciales de usuario
Usuario: Administrador	Interacción Asignada: 4
Prioridad en Negocio: Alta	Puntos Estimado: 4
Riesgo en Desarrollo: Medio	Puntos Reales: 4
Descripción: Como administrador deseo que la aplicación permita recuperar las credenciales mediante él envió a un correo electrónico para ingresar al sistema con la nueva clave creada.	

Cronograma de tareas de ingeniería

Para cumplir con la fecha establecida por la iteración, en la tabla adjunta se establece el cronograma y tarea de ingeniería a desarrollarse.

Cronograma de tareas de ingeniería historia 2

Tarea de ingeniería (código y nombre)	Puntos estimados	Puntos reales	Fecha inicio	Fecha fin
J-1 Generar métodos para la recuperación de credenciales	4	4	07/04/2014	08/04/2014

Tarea de Ingeniería J-1

Tarea de Ingeniería	
Numero Tarea: J-1	Historia de Usuario: Recuperación de Credenciales de usuario

Nombre Tarea: Generar métodos para la recuperación de credenciales	
Tipo Tarea: Creación	Puntos Estimado: 4
Fecha Inicio: 07/04/2014	Fecha Fin: 08/04/2014
Analista Responsable: José Pinto	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones.	

Interfaz de la página de recuperación de clave

Olvido su contraseña?

Correo electrónico:

Usuario:

Prueba de aceptación

Prueba de Aceptación historia 2

Caso de Prueba de Aceptación	
Número caso de Prueba: 5	Historia de Usuario: Recuperación de Credenciales de usuario
Nombre: Prueba Recuperación de Credenciales de usuario	
Descripción: Se probará la recuperación de credenciales del usuario	
Condiciones de Ejecución: El usuario debe tener registrado en el sistema un correo electrónico ya que la clave se le enviara al mismo.	

<p>Entrada / Pasos de Ejecución:</p> <ol style="list-style-type: none"> 1. El usuario administrador ingresa a la página principal del aplicativo que se encuentra en el siguiente link (http://186.42.198.158:8080/Integracion/login.jsf) 2. El usuario administrador selecciona el Link que indica Olvido su contraseña. 3. El usuario administrador ingresa su correo electrónico, nombre de usuario y luego presiona el botón Enviar. 4. El usuario administrador ingresa a su cuenta de correo electrónico y copia la clave que se le envió, con esta contraseña ingresa nuevamente al sistema.
<p>Resultados esperados 1:</p> <ol style="list-style-type: none"> 1. El sistema no permite el envío del correo electrónico, el error que despliega es (Se produjo un error por favor comuníquese con el administrador del sistema!!). 2. El sistema no permite el envío del correo ya que el usuario no tiene registrado ningún correo electrónico válido en la base de datos.
<p>Resultados esperados 2:</p> <ol style="list-style-type: none"> 1. El sistema envía el correo electrónico con las nuevas credenciales, y luego permite el ingreso a la página principal del aplicativo.

Iteración 5

Historia de usuario 3

La tabla adjunta se refiere a la gestión de usuarios para el manejo del sistema de integración y administración de aplicaciones.

Gestión de usuarios

Historia de Usuario	
Número: H3	Nombre Historia Usuario: Gestión de Usuarios
Usuario: Administrador	Interacción Asignada: 5
Prioridad en Negocio: Media	Puntos Estimado: 8
Riesgo en Desarrollo: Bajo	Puntos Reales: 10

Descripción: Como administrador deseo que la aplicación permita crear, modificar y eliminar usuarios para que puedan ingresar a las aplicaciones con las que cuenta la institución.

Cronograma de tareas de ingeniería

Para cumplir con la fecha establecida por la iteración, en la tabla adjunta se establece el cronograma y tarea de ingeniería a desarrollarse.

Cronograma de tareas de ingeniería historia 3

Tarea de ingeniería (código y nombre)		Puntos estimados	Puntos reales	Fecha inicio	Fecha fin
K-1	Mostrar listado de usuarios	1	2	08/04/2014	09/04/2014
K-2	Insertar Usuarios	2	2	09/04/2014	09/04/2014
K-3	Editar Usuarios	2	2	09/04/2014	10/04/2014
K-4	Buscar Usuarios	1	2	10/04/2014	10/04/2014
K-5	Eliminar Usuarios	2	2	10/04/2014	11/04/2014

Tarea de Ingeniería k-1

Tarea de Ingeniería	
Número Tarea: k-1	Historia de Usuario: Gestión de Usuarios
Nombre Tarea: Mostrar listado de usuarios	
Tipo Tarea: Creación	Puntos Estimado: 1
Fecha Inicio: 08/04/2014	Fecha Fin: 09/04/2014
Analista Responsable: Amanda Adriano	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar	

excepciones.

Tarea de Ingeniería k-2

Tarea de Ingeniería	
Número Tarea: k-2	Historia de Usuario: Gestión de Usuarios
Nombre Tarea: Insertar Usuarios	
Tipo Tarea: Creación	Puntos Estimado: 2
Fecha Inicio: 09/04/2014	Fecha Fin: 09/04/2014
Analista Responsable: Amanda Adriano	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones.	

Tarea de Ingeniería k-3

Tarea de Ingeniería	
Número Tarea: k-3	Historia de Usuario: Gestión de Usuarios
Nombre Tarea: Editar Usuarios	
Tipo Tarea: Creación	Puntos Estimado: 2
Fecha Inicio: 09/04/2014	Fecha Fin: 10/04/2014
Analista Responsable: Amanda Adriano	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones.	

Tarea de Ingeniería k-4

Tarea de Ingeniería

Número Tarea: k-4	Historia de Usuario: Gestión de Usuarios	
Nombre Tarea: Buscar Usuarios		
Tipo Tarea: Creación	Puntos Estimado: 1	
Fecha Inicio: 10/04/2014	Fecha Fin: 10/04/2014	
Analista Responsable: Amanda Adriano		
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones.		

Tarea de Ingeniería k-5

Tarea de Ingeniería		
Número Tarea: k-5	Historia de Usuario: Gestión de Usuarios	
Nombre Tarea: Eliminar Usuarios		
Tipo Tarea: Creación	Puntos Estimado: 2	
Fecha Inicio: 10/04/2014	Fecha Fin: 11/04/2014	
Analista Responsable: Amanda Adriano		
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones.		

Interfaz de usuario de la pantalla de agregar usuarios

La tabla adjunta muestra las interfaces para la administración de usuarios, las opciones editar y eliminar seguirán el mismo formato de la interfaz Nuevo Usuario.

Descripción	Interfaz Grafica
Menú principal opción	

<p>Usuarios</p>	<p>Usuarios</p> <p> <input type="button" value="Nuevo Usuario"/> <input type="button" value="Nuevo Cargo"/> <input type="button" value="Nuevo Área"/> </p> <table border="1"> <thead> <tr> <th>Código</th> <th>Cedula</th> <th>Nombres</th> <th>Apellidos</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table> <p> <input type="button" value="Editar"/> <input type="button" value="Eliminar"/> <input type="button" value="Asignar a"/> </p>	Código	Cedula	Nombres	Apellidos				
Código	Cedula	Nombres	Apellidos						
<p>Opción Nuevo Usuario y opción Asignar a</p>	<p>Nuevo Usuario</p> <p> Cedula: <input type="text"/> Nombres: <input type="text"/> Apellidos: <input type="text"/> Email: <input type="text"/> Área: <input type="text" value="v"/> Estado: <input type="text" value="v"/> Cargos: <input type="text" value="v"/> </p> <p> <input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/> </p> <p>Asignar a</p> <p>Miembro de : <input type="text" value="v"/></p> <p> <input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/> </p>								
<p>Opción Nuevo cargo y Nueva Área</p>	<p>Nuevo Cargo</p> <p> Descripción : <input type="text"/> Estado: <input type="text" value="v"/> </p> <p> <input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/> </p> <p>Nuevo Área</p> <p> Descripción : <input type="text"/> Estado: <input type="text" value="v"/> </p> <p> <input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/> </p>								

Prueba de aceptación

Prueba de Aceptación historia 3

Caso de Prueba de Aceptación	
Número caso de Prueba: 6	Historia de Usuario: Gestión de Usuarios
Nombre: Prueba de las opciones de la gestion de usuarios	

Descripción: Se probara lo siguiente: mostrar, insertar, buscar, editar y eliminar usuarios.
Condiciones de Ejecución: El usuario tiene que estar asignado a un grupo con permisos de administrador.
Entrada / Pasos de Ejecución: <ol style="list-style-type: none"> 1. El administrador ingresa a la página principal del aplicativo que se encuentra en el siguiente link (http://186.42.198.158:8080/Integracion/login.jsf) 2. El administrador ingresa su nombre de usuario, contraseña y luego selecciona el botón Autenticar. 3. El administrador selecciona el menú administracion y luego selecciona la opción usuarios. 4. El administrador para ingresar un nuevo usuario debe dar click sobre el botón Nuevo usuario, le desplegara una pantalla con los campos que deben ser ingresados, luego presionar el botón aceptar. 5. El administrador para editar usuarios debe seleccionar de la lista el usuario a ser editado y luego dar click en el botón editar, se desplegara una pantalla en la cual se visualizaran los campos a ser modificados, luego presionar el botón aceptar. 6. El administrador para eliminar usuarios debe seleccionar de la lista el usuario a ser eliminado y luego dar click en el botón eliminar, se desplegara una pantalla en la cual se solicitara autorización para eliminar y luego presionar el botón aceptar, tener en cuenta que para eliminar usuarios estos no deben estar asignado a un grupo.
Resultados esperados 1: <ol style="list-style-type: none"> 1. El sistema cumple satisfactoriamente con todas las pruebas realizadas anteriormente.

Historia de usuario 4

Gestión de Grupos

Historia de Usuario	
Número: H4	Nombre Historia Usuario: Gestión de Grupos
Usuario: Administrador	Interacción Asignada: 5
Prioridad en Negocio: Media	Puntos Estimado: 12

Riesgo en Desarrollo: Bajo	Puntos Reales: 14
Descripción: Como administrador deseo que la aplicación permita crear, modificar y eliminar grupos para asignar a los usuarios que puedan ingresar a las aplicaciones.	

Cronograma de tareas de ingeniería

Para cumplir con la fecha establecida por la iteración, en la tabla adjunta se establece el cronograma y tarea de ingeniería a desarrollarse.

Cronograma de tareas de ingeniería historia 4

Tarea de ingeniería (código y nombre)		Puntos estimados	Puntos reales	Fecha inicio	Fecha fin
L-1	Mostrar listado de grupos	2	2	11/04/2014	11/04/2014
L-2	Insertar grupos	2	3	11/04/2014	14/04/2014
L-3	Editar grupos	2	2	14/04/2014	14/04/2014
L-4	Buscar grupos	2	2	15/04/2014	15/04/2014
L-5	Eliminar grupos	2	2	15/04/2014	15/04/2014
L-6	Asignación de usuarios a Grupos	2	3	16/04/2014	16/04/2014

Tarea de Ingeniería L-1

Tarea de Ingeniería	
Número Tarea: L-1	Historia de Usuario: Gestión de Grupos
Nombre Tarea: Mostrar listado de grupos	
Tipo Tarea: Creación	Puntos Estimado: 2
Fecha Inicio: 11/04/2014	Fecha Fin: 11/04/2014
Analista Responsable: Amanda Adriano	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar	

excepciones.

Tarea de Ingeniería L-2

Tarea de Ingeniería	
Número Tarea: L-2	Historia de Usuario: Gestión de Grupos
Nombre Tarea: Insertar Grupos	
Tipo Tarea: Creación	Puntos Estimado: 3
Fecha Inicio: 11/04/2014	Fecha Fin: 14/04/2014
Analista Responsable: Amanda Adriano	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones.	

Tarea de Ingeniería L-3

Tarea de Ingeniería	
Número Tarea: L-3	Historia de Usuario: Gestión de Grupos
Nombre Tarea: Editar Grupos	
Tipo Tarea: Creación	Puntos Estimado: 2
Fecha Inicio: 14/04/2014	Fecha Fin: 14/04/2014
Analista Responsable: Amanda Adriano	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones.	

Tarea de Ingeniería L-4

Tarea de Ingeniería

Número Tarea: L-4	Historia de Usuario: Gestión de Grupos	
Nombre Tarea: Buscar Grupos		
Tipo Tarea: Creación	Puntos Estimado: 2	
Fecha Inicio: 15/04/2014	Fecha Fin: 15/04/2014	
Analista Responsable: Amanda Adriano		
<p>Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente:</p> <p>Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones.</p>		

Tarea de Ingeniería L-5

Tarea de Ingeniería		
Número Tarea: L-5	Historia de Usuario: Gestión de Grupos	
Nombre Tarea: Eliminar Grupos		
Tipo Tarea: Creación	Puntos Estimado: 2	
Fecha Inicio: 15/04/2014	Fecha Fin: 15/04/2014	
Analista Responsable: Amanda Adriano		
<p>Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente:</p> <p>Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones.</p>		

Tarea de Ingeniería L-6

Tarea de Ingeniería		
Número Tarea: L-6	Historia de Usuario: Gestión de Grupos	
Nombre Tarea: Asignación de usuarios a Grupos		
Tipo Tarea: Creación	Puntos Estimado: 3	
Fecha Inicio: 16/04/2014	Fecha Fin: 16/04/2014	

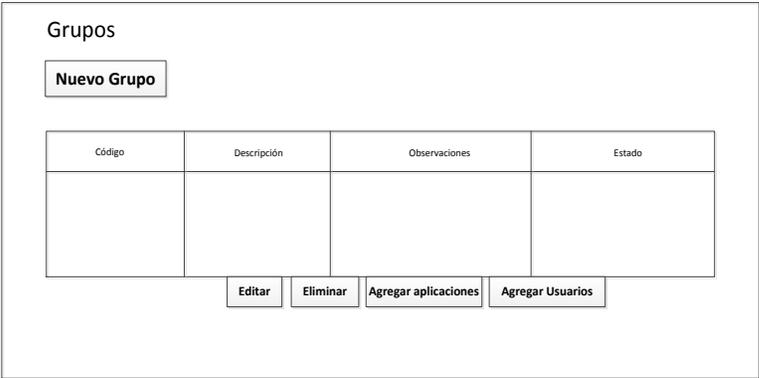
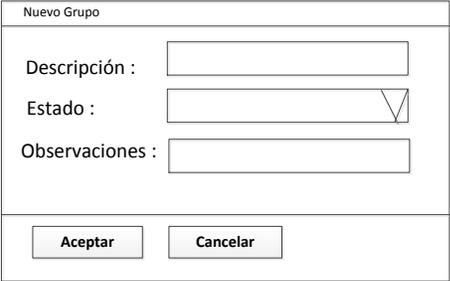
Analista Responsable: Amanda Adriano

Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente:

Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones.

Interfaz de usuario de la página de gestión de grupos

La tabla adjunta muestra las interfaces para la administración de Grupos, las opciones editar y eliminar seguirán el mismo formato de la interfaz Nuevo Grupo.

Descripción	Interfaz Grafica
Menú principal opción Grupos	
Opción Nuevo Grupo	

<p>Opción Agregar Aplicaciones y agregar usuarios</p>	<p>Aplicaciones</p> <div style="display: flex; justify-content: space-between;"> <div style="border: 1px solid black; width: 45%; height: 60px; margin-bottom: 5px;">Desactivadas para el grupo</div> <div style="border: 1px solid black; width: 45%; height: 60px; margin-bottom: 5px;">Activadas para el grupo</div> </div> <div style="display: flex; justify-content: center; gap: 10px; margin: 5px 0;"> -> -> < < </div> <div style="display: flex; justify-content: center; gap: 20px; margin-top: 5px;"> Aceptar Cancelar </div>
	<p>Agregar Usuarios</p> <div style="display: flex; justify-content: space-between;"> <div style="border: 1px solid black; width: 45%; height: 60px; margin-bottom: 5px;">Desactivadas para el grupo</div> <div style="border: 1px solid black; width: 45%; height: 60px; margin-bottom: 5px;">Activadas para el grupo</div> </div> <div style="display: flex; justify-content: center; gap: 10px; margin: 5px 0;"> -> -> < < </div> <div style="display: flex; justify-content: center; gap: 20px; margin-top: 5px;"> Aceptar Cancelar </div>

Prueba de aceptación

Prueba de Aceptación historia 4

Caso de Prueba de Aceptación	
Número caso de Prueba: 7	Historia de Usuario: Gestión de Grupos
Nombre: Prueba de las opciones de la gestion de grupos	
Descripción: Se probara lo siguiente: mostrar, insertar, buscar, editar, eliminar y asignación de usuarios a grupos.	
Condiciones de Ejecución: El usuario tiene que estar asignado a un grupo con permisos de administrador.	
Entrada / Pasos de Ejecución:	
<ol style="list-style-type: none"> 1. El administrador ingresa a la página principal del aplicativo que se encuentra en el siguiente link (http://186.42.198.158:8080/Integracion/login.jsf) 2. El administrador ingresa su nombre de usuario, contraseña y luego selecciona el botón Autenticar. 	

3. El administrador selecciona el menú administracion y luego selecciona la opción grupos.
4. El administrador para ingresar un nuevo grupo debe dar click sobre el botón Nuevo grupo, le desplegara una pantalla con los campos que deben ser ingresados, luego presionar el botón aceptar.
5. El administrador para editar grupos debe seleccionar de la lista el grupo a ser editado y luego dar click en el botón editar, se desplegara una pantalla en la cual se visualizaran los campos a ser modificados, luego presionar el botón aceptar.
6. El administrador para eliminar grupos debe seleccionar de la lista el grupo a ser eliminado y luego dar click en el botón eliminar, se desplegara una pantalla en la cual se solicitara autorización para eliminar y luego presionar el botón aceptar, tener en cuenta que para eliminar grupos estos no deben estar asignado a una aplicación.

Resultados esperados 1:

1. El sistema cumple satisfactoriamente con todas las pruebas realizadas anteriormente.

Iteración 6

Historia de usuario 5

La historia de usuario 5, se refiere a la creación, modificación y eliminación de aplicaciones que se administrarán desde el Módulo de Integración de Aplicaciones.

Gestión de Aplicaciones

Historia de Usuario	
Número: H5	Nombre Historia Usuario: Gestión de Aplicaciones
Usuario: Administrador	Interacción Asignada: 6
Prioridad en Negocio: Media	Puntos Estimado: 10
Riesgo en Desarrollo: Baja	Puntos Reales: 12

Descripción: Como administrador deseo que la aplicación permita crear, modificar y eliminar aplicaciones para que los usuarios de un determinado grupo puedan administrar las aplicaciones.

Cronograma de tareas de ingeniería

Para cumplir con la fecha establecida por la iteración, en la tabla adjunta se establece el cronograma y tarea de ingeniería a desarrollarse.

Cronograma de tareas de ingeniería historia 5

Tarea de ingeniería (código y nombre)		Puntos estimados	Puntos reales	Fecha inicio	Fecha fin
O-1	Mostrar listado de aplicaciones disponibles	1	2	16/04/2014	17/04/2014
O-2	Insertar aplicaciones	2	2	17/04/2014	17/04/2014
O-3	Editar aplicaciones	2	2	17/04/2014	18/04/2014
O-4	Buscar aplicaciones	1	2	18/04/2014	18/04/2014
O-5	Eliminar aplicaciones	2	2	18/04/2014	21/04/2014
O-6	Asignación de Grupos a Aplicaciones	2	2	21/04/2014	21/04/2014

Tarea de Ingeniería O-1

Tarea de Ingeniería	
Número Tarea: O-1	Historia de Usuario: Gestión de Aplicaciones
Nombre Tarea: Mostrar listado de aplicaciones disponibles.	
Tipo Tarea: Creación	Puntos Estimado: 2
Fecha Inicio: 16/04/2014	Fecha Fin: 17/04/2014
Analista Responsable: Amanda Adriano	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar	

excepciones.

Tarea de Ingeniería O-2

Tarea de Ingeniería	
Número Tarea: O-2	Historia de Usuario: Gestión de Aplicaciones
Nombre Tarea: Insertar Aplicaciones	
Tipo Tarea: Creación	Puntos Estimado: 2
Fecha Inicio: 17/04/2014	Fecha Fin: 17/04/2014
Analista Responsable: Amanda Adriano	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones.	

Tarea de Ingeniería O-3

Tarea de Ingeniería	
Número Tarea: O-3	Historia de Usuario: Gestión de Aplicaciones
Nombre Tarea: Editar Aplicaciones	
Tipo Tarea: Creación	Puntos Estimado: 2
Fecha Inicio: 17/04/2014	Fecha Fin: 18/04/2014
Analista Responsable: Amanda Adriano	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones.	

Tarea de Ingeniería O-4

Tarea de Ingeniería	
Número Tarea: O-4	Historia de Usuario: Gestión de Aplicaciones
Nombre Tarea: Buscar Aplicaciones	
Tipo Tarea: Creación	Puntos Estimado: 2
Fecha Inicio: 18/04/2014	Fecha Fin: 18/04/2014
Analista Responsable: Amanda Adriano	
<p>Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente:</p> <p>Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones.</p>	

Tarea de Ingeniería O-5

Tarea de Ingeniería	
Número Tarea: O-5	Historia de Usuario: Gestión de Aplicaciones
Nombre Tarea: Eliminar Aplicaciones	
Tipo Tarea: Creación	Puntos Estimado: 2
Fecha Inicio: 18/04/2014	Fecha Fin: 21/04/2014
Analista Responsable: Amanda Adriano	
<p>Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente:</p> <p>Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones.</p>	

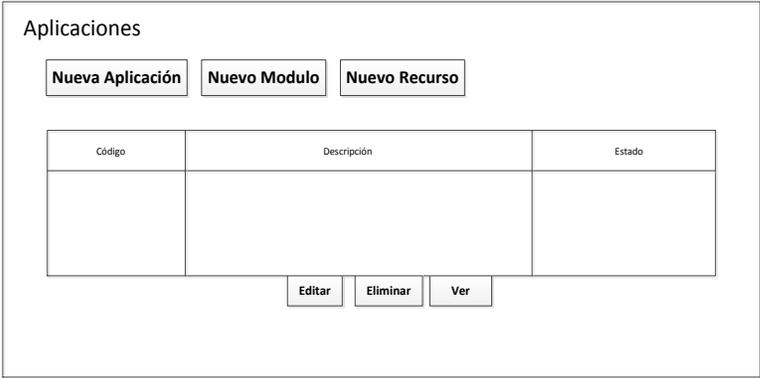
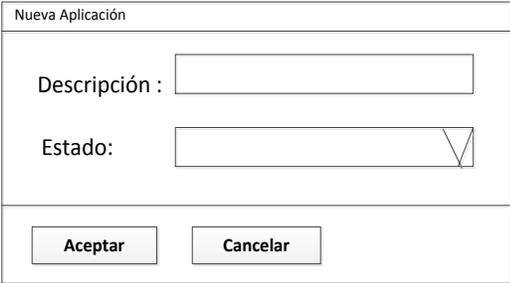
Tarea de Ingeniería O-6

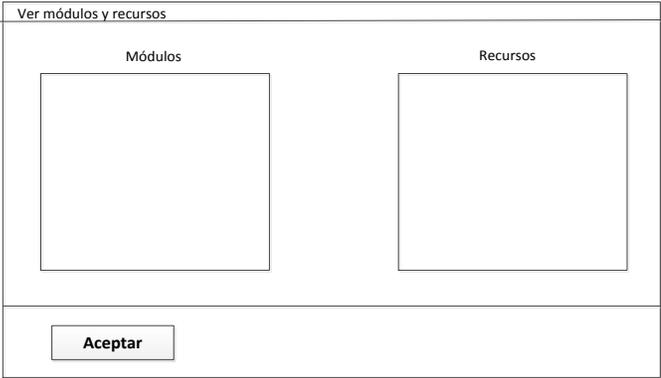
Tarea de Ingeniería	
Número Tarea: O-6	Historia de Usuario: Gestión de Aplicaciones

Nombre Tarea: Asignación de Grupos a Aplicaciones	
Tipo Tarea: Creación	Puntos Estimado: 2
Fecha Inicio: 21/04/2014	Fecha Fin: 21/04/2014
Analista Responsable: Amanda Adriano	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones.	

Interfaz de usuario de la página de gestión de aplicaciones

La tabla adjunta muestra las interfaces para la administración de Aplicaciones, las opciones editar y eliminar seguirán el mismo formato de la interfaz Nueva aplicación.

Descripción	Interfaz Grafica
Menú principal opción Aplicaciones	
Opción Nueva Aplicación	

Opción ver módulos y recursos	
-------------------------------	--

Prueba de aceptación

Prueba de Aceptación historia 5

Caso de Prueba de Aceptación	
Número caso de Prueba: 8	Historia de Usuario: Gestión de Aplicaciones
Nombre: Prueba de las opciones de la administración de aplicaciones	
Descripción: Se probara lo siguiente: mostrar, insertar, buscar, editar, eliminar y asignación de grupos a aplicaciones	
Condiciones de Ejecución: El usuario tiene que estar asignado a un grupo con permisos de administrador.	
Entrada / Pasos de Ejecución:	
<ol style="list-style-type: none"> 1. El administrador ingresa a la página principal del aplicativo que se encuentra en el siguiente link (http://186.42.198.158:8080/Integracion/login.jsf) 2. El administrador ingresa su nombre de usuario, contraseña y luego selecciona el botón Autenticar. 3. El administrador selecciona el menú administracion y luego selecciona la opción aplicaciones. 4. El administrador para ingresar una nueva aplicación debe dar click sobre el botón Nueva aplicación, le desplegara una pantalla con los campos que deben ser ingresados, luego presionar el botón aceptar. 5. El administrador para editar aplicaciones debe seleccionar de la lista la aplicación a ser editada y luego dar click en el botón editar, se desplegara una pantalla en la cual se visualizaran los campos a ser modificados, luego presionar el botón aceptar. 	

6. El administrador para eliminar aplicaciones debe seleccionar de la lista la aplicación a ser eliminada y luego dar click en el botón eliminar, se desplegara una pantalla en la cual se solicitara autorización para eliminar y luego presionar el botón aceptar, tener en cuenta que para eliminar aplicaciones estas no deben tener asignado modulos y tampoco recursos.

Resultados esperados 1:

1. El sistema cumple satisfactoriamente con todas las pruebas realizadas anteriormente.

Historia de usuario 6

Gestión de Semillas para aplicación

Historia de Usuario	
Número: H6	Nombre Historia Usuario: Gestión de Semillas para aplicación
Usuario: Administrador	Interacción Asignada: 6
Prioridad en Negocio: Media	Puntos Estimado: 2
Riesgo en Desarrollo: Baja	Puntos Reales: 4
Descripción: Como administrador deseo que la aplicación permita crear y eliminar la semilla para encriptar las claves de los usuarios.	

Cronograma de tareas de ingeniería

Para cumplir con la fecha establecida por la iteración, en la tabla adjunta se establece el cronograma y tarea de ingeniería a desarrollarse.

Cronograma de tareas de ingeniería historia 6

Tarea de ingeniería (código y nombre)	Puntos estimados	Puntos reales	Fecha inicio	Fecha fin
P-1 Crear semilla para la encriptación de las claves de los usuarios	2	4	21/04/2014	22/04/2014

Tarea de Ingeniería P-1

Tarea de Ingeniería	
Número Tarea: P-1	Historia de Usuario: Gestión de Semillas para aplicación
Nombre Tarea: Asignación de Grupos a Aplicaciones	
Tipo Tarea: Creación	Puntos Estimado: 2
Fecha Inicio: 21/04/2014	Fecha Fin: 22/04/2014
Analista Responsable: Amanda Adriano	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones.	

Prueba de aceptación

Prueba de Aceptación historia 6

Caso de Prueba de Aceptación	
Número caso de Prueba: 9	Historia de Usuario: Gestión de Semillas para aplicación
Nombre: Prueba de creación de semilla y eliminación de semilla	
Descripción: Se probará la creación y eliminación de una semilla cuando se cree o se elimine una aplicación.	
Condiciones de Ejecución: El usuario tiene que estar asignado a un grupo con permisos de administrador.	
Entrada / Pasos de Ejecución: 1. El administrador ingresa a la página principal del aplicativo que se encuentra en el siguiente link (http://186.42.198.158:8080/Integracion/login.jsf)	

2. El administrador ingresa su nombre de usuario, contraseña y luego selecciona el botón Autenticar.
3. El administrador selecciona el menú administracion y luego selecciona la opción aplicaciones.
4. El administrador para ingresar una nueva aplicación debe dar click sobre el botón Nueva aplicación, le desplegara una pantalla con los campos que deben ser ingresados, luego presionar el botón aceptar.
5. El administrador para eliminar aplicaciones debe seleccionar de la lista la aplicación a ser eliminada y luego dar click en el botón eliminar, se desplegara una pantalla en la cual se solicitara autorización para eliminar y luego presionar el botón aceptar, tener en cuenta que para eliminar aplicaciones estas no deben tener asignado modulos y tampoco recursos.

Resultados esperados 1:

1. La semilla se genera satisfactoriamente al momento de crear una nueva aplicación.
2. La semilla se elimina satisfactoriamente al momento de eliminar una aplicación.

Historia de usuario 7

Gestión de Módulos

Historia de Usuario	
Número: H7	Nombre Historia Usuario: Gestión de Módulos.
Usuario: Administrador	Interacción Asignada: 6
Prioridad en Negocio: Media	Puntos Estimado: 8
Riesgo en Desarrollo: Baja	Puntos Reales: 10
Descripción: Como administrador deseo que la aplicación permita crear, modificar y eliminar módulos para tener una estructura organizada de recursos.	

Cronograma de tareas de ingeniería

Para cumplir con la fecha establecida por la iteración, en la tabla adjunta se establece el cronograma y tarea de ingeniería a desarrollarse.

Cronograma de tareas de ingeniería historia 7

Tarea de ingeniería (código y nombre)		Puntos estimados	Puntos reales	Fecha inicio	Fecha fin
Q-1	Mostar listado de módulos	1	2	22/04/2014	23/04/2014
Q-2	Insertar módulos	2	2	23/04/2014	23/04/2014
Q-3	Editar módulos	2	2	23/04/2014	24/04/2014
Q-4	Buscar módulos	1	2	24/04/2014	24/04/2014
Q-5	Eliminar módulos	2	2	24/04/2014	25/04/2014

Tarea de Ingeniería Q-1

Tarea de Ingeniería	
Número Tarea: Q-1	Historia de Usuario: Gestión de Módulos
Nombre Tarea: Mostar listado de módulos	
Tipo Tarea: Creación	Puntos Estimado: 1
Fecha Inicio: 22/04/2014	Fecha Fin: 23/04/2014
Analista Responsable: Amanda Adriano	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones.	

Tarea de Ingeniería Q-2

Tarea de Ingeniería	
Número Tarea: Q-2	Historia de Usuario: Gestión de Módulos
Nombre Tarea: Insertar Módulos	

Tipo Tarea: Creación	Puntos Estimado: 2
Fecha Inicio: 23/04/2014	Fecha Fin: 23/04/2014
Analista Responsable: Amanda Adriano	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones.	

Tarea de Ingeniería Q-3

Tarea de Ingeniería	
Número Tarea: Q-3	Historia de Usuario: Gestión de Módulos
Nombre Tarea: Editar Módulos	
Tipo Tarea: Creación	Puntos Estimado: 2
Fecha Inicio: 23/04/2014	Fecha Fin: 24/04/2014
Analista Responsable: Amanda Adriano	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones.	

Tarea de Ingeniería Q-4

Tarea de Ingeniería	
Número Tarea: Q-4	Historia de Usuario: Gestión de Módulos
Nombre Tarea: Buscar Módulos	
Tipo Tarea: Creación	Puntos Estimado: 1
Fecha Inicio: 24/04/2014	Fecha Fin: 24/04/2014
Analista Responsable: Amanda Adriano	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar	

excepciones.

Tarea de Ingeniería Q-5

Tarea de Ingeniería	
Número Tarea: Q-5	Historia de Usuario: Gestión de Módulos
Nombre Tarea: Eliminar Módulos	
Tipo Tarea: Creación	Puntos Estimado: 2
Fecha Inicio: 24/04/2014	Fecha Fin: 25/04/2014
Analista Responsable: Amanda Adriano	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones.	

Interfaz de usuario de la página de gestión de módulos

La tabla adjunta muestra las interfaces para la administración de módulos, las opciones editar y eliminar seguirán el mismo formato de la interfaz Nueva módulo.

Descripción	Interfaz Grafica								
Menú principal opción Módulos	<p>Módulos</p> <p>Nuevo Módulo</p> <table border="1"><thead><tr><th>Código</th><th>Descripción</th><th>Aplicación</th><th>Estado</th></tr></thead><tbody><tr><td> </td><td> </td><td> </td><td> </td></tr></tbody></table> <p>Editar Eliminar</p>	Código	Descripción	Aplicación	Estado				
Código	Descripción	Aplicación	Estado						

Opción Nuevo Módulo	
---------------------	--

Prueba de aceptación

Prueba de Aceptación historia 7

Caso de Prueba de Aceptación	
Número caso de Prueba: 10	Historia de Usuario: Gestión de Módulos
Nombre: Prueba de las opciones de la administración de modulos	
Descripción: Se probara lo siguiente: mostrar, insertar, buscar, editar, eliminar módulos.	
Condiciones de Ejecución: El usuario tiene que estar asignado a un grupo con permisos de administrador.	
Entrada / Pasos de Ejecución:	
<ol style="list-style-type: none"> 1. El administrador ingresa a la página principal del aplicativo que se encuentra en el siguiente link (http://186.42.198.158:8080/Integracion/login.jsf) 2. El administrador ingresa su nombre de usuario, contraseña y luego selecciona el botón Autenticar. 3. El administrador selecciona el menú administracion y luego selecciona la opción módulos. 4. El administrador para ingresar un nuevo módulo debe dar click sobre el botón Nuevo módulo, le desplegara una pantalla con los campos que deben ser ingresados, luego presionar el botón aceptar. 5. El administrador para editar módulo debe seleccionar de la lista el módulo a ser editada y luego dar click en el botón editar, se desplegara una pantalla en la cual se 	

<p>visualizan los campos a ser modificados, luego presionar el botón aceptar.</p> <p>6. El administrador para eliminar módulos debe seleccionar de la lista el módulo a ser eliminado y luego dar click en el botón eliminar, se desplegara una pantalla en la cual se solicitara autorización para eliminar y luego presionar el botón aceptar, tener en cuenta que para eliminar módulos estas no deben tener asignado recursos.</p>
<p>Resultados esperados 1:</p> <p>1. El sistema cumple satisfactoriamente con todas las pruebas realizadas anteriormente.</p>

Iteración 7

Historia de usuario 8

La historia de usuario 8, se refiere a la creación, modificación y eliminación de recursos (opción del módulo).

Gestión de Recursos.

Historia de Usuario	
Número: H8	Nombre Historia Usuario: Gestión de Recursos
Usuario: Administrador	Interacción Asignada: 7
Prioridad en Negocio: Media	Puntos Estimado: 8
Riesgo en Desarrollo: Baja	Puntos Reales: 10
Descripción: Como administrador deseo que la aplicación permita crear, modificar y eliminar recursos para que los usuarios tengan privilegios específicos sobre las aplicaciones.	

Cronograma de tareas de ingeniería

Para cumplir con la fecha establecida por la iteración, en la tabla adjunta se establece el cronograma y tarea de ingeniería a desarrollarse.

Cronograma de tareas de ingeniería historia 8

Tarea de ingeniería (código y nombre)		Puntos estimados	Puntos reales	Fecha inicio	Fecha fin
R-1	Mostrar listado de recursos	1	2	25/04/2014	25/04/2014
R-2	Insertar recursos	2	2	25/04/2014	28/04/2014
R-3	Editar recursos	2	2	28/04/2014	28/04/2014
R-4	Buscar recursos	1	2	28/04/2014	29/04/2014
R-5	Eliminar recursos	2	2	29/04/2014	29/04/2014

Tarea de Ingeniería R-1

Tarea de Ingeniería	
Número Tarea: R-1	Historia de Usuario: Gestión de Recursos
Nombre Tarea: Mostrar listado de recursos	
Tipo Tarea: Creación	Puntos Estimado: 1
Fecha Inicio: 25/04/2014	Fecha Fin: 25/04/2014
Analista Responsable: Amanda Adriano	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones.	

Tarea de Ingeniería R-2

Tarea de Ingeniería	
Número Tarea: R-2	Historia de Usuario: Gestión de Recursos
Nombre Tarea: Insertar Recursos	
Tipo Tarea: Creación	Puntos Estimado: 2
Fecha Inicio: 25/04/2014	Fecha Fin: 28/04/2014
Analista Responsable: Amanda Adriano	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente:	

Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones.

Tarea de Ingeniería R-3

Tarea de Ingeniería	
Número Tarea: R-3	Historia de Usuario: Gestión de Recursos
Nombre Tarea: Editar Recursos	
Tipo Tarea: Creación	Puntos Estimado: 2
Fecha Inicio: 28/04/2014	Fecha Fin: 28/04/2014
Analista Responsable: Amanda Adriano	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones.	

Tarea de Ingeniería R-4

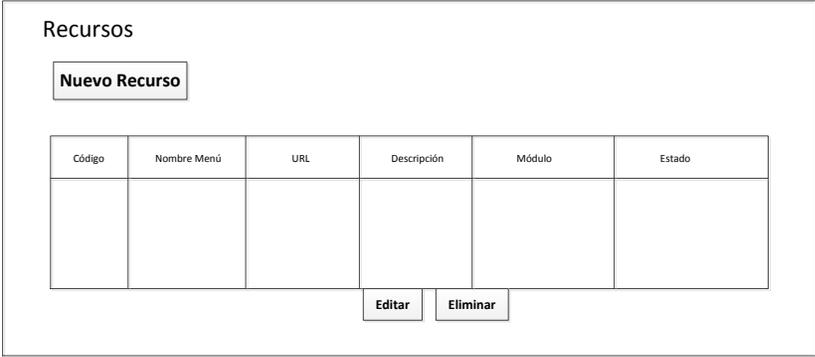
Tarea de Ingeniería	
Número Tarea: R-4	Historia de Usuario: Gestión de Recursos
Nombre Tarea: Buscar Recursos	
Tipo Tarea: Creación	Puntos Estimado: 1
Fecha Inicio: 28/04/2014	Fecha Fin: 29/04/2014
Analista Responsable: Amanda Adriano	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones.	

Tarea de Ingeniería R-5

Tarea de Ingeniería	
Número Tarea: R-5	Historia de Usuario: Gestión de Recursos
Nombre Tarea: Eliminar Recursos	
Tipo Tarea: Creación	Puntos Estimado: 2
Fecha Inicio: 29/04/2014	Fecha Fin: 29/04/2014
Analista Responsable: Amanda Adriano	
<p>Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente:</p> <p>Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones</p>	

Interfaz de usuario de la página de gestión de recursos

La tabla adjunta muestra las interfaces para la administración de recursos, las opciones editar y eliminar seguirán el mismo formato de la interfaz Nuevo recurso.

Descripción	Interfaz Grafica
Menú principal opción Recursos	 <p>The screenshot shows a web interface titled 'Recursos'. At the top left is a button labeled 'Nuevo Recurso'. Below it is a table with the following columns: 'Código', 'Nombre Menú', 'URL', 'Descripción', 'Módulo', and 'Estado'. The table is currently empty. Below the table are two buttons: 'Editar' and 'Eliminar'.</p>

Opción Nuevo Recurso	<div style="border: 1px solid black; padding: 10px; width: fit-content; margin: auto;"> <p style="text-align: center; margin: 0;">Nuevo Recurso</p> <p>Nombre Menú : <input style="width: 150px;" type="text"/></p> <p>URL: <input style="width: 150px;" type="text"/></p> <p>Descripción: <input style="width: 150px;" type="text"/></p> <p>Módulo – Aplicación: <input style="width: 150px;" type="text"/> ▼</p> <p>Estado: <input style="width: 150px;" type="text"/> ▼</p> <div style="display: flex; justify-content: center; gap: 20px; margin-top: 10px;"> <input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/> </div> </div>
-------------------------	--

Prueba de aceptación

Prueba de Aceptación historia 8

Caso de Prueba de Aceptación	
Número caso de Prueba: 11	Historia de Usuario: Gestión de Recursos
Nombre: Prueba de las opciones de la administración de recursos	
Descripción: Se probara lo siguiente: mostrar, insertar, buscar, editar, eliminar recursos.	
Condiciones de Ejecución: El usuario tiene que estar asignado a un grupo con permisos de administrador.	
Entrada / Pasos de Ejecución:	
<ol style="list-style-type: none"> 1. El administrador ingresa a la página principal del aplicativo que se encuentra en el siguiente link (http://186.42.198.158:8080/Integracion/login.jsf) 2. El administrador ingresa su nombre de usuario, contraseña y luego selecciona el botón Autenticar. 3. El administrador selecciona el menú administracion y luego selecciona la opción recursos. 4. El administrador para ingresar un nuevo recurso debe dar click sobre el botón Nuevo recurso, le desplegara una pantalla con los campos que deben ser ingresados, luego presionar el botón aceptar. 5. El administrador para editar recursos debe seleccionar de la lista el recurso a ser editado y luego dar click en el botón editar, se desplegara una pantalla en la cual se 	

<p>visualizan los campos a ser modificados, luego presionar el botón aceptar.</p> <p>6. El administrador para eliminar recursos debe seleccionar de la lista el recurso a ser eliminado y luego dar click en el botón eliminar, se desplegara una pantalla en la cual se solicitara autorización para eliminar y luego presionar el botón aceptar.</p>
<p>Resultados esperados 1:</p> <p>1. El sistema cumple satisfactoriamente con todas las pruebas realizadas anteriormente.</p>

Historia de usuario 9

La tabla adjunta se refiere al a creación, modificación y eliminación de cargos asignados a los usuarios para la administración del Módulo de Integración del Sistema.

Gestión de Cargos.

Historia de Usuario	
Número: H9	Nombre Historia Usuario: Gestión de Cargos
Usuario: Administrador	Interacción Asignada: 7
Prioridad en Negocio: Media	Puntos Estimado: 4
Riesgo en Desarrollo: Baja	Puntos Reales: 6
Descripción: Como administrador deseo que la aplicación permita crear, modificar y eliminar cargos para que los usuarios administren las aplicaciones.	

Cronograma de tareas de ingeniería

Para cumplir con la fecha establecida por la iteración, en la tabla adjunta se establece el cronograma y tarea de ingeniería a desarrollarse.

Cronograma de tareas de ingeniería historia 9

Tarea de ingeniería (código y nombre)		Puntos estimados	Puntos reales	Fecha inicio	Fecha fin
S-1	Mostar listado de cargos	1	1	29/04/2014	29/04/2014
S-2	Insertar cargos	1	2	30/04/2014	30/04/2014
S-3	Editar cargos	1	1	30/04/2014	30/04/2014
S-4	Buscar y eliminar cargos	1	2	30/04/2014	01/05/2014

Tarea de Ingeniería S-1

Tarea de Ingeniería	
Número Tarea: S-1	Historia de Usuario: Gestión de Cargos
Nombre Tarea: Mostrar listado de Cargos	
Tipo Tarea: Creación	Puntos Estimado: 1
Fecha Inicio: 29/04/2014	Fecha Fin: 29/04/2014
Analista Responsable: Amanda Adriano	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones	

Tarea de Ingeniería S-2

Tarea de Ingeniería	
Número Tarea: S-2	Historia de Usuario: Gestión de Cargos
Nombre Tarea: Insertar Cargos	
Tipo Tarea: Creación	Puntos Estimado: 1
Fecha Inicio: 30/04/2014	Fecha Fin: 30/04/2014
Analista Responsable: Amanda Adriano	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar	

excepciones

Tarea de Ingeniería S-3

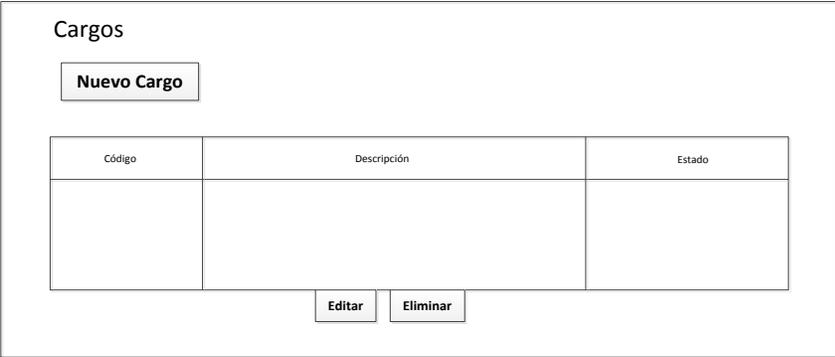
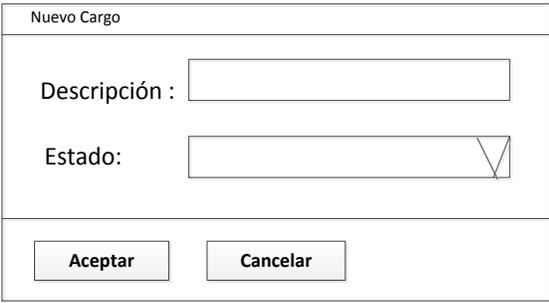
Tarea de Ingeniería	
Número Tarea: S-3	Historia de Usuario: Gestión de Cargos
Nombre Tarea: Editar Cargos	
Tipo Tarea: Creación	Puntos Estimado: 1
Fecha Inicio: 30/04/2014	Fecha Fin: 30/04/2014
Analista Responsable: Amanda Adriano	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones	

Tarea de Ingeniería S-4

Tarea de Ingeniería	
Número Tarea: S-4	Historia de Usuario: Gestión de Cargos
Nombre Tarea: Buscar y eliminar Cargos	
Tipo Tarea: Creación	Puntos Estimado: 1
Fecha Inicio: 30/04/2014	Fecha Fin: 01/05/2014
Analista Responsable: Amanda Adriano	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones	

Interfaz de usuario de la página de gestión de cargos

La tabla adjunta muestra las interfaces para la administración de cargos, las opciones editar y eliminar seguirán el mismo formato de la interfaz Nuevo cargo.

Descripción	Interfaz Grafica
Menú principal opción Cargos	
Opción Nuevo Cargo	

Prueba de aceptación

Prueba de Aceptación historia 9

Caso de Prueba de Aceptación	
Número caso de Prueba: 12	Historia de Usuario: Gestión de Cargos
Nombre: Prueba de las opciones de la administración de cargos	
Descripción: Se probara lo siguiente: mostrar, insertar, buscar, editar, eliminar cargos.	
Condiciones de Ejecución: El usuario tiene que estar asignado a un grupo con permisos de administrador.	
Entrada / Pasos de Ejecución: <ol style="list-style-type: none"> 1. El administrador ingresa a la página principal del aplicativo que se encuentra en el siguiente link (http://186.42.198.158:8080/Integracion/login.jsf) 2. El administrador ingresa su nombre de usuario, contraseña y luego selecciona el 	

<p>botón Autenticar.</p> <ol style="list-style-type: none"> El administrador selecciona el menú administracion y luego selecciona la opción cargos. El administrador para ingresar un nuevo cargo debe dar click sobre el botón Nuevo cargo, le desplegara una pantalla con los campos que deben ser ingresados, luego presionar el botón aceptar. El administrador para editar cargos debe seleccionar de la lista el cargo a ser editado y luego dar click en el botón editar, se desplegara una pantalla en la cual se visualizan los campos a ser modificados, luego presionar el botón aceptar. El administrador para eliminar cargos debe seleccionar de la lista el cargo a ser eliminado y luego dar click en el botón eliminar, se desplegara una pantalla en la cual se solicitara autorización para eliminar y luego presionar el botón aceptar, tener en cuenta que si un cargo esta asignado a uno o varios usuarios no se podrá eliminar.
<p>Resultados esperados 1:</p> <ol style="list-style-type: none"> El sistema cumple satisfactoriamente con todas las pruebas realizadas anteriormente.

Historia de usuario 10

Gestión de Áreas.

Historia de Usuario	
Número: H10	Nombre Historia Usuario: Gestión de Áreas
Usuario: Administrador	Interacción Asignada: 7
Prioridad en Negocio: Media	Puntos Estimado: 8
Riesgo en Desarrollo: Baja	Puntos Reales: 10
Descripción: Como administrador deseo que la aplicación permita crear, modificar y eliminar áreas para que los usuarios administren las aplicaciones.	

Cronograma de tareas de ingeniería

Para cumplir con la fecha establecida por la iteración, en la tabla adjunta se establece el cronograma y tarea de ingeniería a desarrollarse.

Cronograma de tareas de ingeniería historia 10

Tarea de ingeniería (código y nombre)		Puntos estimados	Puntos reales	Fecha inicio	Fecha fin
T-1	Mostar listado de áreas	1	2	01/05/2014	01/05/2014
T-2	Insertar áreas	2	2	01/05/2014	02/05/2014
T-3	Editar áreas	2	2	02/05/2014	02/05/2014
T-4	Buscar áreas	1	2	02/05/2014	05/05/2014
T-5	Eliminar áreas	2	2	05/05/2014	05/05/2014

Tarea de Ingeniería T-1

Tarea de Ingeniería	
Número Tarea: T-1	Historia de Usuario: Gestión de Áreas
Nombre Tarea: Mostar listado de áreas	
Tipo Tarea: Creación	Puntos Estimado: 1
Fecha Inicio: 01/05/2014	Fecha Fin: 01/05/2014
Analista Responsable: Amanda Adriano	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones	

Tarea de Ingeniería T-2

Tarea de Ingeniería	
Número Tarea: T-2	Historia de Usuario: Gestión de Áreas
Nombre Tarea: Insertar Áreas	
Tipo Tarea: Creación	Puntos Estimado: 2
Fecha Inicio: 01/05/2014	Fecha Fin: 02/05/2014

Analista Responsable: Amanda Adriano
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones

Tarea de Ingeniería T-3

Tarea de Ingeniería	
Número Tarea: T-3	Historia de Usuario: Gestión de Áreas
Nombre Tarea: Editar Áreas	
Tipo Tarea: Creación	Puntos Estimado: 2
Fecha Inicio: 02/05/2014	Fecha Fin: 02/05/2014
Analista Responsable: Amanda Adriano	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones	

Tarea de Ingeniería T-4

Tarea de Ingeniería	
Número Tarea: T-4	Historia de Usuario: Gestión de Áreas
Nombre Tarea: Buscar Áreas	
Tipo Tarea: Creación	Puntos Estimado: 1
Fecha Inicio: 02/05/2014	Fecha Fin: 05/05/2014
Analista Responsable: Amanda Adriano	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones	

Tarea de Ingeniería T-5

Tarea de Ingeniería	
Número Tarea: T-5	Historia de Usuario: Gestión de Áreas
Nombre Tarea: Eliminar Áreas	
Tipo Tarea: Creación	Puntos Estimado: 2
Fecha Inicio: 05/05/2014	Fecha Fin: 05/05/2014
Analista Responsable: Amanda Adriano	
<p>Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente:</p> <p>Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones</p>	

Interfaz de la página de gestión de áreas

La tabla adjunta muestra las interfaces para la administración de áreas, las opciones editar y eliminar seguirán el mismo formato de la interfaz Nueva área.

Descripción	Interfaz Grafica
Menú principal opción Áreas	<p>El diagrama muestra una interfaz con el título 'Áreas'. En la parte superior izquierda hay un botón 'Nuevo Área'. Debajo de esto hay una tabla con tres columnas: 'Código', 'Descripción' y 'Estado'. En la parte inferior de la tabla hay dos botones: 'Editar' y 'Eliminar'.</p>

Opción Nueva área	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;"> <p style="text-align: center; margin: 0;">Nueva Área</p> <p>Descripción : <input style="width: 150px;" type="text"/></p> <p>Estado: <input style="width: 150px;" type="text"/></p> <hr/> <p style="text-align: center;"> <input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/> </p> </div>
----------------------	--

Prueba de aceptación

Prueba de Aceptación historia 10

Caso de Prueba de Aceptación	
Número caso de Prueba: 13	Historia de Usuario: Gestión de Áreas
Nombre: Prueba de las opciones de la administración de Áreas	
Descripción: Se probará lo siguiente: mostrar, insertar, buscar, editar, eliminar áreas.	
Condiciones de Ejecución: El usuario tiene que estar asignado a un grupo con permisos de administrador.	
Entrada / Pasos de Ejecución: <ol style="list-style-type: none"> 1. El administrador ingresa a la página principal del aplicativo que se encuentra en el siguiente link (http://186.42.198.158:8080/Integracion/login.jsf) 2. El administrador ingresa su nombre de usuario, contraseña y luego selecciona el botón Autenticar. 3. El administrador selecciona el menú administracion y luego selecciona la opción áreas. 4. El administrador para ingresar una nueva área debe dar click sobre el botón Nueva área, se desplegará una pantalla con los campos que deben ser ingresados, luego presionar el botón aceptar. 5. El administrador para editar áreas debe seleccionar de la lista el área a ser editada y luego dar click en el botón editar, se desplegará una pantalla en la cual se visualizan los campos a ser modificados, luego presionar el botón aceptar. 6. El administrador para eliminar áreas debe seleccionar de la lista el área a ser eliminada y luego dar click en el botón eliminar, se desplegará una pantalla en la 	

cual se solicitara autorización para eliminar y luego presionar el botón aceptar, tener en cuenta que si existe usuarios asignados a una área no permitira eliminar el área seleccionada.

Resultados esperados 1:

1. El sistema cumple satisfactoriamente con todas las pruebas realizadas anteriormente.

Iteración 8

Historia de usuario 11

La tabla adjunta, se refiere a la creación, modificación y eliminación de permisos asignados a los grupos para el acceso a las aplicaciones.

Gestión de Permisos.

Historia de Usuario	
Número: H11	Nombre Historia Usuario: Gestión de Permisos
Usuario: Administrador	Interacción Asignada: 8
Prioridad en Negocio: Media	Puntos Estimado: 4
Riesgo en Desarrollo: Baja	Puntos Reales: 4
Descripción: Como administrador deseo que la aplicación permita crear, modificar y eliminar permisos para que los usuarios puedan ingresar a las diferentes opciones de las aplicaciones.	

Cronograma de tareas de ingeniería

Para cumplir con la fecha establecida por la iteración, en la tabla adjunta se establece el cronograma y tarea de ingeniería a desarrollarse.

Cronograma de tareas de ingeniería historia 11

Tarea de ingeniería (código y nombre)		Puntos estimados	Puntos reales	Fecha inicio	Fecha fin
U-1	Mostar listado de permisos	1	1	05/05/2014	05/05/2014
U-2	Buscar permisos	1	1	06/05/2014	06/05/2014
U-3	Editar permisos	2	2	06/05/2014	06/05/2014

Tarea de Ingeniería U-1

Tarea de Ingeniería	
Número Tarea: U-1	Historia de Usuario: Gestión de Permisos
Nombre Tarea: Mostar listado de permisos	
Tipo Tarea: Creación	Puntos Estimado: 1
Fecha Inicio: 05/05/2014	Fecha Fin: 05/05/2014
Analista Responsable: Amanda Adriano	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones	

Tarea de Ingeniería U-2

Tarea de Ingeniería	
Número Tarea: U-2	Historia de Usuario: Gestión de Permisos
Nombre Tarea: Buscar Permisos	
Tipo Tarea: Creación	Puntos Estimado: 1
Fecha Inicio: 06/05/2014	Fecha Fin: 06/05/2014
Analista Responsable: Amanda Adriano	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones	

Tarea de Ingeniería U-3

Tarea de Ingeniería	
Número Tarea: U-3	Historia de Usuario: Gestión de Permisos
Nombre Tarea: Editar Permisos	
Tipo Tarea: Desarrollador	Puntos Estimado: 2
Fecha Inicio: 06/05/2014	Fecha Fin: 06/05/2014
Analista Responsable: Amanda Adriano	
<p>Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente:</p> <p>Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones</p>	

Interfaz de usuario de la página de gestión de permisos

La tabla adjunta muestra las interfaces para la administración de Permisos.

Descripción	Interfaz Grafica
Menú principal opción Permisos	<p>The screenshot shows a web interface titled 'Permisos'. At the top left is a search bar with a 'Buscar' button. Below it is a table with the following columns: 'Código', 'Nombre Menú', 'Descripción', 'Insertar', 'Actualizar', 'Eliminar', 'Ver', and 'Buscar'. The table is currently empty. Below the table, there are three buttons: 'Editar', 'Permisos masivos', and 'Actualizar'.</p>

Opción Buscar	<div data-bbox="706 262 1205 529"> <p>Grupos - Aplicaciones</p> <p>Grupo: <input type="text"/></p> <p>Aplicación: <input type="text"/></p> <p><input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/></p> </div>
Opción Editar y opción actualizar recursos	<div data-bbox="506 619 940 898"> <p>Editar Permisos</p> <p>Descripción:</p> <p>Insertar <input type="checkbox"/> Eliminar <input type="checkbox"/></p> <p>Actualizar <input type="checkbox"/> Ver <input type="checkbox"/></p> <p>Buscar <input type="checkbox"/></p> <p><input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/></p> </div> <div data-bbox="946 619 1380 898"> <p>Actualizar Recursos</p> <p>Actualizar Recursos insertados ultimadamente para esta aplicación.</p> <p><input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/></p> </div>

Prueba de aceptación

Prueba de Aceptación historia 11

Caso de Prueba de Aceptación	
Número caso de Prueba: 14	Historia de Usuario: Gestión de Permisos
Nombre: Prueba Gestión de Permisos	
Descripción: Se probara la carga de estudiantes, y la sincronización hacia Moodle.	
Condiciones de Ejecución: El usuario tiene que estar asignado a un grupo con permisos de administrador.	
Entrada / Pasos de Ejecución: <ol style="list-style-type: none"> 1. El administrador ingresa a la página principal del aplicativo que se encuentra en el siguiente link (http://186.42.198.158:8080/Integracion/login.jsf) 2. El administrador ingresa su nombre de usuario, contraseña y luego selecciona el botón Autenticar. 3. El administrador selecciona el menú Permisos, y luego presiona el botón Buscar. 	

4. En la pantalla siguiente selecciona el grupo y la aplicación a las cual se le asignaran los permisos, luego presionar el botón aceptar.
5. Seleccionar de la lista desplegable el recurso al cual se le va a dar los permisos, y luego presionar el botón editar, se desplegara una pantalla, seleccionamos el permiso que se desea aplicar y luego presionar el botón aceptar.
6. Para aplicar permisos masivos a los recursos de la aplicación seleccionar el botón Permisos Masivos y luego aplicar el permisos especifico, despues presionar el botón aceptar.
7. Cuando no se visualice un recurso, seleccionar un permiso cualquiera de la lista y luego presionar el botón Actualizar.

Resultados esperados 1:

1. Se cumplieron satisfactoriamente las pruebas realizadas en la funcionalidad descrita anteriormente.

Historia de usuario 12

Diagrama Diurna

Historia de Usuario	
Número: H12	Nombre Historia Usuario: Diagrama Diurna
Usuario: Administrador	Iteración Asignada: 8
Prioridad en Negocio: Alta	Puntos Estimado: 16
Riesgo en Desarrollo: Media	Puntos Reales: 20
Descripción: Como administrador deseo que la aplicación permita generar el diagrama diurna para modelar los datos desde el sistema académico hacia el sistema de integración y visualizar la estructura que tendrá Moodle al sincronizar la información.	

Cronograma de tareas de ingeniería

Para cumplir con la fecha establecida por la iteración, en la tabla adjunta se establece el cronograma y tarea de ingeniería a desarrollarse.

Cronograma de tareas de ingeniería historia 12

Tarea de ingeniería (código y nombre)		Puntos estimados	Puntos reales	Fecha inicio	Fecha fin
V-1	Generar estructura organizacional del Colegio	8	10	06/05/2014	09/05/2014
V-2	Visualizar la estructura organizacional del Colegio	8	10	09/05/2014	13/05/2014

Tarea de Ingeniería V-1

Tarea de Ingeniería	
Número Tarea: V-1	Historia de Usuario: Diagrama Diurna
Nombre Tarea: Generar estructura organizacional del Colegio	
Tipo Tarea: Creación	Puntos Estimado: 8
Fecha Inicio: 06/05/2014	Fecha Fin: 09/05/2014
Analista Responsable: José Pinto	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones.	

Tarea de Ingeniería V-2.

Tarea de Ingeniería	
Numero Tarea: V-2	Historia de Usuario: Diagrama Diurna
Nombre Tarea: Visualizar la estructura organizacional del Colegio	
Tipo Tarea: Creación	Puntos Estimado: 8
Fecha Inicio: 09/05/2014	Fecha Fin: 13/05/2014
Analista Responsable: José Pinto	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar	

excepciones.

Interfaz de usuario de la página de diagrama diurna

Diagramas

Prueba de aceptación

Prueba de Aceptación historia 12

Caso de Prueba de Aceptación	
Número caso de Prueba: 15	Historia de Usuario: Diagrama Diurna
Nombre: Prueba Diagrama Diurna	
Descripción: Se probará la carga de estudiantes, y la sincronización hacia Moodle.	
Condiciones de Ejecución: El usuario tiene que estar asignado a un grupo con permisos que le permitan realizar esta acción.	
Entrada / Pasos de Ejecución: <ol style="list-style-type: none">1. El administrador ingresa a la página principal del aplicativo que se encuentra en el siguiente link (http://186.42.198.158:8080/Integracion/login.jsf)2. El administrador ingresa su nombre de usuario, contraseña y luego selecciona el botón Autenticar.3. Seleccionar el menú INTEGRACIÓN MOODLE DIURNA, luego seleccionar la opción Diagrama Diurna.4. Primeramente se debe seleccionar el botón Generar estructura (Con la ejecución de este botón se extraerá la información desde el web services y se modelarán los datos para el paso a Moodle), y luego el botón Mostrar estructura Moodle (Con la ejecución de este botón se visualizará la estructura académica que se creará en Moodle).	

Resultados esperados 1:

1. El sistema cumple satisfactoriamente con todas las pruebas realizadas anteriormente.

Iteración 9**Historia de usuario 13**

La historia a continuación descrita trata sobre lo que el cliente solicito para la integración de información desde el sistema académico institucional hacia el entorno de aprendizaje Moodle. La tabla adjunta esta detallada la historia de usuario descrita por el cliente para la integración.

Integración de datos de estudiantes a Moodle

Historia de Usuario	
Número: H13	Nombre Historia Usuario: Integración de datos de estudiantes a Moodle.
Usuario: Administrador	Iteración Asignada: 9
Prioridad en Negocio: Alta	Puntos Estimado: 10
Riesgo en Desarrollo: Media	Puntos Reales: 10
Descripción: Como administrador quiero que la información de los estudiantes que se encuentra almacenada en el sistema académico para un año lectivo actual, este también creada en el sistema de entorno de aprendizaje (Moodle), con el objetivo de que el proceso de creación de estudiantes no se lo tenga que realizar manualmente.	

Cronograma de tareas de ingeniería

Para cumplir con la fecha establecida por la iteración, en la tabla adjunta se establece el cronograma y tarea de ingeniería a desarrollarse.

Cronograma de tareas de ingeniería historia 13

Tarea de ingeniería (código y nombre)		Puntos estimados	Puntos reales	Fecha inicio	Fecha fin
W-1	Cargar Estudiantes	2	2	13/05/2014	14/05/2014
W-2	Sincronización Moodle	8	8	14/05/2014	16/05/2014

Tarea de Ingeniería W-1

Tarea de Ingeniería	
Número Tarea: W-1	Historia de Usuario: Integración de datos de estudiantes a Moodle
Nombre Tarea: Cargar Estudiantes	
Tipo Tarea: Creación	Puntos Estimado: 2
Fecha Inicio: 13/05/2014	Fecha Fin: 14/05/2014
Analista Responsable: José Pinto	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones.	

Tarea de ingeniería W-2

Tarea de Ingeniería	
Número Tarea: W-2	Historia de Usuario: Integración de datos de estudiantes a Moodle
Nombre Tarea: Sincronización Moodle	
Tipo Tarea: Creación	Puntos Estimado: 8
Fecha Inicio: 14/05/2014	Fecha Fin: 16/05/2014
Analista Responsable: Amanda Adriano	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear clases en java, crear métodos en java, crear funciones en PostgreSQL.	

Interfaz de usuario de la página de estudiantes

Estudiantes

Cargar Estudiantes Sincronización Moodle

Código	Código Oasis	Nombres	Apellidos	Estado Migración

Prueba de aceptación

Prueba de aceptación historia 13

Caso de Prueba de Aceptación	
Número caso de Prueba: 16	Historia de Usuario: Integración de datos de estudiantes a Moodle
Nombre: Prueba Integración de datos de estudiantes a Moodle	
Descripción: Se probará la carga de estudiantes, y la sincronización hacia Moodle.	
Condiciones de Ejecución: El usuario tiene que estar asignado a un grupo con permisos de administrador.	
Entrada / Pasos de Ejecución: <ol style="list-style-type: none">1. El administrador ingresa a la página principal del aplicativo que se encuentra en el siguiente link (http://186.42.198.158:8080/Integracion/login.jsf)2. El administrador ingresa su nombre de usuario, contraseña y luego selecciona el botón Autenticar.3. El administrador selecciona el menú desplegable Integración Moodle Diurna, luego selecciona la opción Estudiantes.4. El administrador selecciona el botón cargar estudiantes (este botón es opcional al momento de sincronizar).5. El administrador selecciona el botón sincronización Moodle.6. El administrador selecciona un número de cédula de un estudiante migrado.7. El administrador ingresa a la página de Moodle (http://186.42.198.158/moodle/)	

8. El administrador ingresa con las credenciales del estudiante en este caso la cedula de identidad es el user y password, y comprueba que no exista error en el ingreso a la aplicación de Moodle.

Resultados esperados 1:

1. El sistema muestra correctamente la información de los estudiantes a ser sincronizados a Moodle.
2. El sistema registra correctamente la información en Moodle.
3. El sistema Moodle permite ingresar con las credenciales de un estudiante.

Resultados esperados 2:

1. El sistema no permite visualizar la información (No existen registros).
2. El sistema no sincroniza ninguna información hacia Moodle (No existen registros).
3. El sistema Moodle no permite ingresar a la aplicación (No existe estudiante en Moodle).

Resultados esperados 3:

1. El sistema despliega mensaje de error al mostrar información, error producido por que no está disponible base de datos de donde se extraen datos.
2. El sistema despliega mensaje de error al sincronizar información hacia Moodle, error es producido por que no existe conexión hacia la base de datos de Moodle.
3. El sistema Moodle no permite ingresar a la aplicación (No existe estudiante en Moodle).

Historia de usuario 14

La historia a continuación descrita trata sobre lo que el cliente solicito para la integración de información de docentes desde el sistema académico institucional hacia el entorno de aprendizaje Moodle. La tabla adjunta está detallada la historia de usuario descrita por el cliente para la integración.

Integración de datos de docentes a Moodle

Historia de Usuario	
Número: H14	Nombre Historia Usuario: Integración de datos de docentes a Moodle.

Usuario: Administrador	Interacción Asignada: 9
Prioridad en Negocio: Alta	Puntos Estimado: 10
Riesgo en Desarrollo: Media	Puntos Reales: 10
Descripción: Como administrador quiero que la información de los docentes que se encuentra almacenada en el sistema académico para un año lectivo actual, este también creada en el sistema de entorno de aprendizaje (Moodle), con el objetivo de que el proceso de creación de docentes no se lo tenga que realizar manualmente.	
Observaciones: La información de los docentes tiene que estar organizada en Moodle de tal forma que no haya problemas o errores al momento de ingresar y realizar tareas en Moodle.	

Cronograma de tareas de ingeniería

Para cumplir con la fecha establecida por la iteración, en la tabla adjunta se establece el cronograma y tarea de ingeniería a desarrollarse.

Cronograma de tareas de ingeniería historia 14

Tarea de ingeniería (código y nombre)		Puntos estimados	Puntos reales	Fecha inicio	Fecha fin
X-1	Cargar Docentes	2	2	16/05/2014	16/05/2014
X-2	Sincronización Moodle	8	8	16/05/2014	20/05/2014

Tarea de Ingeniería X-1

Tarea de Ingeniería	
Número Tarea: X-1	Historia de Usuario: Integración de datos de docentes a Moodle
Nombre Tarea: Cargar Docentes	
Tipo Tarea: Creación	Puntos Estimado: 2
Fecha Inicio: 16/05/2014	Fecha Fin: 16/05/2014
Analista Responsable: José Pinto	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente:	

Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones.

Tarea de ingeniería X-2

Tarea de Ingeniería	
Número Tarea: I-2	Historia de Usuario: Integración de datos de estudiantes a Moodle
Nombre Tarea: Sincronización Moodle	
Tipo Tarea: Creación	Puntos Estimado: 8
Fecha Inicio: 16/05/2014	Fecha Fin: 20/05/2014
Analista Responsable: Amanda Adriano	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear clases en java, crear métodos en java, crear funciones en PostgreSQL.	

Interfaz de usuario de la página de profesores

Profesores

Código	Código Oasis	Nombres	Apellidos	Estado Migración

Prueba de aceptación

Prueba de aceptación historia 14

Caso de Prueba de Aceptación	
Número caso de Prueba: 17	Historia de Usuario: Integración de datos de docentes a Moodle
Nombre: Prueba Integración de datos de docentes a Moodle	
Descripción: Se probará la carga de información de docentes, y la sincronización hacia Moodle.	
Condiciones de Ejecución: El usuario tiene que estar asignado a un grupo con permisos de administrador.	
Entrada / Pasos de Ejecución: <ol style="list-style-type: none">1. El administrador ingresa a la página principal del aplicativo que se encuentra en el siguiente link (http://186.42.198.158:8080/Integracion/login.jsf)2. El administrador ingresa su nombre de usuario, contraseña y luego selecciona el botón Autenticar.3. El administrador selecciona el menú desplegable Integración Moodle Diurna, luego selecciona la opción Docentes.4. El administrador selecciona el botón cargar docentes (este botón es opcional al momento de sincronizar).5. El administrador selecciona el botón sincronización Moodle.6. El administrador selecciona un número de cédula de un estudiante migrado.7. El administrador ingresa a la página de Moodle (http://186.42.198.158/moodle/)8. El administrador ingresa con las credenciales del docente en este caso la cédula de identidad es el user y password, y comprueba que no exista error en el ingreso a la aplicación de Moodle.	
Resultados esperados 1: <ol style="list-style-type: none">1. El sistema muestra correctamente la información de los docentes a ser sincronizados a Moodle.2. El sistema registra correctamente la información en Moodle.3. El sistema Moodle permite ingresar con las credenciales de un estudiante.	
Resultados esperados 2: <ol style="list-style-type: none">1. El sistema no permite visualizar la información (No existen registros).2. El sistema no sincroniza ninguna información hacia Moodle (No existen registros).	

3. El sistema Moodle no permite ingresar a la aplicación (No existe estudiante en Moodle).
Resultados esperados 3:
4. El sistema despliega mensaje de error al mostrar información, error producido por que no está disponible base de datos de donde se extraen datos.
5. El sistema despliega mensaje de error al sincronizar información hacia Moodle, error es producido por que no existe conexión hacia la base de datos de Moodle.
6. El sistema Moodle no permite ingresar a la aplicación (No existe estudiante en Moodle).

Iteración 10

Historia de usuario 15

La tabla adjunta, se refiere a la sincronización de los datos de cursos, paralelos y materias a Moodle desde el Modulo de Integración de Sistemas.

Integración de cursos, paralelos y materias a Moodle.

Historia de Usuario	
Numero: H15	Nombre Historia Usuario Integración de cursos, paralelos y materias a Moodle
Usuario: Administrador	Interacción Asignada: 10
Prioridad en Negocio: Alta	Puntos Estimado: 20
Riesgo en Desarrollo: Media	Puntos Reales: 16
Descripción: Como administrador deseo que la información tanto de cursos, paralelos, y materias que se encuentran en el sistema académico, sean modelada y sincronizada hacia Moodle para poder ahorrar tiempo en la creación de cada uno de ellos en Moodle.	

Cronograma de tareas de ingeniería

Para cumplir con la fecha establecida por la iteración, en la tabla adjunta se establece el cronograma y tarea de ingeniería a desarrollarse.

Cronograma de tareas de ingeniería historia 15

Tarea de ingeniería (código y nombre)		Puntos estimados	Puntos reales	Fecha inicio	Fecha fin
Y-1	Cargar Categorías/Cursos	10	6	20/05/2014	22/05/2014
Y-2	Sincronización Moodle	10	10	22/05/2014	26/05/2014

Tarea de Ingeniería Y-1

Tarea de Ingeniería	
Numero Tarea: Y-1	Historia de Usuario: Integración de cursos, paralelos, Materias a Moodle
Nombre Tarea: Cargar Categorías/Cursos	
Tipo Tarea: Desarrollador	Puntos Estimado: 10
Fecha Inicio: 20/05/2014	Fecha Fin: 22/05/2014
Analista Responsable: José Pinto	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones.	

Tarea de Ingeniería Y-2

Tarea de Ingeniería	
Numero Tarea: Y-2	Historia de Usuario: Integración de cursos, paralelos, Materias a Moodle
Nombre Tarea: Sincronización Moodle	
Tipo Tarea: Desarrollador	Puntos Estimado: 10
Fecha Inicio: 22/05/2014	Fecha Fin: 26/05/2014
Analista Responsable: José Pinto	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar	

excepciones.

Interfaz de usuario de la página integración de cursos, paralelos, materias a Moodle

Categorías Cursos

Código	Nombres	Código Moodle

Prueba de aceptación

Prueba de aceptación historia 15

Caso de Prueba de Aceptación	
Número caso de Prueba: 18	Historia de Usuario: Integración de cursos, paralelos, Materias a Moodle
Nombre: Prueba Integración de cursos, paralelos, Materias a Moodle	
Descripción: Se probara la carga de información de docentes, y la sincronización hacia Moodle.	
Condiciones de Ejecución: El usuario tiene que estar asignado a un grupo con privilegio para este recurso del sistema.	
Entrada / Pasos de Ejecución: <ol style="list-style-type: none">1. El administrador ingresa a la página principal del aplicativo que se encuentra en el siguiente link (http://186.42.198.158:8080/Integracion/login.jsf)2. El administrador ingresa su nombre de usuario, contraseña y luego selecciona el botón Autenticar.3. El administrador selecciona el menú desplegable Integración Moodle Diurna, luego selecciona la opción Categoría/Cursos.	

<ol style="list-style-type: none"> 4. El administrador selecciona el botón cargar Categoría/Cursos (este botón es opcional al momento de sincronizar). 5. El administrador selecciona el botón sincronización Moodle. 6. El administrador ingresa a la página de Moodle (http://186.42.198.158/moodle/) 7. El administrador ingresa con las credenciales del usuario administrador de Moodle y comprueba la estructura creada.
<p>Resultados esperados 1:</p> <ol style="list-style-type: none"> 4. El sistema muestra correctamente la información de los Cursos, paralelos, materias a ser sincronizados a Moodle. 5. El sistema registra correctamente la información en Moodle. 6. En el sistema Moodle se crea correctamente la estructura de cursos, paralelos, materias.
<p>Resultados esperados 2:</p> <ol style="list-style-type: none"> 1. El sistema no permite visualizar la información (No existen registros). 2. El sistema no sincroniza ninguna información hacia Moodle (No existen registros).
<p>Resultados esperados 3:</p> <ol style="list-style-type: none"> 1. El sistema despliega mensaje de error al mostrar información, error producido por que no está disponible base de datos de donde se extraen datos. 2. El sistema despliega mensaje de error al sincronizar información hacia Moodle, error es producido por que no existe conexión hacia la base de datos de Moodle.

Iteración 11

Historia de usuario 16

La tabla adjunta, se refiere a la matriculación de estudiantes en Moodle, la matriculación se realizara dependiendo de las materias que tenga asignada en el Sistema de Calificación de Docente.

Matricula de estudiantes en Moodle

Historia de Usuario	
Número: H16	Nombre Historia Usuario: Matriculación de estudiantes en Moodle
Usuario: Administrador	Interacción Asignada: 11
Prioridad en Negocio: Alta	Puntos Estimado: 10

Riesgo en Desarrollo: Media	Puntos Reales: 10
Descripción: Como administrador deseo que la aplicación permita matricular estudiantes en Moodle dependiendo de las materias que tenga asignada en el sistema de calificación de docente para que puedan ingresar y visualizar todos los cursos en los cuales se encuentran matriculados.	

Cronograma de tareas de ingeniería

Para cumplir con la fecha establecida por la iteración, en la tabla adjunta se establece el cronograma y tarea de ingeniería a desarrollarse.

Cronograma de tareas de ingeniería historia 16

Tarea de ingeniería (código y nombre)		Puntos estimados	Puntos reales	Fecha inicio	Fecha fin
Z-1	Cargar Matriculas	5	5	26/05/2014	27/05/2014
Z-2	Sincronización Moodle	5	5	28/05/2014	29/05/2014

Tarea de Ingeniería Z-1

Tarea de Ingeniería	
Número Tarea: Z-1	Historia de Usuario: Matrícula de estudiantes en Moodle
Nombre Tarea: Cargar Matriculas	
Tipo Tarea: Desarrollador	Puntos Estimado: 5
Fecha Inicio: 26/05/2014	Fecha Fin: 27/05/2014
Analista Responsable: José Pinto	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones.	

Tarea de Ingeniería Z-2

Tarea de Ingeniería	
Número Tarea: R-2	Historia de Usuario: Matrícula de estudiantes en Moodle
Nombre Tarea: Sincronización Moodle	
Tipo Tarea: Desarrollo	Puntos Estimado: 5
Fecha Inicio: 28/05/2014	Fecha Fin: 29/05/2014
Analista Responsable: José Pinto	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones.	

Interfaz de usuario de la página de matrícula de estudiantes a Moodle

La interfaz para la matriculación de estudiantes a Moodle es la misma que la interfaz realizada para la integración de datos de estudiantes a Moodle, con la diferencia del nombre de los botones por ejemplo: Cargar Estudiantes será Cargar Matricula.

Prueba de aceptación

Prueba de aceptación historia 16

Caso de Prueba de Aceptación	
Número caso de Prueba: 19	Historia de Usuario: Matrícula de estudiantes en Moodle
Nombre: Prueba Matrícula de estudiantes en Moodle	
Descripción: Se probara la carga de información de docentes, y la sincronización hacia Moodle.	
Condiciones de Ejecución: El usuario tiene que estar asignado a un grupo con permisos para este recurso.	
Entrada / Pasos de Ejecución: 1. El administrador ingresa a la página principal del aplicativo que se encuentra en el	

siguiente link (<http://186.42.198.158:8080/Integracion/login.jsf>)

2. El administrador ingresa su nombre de usuario, contraseña y luego selecciona el botón Autenticar.
3. El administrador selecciona el menú desplegable Integración Moodle Diurna, luego selecciona la opción Matriculas.
4. El administrador selecciona el botón cargar Matriculas (este botón es opcional al momento de sincronizar).
5. El administrador selecciona el botón sincronización Moodle.
6. El administrador selecciona un numero de cedula de un estudiante migrado.
7. El administrador ingresa a la página de Moodle (<http://186.42.198.158/moodle/>)
8. El administrador ingresa con las credenciales del estudiante en este caso la cedula de identidad es el user y password, y comprueba que no exista error en el ingreso a la aplicación de Moodle, y también que se visualizen todas las materias en la cual tiene que estar matriculado.

Resultados esperados 1:

1. El sistema muestra correctamente la información de los estudiantes a ser matriculados en Moodle.
2. El sistema registra correctamente la información en Moodle.
3. El sistema Moodle permite ingresar con las credenciales de un estudiante y también permite visualizar las materias en la cual esta matriculado.

Resultados esperados 2:

1. El sistema no permite visualizar la información (No existen registros).
2. El sistema no sincroniza ninguna información hacia Moodle (No existen registros).
3. El sistema Moodle no permite ingresar a la aplicación (No existe estudiante en Moodle) y por consiguiente no se pueden visualizar las materias en la cual esta matriculado.

Resultados esperados 3:

1. El sistema despliega mensaje de error al mostrar información, error producido por que no está disponible base de datos de donde se extraen datos.
2. El sistema despliega mensaje de error al sincronizar información hacia Moodle, error es producido por que no existe conexión hacia la base de datos de Moodle.
3. El sistema Moodle no permite ingresar a la aplicación (No existe estudiante en Moodle).

Historia de usuario 17

La tabla adjunta, se refiere a la asignación de materias a los profesores en Moodle, este proceso se realiza dependiendo de las materias que tenga asignada en el sistema de calificación de docente.

Asignación de materias a profesores en Moodle.

Historia de Usuario	
Numero: H17	Nombre Historia Usuario: Asignación de materias a profesores en Moodle
Usuario: Administrador	Interacción Asignada: 11
Prioridad en Negocio: Alta	Puntos Estimado: 10
Riesgo en Desarrollo: Media	Puntos Reales: 10
Descripción: Como adiestrador deseo que la aplicación asigne los cursos a los docentes sincronizados en Moodle dependiendo de los cursos que tenga asignado en el sistema de calificación del docente de la institución para que Moodle se genere automáticamente esta información.	

Cronograma de tareas de ingeniería

Para cumplir con la fecha establecida por la iteración, en la tabla adjunta se establece el cronograma y tarea de ingeniería a desarrollarse.

Cronograma de tareas de ingeniería historia 17

Tarea de ingeniería (código y nombre)		Puntos estimados	Puntos reales	Fecha inicio	Fecha fin
A1-1	Cargar Asignaciones (Materias/Profesores)	5	5	29/05/2014	30/05/2014
A1-2	Sincronización Moodle	5	5	30/05/2014	02/06/2014

Tarea de Ingeniería A1-1

Tarea de Ingeniería	
Numero Tarea: A1-1	Historia de Usuario: Asignación de materias a profesores en Moodle
Nombre Tarea: Cargar Asignaciones (Materias/Profesores)	
Tipo Tarea: Desarrollo	Puntos Estimado: 5
Fecha Inicio: 29/05/2014	Fecha Fin: 30/05/2014
Analista Responsable: José Pinto	
Descripción: : Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones.	

Tarea de Ingeniería A1-2

Tarea de Ingeniería	
Numero Tarea: A1-2	Historia de Usuario: Asignación de materias a profesores en Moodle
Nombre Tarea: Sincronización Moodle	
Tipo Tarea: Desarrollo	Puntos Estimado: 4
Fecha Inicio: 30/05/2014	Fecha Fin: 02/06/2014
Analista Responsable: José Pinto	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones.	

Interfaz de usuario de la página asignación de materias a profesores en Moodle

La interfaz para la asignación de materias a profesores en Moodle es la misma que la interfaz realizada para la integración de datos de estudiantes a Moodle, con la diferencia del nombre de los botones por ejemplo: Cargar Estudiantes será Cargar Profesores.

Prueba de aceptación

Prueba de aceptación historia 17

Caso de Prueba de Aceptación	
Número caso de Prueba: 20	Historia de Usuario: Asignación de materias a profesores en Moodle
Nombre: Prueba Asignación de materias a profesores en Moodle	
Descripción: Se probará la carga de información de profesores, y la sincronización hacia Moodle.	
Condiciones de Ejecución: El usuario tiene que estar asignado a un grupo con permisos para este recurso.	
Entrada / Pasos de Ejecución: <ol style="list-style-type: none">1. El administrador ingresa a la página principal del aplicativo que se encuentra en el siguiente link (http://186.42.198.158:8080/Integracion/login.jsf)2. El administrador ingresa su nombre de usuario, contraseña y luego selecciona el botón Autenticar.3. El administrador selecciona el menú desplegable Integración Moodle Diurna, luego selecciona la opción Asignación Profesores.4. El administrador selecciona el botón cargar Profesores (este botón es opcional al momento de sincronizar).5. El administrador selecciona el botón sincronización Moodle.6. El administrador selecciona un número de cédula de un profesor migrado.7. El administrador ingresa a la página de Moodle (http://186.42.198.158/moodle/)8. El administrador ingresa con las credenciales del profesor en este caso la cédula de identidad es el user y password, comprueba que no exista error en el ingreso a la aplicación de Moodle, también validar que se visualicen todas las materias a la cual está asignado el profesor.	
Resultados esperados 1: <ol style="list-style-type: none">1. El sistema muestra correctamente la información de los docentes a ser asignados en Moodle.2. El sistema registra correctamente la información en Moodle.3. El sistema Moodle permite ingresar con las credenciales de un docente y también permite visualizar las materias en la cual está asignado.	
Resultados esperados 2:	

1. El sistema no permite visualizar la información (No existen registros).
2. El sistema no sincroniza ninguna información hacia Moodle (No existen registros).
3. El sistema Moodle no permite ingresar a la aplicación (No existe docentes en Moodle) y por consiguiente no se pueden visualizar las materias en la cual esta asignado.

Resultados esperados 3:

1. El sistema despliega mensaje de error al mostrar información, error producido por que no está disponible base de datos de donde se extraen datos.
2. El sistema despliega mensaje de error al sincronizar información hacia Moodle, error es producido por que no existe conexión hacia la base de datos de Moodle.
3. El sistema Moodle no permite ingresar a la aplicación (No existe información docentes en Moodle).

Iteración 12

Historia de usuario 18

La tabla adjunta, se refiere a la generación de reporte de los cursos con su respectivo profesor y respectivo listado de estudiantes.

Gestión de reportes

Historia de Usuario	
Numero: H17	Nombre Historia Usuario: Gestión de reportes
Usuario: Administrador	Interacción Asignada: 12
Prioridad en Negocio: Alta	Puntos Estimado: 4
Riesgo en Desarrollo: Media	Puntos Reales: 4
Descripción: Como administrador deseo que el sistema permita generar reporte de los cursos con su respectivo profesor y respectivo listado de estudiantes para que los docentes lleven un control de los estudiantes matriculados en Moodle.	

Cronograma de tareas de ingeniería

Para cumplir con la fecha establecida por la iteración, en la tabla adjunta se establece el cronograma y tarea de ingeniería a desarrollarse.

Cronograma de tareas de ingeniería historia 18

Tarea de ingeniería (código y nombre)		Puntos estimados	Puntos reales	Fecha inicio	Fecha fin
B1-1	Generar Reporte por cursos	4	4	02/06/2014	03/06/2014

Tarea de Ingeniería B1-2

Tarea de Ingeniería	
Numero Tarea: B1-3	Historia de Usuario: Generar Reporte
Nombre Tarea: Generar Reporte por cursos	
Tipo Tarea: Desarrollo	Puntos Estimado: 4
Fecha Inicio: 02/06/2014	Fecha Fin: 03/06/2014
Analista Responsable: Amanda Adriano	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Crear funciones en PostgreSQL, Crear métodos en Java, Crear clase en Java, registrar excepciones.	

Interfaz de usuario de la página de generar reporte

La interfaz para la generación del reporte dependerá del reporte solicitado.

Prueba de aceptación

Prueba de aceptación historia 18

Caso de Prueba de Aceptación	
Número caso de Prueba: 21	Historia de Usuario: Generar Reporte
Nombre: Prueba Generar Reporte	
Descripción: Se probará la generación de los reportes	
Condiciones de Ejecución: El usuario tiene que estar asignado a un grupo con permisos para	

este recurso
<p>Entrada / Pasos de Ejecución:</p> <ol style="list-style-type: none"> 1. El administrador ingresa a la página principal del aplicativo que se encuentra en el siguiente link (http://186.42.198.158:8080/Integracion/login.jsf) 2. El administrador ingresa su nombre de usuario, contraseña y luego selecciona el botón Autenticar. 3. El administrador selecciona el menú desplegable Reportes 4. El administrador selecciona el botón generar reporte 5. El administrador selecciona el botón sincronización Moodle. 6. El administrador selecciona un numero de cedula de un estudiante migrado. 7. El administrador ingresa a la página de Moodle (http://186.42.198.158/moodle/) 8. El administrador ingresa con las credenciales del docente en este caso la cedula de identidad es el user y password, y comprueba que no exista error en el ingreso a la aplicación de Moodle.
<p>Resultados esperados 1</p> <ol style="list-style-type: none"> 1. El reporte se generar satisfactoriamente.

Metáfora 8

La tabla adjunta, se refiere a la metáfora de implementación de la aplicación en la institución.

Gestión de reportes

Historia de Usuario	
Numero: M8	Nombre Historia Usuario: Implementación de la aplicación
Usuario: Programador	Interacción Asignada: 4
Prioridad en Negocio: Alta	Puntos Estimado: 4
Riesgo en Desarrollo: Media	Puntos Reales: 4
Descripción: Como programador deseo implementar la aplicación desarrollada para la institución para poder ahorrar tiempos en la generación de información en Moodle.	

Cronograma de tareas de ingeniería

Para cumplir con la fecha establecida por la iteración, en la tabla adjunta se establece el cronograma y tarea de ingeniería a desarrollarse.

Cronograma de tareas de ingeniería metáfora 8

Tarea de ingeniería (código y nombre)		Puntos estimados	Puntos reales	Fecha inicio	Fecha fin
C1-1	Instalación y configuración de los servidores de aplicación.	2	2	03/06/2014	04/06/2014
C1-2	Publicación de la aplicación y del servicio web	2	2	04/06/2014	04/06/2014

Tarea de Ingeniería C1-1

Tarea de Ingeniería	
Numero Tarea: C1-1	Historia de Usuario: Implementación de la aplicación
Nombre Tarea: Instalación y configuración de los servidores de aplicación.	
Tipo Tarea: Implementación	Puntos Estimado: 2
Fecha Inicio: 03/06/2014	Fecha Fin: 04/06/2014
Analista Responsable: José Pinto	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Instalar los servidor de aplicación GlassFish y LAMP Configurar los puertos por los que escuchan los servidores Configurar la red entre los servidores de aplicación	

Tarea de Ingeniería C1-2

Tarea de Ingeniería	
Numero Tarea: C1-2	Historia de Usuario: Implementación de la aplicación
Nombre Tarea: Publicación de la aplicación y servicio web	
Tipo Tarea: Implementación	Puntos Estimado: 2

Fecha Inicio: 04/06/2014	Fecha Fin: 04/06/2014
Analista Responsable: José Pinto	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Subir los archivos del proyectos en el servidor de aplicaciones GlassFish Configurar conexiones hacia base de datos Configurar rutas de los archivos log	

Prueba de aceptación

Prueba de aceptación metáfora 8

Caso de Prueba de Aceptación	
Número caso de Prueba: 21	Historia de Usuario: Implementación de la aplicación
Nombre: Prueba Implementación de la aplicación	
Descripción: Se probara el correcto despliegue de la aplicación	
Condiciones de Ejecución: Ninguna	
Entrada / Pasos de Ejecución: <ol style="list-style-type: none"> Se compureba que el aplicativo se despliegue correctamente en ambiente de producción 	
Resultados esperados 1: <ol style="list-style-type: none"> La prueba se cumple satisfactoriamente. 	

Metáfora 9

La tabla adjunta, se refiere a la metáfora de capacitación a usuarios sobre el funcionamiento del sistema.

Capacitación a usuarios

Historia de Usuario	
Número: M9	Nombre Historia Usuario: Capacitación a usuarios
Usuario: Programador	Interacción Asignada: 12
Prioridad en Negocio: Alta	Puntos Estimado: 2
Riesgo en Desarrollo: Media	Puntos Reales: 2
Descripción: Como programador se requiere capacitar al usuario encargado del manejo de la aplicación para evitar causar conflictos al realizar los procesos.	

Cronograma de tareas de ingeniería

Para cumplir con la fecha establecida por la iteración, en la tabla adjunta se establece el cronograma y tarea de ingeniería a desarrollarse.

Cronograma de tareas de ingeniería metáfora 9

Tarea de ingeniería (código y nombre)		Puntos estimados	Puntos reales	Fecha inicio	Fecha fin
B1-1	Capacitación del funcionamiento del sistema	2	2	04/06/2014	05/06/2014

Tarea de Ingeniería B1-2

Tarea de Ingeniería	
Número Tarea: B1-3	Historia de Usuario: Capacitación a usuarios
Nombre Tarea: Capacitación del funcionamiento del sistema	
Tipo Tarea: Desarrollo	Puntos Estimado: 4
Fecha Inicio: 04/06/2014	Fecha Fin: 05/06/2014
Analista Responsable: Amanda Adriano	
Descripción: Para cumplir con la tarea asignada se debe realizar lo siguiente: Realizar una demostración completa del funcionamiento de la aplicación, recalando la	

función que tiene cada una de las opciones desplegadas.

Prueba de aceptación

Prueba de aceptación metáfora 9

Caso de Prueba de Aceptación	
Número caso de Prueba: 22	Historia de Usuario: Capacitación a usuarios
Nombre: Capacitación de funcionamiento del sistema	
Descripción: Se realizara la capacitación al personal encargado de la aplicación.	
Condiciones de Ejecución: Ninguna	
Entrada / Pasos de Ejecución: Ninguna	
Resultados. El personal que custodiara el aplicativo se le impartio todos los conocimientos para manipular la funcionalidades del sistema.	

Sección D

El manual de usuario, manual de instalación y código fuente del aplicativo se entregaron al Colegio Capitán Edmundo Chiriboga y también se encuentra almacenada en el CD que se entrego en conjunto con esta investigación.

BIBLIOGRAFÍA

- 1. Arenas G., Claudia, y otros.** APLICACIONES WEB - SERVIDOR DE APLICACIONES JAVA., Carrera de Ingeniería en Sistemas. Citado el: 10 de 02 de 2008., Sangolquí – Ecuador., TESIS., Escuela Politécnica del Ejercito., Pag 52,53., <http://jano.unicauca.edu.co/apliweb/aservers/sunas/admin1.htm>.
- 2. FERNÁNDEZ TOLEDO, JESÚS.** DEFINICIÓN DE SERVIDORES DE APLICACIONES., Carrera de Ingeniería en Informática., Citado el: 10 de 05 de 2009., Quito – Ecuador., TESIS., Escuela Politécnica Salesiana., Pag 45., <http://www.losteatinos.com/servlets/servlet.html>.
- 3. TORRES COLLAGUAZO, JOSÉ LIZANDRO y VILLAGOMEZ CEVALLOS, JINSON OSWALDO.** Estudio y soporte técnico de Servidores de aplicación., Carrera de Ingeniería de Sistemas., Citado el: 08 de 05 de 2008., Sangolquí – Ecuador., TESIS., Universidad Politécnica del Ejercito., 2013., Pag 25,26., <http://repositorio.espe.edu.ec/bitstream/21000/4431/1/M-ESPEL-0017.pdf>.
- 4. ARQUITECTURA DE GLASSFISH.,**
<http://www.jtech.ua.es/j2ee/restringido/servd-aplic/sesion01-apuntes.pdf>,
[2014-01-27.](http://www.jtech.ua.es/j2ee/restringido/servd-aplic/sesion01-apuntes.pdf)
- 5. ARQUITECTURA SERVIDOR DE APLICACIÓN GERÓNIMO.,**
<http://geronimo.apache.org/GMOxDOC10/arquitectura.html>, 2014-02-08.

6. CARACTERÍSTICAS DE GLASSFISH.,

<http://repositorio.espe.edu.ec/bitstream/21000/4431/1/M-ESPEL-0017.pdf>,
2014-01-25.

7. COEFICIENTE DE PEARSON.,

<http://personal.us.es/vararey/adatos2/correlacion.pdf>, 2014-02-25.

8. COMANDO SAR.,

<http://www.oracle.com/technetwork/es/articles/linux/part3-325564-esa.html>,
2014-02-20.

9. ESTÁNDAR DE CODIFICACIÓN.,

<http://www.oracle.com/technetwork/java/javase/documentation/codeconvtoc-136057.html>, 2014-10-12.

10. MÉTRICAS DE RENDIMIENTO.,

<http://es.agileload.com/agileload/blog/2013/02/18/web-applications-performance-testing-metrics>, 2014-10-05.

11. PARÁMETROS DE COMPARACIÓN.,

<http://www.saber.cic.ipn.mx/cake/SABERsvn/trunk/Repositorios/webVerArchivo/305/1>, 2014-02-17.

12. PRINCIPIOS DE LOS SERVIDORES DE APLICACIÓN.,

<http://www.jtech.ua.es/j2ee/2003-2004/abierto-j2ee-2003-2004/sa/sesion1-apuntes.htm/1946/1/98T00016.pdf>., 2014-01-10.

13. TAMAÑO TIPOS DE DATOS.,

<http://ajpdsoft.com/modules.php?name=News&file=print&sid=499>., 2014-10-27.

14. TIEMPO DE RESPUESTA.,

http://media.cervantesvirtual.com/s3/BVMC_OBRAS/001/a20/c88/2b2/11d/fac/c70/021/85c/e60/64/imes/001a20c8-82b2-11df-acc7-002185ce6064.pdf.,
[2014-02-15.](#)

15. SERVIDOR DE APLICACIONES GLASSFISH.,

http://ddd.uab.cat/pub/trerecpro/2013/hdl_2072_206748/SerraManchadoDavidR-ETISa2009-10.pdf., [2014-01-20.](#)

16. SERVIDOR DE APLICACIONES GERÓNIMO.,

http://ddd.uab.cat/pub/trerecpro/2013/hdl_2072_206748/SerraManchadoDavidR-ETISa2009-10.pdf., [2014-01-30.](#)

17. SERVIDOR DE APLICACIONES JOnAS.,

<http://jonas.ow2.org/current/doc/doc-en/integrated/howto/JonasArchi.html>.,
[2014-02-12.](#)

