



UNIVERSIDAD NACIONAL DE CHIMBORAZO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA EN SISTEMAS Y COMPUTACIÓN

"Trabajo de grado previo a la obtención del
Título de Ingeniero en Sistemas y Computación"

**“ANÁLISIS DE LAS TECNOLOGÍAS AXIS 2 Y METRO 2.0 PARA EL
DESARROLLO DE SERVICIOS WEB SEGUROS EN JAVA, APLICADO AL
MÓDULO DE INTEGRACIÓN Y REPORTES DEL SISTEMA DE
EVALUACIÓN DOCENTE DE LA UNACH”**

AUTOR (ES):

Kléber Cristóbal Bustán Rodríguez
Jorge Braulio Álvarez Sayay

TUTOR:

Ing. Diego Palacios Mgs.

RIOBAMBA-ECUADOR

2016

Los miembros del Tribunal de Graduación del proyecto de investigación de título:

“ANÁLISIS DE LAS TECNOLOGÍAS AXIS 2 Y METRO 2.0 PARA EL DESARROLLO DE SERVICIOS WEB SEGUROS EN JAVA, APLICADO AL MÓDULO DE INTEGRACIÓN Y REPORTES DEL SISTEMA DE EVALUACIÓN DOCENTE DE LA UNACH”

presentado por:

KLÉBER CRISTÓBAL BUSTÁN RODRÍGUEZ

JORGE BRAULIO ÁLVAREZ SAYAY

dirigida por el

ING. DIEGO PALACIOS MGS

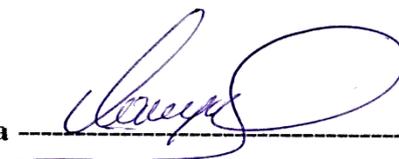
Una vez escuchada la defensa oral y revisado el informe final del proyecto de investigación con fines de graduación escrito en la cual se ha constatado el cumplimiento de las observaciones realizadas, remite la presente para uso y custodia en la biblioteca de la Facultad de Ingeniería de la UNACH.

Para constancia de lo expuesto firman:

Ing. Danny Velasco

Presidente del Tribunal

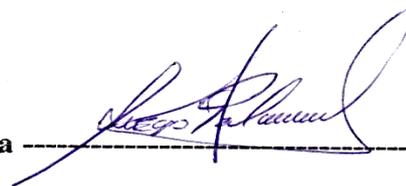
Firma



Ing. Diego Palacios

Miembro del Tribunal

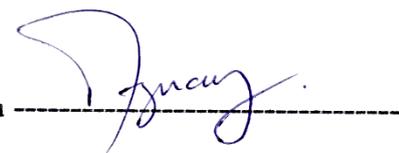
Firma



Ing. Paúl Paguay

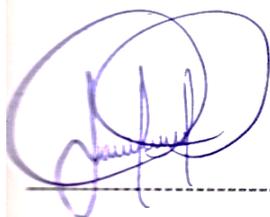
Miembro del Tribunal

Firma



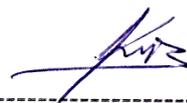
AUTORÍA DE LA INVESTIGACIÓN

“La responsabilidad del contenido de este Proyecto de Graduación, corresponde exclusivamente a: Kléber Cristóbal Bustán Rodríguez y Jorge Braulio Álvarez Sayay, autores del proyecto de investigación, al Ing. Diego Palacios, Director de Tesis, al Ing. Paúl Paguay, Miembro de tribunal; y el patrimonio intelectual de la misma a la Universidad Nacional de Chimborazo”.



Jorge Braulio Álvarez Sayay

0604956409



Kléber Cristóbal Bustán Rodríguez

0604789420

AGRADECIMIENTO

En el presente trabajo de Investigación queremos agradecer primero a Dios por brindarnos salud y vida y por permitirnos hacer realidad nuestro anhelado sueño.

Expresamos nuestro agradecimiento a la **Universidad Nacional de Chimborazo**, la cual nos abrió sus puertas para culminar con éxito una etapa más de nuestras vidas, preparándonos para un futuro competitivo y poder servir a la sociedad con nuestros sólidos conocimientos para el progreso del país.

Un reconocimiento especial a nuestro Director de Tesis, **Ing. Diego Bernardo Palacios Campana** por su calidad humana y todo el apoyo brindado al instruirnos y guiarnos a realizar el presente trabajo investigativo.

Agradecemos al **Ing. Paúl Xavier Paguay Soxo** por todo el apoyo brindado técnico y moral al guiarnos en la culminación del presente trabajo de Investigación.

A los docentes de la Carrera de Ingeniería en Sistemas y Computación porque todos han aportado con sus conocimientos y consejos para nuestra formación profesional.

Para ellos muchas gracias y que Dios os bendiga

Kléber Bustán / Jorge Álvarez

DEDICATORIA

Doy gracias a Dios, por estar conmigo en cada paso que doy, por guiarme e iluminar mi mente y permitirme cumplir una meta más en mi vida.

A mis padres Cristóbal Bustán y Carmen Rodríguez por brindarme incondicionalmente su amor, consejos, ánimos para alcanzar mi más grandioso sueño el ser un profesional y son mi inspiración para seguir adelante y cumplir mis metas.

A mis hermanas Mayra, Fernanda, Maribel, Anabel, Dayana, Liseth, mi sobrino Jordan, a mi cuñado Diego Bucay, amigos y a Sandra Contento por sus consejos constantes y su apoyo incondicional en los momentos más conflictivos y difíciles de mí vida.

A mi amigo Jorge Álvarez por el apoyo brindado, porque sin el equipo que formamos no hubiéramos logrado esta meta muy importante en nuestras vidas.

Y a todas las personas que han sido mi soporte y compañía durante toda mi formación académica.

Kléber Bustán

Este trabajo de investigación os dedico a Dios por haberme permitido llegar hasta este punto y haberme dado salud y vida para lograr mis objetivos, además de su infinita bondad y amor.

A mis padres porque han estado conmigo en cada paso que doy a lo largo de mi vida velando por mi bienestar y educación siendo mis apoyos en todo momento depositando su entera confianza en cada reto que se me presentaba sin dudar si un solo momento en mi capacidad y habilidad.

A mis hermanos, y mi única hermana quienes son mi inspiración para ser mejor cada día.

A mi amigo Kléber Bustán por el apoyo brindado ya que sin el grupo de apoyo que formamos no se hubiera finalizado nuestro trabajo de investigación.

Por ellos soy lo que soy ahora, los quiero mucho.

Jorge Álvarez

ÍNDICE GENERAL

ÍNDICE GENERAL	1
ÍNDICE DE ILUSTRACIONES	6
ÍNDICE DE TABLAS.....	8
ÍNDICE DE ABREVIATURAS Y ACRÓNIMOS	10
RESUMEN.....	12
SUMARY	13
INTRODUCCIÓN	14
CAPÍTULO I.....	16
PROBLEMATIZACIÓN	16
1.1. Identificación y descripción del problema.	16
1.2. Análisis Crítico	17
1.3. Prognosis	17
1.4. Justificación	17
1.5. Delimitación	18
1.6. Formulación del problema	18
1.7. Objetivos	18
1.7.1. General.....	18
1.7.2. Específicos.....	18
1.8. Hipótesis	18
CAPÍTULO II	19
FUNDAMENTACIÓN TEÓRICA.....	19
2. SERVICIOS WEB	19
2.1.1. Generalidades	19
2.1.2. Definición	19
2.1.3. Estándares de los Servicios Web	20
2.1.4. Ventajas y Desventajas	22
2.1.5. Seguridad de los Servicios Web	23
2.2. LENGUAJE DE PROGRAMACION JAVA	25
2.2.1. Características.....	25
2.2.2. Beneficios de Java	26
2.2.3. Tecnologías Java	27
2.2.3.1. Java J2SE (Plataforma Java, Standard Edition).....	27
2.2.3.2. Java J2ME (Plataforma Java, Micro Edition).....	27
2.2.3.3. Java J2EE - JEE (Plataforma Java, Enterprise Edition)	28
2.3. TECNOLOGÍAS PARA SERVICIOS WEB.....	29
2.3.1. AXIS 2	29

2.3.1.1.	Principales características Axis 2	29
2.3.1.2.	Modelo de programación	31
2.3.1.3.	Versiones de Axis	32
2.3.1.4.	Implementaciones de Servicios web Soportadas en Axis 2	32
2.3.1.5.	Servidores Soportados.....	32
2.3.1.6.	Navegadores Soportados.....	32
2.3.1.7.	Ventajas	32
2.3.1.8.	Desventajas.....	33
2.3.2.	METRO 2.0	34
2.3.2.1.	Principales características Metro 2.0.....	34
2.3.2.2.	Versiones de Metro	36
2.3.2.3.	Implementaciones de Servicios web Soportadas en Metro 2.0.....	36
2.3.2.4.	Servidores Soportados.....	36
2.3.2.5.	Navegadores Soportados.....	37
2.3.2.6.	Ventajas	37
2.3.2.7.	Desventajas.....	37
2.3.3.	VULNERABILIDADES EN SERVICIOS WEB	38
2.3.3.1.	Confidencialidad	38
2.3.3.2.	Integridad	40
2.3.3.3.	Disponibilidad	41
CAPÍTULO III.....		42
ANÁLISIS DE LAS TECNOLOGÍAS DE SERVICIOS WEB AXIS 2 Y METRO 2.0.		42
.....		42
3.1.	Diseño de la investigación	42
3.2.	Tipo de investigación.....	42
3.3.	Métodos	42
3.4	Técnicas de investigación.....	42
3.5.	Población y Muestra.....	43
3.6.	Instrumentos para el testeo de seguridad	43
3.6.1.	SoapUI.....	43
3.6.2.	WebInject.....	44
3.6.3.	Testmaker	44
3.7.	Validación de instrumentos	45
3.8.	Selección de la herramienta.....	45
3.8.1.	Hardware	45
3.8.2.	Software	46
3.9.	Escenario de prueba.....	46

3.10.	Proceso de prueba.....	47
3.11.	Test del prototipo de la tecnología de servicio web Axis2.....	48
3.11.1.	Parámetro de Confidencialidad	48
3.11.1.1.	Cross Site Scripting:.....	48
3.11.1.2.	SQL Injection:.....	49
3.11.1.3.	XPath Injection	49
3.11.2.	Parámetro de Integridad	49
3.11.2.1.	Malformed XML.....	49
3.11.2.2.	XML Bomb.....	49
3.11.2.3.	Invalid Types.....	50
3.11.3.	Parámetro de Disponibilidad	50
3.11.3.1.	Denegación de servicios.....	50
3.12.	Test del prototipo de la tecnología de servicio web Metro 2.0.....	50
3.12.1.	Parámetro de Confidencialidad	51
3.12.1.1.	Cross Site Scripting	51
3.12.1.2.	SQL Injection.....	51
3.12.1.3.	XPath Injection	51
3.12.2.	Parámetro de Integridad	51
3.12.2.1.	Malformed XML.....	51
3.12.2.2.	XML Bomb.....	52
3.12.2.3.	Invalid Types.....	52
3.12.3.	Parámetro de Disponibilidad	52
3.12.3.1.	Denegación de servicios.....	52
3.13.	Análisis de resultados	53
3.13.1.	Parámetro de Confidencialidad	53
3.13.2.	Parámetro de Integridad	54
3.13.3.	Parámetro de Disponibilidad	55
3.14.	Resumen del número total de errores de seguridad.....	56
3.15.	Comprobación de hipótesis.....	58
3.15.1.	Hipótesis Nula (Ho)	58
3.15.2.	Hipótesis Alternativa (H1).....	58
3.15.3.	Nivel de Significancia: 0.05.....	58
3.16.	Especificaciones de las regiones de aceptación y rechazo:.....	58
3.16.1.	Tipo de análisis:.....	58
3.16.2.	Especificación de estadístico prueba Z.....	58
3.17.	Recolección y cálculo de datos estadísticos.....	59

3.18.	Evaluación con la prueba Z	59
3.19.	Conclusión de la Hipótesis	60
CAPÍTULO IV		61
DESARROLLO DEL MÓDULO DE INTEGRACIÓN Y REPORTES DEL SISTEMA DE EVALUACIÓN DOCENTE DE LA UNACH.		61
4.1.	Metodología XP	61
4.2.	Desarrollo de los Módulos de Integración y Reportes	62
4.2.1.	Herramientas de desarrollo	62
4.3.	Gestión de Migración y Reportes del Sistema Informático de Evaluación Docente de la UNACH.	63
4.3.1.	Planificación del Proyecto	63
4.3.2.	Integrantes y roles.....	63
4.8.3.	Prototipos.....	63
4.3.3.1.	Migración de datos	64
4.3.3.2.	Generación de reportes:.....	64
4.3.4.	Historias de Usuarios	65
4.3.5.	Plan de Entregas	66
4.3.6.	Incidencia.....	67
4.3.7.	Actividades de Migración.....	68
4.3.8.	Actividades de Reportes	74
4.4.	Implementación	76
4.4.1.	Diagrama de base de datos.....	76
4.4.2.	Diccionario de datos.....	78
4.4.2.1.	Migración por Carreras y Dictados Asignatura.....	78
4.4.2.2.	Creación de Reporte Heteroevaluación finalizadas y Consolidado por docente.....	81
4.4.3.	Funciones	82
4.4.3.1.	Interfaces de Migración	85
4.4.3.2.	Interfaces de Reportes.....	86
4.4.4.	Iteración 1	87
4.4.5.	Iteración 2	89
4.10.1.	Historia de Usuario 1	91
4.10.2.	Historia de Usuario 2	92
4.10.3.	Historia de Usuario 3	93
4.10.4.	Historia de Usuario 4	93
4.10.5.	Historia de Usuario 5	94

CONCLUSIONES	95
RECOMENDACIONES	96
BIBLIOGRAFIA	97
ANEXOS	101

ÍNDICE DE ILUSTRACIONES

Ilustración 1: Funcionalidad del Servicio Web.....	19
Ilustración 2: Interacción de estándares de servicios web	22
Ilustración 3 Logo de Java	25
Ilustración 4: Tecnologías JAVA.....	27
Ilustración 5: Logo de Axis 2.....	29
Ilustración 6: Logo Metro 2.0	34
Ilustración 7: Características de los servicios web	35
Ilustración 8: Ataque de Cross Site Scripting	38
Ilustración 9: Ataque de SQL Injection	38
Ilustración 10: Ataque de XPath Injection.....	39
Ilustración 11: Ataque de Malformed XML	40
Ilustración 12: Ataque de XML Bomb	40
Ilustración 13: Ataque de Invalid Types	41
Ilustración 14: Error de denegación de servicios	41
Ilustración 15: Logo de SoapUI.....	43
Ilustración 16: Logo de WebInject	44
Ilustración 17: Logo de Testmaker PushtoTest	44
Ilustración 18: Validación de Instrumentos para testeo de seguridad.....	45
Ilustración 19: Prototipos Axis 2 y Metro 2.0 creados para medir la seguridad en SW	47
Ilustración 20: Test de Prototipo Axis2	48
Ilustración 21: Cross Site Scripting aplicado en Axis 2	48
Ilustración 22: SQL Injection aplicado en Axis 2.....	49
Ilustración 23: Xpath Injection aplicado en Axis 2	49
Ilustración 24: Malformed XML aplicado en Axis 2.....	49
Ilustración 25: XMLBomb aplicado en Axis 2.....	49
Ilustración 26: Invalid Types aplicado en Axis 2	50
Ilustración 27: Errores de denegación de servicio en Axis 2.....	50
Ilustración 28: Test Prototipo Metro 2.0.....	50
Ilustración 29: Zona de Aceptación	58
Ilustración 30: Fases de la Metodología XP	61
Ilustración 31: Modulo de Migración	64
Ilustración 32: Generación de reportes	64
Ilustración 33: Generación de reportes por carrera de los estudiantes.....	64
Ilustración 34: Generación de reportes de los docentes por carrera	65
Ilustración 35: Plan de Entregas Iteración 1	66
Ilustración 36: Plan de Entregas Iteración 2	67
Ilustración 37: Plan de Entregas Iteración 2	67
Ilustración 38: Esquema Base de Datos-Postgresql.....	76
Ilustración 39 : Diagrama Entidad Relación BD	77
Ilustración 40: Migración Carrera.....	85
Ilustración 41: Migración por Dictados Asignatura.....	85
Ilustración 42: Generación de Reporte Administrador	86
Ilustración 43: Generación de Reporte Administrador	86
Ilustración 44: Sistema SICOA de la UNACH.....	122
Ilustración 45: Base de datos SICOA Modelo Distributivos docentes	123

Ilustración 46: Base de datos SICOA Modelo Matriculas.....	124
Ilustración 47: Sistema de Evaluación Docente UNACH	125
Ilustración 48: Base de Datos Evaluación_unach.....	126
Ilustración 49: Integración del Sistema de Evaluación Docente con el Sistema SICOA-UNACH.	127
Ilustración 50: Servicio Web utilizado para la Integración	128

ÍNDICE DE TABLAS

Tabla 1: Características del lenguaje Java	25
Tabla 2: Hardware utilizado para realizar pruebas de seguridad del SW	45
Tabla 3: Software utilizado para realizar pruebas de seguridad en SW	46
Tabla 4: Parámetros de Confidencialidad de la Tecnología Metro 2.0.....	53
Tabla 5: Parámetros de Confidencialidad Tecnología Axis 2	53
Tabla 6: Parámetros de Integridad de la Tecnología Metro 2.0.....	54
Tabla 7: Parámetros de Integridad de la tecnología Axis 2	55
Tabla 8: Parámetro de Disponibilidad en la Tecnología Metro 2.0.....	55
Tabla 9: Parámetro de Disponibilidad en la Tecnología Axis 2	56
Tabla 10: Resumen del número total de errores de seguridad en Axis 2 y Metro 2.0.....	57
Tabla 11: Recolección de datos Estadísticos	59
Tabla 12: Herramientas Utilizadas para el desarrollo de los módulos de Integración y Reportes	62
Tabla 13: Integrantes y Roles	63
Tabla 14: Historias de Usuarios.....	65
Tabla 15: Prototipos de Migración y Consumo de servicios web en semanas	66
Tabla 16: Plan de Entrega Iteración 2.....	66
Tabla 17: Plan de entrega iteración 2.....	67
Tabla 18: Proceso Migrar Facultades	69
Tabla 19: Proceso Migrar Carreras	69
Tabla 20: Proceso Migrar Periodos.....	70
Tabla 21: Proceso Migrar Docentes.....	70
Tabla 22: Proceso Migrar Estudiantes	71
Tabla 23: Proceso Migrar Asignaturas	71
Tabla 24: Proceso Migrar Niveles	72
Tabla 25: Proceso Migrar Paralelos	72
Tabla 26: Proceso Migrar Dictados Asignatura.....	73
Tabla 27: Proceso Actividades Reportes	74
Tabla 28: Proceso Creación de servicio web con seguridad.....	75
Tabla 29: Facultad.....	78
Tabla 30: Carrera	78
Tabla 31: Periodo.....	78
Tabla 32: Estudiante	79
Tabla 33: Docente	79
Tabla 34: Asignatura.....	80
Tabla 35: Nivel	80
Tabla 36: Paralelo	80
Tabla 37: Dictado_asignatura	80
Tabla 38: Evaluación	81
Tabla 39: Coevaluación	81
Tabla 40: Evaluación_hetero_docencia	81
Tabla 41: Evaluación_docencia	82
Tabla 42: Función de porcentaje de autodirección y gestión.....	82
Tabla 43: Función de porcentaje auto docencia.....	82
Tabla 44: Función de porcentaje auto investigación.....	83

Tabla 45: Función para calcular el promedio autodirección.....	83
Tabla 46: Funciones de valor máximo auto dirección y gestión	83
Tabla 47: Funciones de valor máximo auto investigación.....	83
Tabla 48: Funciones de valor máximo dirección docencia.....	84
Tabla 49: Funciones de valor máximo Evaluaciones	84
Tabla 50: Iteración 1, Historia 1	87
Tabla 51: Iteración 1, Historia 2	88
Tabla 52: Iteración 2, Historia 1	89
Tabla 53: Iteración 2, Historia 2	89
Tabla 54: Iteración 2, Historia 3	90
Tabla 55: Prueba Historia 1 - Migración por carreras	91
Tabla 56: Prueba Historia 2 - Migración de dictados por asignatura	92
Tabla 57: Prueba Historia 3 - Reporte heteroevaluaciones.....	93
Tabla 58: Prueba Historia 4 - Reporte consolidado por docente	93
Tabla 59: Prueba Historia 5 - Servicio Web del listado de estudiantes evaluados	94

ÍNDICE DE ABREVIATURAS Y ACRÓNIMOS

UNACH	Universidad Nacional de Chimborazo
SICOA	Sistema de Control Académico
UTECA	Unidad Técnica Académica
CISYC	Carrera de Ingeniería en Sistemas y Computación
ADB	Axis2 DataBinding
SDO	Objetos de datos de servicio
DDoS	Distribuid Denial of Service
DTIC	Dirección de Tecnologías de la Información y Comunicación
FTP	File Transfer Protocol
GB	Gigabyte
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
IIS	Internet Information Server
JAX	Java API for XML
JAXB	Java Architecture for XML Binding
JDK	Java Development Kit
JSF	Java Server Face
MVC	Model View Controller
OASIS	Advancing Open Standards for the Information Society
OASIS	Orion Academic System with Internet Services
PDF	Portable Document Format
POJO	Plain Old Java Objects
RAM	Random Access Memory
RPC	Remote Procedure Call
SAAJ	SOAP with Attachments API for Java
SOAP	Simple Object Access Protocol
SOAPUI	Simple Object Access Protocol User Interface
SSL	Security Sockets Layer

UDDI	Universal Discovery, Description and Integration
VM	Virtual Machine
WCF	Windows Communication Foundation
WS	Web service
WSDL	Web Services Definition Language
WSIT	Web Service Interoperability Technologies
XHTML	eXtensible HyperText Markup Language
XML	eXtensible Markup Language
XP	Extreme Programming
XSD	XML Schema
SIAE	Sistema Inteligente de Análisis Estadístico

RESUMEN

La presente investigación tiene como objetivo estudiar las tecnologías de Servicio Web Axis 2 y Metro 2.0 para el desarrollo de servicios web seguros en java aplicado al Sistema Informático de Evaluación Docente de la Universidad Nacional de Chimborazo, es determinar cuál de las tecnologías de Servicio Web ofrece un menor número de errores de seguridad.

Para su realización se creó un prototipo de Servicio Web sobre la tabla Facultad para cada una de las tecnologías de servicio web que se mencionó anteriormente, luego se procedió a realizar el respectivo test de seguridad utilizando la Herramienta SoapUI, además se analizó la cantidad de errores de seguridad de los prototipos llegando a comprobar que la incidencia de seguridad de las Tecnologías de Servicio Web utiliza estadística descriptiva, el Sistema Inteligente de Análisis Estadístico (SIAE) y la prueba Z.

De acuerdo a los resultados del test de seguridad y su análisis se obtuvo los siguientes resultados: La tecnología Axis 2 en el parámetro de Confidencialidad obtuvo un 0% en errores de seguridad frente a la tecnología Metro 2.0, en lo referente al parámetro de Integridad la tecnología de Axis 2 de igual manera posee un 0% en errores de seguridad en relación a la tecnología Metro 2.0 y en el parámetro de Disponibilidad la tecnología Axis 2 decrece un 0.16% en errores de seguridad en medida a la tecnología Metro 2.0, evidenciando que la tecnología Axis 2 difiere un menor número de errores de seguridad por lo que se acepta la hipótesis; la tecnología de servicios web Axis 2 presenta un menor número de errores de seguridad en comparación a la tecnología Metro 2.0.

Por lo tanto, se procede a desarrollar los Módulos de integración y reportes del Sistema Informático de Evaluación Docente de la Universidad Nacional de Chimborazo, utilizando la metodología XP, para el seguimiento de las iteraciones con sus respectivas historias de usuario y pruebas de aceptación de acuerdo a la planificación, se utilizó NetBeans 8.1 como IDE de desarrollo, Apache Tomcat 8.0 como servidor de aplicaciones, Postgresql 9.5 como motor de Base de Datos.

Palabras Claves:

<UNIVERSIDAD NACIONAL DE CHIMBORAZO>, <SISTEMA INTELIGENTE DE ANALISIS ESTADISTICO>, <METODOLOGIA XP>, <METRO 2.0>, <AXIS 2>, <SEGURIDAD>, <SERVICIOS WEB>, <INTEGRIDAD>, <CONFIDENCIALIDAD>, <DISPONIBILIDAD>.

SUMMARY

Abstract

This research aims to study the technologies axis 2 web service and metro 2.0 for developing secure web services in java applied to computer evaluation system of the National University of Chimborazo, to determine which of the technologies of web service offers fewer security errors. To make this possible, a prototype web service on the faculty table for each of the technologies web service mentioned above, was created, then proceeded to the respective safety tests using the soapUI tool, plus the number of errors was analyzed security of the prototypes coming to see that the security incident technologies web service uses descriptive statistics, intelligent statistical analysis system (EMIS) and Z test. According to the results of safety tests and analysis, the following results were obtained: the axis 2 technology in the confidential parameter obtained a 0% errors of security against metro 2.0 technology, in relation to the parameter of integrity technology axis 2 likewise has a 0% errors of security in relation to the metro 2.0 technology, showing that axis 2 technology differs fewer security errors compared to metro 2.0 technology. Therefore, the integration modules and reports computer teacher evaluation system of the National University of Chimborazo are developed, using the XP methodology for tracking iterations with their user stories and acceptance tests according planning. NetBeans IDE 8.1 was used as development, Apache Tomcat 8.0as application server, as Postgresql 9.5 database engine.



SIGNATURE

Reviewed by Solis, Hugo
Language Center Teacher



INTRODUCCIÓN

En la actualidad el uso de los servicios web se ha vuelto popular debido a que estos permiten la interacción de aplicaciones web desarrollados en plataformas heterogéneas, haciendo que estas sean interoperables, escalables, fáciles de mantener e integrar con sistemas y fuentes de datos existentes. Dado estas condiciones se han desarrollado tecnologías de servicios web que permiten implementar aplicaciones de este tipo.

Mediante las tecnologías de servicios web se pueden desarrollar tanto el servicio como el cliente, es decir el emisor y el receptor, que son fundamentales en toda aplicación interoperable. Dentro del desarrollo de este tipo de aplicaciones el cliente del servicio web juega un papel vital debido a que este es el encargado de entender los contratos expuestos por los servicios web y con ello lograr una comunicación directa entre aplicaciones desarrolladas en plataformas heterogéneas.

La presente investigación pretende realizar un análisis entre las tecnologías de servicios web Axis 2 y Metro 2.0 orientados a la seguridad desde el lado del cliente, para el desarrollo de aplicaciones que consumen servicios web seguros; contemplando el posterior desarrollo del Módulo de Integración y Reportes del sistema de Evaluación Docente para la Universidad Nacional de Chimborazo (UNACH) con la tecnología que brinde mejores prestaciones.

Este proyecto de investigación está estructurado en cuatro capítulos.

En el **Capítulo I**, se tratará sobre el marco referencial, en el cual se encuentra descrita de manera general los antecedentes, la justificación del proyecto de tesis, los objetivos a alcanzar y la hipótesis a demostrar con el desarrollo de la misma.

En el **Capítulo II**, se detallarán las definiciones conceptuales de los servicios web, interoperabilidad entre aplicaciones informáticas, lenguaje de programación java, tecnologías de servicios web Axis 2 y Metro 2.0, estos dos últimos enfocados en la seguridad desde el cliente del servicio web.

En el **Capítulo III**, se enfoca en la realización del análisis comparativo entre las tecnologías de servicios web Axis 2 y Metro 2.0 enfocados en la seguridad desde el cliente, teniendo como objetivo demostrar las fortalezas y debilidades mediante los parámetros de comparación establecidos en este capítulo.

En el **Capítulo IV**, con la tecnología que se seleccione en el Capítulo III, es decir, el que brinde mayores beneficios, se procederá al desarrollo del Módulo de Integración y Reportes del sistema de Evaluación Docente para la Universidad Nacional de Chimborazo; este capítulo se constituye por la documentación del sistema aplicado la metodología Programación Extrema (XP) logrando de esta forma relacionar lo investigativo con lo práctico.

La presente investigación finaliza emitiendo las conclusiones y recomendaciones que son resultado del trabajo realizado.

El presente trabajo, servirá de soporte a la hora de seleccionar entre las dos tecnologías de servicios web mencionados anteriormente y decidir cuál es el más adecuado para el desarrollo de aplicaciones que consuman servicios web seguros.

CAPÍTULO I

PROBLEMATIZACIÓN

1.1. Identificación y descripción del problema.

Si bien pueden existir múltiples definiciones del término “servicios web”, una definición de sencillo entendimiento puede ser que los servicios web son un conjunto de aplicaciones o de tecnologías que interoperan, intercambian información y utilizan la información intercambiada, con el objetivo de ofrecer servicios mediante la interoperabilidad basada en la web. Esta interoperabilidad permite que las aplicaciones se comuniquen de forma independiente de la plataforma o de lenguajes de programación usados en cada una de las aplicaciones, deberían contener protocolos de seguridad para que contribuyan directamente a la eliminación de lo que actualmente sucede con las aplicaciones en la web, que son las constantes violaciones de confidencialidad, violación de seguridad, ataques SQL inyección, buffer overflow y ataques de repetición.

La seguridad en los servicios web es muy importante al instante de envía y recibir información mediante la capa de transporte y mensajería por lo tanto se debería utilizar XML Encryption, XML Signature y WS Security para una mejor privacidad de datos.

Actualmente la Universidad Nacional de Chimborazo posee información sobre las evaluaciones que se realizan conjuntamente con los estudiantes, administrada por la aplicación informática que gestiona el proceso de evaluación docente, donde la aplicación no se encuentra integrada en su totalidad con el sistema SICOA.

El proceso para realizar una nueva evaluación a los docentes se realiza de la siguiente forma, el departamento de evaluación y acreditación solicita al departamento de UTECA información actualizada del docente, el archivo que entrega el sistema SICOA es un documento de Excel que contiene los siguientes campos: Nombre y Apellido del docente, dirección, teléfono, materia, paralelo, facultad y carrera, con esta información proporcionada por el DEA transcriben estos datos en su sistema el cual conlleva mucho tiempo y personal.

Por lo que nace la necesidad de realizar el Análisis de las tecnologías Axis 2 y Metros 2.0 para el desarrollo de servicios web seguros en java, aplicado al módulo de integración y reportes del sistema de evaluación docente de la UNACH.

1.2. Análisis Crítico

Controlar la seguridad de los servicios web publicados en internet no es tarea sencilla. Los cortafuegos IP más avanzados que filtran la capa de aplicación no son de gran ayuda, las peticiones HTTP correctas las tiene que dejar pasar. Con otros protocolos TCP/IP es más fácil, ya que utiliza el protocolo adecuado o se frenan, pero en caso de SOAP sobre HTTP, aunque se utilice el protocolo adecuado, la validez de los datos XML no la filtran los cortafuegos. Siempre hay que tener en cuenta todas las facetas de la seguridad a la hora de publicar un servicio web en internet, ya que los atacantes utilizarán el punto más débil de todo proceso e infraestructura para colarse y dañar.

1.3. Prognosis

Las tecnologías de Axis 2 o Metro 2.0 con respecto a la seguridad, facilitara el consumo de servicios web encriptados para el módulo de integración y reportes del Sistema de Evaluación Docente de la UNACH y de esta manera disminuirán la vulnerabilidad de información.

1.4. Justificación

Las tecnologías de servicios web proporcionan un entorno de ejecución, lo cual permite una comunicación exitosa y rápida con aplicaciones desarrolladas en plataformas heterogéneas. El principal objetivo de estas tecnologías es desarrollar aplicaciones web interoperables de manera rápida, fácil y con la máxima reutilización de código. Las tecnologías estables en el mercado son Axis 1.x, Axis 2, Celtix, Glue, JBossWS, Xfire 1.2 y Metro 2.0, de los cuales según el Proyecto de Entorno de desarrollo de software “CODEHAUSE” (BIJOY, 2014), Axis 2 y Metro 2.0 son las que proveen más funcionalidades.

El presente análisis pretende estudiar las tecnologías de consumo de los servicios web, Axis 2 y Metro 2.0 para el desarrollo de aplicaciones que consuman servicios web seguros, enfocándonos principalmente en el incremento del desempeño de las aplicaciones web interoperables, ambas tecnologías son similares y presentan un gran crecimiento y acogida entre los desarrolladores de aplicaciones web, por lo cual se considera relevante el poder determinar por medio de esta investigación el comportamiento de estas tecnologías que permitan describir con claridad la mejor opción para el incremento del desempeño de una aplicación web interoperable. Los módulos de Integración y Reportes se va desarrollar en la Universidad Nacional de Chimborazo, Departamento de Evaluación y Acreditación, la misma que estará integrada al Sistema de Control Académico SICOA, además va estar desarrollada con servicios web seguros el cual, evitará las constantes violaciones de confidencialidad,

violación de seguridad, ataques SQL inyección, buffer overflow y ataques de repetición, de esta forma la información se trasladara de una forma segura, con la finalidad de cumplir con uno de los indicadores que el CEAACES requiere al momento de evaluar a la institución.

1.5. Delimitación

El estudio de la investigación se basará en el Análisis de las tecnologías Axis 2 y Metro 2.0 con respecto a la seguridad de servicios web aplicado a los módulos de integración y reportes del SISTEMA DE EVALUACIÓN DOCENTE DE LA UNIVERSIDAD NACIONAL DE CHIMBORAZO, que se encuentra ubicado en la Avenida Antonio José de Sucre, Km 1 ½ vía a Guano en la ciudad de Riobamba provincia de Chimborazo.

1.6. Formulación del problema

¿Cómo inciden las tecnologías de Axis 2 o Metro 2?0? con respecto a la seguridad en los servicios web, en los módulos de integración y reportes del SISTEMA DE EVALUACIÓN DOCENTE DE LA UNACH?

1.7. Objetivos

1.7.1. General

Analizar las tecnologías Axis 2 y Metro 2.0 para el desarrollo de servicios web seguros en java, aplicado en los módulos de integración y reportes del sistema de evaluación docente de la UNACH.

1.7.2. Específicos

- ✓ Describir las características de las tecnologías Axis 2 y Metro 2.0.
- ✓ Comparar las tecnologías Axis 2 y Metro 2.0 con respecto a la seguridad.
- ✓ Desarrollar el módulo de reportes y el módulo de integración del sistema de evaluación docente con el sistema de control académico SICOA de la UNACH.

1.8. Hipótesis

La tecnología de servicios web Axis 2 presenta un menor número de errores de seguridad frente a la tecnología Metro 2.0, para el desarrollo de servicios web aplicado al Sistema Informático de Evaluación docente de la UNACH.

CAPÍTULO II

FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se detallan las definiciones conceptuales de servicios web incluyendo sus ventajas, desventajas, seguridad y despliegue en los servidores web. Además, se dará a conocer las tecnologías de servicios web Axis 2 y Metro 2.0, utilizados en la presente investigación.

2. SERVICIOS WEB

2.1.1. Generalidades

En la actualidad la globalidad de la información hace que la forma de comunicación entre las personas y los negocios que realizan estas cambien rotundamente, haciendo que las aplicaciones web tradicionales no sean suficientes para esta demanda, bajo este contexto surge la necesidad de desarrollar aplicaciones que se integren a otras independientemente de su plataforma de desarrollo. Para lograr esto, las empresas de desarrollo de software más grandes en el mundo, tales como Microsoft, IBM, Oracle han desarrollado un lenguaje común que permita el intercambio de información basándose en estándares existentes, surgiendo así los servicios web. (IBM, 2013)

2.1.2. Definición

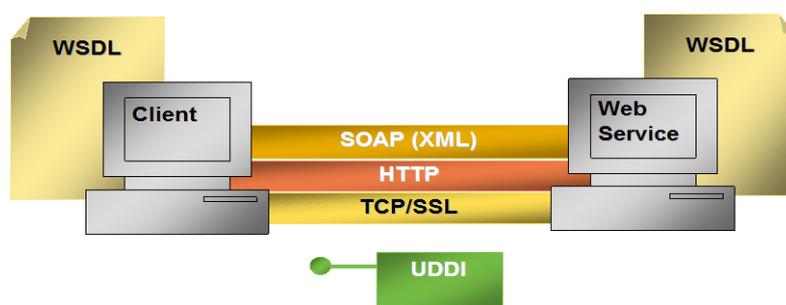


Ilustración 1: Funcionalidad del Servicio Web
Fuente: Pablo, (Peris Soler, 2014)

Observando la ilustración 1 existen varias definiciones sobre servicios web como son:

W3C.- Un servicio Web es un sistema de software diseñado para apoyar la interoperabilidad de máquina a máquina sobre una red de interacción. Tiene una interfaz descrita en un formato procesable por máquina (específicamente WSDL). Otros sistemas interactúan con el servicio Web en la forma prescrita por su descripción utilizando mensajes SOAP, típicamente

transportados usando HTTP con una serialización XML en conjunción con otros estándares relacionados con la web. (W3C, 2012)

Sun Microsystems. - Un servicio Web describe la funcionalidad específica del negocio expuesta por una compañía, generalmente a través de una conexión de Internet, con el fin de proporcionar una manera para que otra compañía, o programa informático utilice el servicio. (webservice, 2011)

Microsoft. - Los servicios Web son componentes de un servidor web a los que puede llamar una aplicación cliente realizando solicitudes HTTP a través de Internet.

De lo expuesto anteriormente, un servicio web es un componente de software que expone funcionalidades de negocio a través de una red, para que sean consumidas por una aplicación cliente mediante peticiones HTTP. (Newcomer, 2012)

2.1.3. Estándares de los Servicios Web

Los servicios web para lograr un funcionamiento y una comunicación se basan en los siguientes estándares web:

XML: eXtensible Markup Language

El Extensible Markup Language (XML) es un simple formato basado en texto para representar información estructurada: documentos, datos, configuración, libros, transacciones, facturas y mucho más. Físicamente, el documento está compuesto de unidades llamadas entidades, todo documento tiene una entidad raíz. En servicios web XML, todos los datos que se intercambiarán están formateados con etiquetas XML. (estandarwebservice, 2012)

SOAP: Simple Object Access Protocol

SOAP es un protocolo liviano, basado en XML, para el intercambio de información estructurada en un ambiente descentralizado y distribuido. Se encuentra en el núcleo de los servicios web proporcionando un mecanismo estándar de empaquetar mensajes.

Algunas de las mayores Compañías que soportan SOAP son Microsoft, IBM y Oracle. (González, 2014)

WSDL: Web Services Definition Language

A medida que los protocolos de comunicación y formatos de mensaje se estandarizados en la comunidad web, se hace cada vez más posible e importante ser capaz de describir las comunicaciones de una manera estructurada. WSDL aborda esta necesidad definiendo una gramática XML para describir servicios de red como colecciones de extremos de comunicación capaces de intercambiar mensajes. Las definiciones de servicio WSDL proporcionan documentación para sistemas distribuidos y sirven como una receta para la automatización de los detalles involucrados en las aplicaciones de comunicación.

Es importante observar que WSDL no introduce un lenguaje de definición de tipo nuevo. WSDL reconoce la necesidad de sistemas de tipos ricos para describir formatos de mensaje, y es compatible con la especificación de esquemas XML (XSD) como su sistema de tipo canónico. Además, WSDL define un mecanismo de enlace común. Esto se utiliza para enlazar un protocolo específico, o formato de datos o la estructura de un mensaje abstracto, operación, del extremo. Se permite la reutilización de definiciones abstractas. (Meredith, 2012)

Estructura del Documento WSDL. - Un documento WSDL según (Meredith, 2012) es simplemente un conjunto de definiciones. Hay un elemento de definiciones en la raíz, y definiciones internas. La Gramática es la siguiente:

```
<definitions>
<types>los_tipos_de_datos... </types>
<message> las_definiciones_del_mensaje... </message>
<portType> las_definiciones_de_operación ... </portType>
<binding>las_definiciones_de_protocolo... </binding> <service>direcciones_relacionadas ...
</service>
</definitions>
```

UDDI: Universal Discovery, Description and Integration

Es una especificación industrial para publicar y localizar información acerca de los servicios Web. Se puede utilizar los servicios UDDI para publicar, descubrir, compartir e interactuar con los servicios Web dentro de la empresa o entre los socios comerciales. Con los servicios UDDI, los desarrolladores pueden interactuar con los servicios web directamente a través de sus herramientas de desarrollo y aplicaciones de negocio. UDDI adopta un enfoque que se basa en un registro distribuido de las empresas y su servicio descripciones implementado en un formato XML común. Un proveedor de servicios aloja un servicio Web y lo hace accesible

con protocolos como SOAP/HTTP y SOAP/JMS. El servicio Web se describe mediante los documentos WSDL que se almacenan en el servidor del proveedor o en un depósito especial. Los servicios de empresa UDDI hacen referencia a los documentos WSDL (documentos de servicio) y tModels (documentos de enlace). Estos punteros permiten que los solicitantes de servicio encuentren servicios Web. (Committee, 2013)

Interacción de estándares de la web

Según la ilustración 2, las empresas que desean exponer funcionalidades para que estas sean consumidas por clientes, realizan en primera instancia el registro mediante el WSDL en un repositorio UDDI, una vez ahí, el consumidor realizara una búsqueda en el repositorio para obtener WSDL publicado, y con esto realiza una petición SOAP (XML) vía HTTP al servicio web publicado esperando una respuesta del mismo tipo. En la ilustración 2 se puede observar de forma gráfica la interacción de los estándares.

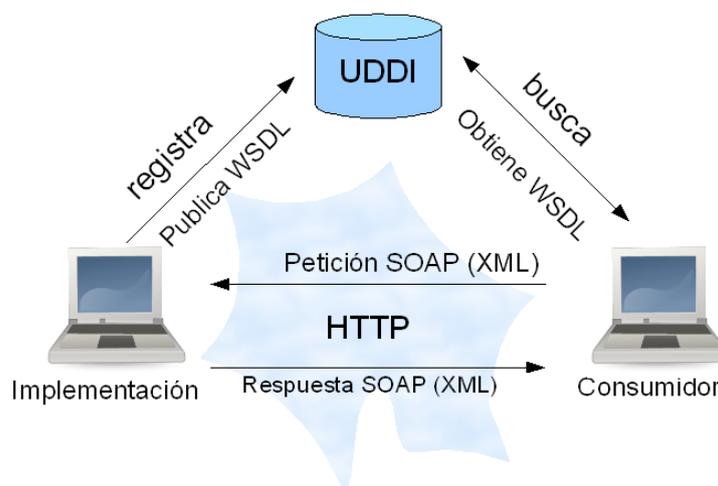


Ilustración 2: Interacción de estándares de servicios web
Fuente: Labra, (José Emilio,2014)

2.1.4. Ventajas y Desventajas

Ventajas

Los servicios web ofrecen muchos beneficios frente a otras arquitecturas de software distribuidas, entre las principales ventajas se puede destacar.

Interoperabilidad. - Este es el beneficio más importante de los Servicios Web. Estos suelen trabajar fuera de las redes privadas, ofreciendo a los desarrolladores una ruta no propietaria a sus soluciones. Los servicios desarrollados son probables, por lo tanto, pueden tener una vida

más larga que ofrecer. Además, permiten que los desarrolladores usen sus lenguajes de programación preferidos. Además, gracias a la utilización de métodos de comunicaciones basadas en estándares, los servicios web son prácticamente independientes de la plataforma.

Independencia. - Totalmente independientes de la plataforma, no hay restricciones en cuanto a la plataforma en la que pueden ser desarrollados, las aplicaciones que utilizan los servicios web pueden ejecutarse en cualquier plataforma.

Basados en estándares de la web. - Los servicios web pueden ser desarrollados como componentes de aplicación débilmente acoplados utilizando cualquier lenguaje de programación, cualquier protocolo o plataforma.

Integración con sistemas existentes. - Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados. (webservice, 2011)

Desventajas.

Reducción de Velocidad de Transmisión. - Los servicios Web utilizan protocolos de texto plano que utilizan un método bastante detallado para identificar los datos. Esto significa que las solicitudes de servicio Web son más grandes que las solicitudes codificadas con un protocolo binario. El tamaño extra es realmente sólo un problema a través de conexiones de baja velocidad o en conexiones extremadamente ocupados.

Implementación de Seguridad. - La creación de servicios web seguros, es una programación avanzada, es decir que se requiere de un gran conocimiento previo para poder configurarlos e implementarlos en un ambiente de producción. (technology, 2016)

2.1.5. Seguridad de los Servicios Web

La especificación SOAP básica no establece la protección de mensajes, y deja esas especificaciones extendidas. El problema nace en la naturaleza de una aplicación de servicios web. En la mayoría de los casos, se trata con SOAP sobre HTTP, lo que significa que cada mensaje debe pasar por uno o más nodos intermedios, cualquiera de los cuales puede leer y/o alterar un mensaje. Y eso suponiendo que la solicitud SOAP misma fuera directa. En algunos casos, los mensajes SOAP están diseñados específicamente para viajar a través de más de un

modo antes de llegar a su destino final. El resultado final es que se necesita evitar que alguien que no sea el destinatario deseado lea información confidencial, a fin de evitar la interceptación. Además, es necesaria una manera de evitar que alguien que no sea el remitente deseado envíe mensajes, a fin de evitar el acceso no autorizado. (M. Piattini, 2012)

Para lograr una seguridad integrada, que implique una autenticación, autorización y encriptación de los servicios web, se han implementado diferentes métodos tales como: especificaciones de seguridad SOA WS-Security y servicios web sobre SSL, este último método, será utilizado en el desarrollo de los servicios web WCF para la presente investigación. (M. Piattini, 2012)

La especificación SOAP. - Hay tres grandes problemas en la protección de intercambios de mensajes SOAP, y WS-Security brinda respuestas para todos ellos, aunque no directamente. Es, de hecho, una especificación que habla no sobre cómo proteger el mensaje, sino cómo hacer saber al destinatario que se ha protegido el mensaje. Para realizar la protección real, WS-Security referencia especificaciones adicionales. (Center, 2015)

Servicios web sobre SSL. - La seguridad de servicios web sobre SSL (Security Sockets Layer) se concentra en la seguridad a nivel de transporte, estableciendo un canal seguro, por el cual se transmiten los datos, es decir que la información se transmite mediante HTTPS. Para lograr este nivel de seguridad se debe en primera instancia registrar el certificado SSL en un servidor web, agregar el enlace SSL al sitio web, configurar el servicio web para que haga uso de HTTPS y dentro de este último se establece el método de autenticación, que permite un nivel superior de seguridad. Cabe aclarar que a diferencia de la seguridad SOA WS-Security, SSL se configura en el servidor y cliente, sin afectar de manera exclusiva a los mismos. Para el desarrollo del aplicativo de la presente investigación se hará uso de este tipo de seguridad, debido a que provee un nivel más óptimo de rendimiento y seguridad. Mediante la seguridad en los servicios web, se pretende garantizar una entrega y recepción de información fiable, confiable, autorizada y autenticada. (Sysmatec, 2016)

2.2. LENGUAJE DE PROGRAMACION JAVA



Ilustración 3 Logo de Java
Fuente: (Java, Tecnología Java, 2016)

Java es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems. Se popularizó a partir del lanzamiento de su primera versión comercial de amplia difusión, la JDK 1.0 en 1996. Java es rápido, seguro y fiable además en la ilustración 3 observamos su logo. (Oracle, 2016)

2.2.1. Características

Según (Sun, 2016), se puede leer que Java es: "Un lenguaje simple, orientado al objeto, distribuido, interpretado, sólido, seguro, de arquitectura neutral, portable, de alto desempeño, de multihilos y dinámico", también en la tabla 1 observamos las siguientes características de java.

Tabla 1: Características del lenguaje Java

Simple	Basado en el lenguaje C++
Orientado al objeto	Java da buen soporte a las técnicas de desarrollo Programación Orientada a Objetos (POO) y a la reutilización de componentes de software.
Distribuido	Java se ha diseñado para trabajar en ambiente de redes y contienen una gran biblioteca de clases para la utilización del protocolo TCP/IP, incluyendo HTTP y FTP. El código Java se puede manipular a través de recursos URL.
Interpretado	El compilador Java traduce cada fichero fuente de clases a código de bytes (Bytecode), que puede ser interpretado por todas las máquinas que den soporte a un visualizador que funcione con Java.
Sólido	El código Java no se quiebra fácilmente ante errores de programación.

Seguro	Java evita la manipulación de código. Actualmente se está trabajando en la encriptación del código.
De arquitectura neutral	El compilador crea códigos de byte (Bytecode) que se envía al visualizador solicitado y se interpreta en la máquina que posee un intérprete de Java o dispone de un visualizador que funciona con Java.
Portable	Al ser de arquitectura neutral es altamente portable.
Alto Desempeño	El compilador Java suele ofrecer la posibilidad de compilar Bytecode en código máquina de determinadas plataformas.
Multihilos	Java puede aplicarse a la realización de aplicaciones en las que ocurra más de una cosa a la vez.
Dinámico	Java utiliza un sistema de interfaces que permite aligerar esta dependencia. Como resultado, los programas Java pueden permitir nuevos métodos y variables en un objeto de biblioteca sin afectar a los objetos dependientes.

Fuente: Saula, (Java, Tecnología Java, 2016)

2.2.2. Beneficios de Java

Java se ha convertido en un valor impagable para los desarrolladores, ya que les permite:

- Escribir software en una plataforma y ejecutarla virtualmente en otra
- Crear programas que se puedan ejecutar en un explorador y acceder a servicios Web disponibles
- Desarrollar aplicaciones de servidor para foros en línea, almacenes, encuestas, procesamiento de formularios HTML y mucho más
- Combinar aplicaciones o servicios que utilizan el lenguaje Java para crear aplicaciones o servicios con un gran nivel de personalización
- Escribir aplicaciones potentes y eficaces para teléfonos móviles, procesadores remotos, microcontroladores, módulos inalámbricos, sensores, gateways, productos de consumo y prácticamente cualquier otro dispositivo electrónico. (Java, Conozca más sobre la tecnología Java, 2016)

2.2.3. Tecnologías Java

De acuerdo a la ilustración 4 las tres tecnologías de la plataforma Java facilita el proceso para que desarrolladores de software, prestadores de servicios y fabricantes de dispositivos enfoquen mercados específicos: (Canut, 2012)

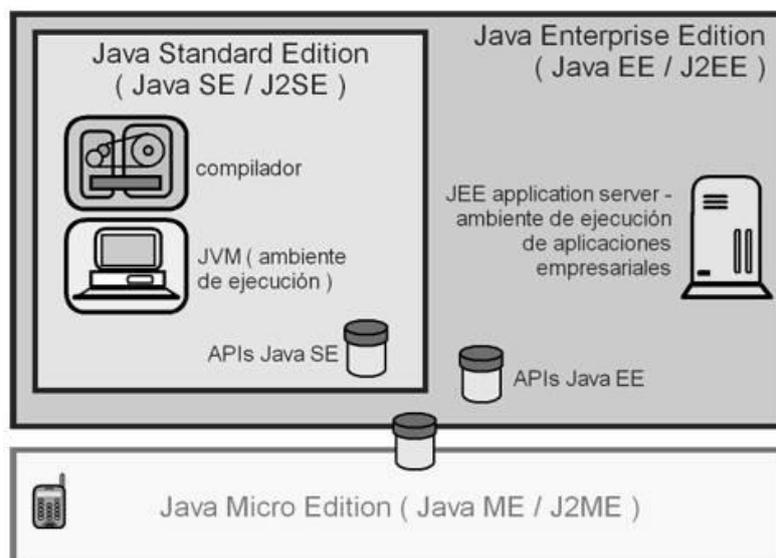


Ilustración 4: Tecnologías JAVA
Fuente: Ganell, (Canut, 2012)

2.2.3.1. Java J2SE (Plataforma Java, Standard Edition)

J2SE permite desarrollar y desplegar aplicaciones Java en desktops y servidores, como también en entornos incorporados y en tiempo real. J2SE incluye clases que soportan el desarrollo de servicios web Java y proporciona la base para la Plataforma Java, Enterprise Edition (Java EE). Java SE 6 ("Mustang") es la versión actual de la plataforma Java SE. Muchos desarrolladores Java usan Java SE 5, también conocido como Java 5.0 o "Tiger". (Glance, 2015)

2.2.3.2. Java J2ME (Plataforma Java, Micro Edition)

J2ME proporciona un entorno para aplicaciones que operan en una gama amplia de dispositivos móviles e incorporados, como teléfonos móviles, PDAs e impresoras. La plataforma Java ME incluye interfaces de usuario flexibles, un modelo robusto de seguridad, una gama amplia de protocolos de red incorporados y amplio soporte para aplicaciones conectadas en red y offline que pueden ser descargadas dinámicamente. Las aplicaciones basadas en las especificaciones de Java ME se escriben una única vez para una gama amplia de dispositivos, pero aprovechan las posibilidades nativas de cada dispositivo. (Java Platform, 2015)

2.2.3.3. Java J2EE - JEE (Plataforma Java, Enterprise Edition)

J2EE es una plataforma de programación para desarrollar y ejecutar software de aplicaciones con arquitecturas de múltiples capas distribuidas. Se basa ampliamente en componentes modulares de software ejecutándose sobre un servidor de aplicaciones. Basado en Java SE, J2EE proporciona APIs (Application Programming Interface) de comunicaciones, servicios web, modelos de componentes y gestión para implementar aplicaciones Web 2.0 de nivel empresarial.

Características

- Plataforma abierta y estándar
- Modelo de aplicaciones distribuidas
- Componente modulares y estandarizados

Elementos

- Componentes (cliente, web y negocio)
- Contenedores (aplicaciones, applets, web)
- Servicios (configurables y no configurables)
- Servidores de aplicaciones
- Servicios Web

Para el desarrollo de Servicios Web se han ideado varias tecnologías que facilitan el desarrollo de los mismos como Axis 2, Metro 2.0. (Java J2EE, 2015)

2.3. TECNOLOGÍAS PARA SERVICIOS WEB

2.3.1. AXIS 2



Ilustración 5: Logo de Axis 2

Fuente: Apache software foundation, (Apache, 2016)

Mediante la ilustración 5 Apache Axis2™ es un motor de servicios web / SOAP / WSDL, la sucesora de la ampliamente utilizada Apache Axis pila SOAP. Hay dos implementaciones del motor de servicios web Apache Axis2 - Apache Axis2 / Java y Apache Axis2 / C.

Apache Axis 2 es una tecnología basada en java, tanto del cliente como del servidor de la ecuación de servicios Web. Diseñado para aprovechar las lecciones aprendidas de Apache Axis 1.0, Apache Axis2 proporciona un modelo de objetos completo y una arquitectura modular que hace fácil agregar funcionalidad y soporte para los nuevos servicios de la Web relacionados con las especificaciones y recomendaciones. (Apache, 2016)

2.3.1.1. Principales características Axis 2

Axis2 viene con muchas nuevas características, mejoras e implementaciones de la especificación de la industria. Las características principales que se ofrecen son los siguientes:

- **Velocidad.** - Axis2 utiliza su propio modelo de objetos y StAX (API para XML Streaming) analizar para lograr velocidad significativamente mayor que las versiones anteriores de Apache Axis.
- **Bajo los pies de impresión de memoria.** - Axis2 fue diseñado de impresión de bajo mantenimiento de la planta del pie hasta la memoria en mente.
- **AXIOM** - Axis2 viene con su propio peso ligero modelo de objetos, AXIOM, para el procesamiento de mensajes que es extensible, gran rendimiento y es promotora conveniente. (Apache, 2016)
- **Despliegue en caliente.** - Axis2 está equipado con la capacidad de despliegue de servicios Web y manipuladores mientras el sistema está en funcionamiento. En otras

palabras, los nuevos servicios pueden ser añadidos al sistema sin tener que apagar el servidor. Simplemente coloque el archivo de servicio web requerido en el directorio de servicios en el repositorio, y el modelo de implementación desplegará automáticamente el servicio y hacer que esté disponible para su uso.

- **Servicios web asíncronos.** - Axis2 ahora soporta servicios web asíncronos y asíncrona invocación de servicios Web que utilizan los clientes no bloqueante y transportes.
- **MEP Soporte.** - Axis2 ahora es muy útil con la flexibilidad necesaria para soportar los patrones de mensajes de Exchange (MEPS) con soporte incorporado para que los eurodiputados básicos definidos en WSDL 2.0.
- **Flexibilidad.** - La arquitectura Axis2 le da al desarrollador completa libertad para insertar las extensiones en el motor de procesamiento de cabecera personalizada, gestión del sistema, y cualquier otra cosa que se pueda imaginar.
- **Estabilidad.** - Axis2 define un conjunto de interfaces publicadas que cambian de forma relativamente lenta en comparación con el resto del Eje.
- **Componente orientado Despliegue.** - Se puede definir fácilmente reutilizables redes de los manipuladores para implementar patrones comunes de procesamiento para sus aplicaciones, o para distribuir a los socios.
- **Marco de Transporte.** - Tenemos una abstracción limpia y sencilla para la integración y el uso de Transportes (es decir, los remitentes y los oyentes de SOAP a través de varios protocolos como SMTP, FTP, middleware orientado a mensajes, etc.) y el núcleo del motor es completamente transporte- independiente.
- **Apoyo WSDL.** - Axis2 apoya la Web Service Description Language, versión 1.1 y 2.0 , que le permite crear fácilmente talones de acceder a servicios remotos, y también para exportar automáticamente descripciones legibles por máquinas de sus servicios desplegados desde Axis2. (Axis2, 2016)

- **Complementos.** - Varios servicios Web especificaciones se han incorporado, entre otras WSS4J para la seguridad (Apache Rampart), Sandesha para la mensajería fiable, Kandula que es una encapsulación de WS-Coordination y WS-AtomicTransaction y WS-BusinessActivity.
- **Composición y extensibilidad.** - Módulos y fases mejorar el apoyo a la componibilidad y extensibilidad. Módulos soportan componibilidad y también pueden apoyar a las nuevas especificaciones de WS * de una manera sencilla y limpia. Sin embargo, ellos no son caliente de despliegue a medida que cambian el comportamiento global del sistema. (Apache, 2016)

2.3.1.2. Modelo de programación

- Mejorado, la API cliente XML centrada incluido el pleno WSDL y apoyo a las políticas.
- Apoyo a los servicios de estilo JAXWS y clientes.
- Apoyo a los servicios POJO y primavera y clientes.
- Soporte para cualquier patrón de intercambio de mensajes.
- Llamadas síncronas y asíncronas.
- Archivado modelo de implementación de servicios de apoyo encapsulación de servicio completo con soporte para versiones.
- Archivado modelo de implementación del módulo de soporte extensibilidad controlada con el apoyo de versiones.
- El despliegue en caliente.
- WS-Policy extensiones de generación de código impulsado.
- Servicio flexible modelo de ciclo de vida.
- Apoyo automático a la invocación de estilo POX (REST) de los servicios.
- Apoyo para la consulta de WSDL de un servicio (utilizando WSDL), esquema (usando Xsd) y políticas (usando Política).
- WSDL 2.0
- Implementadores personalizados.
- Serialización binaria (conjunto de información rápida).
- El apoyo JSON.
- Proveedor de soporte EJB. (Apache, 2016)

2.3.1.3. Versiones de Axis

- Versión Axis 1
- Versión Axis 2

2.3.1.4. Implementaciones de Servicios web Soportadas en Axis 2

- SOAP 1.1 y 1.2
- Mecanismo de transmisión de mensajes Optimization (MTOM), XML Empaquetado Optimizado (XOP) y SOAP con archivos adjuntos
- WSDL 1.1, que incluye tanto SOAP y enlaces HTTP
- WS-Addressing (presentación y última)
- WS-Policy
- 1.1 SAAJ (Apache, 2016)

2.3.1.5. Servidores Soportados

Axis 2 proporciona integración con los siguientes servidores:

- Apache Tomcat 4.1 - 6.0 - 8.0
- JBoss 3.2 - 4.2.x

2.3.1.6. Navegadores Soportados

- Internet Explorer
- Firefox
- Chrome
- Opera

2.3.1.7. Ventajas

- Enviar mensajes SOAP.
- Recibir y procesar mensajes SOAP.
- Crear un servicio Web y exponerlo a partir de una básica clase Java, en inglés referenciada como POJO.
- Crear clases de implementación del lado del cliente y del lado del servidor empleando WSDL.
- Recibir fácilmente el WSDL por un servicio.

- Enviar y recibir mensajes SOAP con adjuntos.
- Crear o utilizar servicios que aprovechen las ventajas que poseen las recomendaciones de WS-Security, WS-ReliableMessaging, WS-Addressing, WS-Coordination, and WS-Atomic Transaction. (S.R.L, 2012)

2.3.1.8. Desventajas

- El WSDL que genera Axis2 es bastante complejo (hay que tenerlo en cuenta a la hora de que alguien tenga que crear una aplicación cliente que se conecte a nuestro WS a partir del WSDL que nosotros le proporcionemos). (Axis2, 2016)

2.3.2. METRO 2.0



Ilustración 6: Logo Metro 2.0
Fuente: Glassfish, (Metro 2.0, 2012)

Metro es un framework de servicios web, de alto rendimiento, extensible y fácil de usar. Se trata de una ventanilla única para todas sus necesidades de servicio web, desde el servicio web más sencillo hasta un servicio web fiable, que implica incluso servicios .Net. El framework de servicios web Metro es una parte de la comunidad GlassFish, pero puede también ser utilizado fuera de GlassFish, además en la ilustración 6 apreciamos el logo de Metro 2.0 (java.net, 2009).

Los componentes de metro incluyen WSIT, JAXB y JAX-WS, como pilares para la creación de servicios y clientes interoperables.

2.3.2.1. Principales características Metro 2.0

2.3.2.1.1. Web Service Interoperability Technologies (WSIT)

Durante varios años Sun actualmente Oracle ha trabajado estrechamente con Microsoft para garantizar la interoperabilidad de las tecnologías de servicios web de la empresa, tales como la seguridad, la mensajería fiable y transacciones atómicas. La parte de Metro se conoce como WSIT (Web Service Interoperability Technologies). Subsistema WSIT Metro es una implementación de una serie de especificaciones de servicios web abiertas para apoyar las funciones de empresa. Además de la seguridad, la mensajería fiable y transacciones atómicas, Metro incluye un programa previo y la tecnología de configuración. Ilustración 7, se puede apreciar las características de metro. (WSIT, 2013)

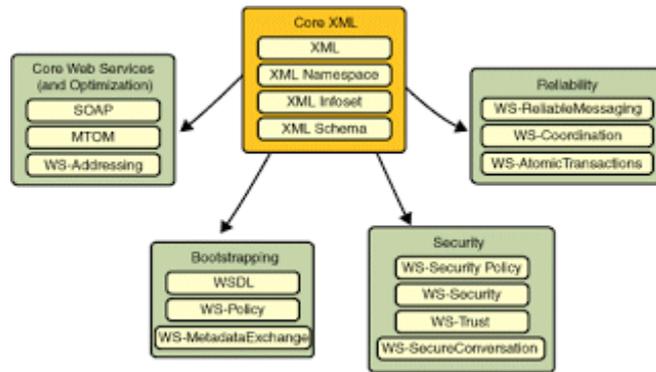


Ilustración 7: Características de los servicios web metro 2.0
Fuente: Metro, (Working, 2016)

Comenzando por la compatibilidad con XML integrada en la plataforma Java, Metro utiliza o amplía funciones existentes y añade compatibilidad adicional para servicios web interoperables. Las funciones son las siguientes:

- Secuencia de arranque y configuración.
- Tecnología de optimización de mensajes.
- Tecnología de mensajería fiable.
- Tecnología de Seguridad.

2.3.2.1.2. Java Architecture XML Binding (JAXB)

Forma parte Metro, como proyecto para JAXB proporcionado implementación de referencia es desarrollado por JCP para JAXB.

JAXB es un acrónimo de la arquitectura Java para XML Binding. JAXB proporciona API para acceder y procesar un documento XML. JAXB es intermediario entre instancias de documentos XML y Java. Paso importante en el uso de JAXB es la creación de los POJOs Java. Esquema XML puede ser utilizado por un compilador de unión JAXB para generar clases Java. Esas clases java generados coinciden con el esquema XML y serán cargados en tiempo de ejecución de la aplicación. Si no se tiene el esquema XML, entonces podemos se puede codificar manualmente las clases POJO y anotar con anotaciones JAXB. Se puede crear instancias de esas clases luego leen o escriben en XML. XJC - com.sun.tools.xjc.XJCTask es el compilador de enlace proporcionado por el Metro. Las clases JAXB generados a partir de un esquema XML son POJOs con las anotaciones JAXB estándar. Ellos están destinados a ser compatible con otras implementaciones JAXB. (Binding, 2013)

2.3.2.2. Versiones de Metro

- Versión GlassFish Metro 1.5
- Versión GlassFish Metro 2.0
- Versión GlassFish Metro 2.2
- Versión GlassFish Metro 2.3

2.3.2.3. Implementaciones de Servicios web Soportadas en Metro 2.0

La especificación JSR 224 – Java API for XML-Based Web Services establece las bases para trabajar con servicios web que utilizan XML para comunicarse. Como pasa con todas las especificaciones puede haber varias implementaciones, pero siempre hay una que se elige como la implementación de referencia (RI – Reference Implementation).

En el caso de ésta especificación la implementación de referencia la aporta Metro. La RI que vamos a utilizar es la versión JAX-WS 2.0, por que se acerca mucho a la versión 2.1.5 que es la que incluye de saque el servidor de aplicaciones Weblogic 11g (parche 10.3.6), que es el que estamos utilizando para realizar estas pruebas.

En teoría, se podría utilizar cualquier versión de la implementación de referencia teniendo en cuenta que si difiere de la que incorpora weblogic habrá que incluir las librerías correctas en la aplicación y tal vez marcar la opción prefer-web-inf-classes en el descriptor weblogic.xml de nuestra aplicación. (Tecnicomio, 2013)

2.3.2.4. Servidores Soportados

Metro 2.0 viene incluido con numerosos servidores de aplicaciones tales como:

- Glassfish
- Sun Java System Application Server Platform Edition 9.x
- Oracle WebLogic Server
- JBoss (solo versión 5.x y siguientes)
- TmaxSoft JEUS 6.x

La implantación de referencia JAXB desarrollada para Metro se usa en casi cualquier framework de servicio web Java (Apache Axis2, Codehaus Xfire, Apache CFX) y servidores de aplicaciones. (Dennis Sosnoski, 2012)

2.3.2.5. Navegadores Soportados

- Internet Explorer
- Firefox
- Chrome
- Opera

2.3.2.6. Ventajas

- Una de las ventajas más importantes de Metro 2.0 es que está incorporado en el servidor Glassfish del IDE de desarrollo Netbeans.
- Es más fácil de configurar utilizando conjuntos de políticas.
- Admite manejadores JAX-WS.
- Evita la necesidad de instalar un repositorio SDO.
- Ofrece la ventaja de proporcionar, en forma general, resultados más consistentes que los de las pruebas ejecutadas en una red. (Sosnoski, 2012)

2.3.2.7. Desventajas

- La configuración de WS-Security puede ser compleja y que a veces añade mucho volumen a los mensajes que se intercambian.
- La desventaja más significativa es que combina la sobrecarga de procesamiento del cliente y del servidor, lo que imposibilita que se evalúen separadamente. (Sosnoski, 2012)

2.3.3. VULNERABILIDADES EN SERVICIOS WEB

Según (Commons, 2014) los primeros ataques a la red aprovecharon las vulnerabilidades relacionadas con la implementación de conjuntos de protocolos TCP/IP. Al corregirlas gradualmente, los ataques se dirigieron a las capas de aplicaciones y a la Web en particular, ya que la mayoría de las empresas abrieron sus sistemas de firewall al tráfico en Internet.

A continuación, se detalla las definiciones de los indicadores con los que se realizó el respectivo análisis de las tecnologías Axis 2 y metro 2.0.

2.3.3.1. Confidencialidad

- Cruzar sitio en secuencia de comandos (Cross Site Scripting)

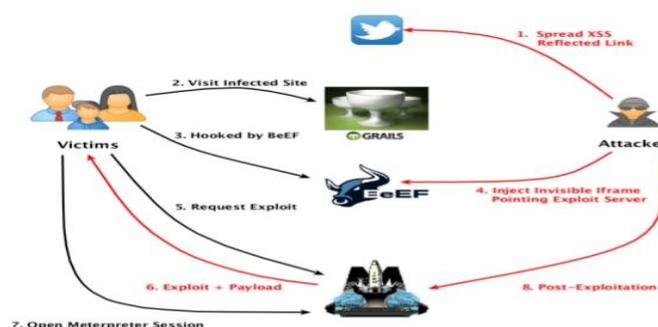


Ilustración 8: Ataque de Cross Site Scripting

Fuente: SOAPUI, (Scripting, 2015)

El Cross-Site-Scripting es una vulnerabilidad que aprovecha la falta de mecanismos de filtrado en los campos de entrada y permiten el ingreso y envío de datos sin validación alguna, aceptando el envío de scripts completos, pudiendo generar secuencias de comandos maliciosas que impacten directamente en el sitio o en el equipo de un usuario, en la ilustración 8 observamos cómo se realiza el ataque. (Ramos, 2014)

- Inyección SQL (SQL Injection)

SQL Injection

User Name:

Password:

```
select * from users where user_id='shivam' and password='owasp';
```

User Name:

Password:

```
select * from users where user_id=' ' OR 1=1; /*and password='owasp'; */
```

Ilustración 9: Ataque de SQL Injection

Fuente: SOAPUI, (Injection S. , 2015)

La inyección de código SQL es un ataque en el cual se inserta código malicioso en las cadenas que posteriormente se pasan a una instancia de SQL Server para su análisis y ejecución. Todos los procedimientos que generan instrucciones SQL deben revisarse en busca de vulnerabilidades de inyección de código, ya que SQL Server ejecutará todas las consultas recibidas que sean válidas desde el punto de vista sintáctico. Un atacante cualificado y con determinación puede manipular incluso los datos con parámetros, también la ilustración 9 observamos un ejemplo de cómo se realiza el ataque (TechNet, 2016)

- **Inyección Xpath (XPath Injection)**



Ilustración 10: Ataque de XPath Injection
Fuente: SOAPUI, (Injection X. , 2015)

Es un lenguaje que permite construir expresiones que recorren y procesan un documento XML. La idea es parecida a las expresiones regulares para seleccionar partes de un texto sin atributos (plain text). XPath permite buscar y seleccionar teniendo en cuenta la estructura jerárquica del XML. Además en la ilustración 10 observamos un claro ejemplo de cómo se realiza el ataque (REBERTO, 2012)

- **Tipos Inválidos (Invalid Types)**

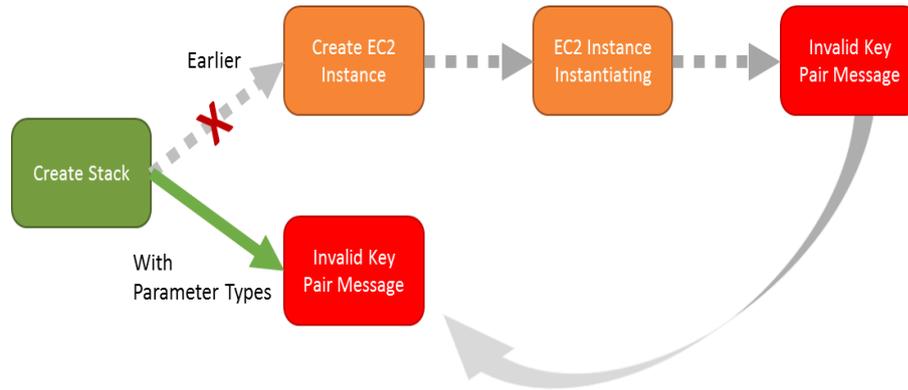


Ilustración 13: Ataque de Invalid Types
Fuente: SOAPUI, (Types, 2015)

Al implementar servicios web que es fácil olvidar la manipulación de valores que no esperas, especialmente si la entrada está restringida ya en el lado del cliente. Los tipos válidos de análisis se emplean para ayudar a usted para asegurarse de que el servidor se encarga de este tipo de situaciones con gracia. Además en la ilustración 13 observamos cómo se realiza el ataque. (SmartBear, 2015)

2.3.3.3. Disponibilidad

- **Error de denegación de servicio.**



Ilustración 14: Error de denegación de servicios
Fuente: REALNET, (API, 2015)

Un ataque de denegación de servicio ó DDoS, se define principalmente como un tipo de ataque informático especialmente dirigido a redes de computadoras. Tiene como objetivo lograr que un servicio específico o recurso de la red, quede completamente inaccesible a los usuarios legítimos de la red. Conjuntamente en la ilustración 14 observamos un claro ejemplo de cómo se realiza el ataque. (Culturacion, 2014)

CAPÍTULO III

ANÁLISIS DE LAS TECNOLOGÍAS DE SERVICIOS WEB AXIS 2 Y METRO 2.0.

3.1. Diseño de la investigación

El presente trabajo se consideró como una investigación de tipo documental bibliográfica ya que es basado en la búsqueda, recuperación, análisis, crítica e interpretación de información a través de fuentes secundarias. Según (Casas, 2015) los tres principios de la seguridad son Confidencialidad (C), Integridad (I) y Disponibilidad (D), los mismos que se utilizó en el desarrollo de servicios web seguros para comprobar los siguientes indicadores.

- **Confidencialidad**

Esta propiedad asegura que el contenido incluido en los mensajes que se intercambien entre servidor y cliente se mantenga como información confidencial.

- **Integridad**

Esta propiedad garantiza que la información que se ha recibido, sea la misma que se envió desde el cliente.

- **Disponibilidad**

Esta propiedad garantiza que el servicio web tenga una alta disponibilidad.

3.2. Tipo de investigación

Los tipos de investigación que se utilizaron son: la investigación descriptiva y aplicada, porque se basa en estudios previos y procesos aplicados en otros entornos similares al de este trabajo, el objetivo es determinar cuál de las dos tecnologías de servicio web ofrece un mejor nivel de seguridad.

3.3. Métodos

En la presente investigación se aplicó el método deductivo, ya que se estudió las variables de investigación desde lo general hasta llegar a deducir conclusiones óptimas, se consideraron sucesos significativos para el objeto de estudio.

3.4 Técnicas de investigación

- Observaciones
- Entrevistas

3.5. Población y Muestra

La población objeto está conformada por un total de 633 scripts los cuales se lograron generar para el análisis del test de seguridad, al especificar el tamaño de muestra nosotros procuramos que dicha información sea válida y confiable, por lo tanto, el tamaño de la muestra está limitado para el objeto de estudio, el número de la muestra será de 633 scripts los cuales son significativos para el estudio de la investigación.

Para la cual utilizaremos la herramienta que tenga mayor valoración de acuerdo a la Ilustración número 28 Validación de Instrumentos, y la herramienta debe proporcionar una configuración dentro de los siguientes parámetros que se detallan a continuación:

- Sql Injection
- XPath Injection
- Cross Site Scripting
- Malformed XML
- XML Bomb
- Invalid Types

3.6. Instrumentos para el testeo de seguridad

Para el testeo de la seguridad de los servicios web se seleccionará una de las siguientes herramientas de escaneo de vulnerabilidad de acuerdo a su máxima valoración:

3.6.1. SoapUI



Ilustración 15: Logo de SoapUI
Fuente: SOAPUI, (SMART, 2016)

SoapUI es una aplicación muy versátil que nos permite probar, simular y generar código de servicios web de forma ágil, partiendo del contrato de los mismos en formato WSDL y con vínculo SOAP sobre HTTP. SoapUI tiene dos distribuciones: soapUI freeware (GNU LGPL y opensource java) y soapUIPro (comercial), en versión de escritorio, online y plugin para varios IDE. Conjuntamente observamos el logo en la ilustración 15. (Puebla, 2012)

3.6.2. WebInject



Ilustración 16: Logo de WebInject
Fuente: WEBINJECTED, (technology, 2016)

WebInject es una herramienta de prueba extremadamente ligera, que puede realizar pruebas automatizadas de servicios web y aplicaciones web, WebInject puede probar el servicio Web XML / SOAP. Es la primera herramienta escrito en Perl, aunque su autor proporciona una sencilla interfaz de usuario Perl / Tk, al menos simplificar la aplicación de la prueba (para algunas personas que no quieren gastar demasiado tiempo en la línea de comandos).

Si usted no está familiarizado con Perl, no tenga miedo, ya que WebInject trabajo sin ningún código Perl. Además, en la ilustración 16 observamos su logo. (muyiyangfeng, 2014)

3.6.3. Testmaker



Ilustración 17: Logo de Testmaker PushtoTest
Fuente: PUSHTOTEST, (Security, 2015)

TestMaker es una herramienta de prueba que se realiza mediante un script llamado "agente de ensayo (agentes de prueba), para leer definición WSDL y automáticamente crear la estructura básica de un agente de prueba para proporcionar un" agente Wizard (Asistente de Agente). Debe tenerse en cuenta que: TestMaker no sólo los servicios web de prueba; También se puede utilizar para probar aplicaciones Web. TestMaker también es una herramienta de monitorización de red que puede supervisar el tráfico HTTP entre el objetivo de las aplicaciones Web navegador y generar casos de prueba del proceso interactivo. Adjuntado su logo en la ilustración 17. (muyiyangfeng, 2014)

3.7. Validación de instrumentos

Los instrumentos que se muestra en la ilustración 18 son open source ya que son utilizados a diario para testear la seguridad en los servicios web de acuerdo con (muyiyangfeng, 2014) y (InfoWorld, 2012), la opción de muchos expertos en relación a las mejores herramientas para pruebas sobre seguridad en servicios web ha ponderado varios aspectos que son: Características, documentación, etc.; de los softwares comparados y el que tiene mejor puntuación en todos los aspectos es SoapUI en la versión 1.6.

InfoWorld Scorecard	Features (20.0%)	Value (10.0%)	Portability (15.0%)	Scalability (20.0%)	Documentation (20.0%)	Ease of use (15.0%)	Overall Score (100%)
Eviware SoapUI 1.6	9.0	9.0	9.0	9.0	9.0	9.0	9.0 ★★★★★
PushToTest TestMaker	9.0	9.0	9.0	9.0	9.0	7.0	8.7 ★★★★☆
WebInject	8.0	9.0	9.0	8.0	8.0	8.0	8.3 ★★★★☆

Ilustración 18: Validación de Instrumentos para testeo de seguridad
Fuente: INFOWORLD, (Grehan, 2012)

3.8. Selección de la herramienta

Mediante la ilustración 28 se seleccionó e instaló en un ordenador la herramienta SoapUI para utilizarlos con los scripts para el testeo de seguridad en el servicio web creado con la tecnología Axis 2 y Metro 2.0, para la cual ambas tecnologías contarán con el mismo hardware y software de las siguientes características:

3.8.1. Hardware

En la tabla 2 se indica el hardware que se utilizará para realizar las pruebas de seguridad del Servicio Web con las dos tecnologías de servicios web.

Tabla 2: Hardware utilizado para realizar pruebas de seguridad del SW

Cantidad	Equipo	Marca	Modelo	Especificaciones
1	Ordenador	HP	DV4	Procesador Core 2 Duo 2.2GHz
				4GB Memoria Ram
				320GB Disco Duro

Fuente: Kléber Bustán/Jorge Álvarez

3.8.2. Software

En la tabla 3 se indica el software que se utilizará para realizar las diferentes pruebas de seguridad en el Servicio Web con las 2 tecnologías de servicios web.

Tabla 3: Software utilizado para realizar pruebas de seguridad en SW

Software	Nombre	Descripción
	WINDOWS 10	Sistema operativo base que se utilizó para utilizar las herramientas de desarrollo.
	SOAPUI	Aplicación para test de seguridad servicios web.
	APACHE TOMCAT 8.0.35	Servidor de aplicaciones Java
	POSTGRESQL 9.4	Sus características técnicas la hacen una de las bases de datos más potentes y robustos del mercado.
	NETBEANS 8.1	NetBeans IDE le permite desarrollar rápida y fácilmente aplicaciones de escritorio, móviles y aplicaciones web, así como aplicaciones HTML5 con HTML, JavaScript y CSS.

Fuente: Kléber Bustán/Jorge Álvarez

3.9. Escenario de prueba

Se desarrolló dos prototipos de aplicaciones de servicios web, la primera utilizando la tecnología de servicio web Axis 2 y la segunda utilizando la tecnología de servicio web Metro 2.0, como se muestra en la ilustración 20 para lo cual se construyó un Web Método que realiza consultas de los registros de la tabla facultad dado su código.

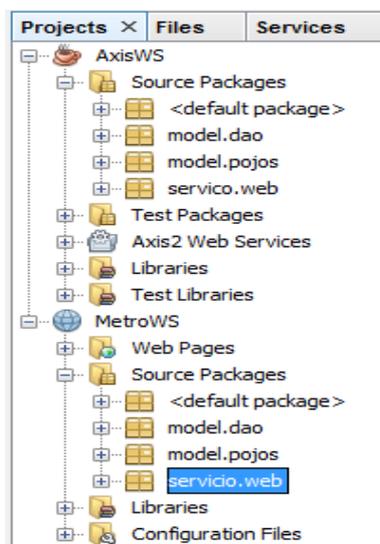


Ilustración 19: Prototipos Axis 2 y Metro 2.0 creados para medir la seguridad en SW
Fuente: Kléber Bustán/Jorge Álvarez

3.10. Proceso de prueba

Como se muestra en la ilustración 19 las dos aplicaciones de servicio web, se aplicó la seguridad de UsernameToken presente en las dos tecnologías; Se tomó la tabla facultad la cual tiene la siguiente información: id, nombre, descripción: La seguridad UsernameToken se utiliza para identificar al solicitante mediante un nombre de usuario y contraseña el cual es necesario para autenticar esa entidad frente al proveedor de servicio web.

En la herramienta SoapUI, se procede a dividir en los siguientes parámetros para el proceso del test seguridad como se detallan a continuación:

Confidencialidad

- Cross Site Scripting
- SQL Injection
- XPath Injection

Integridad

- Malformed XML
- XML Bomb
- Invalid Types

Disponibilidad

- Errores de denegación de servicio.

Empleando la herramienta SoapUI se procedió a evaluar cada prototipo de prueba, en los parámetros anteriormente mencionados, seguidamente se muestra los resultados en tiempo real de cada prototipo.

3.11. Test del prototipo de la tecnología de servicio web Axis2

Se utilizó la herramienta SoapUI para realizar el test de seguridad en servicios web con la tecnología Axis como se muestra en la ilustración 20:

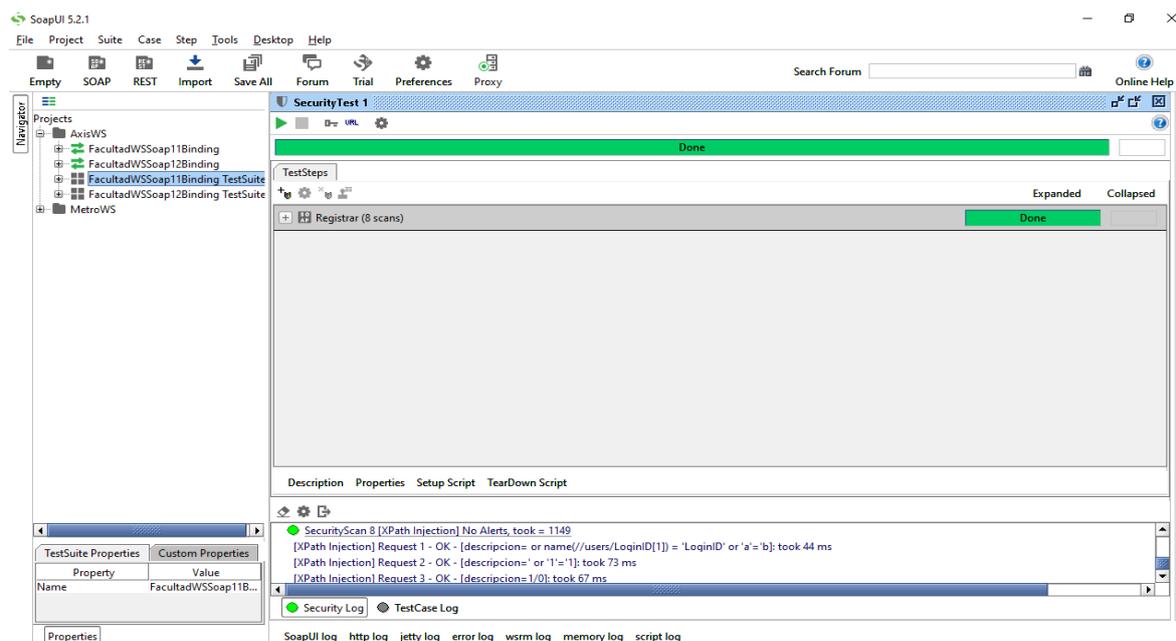


Ilustración 20: Test de Prototipo Axis2
Fuente: Kléber Bustán/Jorge Álvarez

3.11.1. Parámetro de Confidencialidad

A continuación, se detalló los indicadores del parámetro confidencialidad.

3.11.1.1. Cross Site Scripting:

Con el script Cross Site Scripting en la ilustración 21 no se encontró errores de seguridad en la tecnología Axis 2.



Ilustración 21: Cross Site Scripting aplicado en Axis 2
Fuente: Kléber Bustán/Jorge Álvarez

3.11.1.2. SQL Injection:

Mediante el script Sql Injection en la ilustración 22 no se encontró errores de seguridad en la tecnología Axis 2.



Ilustración 22: SQL Injection aplicado en Axis 2
Fuente: Kléber Bustán/Jorge Álvarez

3.11.1.3. XPath Injection

Con el script XPath Injection en la ilustración 23 no se encontró errores de seguridad en la tecnología Axis 2.



Ilustración 23: Xpath Injection aplicado en Axis 2
Fuente: Kléber Bustán/Jorge Álvarez

3.11.2. Parámetro de Integridad

A continuación, se procedió a detallar todos los indicadores del parámetro de integridad.

3.11.2.1. Malformed XML

En el script Malformed XML en la ilustración 24 no se encontró errores de seguridad en la tecnología Axis 2.



Ilustración 24: Malformed XML aplicado en Axis 2
Fuente: Kléber Bustán/Jorge Álvarez

3.11.2.2. XML Bomb

Con el script XML Bomb en la ilustración 25 no se encontró errores de seguridad en la tecnología Axis 2.

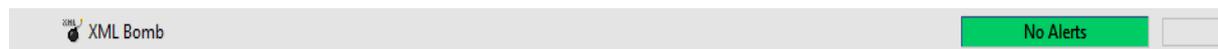


Ilustración 25: XMLBomb aplicado en Axis 2
Fuente: Kléber Bustán/Jorge Álvarez

3.11.2.3. Invalid Types

Con el script Invalid Types en la ilustración 26 no se encontró errores de seguridad en la tecnología Axis 2.



Ilustración 26: Invalid Types aplicado en Axis 2
Fuente: Kléber Bustán/Jorge Álvarez

3.11.3. Parámetro de Disponibilidad

Seguidamente se detalló el parámetro de disponibilidad.

3.11.3.1. Denegación de servicios

En la ilustración 27 se indica el resultado del número de errores de denegación de servicio en la tecnología Axis2.

	Min	Max	Median	Last	Count	TPS	Err	Err %
Registrar	2	12358	1006	2	1631	0,00	1	0,06

Ilustración 27: Errores de denegación de servicio en Axis 2
Fuente: Kléber Bustán/Jorge Álvarez

3.12. Test del prototipo de la tecnología de servicio web Metro 2.0

Se utilizó la herramienta SoapUI para el testeo de seguridad en servicios web en la tecnología Metro 2.0 como muestra en la ilustración 28.

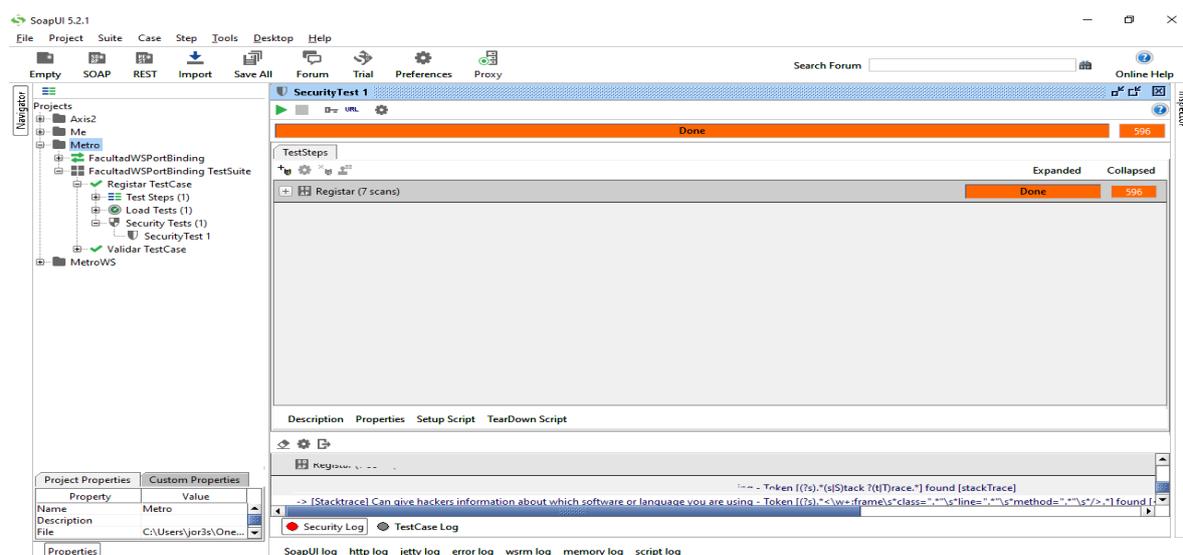


Ilustración 28: Test Prototipo Metro 2.0
Fuente: Kléber Bustán/Jorge Álvarez

3.12.1. Parámetro de Confidencialidad

A continuación, se detalla cada uno de los indicadores dentro del parámetro de confidencialidad aplicado a la tecnología de servicios web Metro 2.0.

3.12.1.1. Cross Site Scripting

Mediante el script Cross Site Scripting en la ilustración 47 se obtuvo 368 errores de seguridad en la tecnología Metro 2.0.



Ilustración 47: Cross Site Scripting aplicado en Metro 2.0
Fuente: Kléber Bustán/Jorge Álvarez

3.12.1.2. SQL Injection

Con el script SQL Injection en la ilustración 48 se obtuvo 56 errores de seguridad en la tecnología Metro 2.0.



Ilustración 48: SQL Injection aplicado en Metro 2.0
Fuente: Kléber Bustán/Jorge Álvarez

3.12.1.3. XPath Injection

Con el script XPath Injection en la ilustración 49 se obtuvo 40 errores de seguridad en la tecnología Metro 2.0.



Ilustración 49: XPath Injection aplicado en Metro 2.0
Fuente: Kléber Bustán/Jorge Álvarez

3.12.2. Parámetro de Integridad

3.12.2.1. Malformed XML

En el script Malformed XML en la ilustración 50 se obtuvo 36 errores de seguridad en la tecnología Metro 2.0.



Ilustración 50: Malformed XML aplicado en Metro 2.0
Fuente: Kléber Bustán/Jorge Álvarez

3.12.2.2. XML Bomb

En el script XML Bomb en la ilustración 51 no se encontraron errores de seguridad en la tecnología Metro 2.0.



Ilustración 51: XML Bomb aplicado en Metro 2.0
Fuente: Kléber Bustán/Jorge Álvarez

3.12.2.3. Invalid Types

Con el script de Invalid Types en la ilustración 52 se obtuvo 96 errores de seguridad en la tecnología Metro 2.0 como se muestra en la siguiente ilustración a continuación:



Ilustración 52: Invalid Types aplicado en Metro 2.0
Fuente: Kléber Bustán/Jorge Álvarez

3.12.3. Parámetro de Disponibilidad

Procedemos a detallar el indicador dentro del parámetro de disponibilidad.

3.12.3.1. Denegación de servicios

A continuación, se detalla el número de errores de denegación de servicio en la tecnología Metro 2.0 como se visualiza en la ilustración 53.

	Min	Max	Median	Last	Count	TPS	Err	Err %
Registrar	10	3236	83	11	2283	0,00	0	0,00

Ilustración 53: Errores de denegación de servicios en Metro 2.0
Fuente: Kléber Bustán/Jorge Álvarez

3.13. Análisis de resultados

Se obtuvo los siguientes resultados luego de las pruebas realizadas con cada una de las tecnologías de servicio web.

3.13.1. Parámetro de Confidencialidad

En la tabla 4 se visualiza el total de errores en el parámetro de confidencialidad de la tecnología Metro 2.0

Tabla 4: Parámetros de Confidencialidad de la Tecnología Metro 2.0

Metro		Cantidad de errores de seguridad
Web Método sobre la tabla Facultad:	Cross Site Scripting	368
	SQL Injection	56
	XPath Injection	40
Total, de errores de seguridad		464

Fuente: Kléber Bustán/Jorge Álvarez

En la tabla 5 se visualiza el total de errores en el parámetro de confidencialidad de la tecnología Axis 2.

Tabla 5: Parámetros de Confidencialidad Tecnología Axis 2

Axis2		Cantidad de errores de seguridad
Web Método sobre la tabla Facultad:	Cross Site Scripting:	0
	SQL Injection	0
	XPath Injection	0
Total, de errores de seguridad		0

Fuente: Kléber Bustán/Jorge Álvarez

A continuación, se observa los valores en cuanto al indicador antes mencionado, estableciendo así cuál de las dos tecnologías de servicio web contiene un número mayor de errores de seguridad, como se observa en la Ilustración 54.

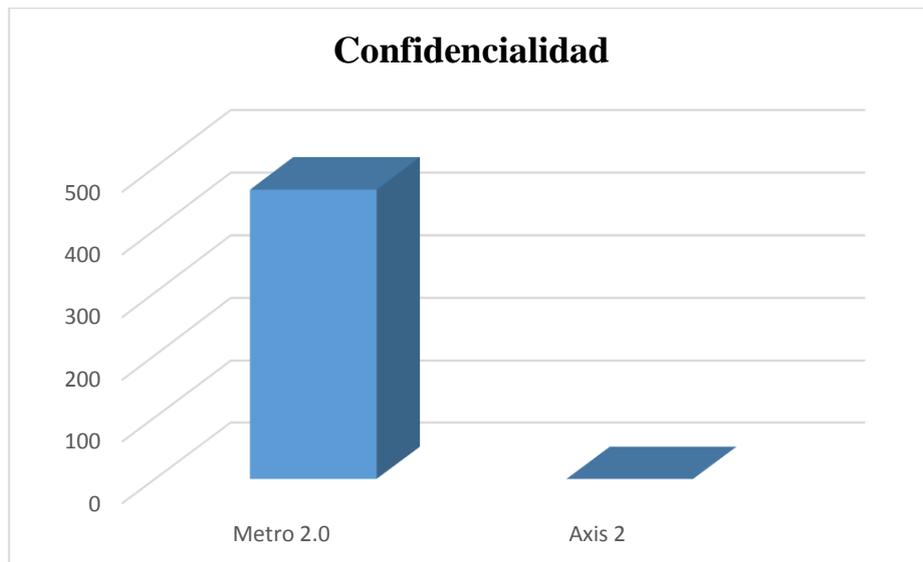


Ilustración 54: Resumen de los Parámetros de Confidencialidad Metro 2.0 y Axis 2
Fuente: Kléber Bustán/Jorge Álvarez

A partir de los datos observados en cuanto al parámetro de Confidencialidad la tecnología Metro 2.0 posee un número mayor de errores de seguridad que la tecnología Axis 2.

3.13.2. Parámetro de Integridad

En la tabla 6 se muestra el total de errores de seguridad en el parámetro de integridad en la tecnología Metro 2.0.

Tabla 6: Parámetros de Integridad de la Tecnología Metro 2.0

Metro		Cantidad de errores de seguridad
Web Método sobre la tabla Facultad:	Malformed XML	36
	XML Bomb	0
	Invalid Types	96
Total, de errores de seguridad		132

Fuente: Kléber Bustán/Jorge Álvarez

En la tabla 7 se muestra el total de errores de seguridad en el parámetro de integridad de la tecnología de servicios web Axis 2.

Tabla 7: Parámetros de Integridad de la tecnología Axis 2

Axis2		Cantidad de errores de seguridad
Web Método sobre la tabla Facultad:	Malformed XML	0
	XML Bomb	0
	Invalid Types	0
Total, de errores de seguridad		0

Fuente: Kléber Bustán/Jorge Álvarez

A partir de los valores en cuanto al indicador antes mencionado, se estableció cuál de las dos tecnologías de servicio web contiene un número mayor de errores de seguridad como se observa en la ilustración 55 el resumen de los parámetros de las tecnologías Axis y Metro 2.0.

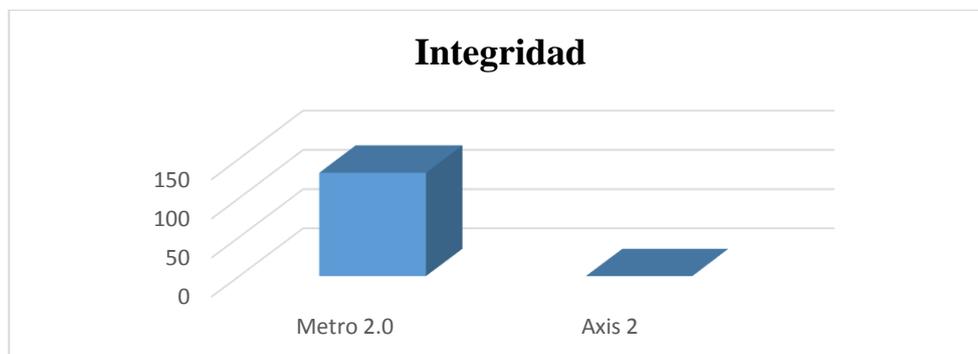


Ilustración 55: Resumen de los Parámetros de Integridad de las tecnologías Metro 2.0 y Axis 2

Fuente: Kléber Bustán/Jorge Álvarez

Los datos presentes en el parámetro de Integridad nos indicaron que la tecnología Metro 2.0 de igual manera tiene un número mayor de errores de seguridad que la tecnología Axis 2.

3.13.3. Parámetro de Disponibilidad

En la tabla 8 se muestra el total de errores de seguridad en el parámetro de disponibilidad en la tecnología Metro 2.0 con un tiempo de prueba de 2 minutos.

Tabla 8: Parámetro de Disponibilidad en la Tecnología Metro 2.0

Metro	Errores en tiempo de respuesta
Facultad: Web método buscar	0
Tiempo de prueba	2m

Fuente: Kléber Bustán/Jorge Álvarez

En la tabla 9 se muestra el total de errores de seguridad en el parámetro de disponibilidad en la tecnología Axis 2 con un tiempo de prueba de 2 minutos.

Tabla 9: Parámetro de Disponibilidad en la Tecnología Axis 2

Axis2	Errores en tiempo de respuesta	
Facultad: Web método buscar	1	
Tiempo de prueba	2m	

Fuente: Kléber Bustán/Jorge Álvarez

A continuación, se observa los valores en cuanto al indicador antes mencionado, estableciendo así cuál de las dos tecnologías de servicio web contiene un número mayor de errores de seguridad, como se observa en la ilustración 56 el resumen de los parámetros de las tecnologías Axis y Metro 2.0.

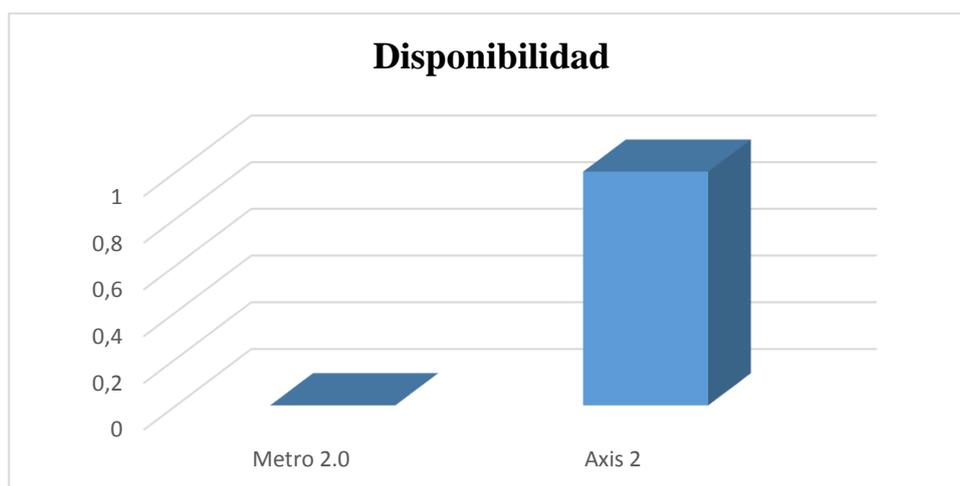


Ilustración 56: Resumen de los Parámetros de Disponibilidad de las tecnologías Metro 2.0 y Axis 2

Fuente: Kléber Bustán/Jorge Álvarez

En cuanto al parámetro de Disponibilidad la tecnología Metro 2.0 no obtuvo ningún error mientras que la tecnología Axis 2 solamente obtuvo un error.

3.14. Resumen del número total de errores de seguridad

En la tabla 10 se indica un resumen del número total de errores de seguridad para las tecnologías de servicio web Axis 2 y Metro 2.0, para cada uno de los parámetros: **Confidencialidad, Integridad y Disponibilidad.**

Tabla 10: Resumen del número total de errores de seguridad en Axis 2 y Metro 2.0

Parámetros de Seguridad	Cantidad de errores de seguridad		Resultados en porcentaje	
	Axis 2	Metro 2.0	Axis 2	Metro 2.0
Confidencialidad	0	464	0%	73,3%
Integridad	0	132	0%	20,9%
Disponibilidad	1	0	0,16%	0%
Total	1	596	0,16%	94,2%
Media	0,33	198,67		
Desviación Estándar	0,58	239,08		

Fuente: Kléber Bustán/Jorge Álvarez

A continuación, se observa los valores en cuanto a los indicadores antes mencionados, estableciendo así cuál de las dos tecnologías de servicio web contiene un menor número de errores de seguridad como se visualiza en la ilustración 57 el resumen general de errores de las tecnologías.

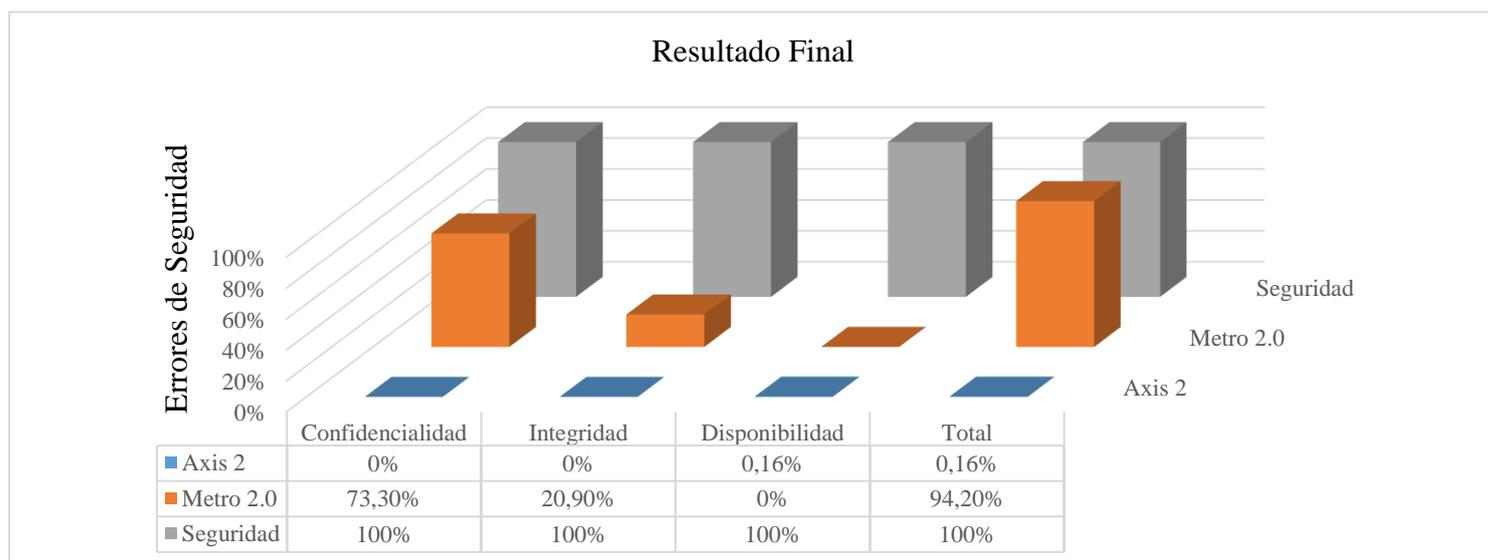


Ilustración 57: Resumen General de Errores de Seguridad de las tecnologías Axis 2 y Metro 2.0

Fuente: Kléber Bustán/Jorge Álvarez

3.15. Comprobación de hipótesis

3.15.1. Hipótesis Nula (H₀)

La tecnología de servicios web Axis 2 no difiere en el número de errores de seguridad frente a la tecnología Metro 2.0.

3.15.2. Hipótesis Alternativa (H₁)

La tecnología de servicios web Axis 2 presenta un menor número de errores de seguridad frente a la tecnología Metro 2.0.

3.15.3. Nivel de Significancia: 0.05

3.16. Especificaciones de las regiones de aceptación y rechazo:

A un nivel $\alpha = 0,05 = \pm 1,96$

3.16.1. Tipo de análisis:

Análisis a dos colas, si H₀ es de la forma $\bar{x}_1 = \bar{x}_2$ y H₁ es de la forma $\bar{x}_1 \neq \bar{x}_2$

3.16.2. Especificación de estadístico prueba Z

Se utilizará la prueba: $Z = \frac{\bar{x}_1 - \bar{x}_2}{\sigma_{\bar{x}_1 - \bar{x}_2}}$ donde $\sigma_{\bar{x}_1 - \bar{x}_2} = \sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}$

3.16.3. Especificaciones de las regiones de aceptación y rechazo:

A un nivel $\alpha = 0,05 = \pm 1,96$ en la ilustración 29 se observa la zona de aceptación de la hipótesis

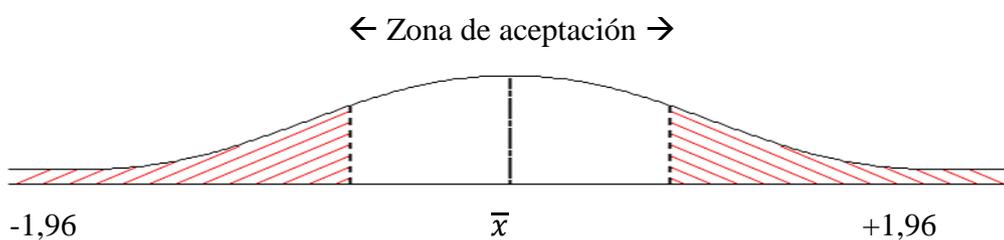


Ilustración 29: Zona de Aceptación
Fuente: Kléber Bustán/Jorge Álvarez

3.17. Recolección y cálculo de datos estadísticos.

En la tabla 11 se realizó la recolección y cálculo de datos estadísticos como son: la desviación estándar, tamaño de la muestra y la media para realizar el test con la prueba Z con las tecnologías Axis 2 y Metro 2.0.

Tabla 11: Recolección de datos Estadísticos

	Axis 2	Metro 2.0
σ = Desviación estándar	0,58	239,08
n = Tamaño de la muestra	633	633
\bar{x} = Media	0,33	198,67

Fuente: Kléber Bustán/Jorge Álvarez

3.18. Evaluación con la prueba Z

Se utilizó la prueba Z (Tomando en cuenta que los datos son mayores a 30 y teniendo la desviación sigma o estándar) para la comprobación de la hipótesis se utilizarán las siguientes formulas:

Para encontrar la desviación sigma se utilizó:

$$\sigma_{\bar{x}_1 - \bar{x}_2} = \sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}$$

La desviación sigma es = 9,50

Para encontrar la Z se utilizó la siguiente formula:

$$Z = \frac{\bar{x}_1 - \bar{x}_2}{\sigma_{\bar{x}_1 - \bar{x}_1}}$$

Por lo tanto, Z es = -20,872

3.19. Conclusión de la Hipótesis

Aplicado la prueba **Z** se comprueba la hipótesis **H1**.

Porque: La tecnología de servicios web Axis 2 presenta un menor número de errores frente a la tecnología Metro 2.0. al estar en el intervalo el valor de $Z = -20,872$ por lo que se establece que la tecnología Axis 2 es la mejor a utilizar para el desarrollo del servicio web aplicado al Sistema Informático de Evaluación docente de la UNACH.

CAPÍTULO IV

DESARROLLO DEL MÓDULO DE INTEGRACIÓN Y REPORTE DEL SISTEMA DE EVALUACIÓN DOCENTE DE LA UNACH.

La Tecnología de servicios web más apropiada para el desarrollo de los módulos de Integración y Reportes con respecto a la seguridad es Axis 2.

Por tal razón esta tecnología de servicios web se utilizará para el desarrollo de los módulos de Integración y Reportes con una metodología de desarrollo de software eXtreme Programming (XP).

La Integración con el Sistema SICOA y el módulo de reportes se desarrolló en la Universidad Nacional de Chimborazo, y lo utilizara el Departamento de Evaluación y Acreditación, el mismo que ayudara a llevar un Control de las Evaluaciones realizadas por los Administrativos, Directores, Docentes, Estudiantes y todos los procesos que se lleven a cabo en esta Gestión de Evaluaciones.

Todos estos temas, conceptos, metodologías y características serán tratados en el transcurso del presente capítulo y el desarrollo de los Módulos en sí.

4.1. Metodología XP

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico, en la ilustración 30 se muestra las fases de la tecnología XP.



Ilustración 30: Fases de la Metodología XP
Fuente: INGENIERÍA DE SOFTWARE, (Beck, 2016)

4.2. Desarrollo de los Módulos de Integración y Reportes

4.2.1. Herramientas de desarrollo

Para el desarrollo de los módulos de Integración y Reportes del sistema de evaluación docente de la UNACH se utilizará las siguientes tecnologías y herramientas que se indican en la tabla 12 a continuación:

Tabla 12: Herramientas Utilizadas para el desarrollo de los módulos de Integración y Reportes

HERRAMIENTA	CONCEPTO	VERSIÓN UTILIZADA
<p>POSTGRESQL</p> 	<p>Sus características técnicas la hacen una de las bases de datos más potentes y robustos del mercado.</p>	<p>Postgres v9.4</p>
<p>NETBEANS</p> 	<p>NetBeans IDE le permite desarrollar rápida y fácilmente aplicaciones de escritorio, móviles y aplicaciones web, así como aplicaciones HTML5 con HTML, JavaScript y CSS.</p>	<p>Netbeans v8.2</p>
<p>IREPORT</p> 	<p>iReport es el código abierto diseñador de informes libre para JasperReports y JasperReports Server. Crea diseños muy sofisticados. Accede a sus datos a través de JDBC, TableModels, JavaBeans, fuentes XML, Hibernate, CSV y personalizados.</p>	<p>Ireport V3.6.0</p>
<p>.NET WEB SERVICE STUDIO</p> 	<p>.NET Servicio web Studio es una herramienta para invocar webMethods forma interactiva.</p>	<p>.Net Web Service V2.0</p>

Fuente: Kléber Bustán /Jorge Álvarez

4.3. Gestión de Migración y Reportes del Sistema Informático de Evaluación Docente de la UNACH.

4.3.1. Planificación del Proyecto

La planificación del proyecto se realizó tras el estudio del problema y los requerimientos, mediante la representación de las historias se efectuó la planificación inicial la cual fue variando en el transcurso de la misma y mejorando las historias en base a la concepción del problema.

4.3.2. Integrantes y roles

Con la participación del Director del proyecto, los miembros, los usuarios y desarrolladores, se formó el equipo encargado de la integración y reportes del Sistema Informático de Evaluación Docente.

Esto implicó que los diseños deberán ser sencillos y claros, los usuarios dispondrán de versiones de prueba del software para que puedan participar en el proceso de desarrollo mediante sugerencias y aportaciones, dicho equipo de trabajo se ve ilustrado en la tabla 13 definiendo Integrantes y Roles.

Tabla 13: Integrantes y Roles

Miembro	Grupo	Roles XP	Metodología
Jorge Álvarez	Tesista	Rastreador, Testeador, Programador	Xp
Kléber Bustán	Tesista	Rastreador, Testeador, Programador	
Ing. Diego Palacios		Consultor	
Ing. Paúl Paguay		Consultor	

Fuente: Kléber Bustán/Jorge Álvarez

4.8.3. Prototipos

Las interfaces de usuario son las más importantes ya que de esto dependerá el entendimiento fácil y rápido por parte del usuario al comenzar a manejar el sistema.

Se pretende que la interfaz del usuario sea amigable, sencilla y funcional con un alto grado de comprensión, por tal razón se crearon los prototipos generales del sistema.

A continuación, se realizará una breve descripción del proceso principal.

4.3.3.1. Migración de datos

Mediante la ilustración 31 observamos un prototipo de menú para seleccionar la migración de acuerdo a la necesidad del administrador del Sistema Informativo de Evaluación Docente de la UNACH.



Ilustración 31: Módulo de Migración
Fuente: Kléber Bustán/Jorge Álvarez

En el apartado **Migrar carrera** se procederán a migrar todas las carreras que posea actualmente la UNACH de todas las facultades disponibles que se encuentren.

En el apartado **Migrar Dictado** se procederán a migrar las asignaturas que contengan las carreras durante el periodo actual.

4.3.3.2. Generación de reportes:

El Menú selección de navegación para los reportes dentro del periodo actual se indica en la ilustración 32.

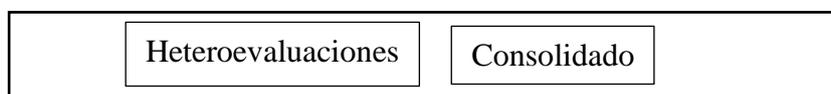


Ilustración 32: Generación de reportes
Fuente: Kléber Bustán/Jorge Álvarez

En la ilustración 33 se indica una pantalla para listar los reportes por carrera de los estudiantes.

Periodo	<input type="text" value="Agosto-Abril"/>			
Facultad	<input type="text" value="Ingenieria"/>			
Escuela	<input type="text" value="Sistemas"/>			
<input type="button" value="Exportar PDF"/>				
Ingenieria	Sistemas	Juan Perez	Base de Datos	Si
Ingenieria	Sistemas	Luis Tapia	Ciencias	No
Ingenieria	Sistemas	Jonny Bravo	Electricidad	No

Ilustración 33: Generación de reportes por carrera de los estudiantes
Fuente: Kléber Bustán/Jorge Álvarez

En la ilustración 34 se indica una pantalla para listar los reportes por docentes.

Periodo	Agosto-Abril			
Facultad	Ingeniería			
Escuela	Sistemas			
Exportar PDF				
Facultad	Escuela	Nombres y Apellidos	Heteroevaluacion	Autoevaluacion
Ingeniería	Sistemas	Juan Perez	si	Si
Ingeniería	Sistemas	Luis Tapia	si	No
Ingeniería	Sistemas	Jonny Bravo	si	No

Ilustración 34: Generación de reportes de los docentes por carrera

Fuente: Kléber Bustán/Jorge Álvarez

4.3.4. Historias de Usuarios

Las historias de usuarios tienen como finalidad ver las necesidades del sistema por lo tanto se realizarán descripciones cortas y escritas en el lenguaje del usuario sin terminología, detallando el tiempo que conllevara la implementación, así como la estimación del riesgo de dicha historia de usuario.

Cada historia de usuario se divide en actividades panificables y medibles para su realización, estas forman una interacción, este plan nos indicara diferentes interacciones del sistema. La realización de este plan debe tener muy en cuenta la prioridad de los usuarios para satisfacerles en mayor medida como se muestra en la tabla 14.

Tabla 14: Historias de Usuarios

Nº	NOMBRE	PRIORIDAD	RIESGO	ESFUERZO	ITERACION
1	Migración por Carreras	Alta	Alto	Alto	1
2	Migración de Dictados por Asignatura	Alta	Alto	Alto	1
3	Emisión del reporte evaluación por asignaturas (Heteroevaluación)	Alta	Medio	Alto	2
4	Emisión del reporte evaluación promedio por docente (consolidado)	Alta	Medio	Alto	2
5	Creación del servicio web con seguridad (Listado de Estudiantes Evaluados por Carrera)	Alta	Alto	Alto	2

Fuente: Kléber Bustán/Jorge Álvarez

4.3.5. Plan de Entregas

El plan de entregas se utilizó para crear planes por cada iteración, con cada historia de usuario previamente evaluada en tiempo de desarrollo ideal, el usuario agrupará en orden de importancia.

Para realizar el plan de entregas se hará en función de dos parámetros: tiempo de desarrollo y grado de importancia para el usuario. Las iteraciones individuales son planificadas en detalle antes de que comience cada iteración como se puede apreciar en las tablas 15, 16 y 17. También en las ilustraciones 35, 36 y 37 observamos los resultados en barras.

Tabla 15: Prototipos de Migración y Consumo de servicios web en semanas

Iteración 1	Duración en semanas
Migración por Carreras	8
Migración de Dictados por Asignatura	7

Fuente: Kléber Bustán /Jorge Álvarez

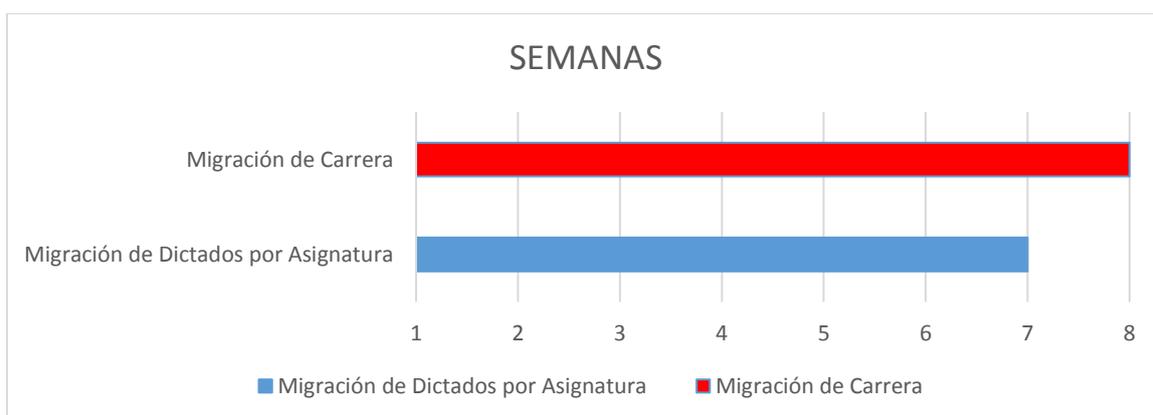


Ilustración 35: Plan de Entregas Iteración 1

Fuente: Kléber Bustán/Jorge Álvarez

Iteración 2

Tabla 16: Plan de Entrega Iteración 2

Historia de Usuario	Duración en semanas
Emisión del reporte evaluación por asignaturas (Heteroevaluación)	2
Emisión del reporte evaluación promedio por docente (consolidado)	2

Fuente: Kléber Bustán/Jorge Álvarez

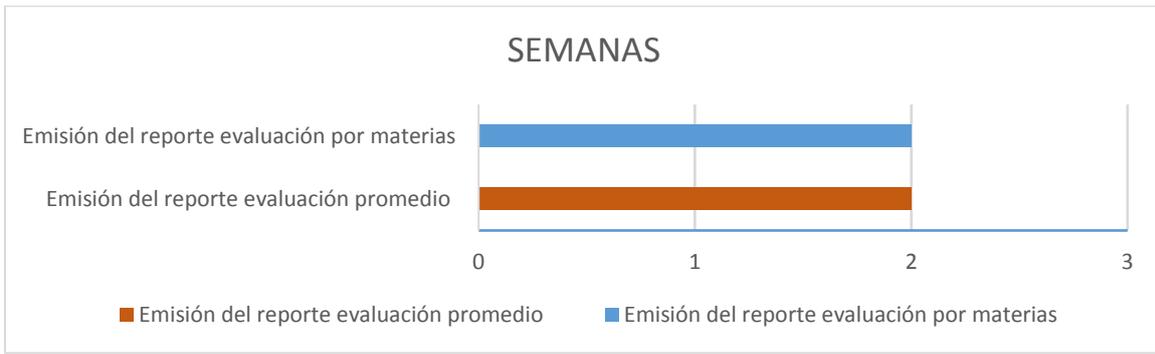


Ilustración 36: Plan de Entregas Iteración 2
Fuente: Kléber Bustán/Jorge Álvarez

Tabla 17: Plan de entrega iteración 2

Historia de Usuario	Duración en semanas
Creación del servicio web con seguridad (Listado de Estudiantes Evaluados por Carrera)	3

Fuente: Kléber Bustán/Jorge Álvarez

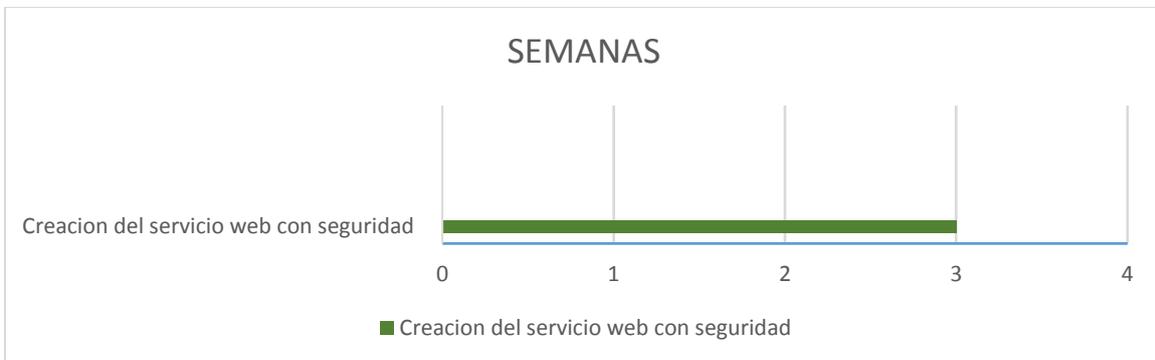


Ilustración 37: Plan de Entregas Iteración 2
Fuente: Kléber Bustán/Jorge Álvarez

4.3.6. Incidencia

Iteración 1: Se realizó un prototipo de la base datos acorde al reglamento de evaluaciones de la UNACH certificado por las autoridades para luego implementarla en el motor de base de datos Postgresql, a continuación, se procedió a crear en Java los respectivos métodos get () y set (), funciones (insertar, actualizar, eliminar, obtener), los controladores y vistas para visualizar la Integración.

Para la integración del sistema de evaluación docente con el sistema SICOA, se requiere consumir servicios web para lo cual se procedió a realizar el trámite correspondiente con el respectivo formato de solicitud (**VER ANEXO 5**) indicando las especificaciones técnicas de los servicios web, este proceso tuvo una duración de aproximadamente 4 semanas.

Para la entrega de la dirección de los servicios web el departamento de UTECA realizó la respectiva acta entrega-recepción (**VER ANEXO 6**) para poder empezar a trabajar en el consumo de los servicios web.

A continuación, se procede a crear las respectivas vistas con la opción migración Carrera y Migrar dictados con sus respectivos parámetros de entrada para las tablas facultad, carrera, periodo, estudiante, docente, asignatura, nivel, paralelo, dictado_asignatura consumiendo de este modo datos reales desde la base de datos SICOA a la base de datos Evaluación.

Una vez consumidos los servicios web, la visión de los miembros del equipo puede ser interpretada de distinta forma que la del usuario es por ello que es importante las sugerencias del usuario administrador, por ende, se sugirió realizar dos vistas para que el administrador pueda conocer las migraciones que se desarrollan durante el periodo vigente, además podrá visualizar las migraciones de las asignaturas realizadas en los distintos periodos académicos.

Iteración 2: Con esta iteración se pretende generar los respectivos reportes para las diferentes evaluaciones y desplegar el servicio web con seguridad.

Por requerimiento del departamento de evaluación se generó un reporte con la nómina de los estudiantes que finalizaron las evaluaciones con su respectiva carrera y periodo, también se generó un reporte por docente, carrera y periodo con su respectivo promedio de evaluación.

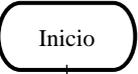
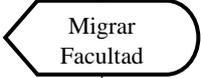
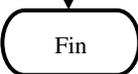
Además, el departamento de evaluación solicitó la creación de un servicio web con seguridad la cual será consumida por el mismo, para el proceso de habilitación de las matrículas en línea.

4.3.7. Actividades de Migración

Las actividades del Módulo de Migración fueron divididas en varios procesos que serán reflejados mediante flujos de procesos.

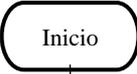
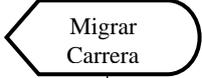
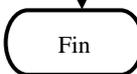
- En la tabla 18 observamos el PROCESO MIGRAR FACULTADES.
- En la tabla 19 observamos el PROCESO MIGRAR CARRERAS.
- En la tabla 20 observamos el PROCESO MIGRAR PERIODOS.
- En la tabla 21 observamos el PROCESO MIGRAR DOCENTES.
- En la tabla 22 observamos el PROCESO MIGRAR ESTUDIANTES.
- En la tabla 23 observamos el PROCESO MIGRAR ASIGNATURAS.
- En la tabla 24 observamos el PROCESO MIGRAR NIVELES.
- En la tabla 25 observamos el PROCESO MIGRAR PARALELOS.
- En la tabla 26 observamos el PROCESO MIGRAR DICTADOS ASIGNATURA.

Tabla 18: Proceso Migrar Facultades
PROCESO MIGRAR FACULTADES

	Actividad	Flujograma	IN	OUT	Responsable	Observación
	Inicio					
1	Actividad				Administrador	
2					Administrador	
3					Administrador	
	Fin					

Fuente: Kléber Bustán/Jorge Álvarez

Tabla 19: Proceso Migrar Carreras
PROCESO MIGRAR CARRERAS

	Actividad	Flujograma	IN	OUT	Responsable	Observación
	Inicio					
1	Actividad				Administrador	
2					Administrador	
3					Administrador	
	Fin					

Fuente: Kléber Bustán/Jorge Álvarez

Tabla 20: Proceso Migrar Periodos

PROCESO MIGRAR PERIODOS						
	Actividad	Flujograma	IN	OUT	Responsable	Observación
	Inicio					
1	Actividad				Administrador	
2					Administrador	
3					Administrador	
	Fin					

Fuente: Kléber Bustán/Jorge Álvarez

Tabla 21: Proceso Migrar Docentes

PROCESO MIGRAR DOCENTES						
	Actividad	Flujograma	IN	OUT	Responsable	Observación
	Inicio					
1	Actividad				Administrador	
2					Administrador	
3					Administrador	
	Fin					

Fuente: Kléber Bustán/Jorge Álvarez

Tabla 22: Proceso Migrar Estudiantes

PROCESO MIGRAR ESTUDIANTES						
	Actividad	Flujograma	IN	OUT	Responsable	Observación
	Inicio					
1	Actividad				Administrador	
2					Administrador	
3					Administrador	
	Fin					

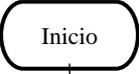
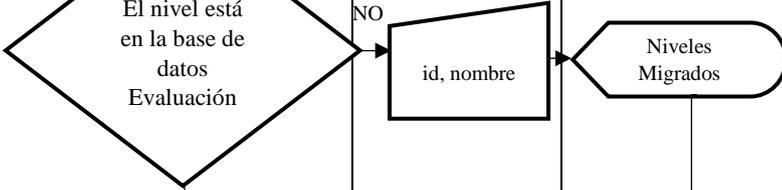
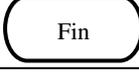
Fuente: Kléber Bustán/Jorge Álvarez

Tabla 23: Proceso Migrar Asignaturas

PROCESO MIGRAR ASIGNATURAS						
	Actividad	Flujograma	IN	OUT	Responsable	Observación
	Inicio					
1	Actividad				Administrador	
2					Administrador	
3					Administrador	
	Fin					

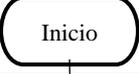
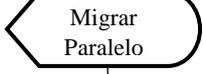
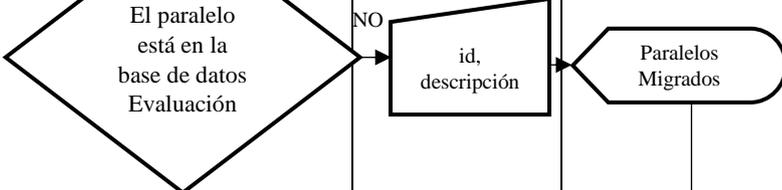
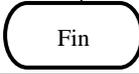
Fuente: Kléber Bustán/Jorge Álvarez

Tabla 24: Proceso Migrar Niveles
PROCESO MIGRAR NIVELES

Actividad	Flujograma	IN	OUT	Responsable	Observación
Inicio					
1				Administrador	
2				Administrador	
3				Administrador	
Fin					

Fuente: Kléber Bustán/Jorge Álvarez

Tabla 25: Proceso Migrar Paralelos
PROCESO MIGRAR PARALELOS

Actividad	Flujograma	IN	OUT	Responsable	Observación
Inicio					
1				Administrador	
2				Administrador	
3				Administrador	
Fin					

Fuente: Kléber Bustán/Jorge Álvarez

Tabla 26: Proceso Migrar Dictados Asignatura

PROCESO MIGRAR DICTADOS_ASIGNATURA						
	Actividad	Flujograma	IN	OUT	Responsable	Observación
	Inicio					
1	Actividad				Administrador	
2		 Los dictados están en la base de datos Evaluación	NO	 id_perodo, id_carrera, id_docente, id_asignatura , id_nivel, id_paralelo	Administrador	
3		SI		 Dictados_ Asignatura Migrados	Administrador	
	Fin					

Fuente: Kléber Bustán/Jorge Álvarez

4.3.8. Actividades de Reportes

Las actividades del módulo de reportes fueron divididas en varios procesos que serán reflejados mediante flujos de procesos como se observa en las tablas 27 y 28.

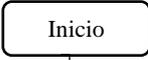
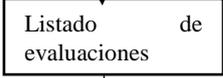
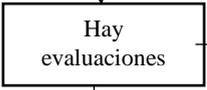
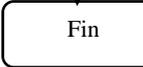
- Emisión del reporte evaluación por asignaturas (heteroevaluación).
- Emisión del reporte evaluación promedio por docente (consolidado).
- Creación del servicio web con seguridad (Listado de Estudiantes Evaluados por Carrera).

Tabla 27: Proceso Actividades Reportes

ACTIVIDADES DE EMISIÓN REPORTES (CONSOLIDADO Y HETEROEVALUACIONES)						
	Actividad	Flujograma	IN	OUT	Responsable	Observaciones
	Inicio				Tutor	
1					Tutor	
2					Tutor	
3					Tutor	
4					Tutor	
	Fin					

Fuente: Kléber Bustán/Jorge Álvarez

Tabla 28: Proceso Creación de servicio web con seguridad

CREACIÓN DEL SERVICIO WEB CON SEGURIDAD (LISTADO DE ESTUDIANTES EVALUADOS POR CARRERA)						
	Actividad	Flujograma	IN	OUT	Responsable	Observaciones
	Inicio				Tutor	
1					Tutor	
2					Tutor	
3					Tutor	
4					Tutor	
	Fin					

Fuente: Kléber Bustán/Jorge Álvarez

4.4. Implementación

4.4.1. Diagrama de base de datos

Nuestra base de datos consta de 1 esquema y 64 funciones:

Como se observa en la ilustración 38 tenemos nuestro esquema **Public**: donde se maneja todo lo concerniente con la seguridad, creación de menús, y todo lo que tiene que ver con la migración de Facultades, Carreras, Periodos, Estudiantes, Docentes, Asignaturas, Niveles, Paralelos, Dictado Asignatura, Autoevaluaciones, Coevaluaciones y Heteroevaluaciones.

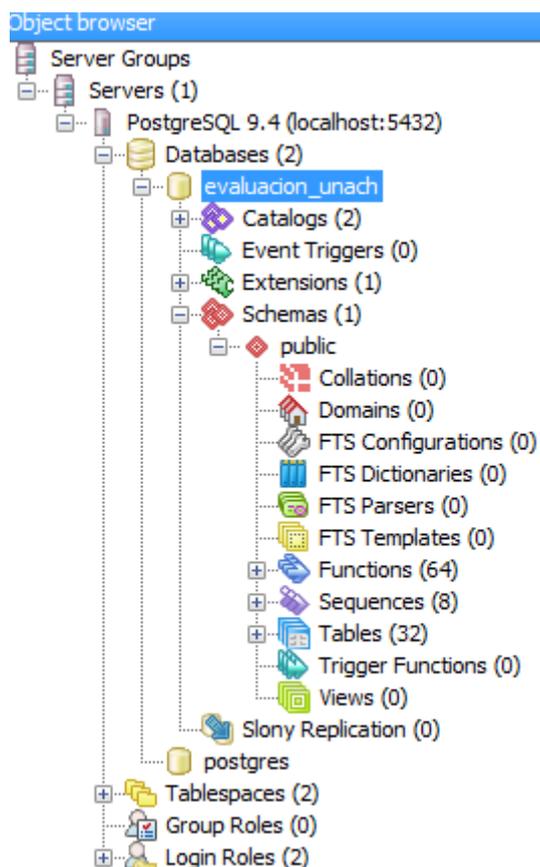


Ilustración 38: Esquema Base de Datos-Postgresql
Fuente: Kléber Bustán/Jorge Álvarez

DIAGRAMA ENTIDAD RELACIÓN DE LA BASE DE DATOS DEL SISTEMA DE EVALUACIÓN DOCENTE - UNACH

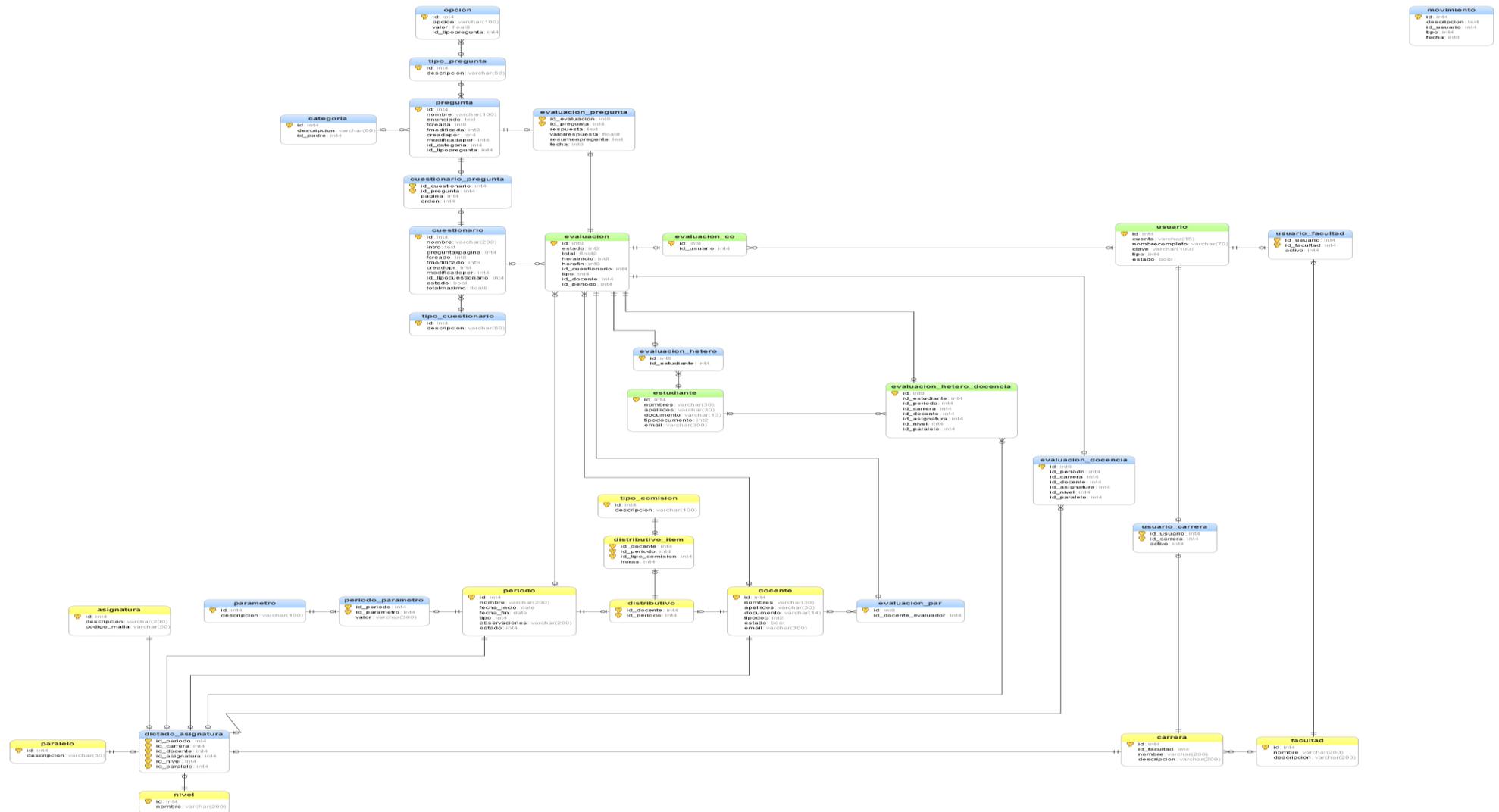


Ilustración 39 : Diagrama Entidad Relación BD
Autores: Kléber Bustán/Jorge Álvarez

4.4.2. Diccionario de datos

El diccionario de datos permite guardar la estructura de la base de datos, es decir se define como se almacena y accede a la información.

4.4.2.1. Migración por Carreras y Dictados Asignatura

NOMBRE DE LA TABLA: facultad, es la tabla que permite registrar todas las facultades que se encuentran en la institución como se observa en la tabla 29.

Tabla 29: Facultad

Nombre de la Columna	Tipo de Dato	Clave Primaria	Valores Nulos	Auto Incremental
id	int4	SI	NO	NO
nombre	varchar	NO	NO	NO
descripción	varchar	NO	NO	NO

Fuente: Kléber Bustán/Jorge Álvarez

NOMBRE DE LA TABLA: carrera, es la tabla que permite registrar todas las carreras que se encuentran distribuidas en cada facultad de la institución como se observa en la tabla 30.

Tabla 30: Carrera

Nombre de la Columna	Tipo de Dato	Clave Primaria	Valores Nulos	Auto Incremental
id	int4	SI	NO	NO
id_facultad	Int4	NO	NO	NO
nombre	varchar	NO	NO	NO
descripción	varchar	NO	NO	NO

Fuente: Kléber Bustán/Jorge Álvarez

NOMBRE DE LA TABLA: periodo, es la tabla que permite registrar todos los periodos académicos de la institución como se observa en la tabla 31.

Tabla 31: Periodo

Nombre de la Columna	Tipo de Dato	Clave Primaria	Valores Nulos	Auto Incremental
id	int4	SI	NO	NO
nombre	varchar	NO	NO	NO
fecha_inicio	date	NO	NO	NO

fecha_fin	date	NO	NO	NO
tipo	int4	NO	NO	NO
observaciones	varchar	NO	NO	NO
estado	int4	NO	NO	NO

Fuente: Kléber Bustán/Jorge Álvarez

NOMBRE DE LA TABLA: estudiante, es la tabla que permite registrar todos los estudiantes de las distintas Carreras de la institución como se observa en la tabla 32.

Tabla 32: Estudiante

Nombre de la Columna	Tipo de Dato	Clave Primaria	Valores Nulos	Auto Incremental
id	int4	SI	NO	NO
nombres	varchar	NO	NO	NO
apellidos	varchar	NO	NO	NO
documento	varchar	NO	NO	NO
tipo_documento	int4	NO	NO	NO
Email	varchar	NO	NO	NO

Fuente: Kléber Bustán/Jorge Álvarez

NOMBRE DE LA TABLA: docente, es la tabla que permite registrar todos los docentes de las distintas Carreras de la institución como se observa en la tabla 33.

Tabla 33: Docente

Nombre de la Columna	Tipo de Dato	Clave Primaria	Valores Nulos	Auto Incremental
id	int4	SI	NO	NO
nombres	varchar	NO	NO	NO
apellidos	varchar	NO	NO	NO
documento	varchar	NO	NO	NO
tipo_documento	int4	NO	NO	NO
estado	boolean	NO	NO	NO
Email	varchar	NO	NO	NO

Fuente: Kléber Bustán/Jorge Álvarez

NOMBRE DE LA TABLA: asignatura, es la tabla que permite registrar todas las asignaturas de las Carreras de la institución como se observa en la tabla 34.

Tabla 34: Asignatura

Nombre de la Columna	Tipo de Dato	Clave Primaria	Valores Nulos	Auto Incremental
Id	int4	SI	NO	NO
descripción	varchar	NO	NO	NO
codigo_malla	varchar	NO	NO	NO

Fuente: Kléber Bustán/Jorge Álvarez

NOMBRE DE LA TABLA: nivel, es la tabla que permite registrar todos los niveles de las distintas carreras de la institución como se observa en la tabla 35.

Tabla 35: Nivel

Nombre de la Columna	Tipo de Dato	Clave Primaria	Valores Nulos	Auto Incremental
id	int4	SI	NO	NO
nombre	varchar	NO	NO	NO

Fuente: Kléber Bustán/Jorge Álvarez

NOMBRE DE LA TABLA: paralelo, es la tabla que permite registrar todos los paralelos de los distintos niveles de la institución como se observa en la tabla 36.

Tabla 36: Paralelo

Nombre de la Columna	Tipo de Dato	Clave Primaria	Valores Nulos	Auto Incremental
id	int4	SI	NO	NO
nombre	varchar	NO	NO	NO

Fuente: Kléber Bustán/Jorge Álvarez

NOMBRE DE LA TABLA: dictado_asignatura, es la tabla que permite registrar todos los dictados por asignatura de los distintos niveles de la institución como se observa en la tabla 37.

Tabla 37: Dictado_asignatura

Nombre de la Columna	Tipo de Dato	Clave Primaria	Valores Nulos	Auto Incremental
id_periodo	int4	SI	NO	NO
id_carrera	int4	SI	NO	NO
id_docente	int4	SI	NO	NO

id_asignatura	int4	SI	NO	NO
id_nivel	int4	SI	NO	NO
id_paralelo	int4	SI	NO	NO

Fuente: Kléber Bustán/Jorge Álvarez

4.4.2.2. Creación de Reporte Heteroevaluación finalizadas y Consolidado por docente.

NOMBRE DE LA TABLA: **evaluación**, esta tabla permite almacenar datos de las evaluaciones como se observa en la tabla 38.

Tabla 38: Evaluación

Nombre de la Columna	Tipo de Dato	Clave Primaria	Valores Nulos	Auto incremental
Id	Bigint	SI	NO	NO
estado	smallint	NO	NO	NO
total	double precisión	NO	NO	NO
hora_inicio	Bigint	NO	NO	NO
hora_fin	Bigint	NO	NO	NO
id_cuestionario	integer	NO	NO	NO
tipo	integer	NO	NO	NO
id_docente	integer	NO	NO	NO
id_periodo	integer	NO	NO	NO

Fuente: Kléber Bustán/Jorge Álvarez

NOMBRE DE LA TABLA: **evaluación_co**, esta tabla permite almacenar datos de la evaluación de los docentes como se observa en la tabla 39.

Tabla 39: Coevaluación

Nombre de la Columna	Tipo de Dato	Clave Primaria	Valores Nulos	Auto incremental
Id	Bigint	SI	NO	NO
id_usuario	integer	NO	NO	NO

Fuente: Kléber Bustán/Jorge Álvarez

NOMBRE DE LA TABLA: **evaluación_hetero_docencia**, esta tabla permite almacenar datos de la evaluación de los estudiantes como se observa en la tabla 40.

Tabla 40: Evaluación_hetero_docencia

Nombre de la Columna	Tipo de Dato	Clave Primaria	Valores Nulos	Auto incremental
id	Bigint	SI	NO	NO
id_estudiante	integer	NO	NO	NO
id_periodo	integer	NO	NO	NO
id_carrera	integer	NO	NO	NO
id_docente	integer	NO	NO	NO
id_asignatura	integer	NO	NO	NO
id_nivel	integer	NO	NO	NO

id_paralelo	integer	NO	NO	NO
-------------	---------	----	----	----

Fuente: Kléber Bustán/Jorge Álvarez

NOMBRE DE LA TABLA: evaluaci3n_docencia esta tabla permite almacenar datos de las evaluaciones realizadas como se observa en la tabla 41.

Tabla 41: Evaluaci3n_docencia

Nombre de la Columna	Tipo de Dato	Clave Primaria	Valores Nulos	Auto incremental
id	bigint	SI	NO	NO
id_periodo	integer	NO	NO	NO
id_carrera	integer	NO	NO	NO
id_docente	integer	NO	NO	NO
id_asignatura	integer	NO	NO	NO
id_nivel	integer	NO	NO	NO
id_paralelo	integer	NO	NO	NO

Fuente: Kléber Bustán/Jorge Álvarez

4.4.3. Funciones

Las funciones permitieron capturar datos para el proceso de la creaci3n de los reportes.

NOMBRE DE LA FUNCI3N: fn_util_parametro_autodireciongestion esta funci3n captura el valor del parámetro de auto direcci3n y gesti3n como se observa en la tabla 42.

Tabla 42: Funci3n de porcentaje de autodirecci3n y gesti3n

Nombre de Parámetro	Tipo de Parámetro	Tipo de Dato
codperiodo	IN	Integer
parametro_autodireciongestion	OUT	Integer

Fuente: Kléber Bustán/Jorge Álvarez

NOMBRE DE LA FUNCI3N: fn_util_parametro_autodocencia esta funci3n captura el valor del parámetro de auto docencia como se observa en la tabla 43.

Tabla 43: Funci3n de porcentaje auto docencia

Nombre de Parámetro	Tipo de Parámetro	Tipo de Dato
codperiodo	IN	integer
parametro_autodireciongestion	OUT	integer

Fuente: Kléber Bustán/Jorge Álvarez

NOMBRE DE LA FUNCI3N: fn_util_parametro_autoinvestigacion esta funci3n captura el parámetro de auto investigaci3n como se observa en la tabla 44.

Tabla 44: Función de porcentaje auto investigación

Nombre de Parámetro	Tipo de Parámetro	Tipo de Dato
Ddocumento	IN	character varying
_codperiodo	IN	integer
Promedio	OUT	double precision

Fuente: Kléber Bustán/Jorge Álvarez

NOMBRE DE LA FUNCIÓN: fn_util_get_promedio_autodireccion esta función captura los promedios de auto dirección como se observa en la tabla 45.

Tabla 45: Función para calcular el promedio autodirección

Nombre de Parámetro	Tipo de Parámetro	Tipo de Dato
Ddocumento	IN	character varying
_codperiodo	IN	Integer
Promedio	OUT	double precision

Fuente: Kléber Bustán/Jorge Álvarez

NOMBRE DE LA FUNCIÓN: fn_get_resultado_autodirecciongestion esta función captura el valor del resultado de auto dirección y gestión como se observa en la tabla 46.

Tabla 46: Funciones de valor máximo auto dirección y gestión

Nombre de Parámetro	Tipo de Parámetro	Tipo de Dato
Ddocumento	IN	character varying
Codperiodo	IN	Integer
Total	OUT	Float

Fuente: Kléber Bustán/Jorge Álvarez

NOMBRE DE LA FUNCIÓN: fn_get_resultado_autoinvestigacion esta función captura el valor del resultado de auto investigación como se visualiza en la tabla 47.

Tabla 47: Funciones de valor máximo auto investigación

Nombre de Parámetro	Tipo de Parámetro	Tipo de Dato
Ddocumento	IN	character varying
Codperiodo	IN	Integer
Total	OUT	Float

Fuente: Kléber Bustán/Jorge Álvarez

NOMBRE DE LA FUNCIÓN: fn_get_resultado_direcdocencia esta función captura el valor del resultado de la dirección docencia como se visualiza en la tabla 48.

Tabla 48: Funciones de valor máximo dirección docencia

Nombre de Parámetro	Tipo de Parámetro	Tipo de Dato
Ddocumento	IN	character varying
Codperiodo	IN	Integer
Total	OUT	Float

Fuente: Kléber Bustán/Jorge Álvarez

NOMBRE DE LA FUNCIÓN: fn_util_resultado_consolidado esta función devuelve el valor del resultado de las evaluaciones de manera consolidada como se visualiza en la tabla 49.

Tabla 49: Funciones de valor máximo Evaluaciones

Nombre de Parámetro	Tipo de Parámetro	Tipo de Dato
ddocumento	IN	character varying
codperiodo	IN	integer
nperiodo	OUT	character varying
nfacultad	OUT	character varying
ncarrera	OUT	character varying
nombres	OUT	character varying
apellidos	OUT	character varying
autodocencia	OUT	double precisión
autoinvestigacion	OUT	double precisión
autodireciongestion	OUT	double precisión
direcdocencia	OUT	double precisión
direcivestigacion	OUT	double precisión
direcdireciongestion	OUT	double precisión
hetedocencia	OUT	double precisión
hetedireciongestion	OUT	double precisión
pautodocencia	OUT	Integer
pautoinvestigacion	OUT	Integer
pautodireciongestion	OUT	Integer
pparesdocencia	OUT	Integer
pparesinvestigacion	OUT	Integer
pparesdireciongestion	OUT	Integer
pdirecdocencia	OUT	Integer
pdirecivestigacion	OUT	Integer

Fuente: Kléber Bustán/Jorge Álvarez

4.9.3. Prototipos interfaces de usuario finales

4.4.3.1. Interfaces de Migración

A continuación, se visualiza en la ilustración 40 la interfaz de los resultados de la migración por carrera en el sistema de evaluación docente de la UNACH.

Sistema de Evaluación UNACH

Inicio Cambio de clave Administrar Cuestionarios Migrar Carreras Migrar Dictado Reportes

Usuario

Nombre

Perfil

Cerrar Sesión

FACULTAD Seleccione la Facultad

CARRERA Seleccione un Carrera

PERIODO Seleccione un Periodo

Migración Carrera

Resultado									
Resultado:	Estudiantes	Docentes	Asignaturas	Niveles	Paralelos	Dictados	AutoEvaluaciones	CoEvaluaciones	HeteroEvaluaciones
Error	Migrados: 0	Migrados: 0	Migradas: 0	Migrados: 0	Migrados: 0	Migrados: 0	Migradas: 0	Migradas: 0	Migradas: 0

Todos los derechos reservados
UNACH - 2015

Ilustración 40: Migración Carrera
Fuente: Kléber Bustán/Jorge Álvarez

También se creó una vista para administrar la migración por dictados asignatura a continuación se detalla el resultado en la ilustración 41 Migración por Dictados Asignatura.

Sistema de Evaluación UNACH

Inicio Cambio de clave Administrar Cuestionarios Migrar Carreras Migrar Dictado Reportes

Usuario

Nombre

Perfil

Cerrar Sesión

Datos de la Carrera a Migrar

FACULTAD Seleccione la Facultad

CARRERA Seleccione un Carrera

PERIODO Seleccione un Periodo

Dictado de Asignaturas

Docente Asignatura Nivel Paralelo

Sin registros

Resultado

Todos los derechos reservados
UNACH - 2015

Ilustración 41: Migración por Dictados Asignatura
Fuente: Kléber Bustán/Jorge Álvarez

4.4.3.2. Interfaces de Reportes

Con la descripción detallada en las historias de usuarios, actividades planificadas y con los diagramas de procesos se definió interfaces de usuarios finales, las cuales serán implantadas en el sistema.

Prototipo de la interfaz final para la generación de los reportes consolidados de evaluación docente durante el un periodo académico como se visualiza en la ilustración 42.

Sistema de Evaluación UNACH

Mensaje Docente Encontrado!!

Inicio Cambio de clave Administrar Cuestionarios Migrar Carreras Migrar Dictado Reportes

Usuario: ADMINISTRADOR

Nombre: Administrador
Perfil: Administrador
Cerrar Sesión

EVALUACION INTEGRAL

Buscar: 0602724775 Consultar

C.I.	0602724775
Nombres	PAUL XAVIER
Apellidos	PAGUAY SOXO
Período	ABRIL 2016 - AGOSTO 2016

Imprimir

Todos los derechos reservados UNACH - 2015

Ilustración 42: Generación de Reporte Administrador

Fuente: Kléber Bustán/Jorge Álvarez

En la ilustración 43 observamos el prototipo de la interfaz final de la generación de los reportes de las Heteroevaluaciones finalizadas por paralelo y nivel durante el periodo académico.

Sistema de Evaluación UNACH

Inicio Cambio de clave Administrar Cuestionarios Migrar Carreras Migrar Dictado Reportes

Usuario: ADMINISTRADOR

Nombre: Administrador
Perfil: Administrador
Cerrar Sesión

GESTION DE HETEROEVALUACIONES

Período: ABRIL 2016 - AGOSTO 2016

Facultad: FACULTAD DE INGENIERÍA

Carrera: INGENIERIA EN SISTEMAS Y COMPUTACION

Nivel: OCTAVO SEMESTRE

Paralelo: A

Imprimir

Todos los derechos reservados UNACH - 2015

Ilustración 43: Generación de Reporte Administrador

Fuente: Kléber Bustán/Jorge Álvarez

4.4.4. Iteración 1

En la iteración 1 observamos las tablas 50 y 51.

Tabla 50: Iteración 1, Historia 1

Historia de Usuarios	
Numero: 1	Usuario: Administrador
Nombre historia: Migrar Carreras.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Esfuerzo: Alto	Iteración asignada: 1
Programador responsable: Kléber Bustán/Jorge Álvarez	
Descripción: El administrador una vez que ingrese al sistema podrá realizar la migración por Facultad, Carrera y Periodo, devolviendo como resultado el número de estudiantes, docentes, asignaturas, niveles, paralelos, dictados, autoevaluaciones, coevaluaciones y heteroevaluaciones.	
Observaciones: Se necesitan varios parámetros para generar la respectiva migración el cual son los siguientes: Facultad, Carrera, Periodo.	

Sistema de Evaluación UNACH

Inicio Cambio de clave Administrar Cuestionarios Migrar Carreras Migrar Dictado Reportes

Usuario

Nombre

Perfil

Cerrar Sesión

FACULTAD Seleccione la Facultad

CARRERA Seleccione un Carrera

PERIODO Seleccione un Periodo

Migracion Carrera

Resultado									
Resultado:	Estudiantes Migrados:	Docentes Migrados:	Asignaturas Migradas:	Niveles Migrados:	Paralelos Migrados:	Dictados Migrados:	AutoEvaluaciones Migradas:	CoEvaluaciones Migradas:	HeteroEvaluaciones Migradas:
Error	0	0	0	0	0	0	0	0	0

Todos los derechos reservados
UNACH - 2015

Fuente: Kléber Bustán/Jorge Álvarez

Tabla 51: Iteración 1, Historia 2

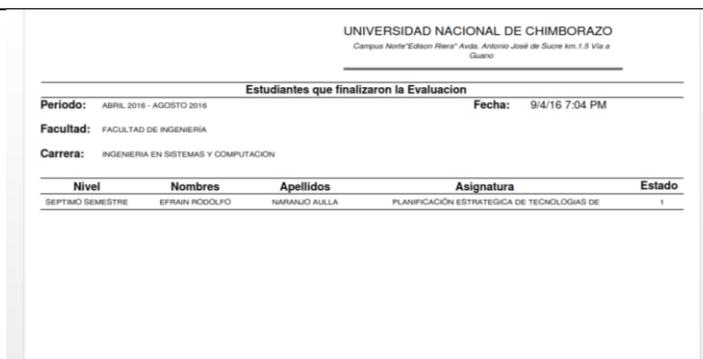
Historia de Usuarios	
Numero: 2	Usuario: Administrador
Nombre historia: Migrar Dictado.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Esfuerzo: Alto	Iteración asignada: 1
Programador responsable: Kléber Bustán/Jorge Álvarez	
Descripción: El administrador una vez que ingrese al sistema podrá realizar la migración por Dictado Asignatura dado la Facultad, Carrera y Periodo, obteniendo como resultado los docentes, asignaturas, niveles, paralelos.	
Observaciones: Se necesitan varios parámetros para generar la respectiva migración por dictados el cual son los siguientes: Facultad, Carrera, Periodo.	

Fuente: Kléber Bustán/Jorge Álvarez

4.4.5. Iteración 2

En la iteración 2 observamos las tablas 52, 53 y 54.

Tabla 52: Iteración 2, Historia 1

Historia de Usuarios	
Numero: 1	Usuario: Administrador
Nombre historia: Emisión de reportes de las Heteroevaluaciones finalizadas durante el periodo académico ().	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Esfuerzo: Medio	Iteración asignada: 2
Programador responsable: Kléber Bustán/Jorge Álvarez	
Descripción: El administrador procede a generar los reportes de las Heteroevaluaciones por carrera solicitado por el director de escuela.	
Observaciones: Se necesitan varios parámetros para generar un reporte en formato pdf: Periodo, Facultad, Escuela, Nivel, Paralelo.	
	

Fuente: Kléber Bustán/Jorge Álvarez

Tabla 53: Iteración 2, Historia 2

Historia de Usuarios	
Numero: 2	Usuario: Administrador
Nombre historia: Emisión de reportes de evaluación dado un docente durante el periodo académico.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Esfuerzo: Medio	Iteración asignada: 2
Programador responsable: Kléber Bustán/Jorge Álvarez	

4.10. Pruebas

Las pruebas son muy importantes y son creadas a partir de las historias de usuario, el usuario debe especificar los aspectos que se van a probar cuando una historia de usuario ha sido realizada correctamente, estas pruebas de usuario pueden tener una o más pruebas para garantizar el correcto funcionamiento, cada prueba realizada representa una salida del sistema.

A continuación, se muestra las pruebas realizadas a cada una de las historias de los usuarios del sistema con su respectiva tabla de pruebas como se visualiza en las tablas 55, 56, 57, 58 y 59.

4.10.1. Historia de Usuario 1

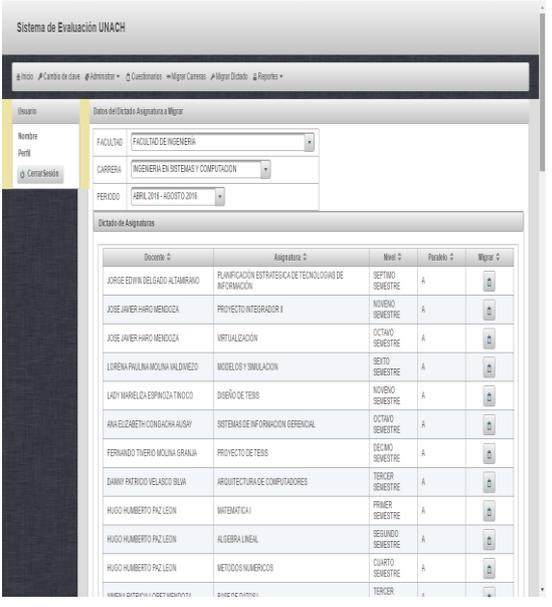
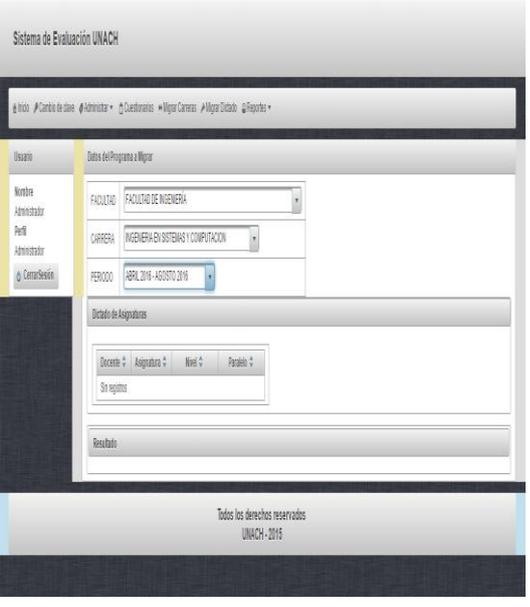
Tabla 55: Prueba Historia 1 - Migración por carreras

	Descripción	Autor
	Pruebas	Jorge Álvarez/Kléber Bustán
MIGRACIÓN POR CARRERAS		
Descripción	El Administrador ingresa al sistema y se ubica en la opción del menú superior migrar carrera para poder realizar la migración.	
Condiciones de Ejecución.	El Administrador debe constar en la base de datos del sistema para poder realizar la respectiva migración por facultad, carrera y periodo.	
Entrada	El Administrador, una vez que ingresa al sistema escoge la opción migrar carrera.	
Resultado Esperado	Al dar un clic en el botón migrar carreras se realiza la migración, exitosamente. Caso contrario como resultado me da como resultado un error.	
Evaluación de la Prueba	<p>Exitosa</p>  <p>The screenshot shows the 'Migración Carreras' form with the following data: FACULTAD: FACULTAD DE INGENIERIA, CARRERA: INGENIERIA EN SISTEMAS Y COMPUTACION, PERIODO: ABRIL 2016 - AGOSTO 2016. The 'Resultado' table shows: Resultado: Correcto, Estudiantes Migrados: 176, Docentes Migrados: 35, Asignaturas Migradas: 62, Niveles Migrados: 10, Paralelos Migrados: 10, Ciclos Migrados: 62, AutoEvaluaciones Migradas: 71, Criterios Migrados: 71, Hechos Migrados: 1017.</p>	<p>Fallida</p>  <p>The screenshot shows the 'Migración Carreras' form with the same data as the successful case. The 'Resultado' table shows: Resultado: Error, Estudiantes Migrados: 0, Docentes Migrados: 0, Asignaturas Migradas: 0, Niveles Migrados: 0, Paralelos Migrados: 0, Ciclos Migrados: 0, AutoEvaluaciones Migradas: 0, Criterios Migrados: 0, Hechos Migrados: 0.</p>

Fuente: Kléber Bustán/Jorge Álvarez

4.10.2. Historia de Usuario 2

Tabla 56: Prueba Historia 2 - Migración de dictados por asignatura

	Descripción	Autor
	Pruebas	Jorge Álvarez/Kléber Bustán
MIGRACIÓN DE DICTADOS POR ASIGNATURA		
Descripción	El Administrador dependiendo del listado podrá imprimir un reporte.	
Condiciones de Ejecución.	El Administrador debe constar en la base de datos del sistema para poder generar un reporte.	
Entrada	El Administrador, una vez que ingresa al sistema escoge la opción imprimir.	
Resultado Esperado	Tras seleccionar la acción que requieren se imprime, exitosamente el reporte.	
Evaluación de la Prueba	<p>Exitosa</p> 	<p>Fallida</p> 

Fuente: Kléber Bustán/Jorge Álvarez

4.10.3. Historia de Usuario 3

Tabla 57: Prueba Historia 3 - Reporte heteroevaluaciones

	Descripción	Autor
	Pruebas	Jorge Álvarez/Kléber Bustán
EMISIÓN DEL REPORTE EVALUACIÓN POR ASIGNATURAS (HETEROEVALUACION)		
Descripción	El Administrador dependiendo del listado podrá imprimir un reporte.	
Condiciones de Ejecución.	El Administrador debe constar en la base de datos del sistema para poder generar un reporte.	
Entrada	El Administrador, una vez que ingresa al sistema escoge la opción imprimir.	
Resultado Esperado	Tras seleccionar la acción que requieren se imprime, exitosamente el reporte.	
Evaluación de la Prueba	Exitosa 	Fallida 

Fuente: Kléber Bustán/Jorge Álvarez

4.10.4. Historia de Usuario 4

Tabla 58: Prueba Historia 4 - Reporte consolidado por docente

	Descripción	Autor
	Pruebas	Jorge Álvarez/Kléber Bustán
EMISIÓN DEL REPORTE EVALUACIÓN PROMEDIO POR DOCENTE (CONSOLIDADO)		
Descripción	El Administrador dado el número del documento de identificación podrá imprimir el reporte consolidado del docente.	
Condiciones de Ejecución.	El Administrador debe constar en la base de datos del sistema para poder generar un reporte.	
Entrada	El Administrador, una vez que ingresa al sistema escoge la opción imprimir.	
Resultado Esperado	Tras seleccionar la acción que requieren se imprime, exitosamente el reporte.	

CONCLUSIONES

- El estudio de las características de las tecnologías Axis 2 permitió determinar que la tecnología de Servicio Web Axis 2 posee un menor número de errores de seguridad con un 0,16% frente a Metro 2.0 que obtuvo un 94,20%, por lo tanto, Axis 2 ofrece un 94,04% en comparación a Metro 2.0 de mejora en errores de seguridad estableciendo como una herramienta muy útil en el intercambio de información entre cliente y servidor.
- Se seleccionó la tecnología de Servicio Web Axis 2 para el desarrollo del servicio web seguro, con la información de las evaluaciones del sistema informático de evaluación docente, para el sistema SICOA, debido a que los 2 sistemas estarán a la par compartiendo información con seguridad, al cual se logró integrar el estudio de las tecnologías en un caso aplicativo real, en beneficio de la UNACH.
- Mediante el consumo de servicios web se logró la integración del Sistema Informático de Evaluación docente con el sistema SICOA de la UNACH, permitiendo así reducir la duplicidad de información.
- Se desarrolló el nuevo método web de información sobre las evaluaciones para los estudiantes mediante la tecnología de Servicio Web Axis 2 aplicando la seguridad, necesarios para la verificación de las Heteroevaluaciones la misma que activará la matrícula online.

RECOMENDACIONES

- Recomendamos especificar de forma detallada los servicios web al instante de solicitar al departamento de UTECA para un mejor alcance en un futuro proyecto.
- Que el resultado de la tesis **“ANÁLISIS DE LAS TECNOLOGÍAS AXIS 2 Y METRO 2.0 PARA EL DESARROLLO DE SERVICIOS WEB SEGUROS EN JAVA, APLICADO AL MÓDULO DE INTEGRACIÓN Y REPORTE DEL SISTEMA DE EVALUACIÓN DOCENTE DE LA UNACH”** sirva como ejemplo para futuras aplicaciones con seguridad.
- Para sistemas como la presente investigación se recomienda crear Servicios Web con la tecnología Axis 2 ya que posee un menor número de errores de seguridad y utilizar el servidor Apache Tomcat al tener más compatibilidad con la misma.
- Se recomienda investigar a futuro sobre los demás mecanismos de seguridad que tiene la tecnología de Servicio Web Axis 2.
- Se invita para futuras investigaciones realizar un análisis de la Tecnología Axis 2 con respecto a otros factores como es el caso de rendimiento y productividad.

BIBLIOGRAFIA

- Apache. (21 de 10 de 2016). *Welcome to Apache Axis2/Java*. Obtenido de <http://axis.apache.org/axis2/java/core/>
- API, E. M. (17 de 09 de 2015). *Disponibilidad de características del servicio Web de API de Exchange y de la API administrada de EWS*. Obtenido de [https://msdn.microsoft.com/es-es/library/office/dn602479\(v=exchg.150\).aspx](https://msdn.microsoft.com/es-es/library/office/dn602479(v=exchg.150).aspx)
- Axis2, A. (12 de 09 de 2016). *Apache Axis2/Java - Next Generation Web Services*. Obtenido de Apache Axis2/Java - Next Generation Web Services: <http://axis.apache.org/axis2/java/core/>
- Beck, K. (12 de 05 de 2016). *PROGRAMACION EXTREMA XP* . Obtenido de http://ingenieriadesoftware.mex.tl/52753_XP---Extreme-Programing.html
- BIJOY, C. (17 de 06 de 2014). *CoderPanda*. Obtenido de CoderPanda: <http://www.coderpanda.com/jax-ws-tutorial/>
- Binding, J. A. (20 de 03 de 2013). *ORACLE*. Obtenido de <http://www.oracle.com/technetwork/articles/javase/index-140168.html>
- Bomb, X. (16 de 04 de 2015). *SOAPUI by SMART BEAR*. Obtenido de <https://www.soapui.org/security-testing/security-scans/xml-bomb.html>
- Canut, C. G. (11 de 06 de 2012). *Desarrollos de proyectos con tecnología java*. Obtenido de <http://www3.uji.es/~belfern/pdf/libroJavaConTapa.pdf>
- Casas, P. (19 de 11 de 2015). *Seguridad en Cómputo*. Obtenido de El Triángulo de la Seguridad: <http://blogs.acatlan.unam.mx/lasc/2015/11/19/el-triangulo-de-la-seguridad/>
- Center, I. K. (28 de 02 de 2015). *IBM* . Obtenido de https://www.ibm.com/support/knowledgecenter/es/SSKM8N_8.0.0/com.ibm.etools.mft.doc/ac55910_.htm
- Committee, O. U. (01 de 04 de 2013). *UDDI*. Obtenido de XML.org.: <http://uddi.xml.org/>
- Commons, C. (23 de 06 de 2014). *KIOSKEA*. Obtenido de KIOSKEA: <http://es.ccm.net/contents/16-ataques-al-servidor-web>
- Culturacion, S. (18 de 04 de 2014). *Ataque de denegación de servicio*. Obtenido de Ataque de denegación de servicio: <http://culturacion.com/que-es-una-denegacion-de-servicio/>
- Dennis Sosnoski, A. C. (03 de 11 de 2012). *developerWorks®*. Obtenido de developerWorks®: <http://www.ibm.com/developerworks/library/j-jws9/>
- estandarwebservice, L. d. (25 de 07 de 2012). *Estandares de servicio web*. Obtenido de Estandares de servicio web: <http://bit.ly/1XQOfYH>
- Glance, J. S. (02 de 10 de 2015). *Oracle java J2SE*. Obtenido de <http://www.oracle.com/technetwork/java/javase/overview/index.html>

- González, B. (07 de 07 de 2014). *SOAP*. Obtenido de (Simple Object Access Protocol): <http://www.desarrolloweb.com/articulos/1557.php>
- Grehan, R. (17 de 05 de 2012). *Three open source Web service testing tools get high marks*. Obtenido de <http://www.infoworld.com/article/2663940/application-development/three-open-source-web-service-testing-tools-get-high-marks.html?page=2>
- IBM, W. S. (2013). *Servicio Web*. Quito - Ecuador: IBM Developers. Obtenido de <http://www.ondemand.es/actualidad/servicios-web/>
- InfoWorld, R. G. (11 de 07 de 2012). *Three open source Web service testing tools get high marks*. Obtenido de Three open source Web service testing tools get high marks: <http://www.infoworld.com/article/2663940/application-development/three-open-source-web-service-testing-tools-get-high-marks.html?page=2>
- Injection, S. (15 de 04 de 2015). *SOAPUI by SMART BEAR*. Obtenido de <https://www.soapui.org/security-testing/security-scans/sql-injection.html>
- Injection, X. (18 de 04 de 2015). *SOAPUI by SMART BEAR*. Obtenido de <https://www.soapui.org/security-testing/security-scans/xpath-injection.html>
- Java. (12 de 05 de 2016). *Conozca más sobre la tecnología Java*. Obtenido de <https://www.java.com/es/about/>
- Java. (20 de 02 de 2016). *Tecnología Java*. Obtenido de <https://www.java.com/es/about/>
- Java J2EE, O. (06 de 11 de 2015). *Compatibilidad con J2EE*. Obtenido de <https://docs.oracle.com/cd/E19528-01/820-0498/fxxwo/index.html>
- Java Platform, M. E. (12 de 04 de 2015). *Oracle micro edition (java me)*. Obtenido de <http://www.oracle.com/technetwork/java/embedded/javame/index.html>
- java.net, M. 2. (10 de 12 de 2009). *GlassFish » Metro*. Obtenido de GlassFish » Metro: <https://metro.java.net/2.0/>
- M. Piattini, C. G.-M. (15 de 01 de 2012). *Seguridad en Servicios Web*. Obtenido de Departamento de Informática - Universidad de Castilla-La Mancha, España.: <https://goo.gl/5i4BY7>
- Meredith, G. (15 de 03 de 2012). *Web Services Description Language* . Obtenido de (WSDL) 1.1: <https://www.w3.org/TR/wsdl>
- Metro 2.0, G. (02 de 10 de 2012). *GlassFish » Metro 2.0*. Obtenido de <https://metro.java.net/2.0/>
- muyiyangfeng. (04 de 03 de 2014). *Programmer share*. Obtenido de Programmer share: <http://www.programmershare.com/3702163/>
- Newcomer, E. (09 de 07 de 2012). *Introducción a los servicios web*. Obtenido de Microsoft .Net: <http://geneura.ugr.es/~jmerelo/ws/>
- Oracle, J. (25 de 08 de 2016). *Qué es Java*. Obtenido de https://www.java.com/es/about/whatis_java.jsp

- Peris Soler, N. B. (08 de 11 de 2014). *Universidad Politécnica de Valencia*. Obtenido de Facultad de Informática: <https://goo.gl/3iaiD8>
- Puebla, I. G. (28 de 12 de 2012). *Adictos al trabajo*. Obtenido de Adictos al trabajo: <https://www.adictosaltrabajo.com/tutoriales/introduccion-soap-ui/>
- Ramos, K. A. (16 de 11 de 2014). *DesarrolloWeb.com*. Obtenido de DesarrolloWeb.com: <http://www.desarrolloweb.com/articulos/definicion-y-a-quien-afecta-croos-site-scripting.html>
- REBERTO. (08 de 10 de 2012). *ArgentinaSecurity*. Obtenido de ArgentinaSecurity: <http://argentinasec.blogspot.com/2008/07/ntroduccion-al-xpath-injection-by.html>
- Rouse, M. (20 de 11 de 2012). *Ataque de denegación de servicio (DDoS)*. Obtenido de Ataque de denegación de servicio (DDoS): <http://searchdatacenter.techtarget.com/es/definicion/Ataque-de-denegacion-de-servicio>
- S.R.L, M. (03 de 09 de 2012). *MAYPUN SOLUCIONES INFORMÁTICAS*. Obtenido de MAYPUN SOLUCIONES INFORMÁTICAS: <http://maypun.blogspot.com/2009/09/apache-axis2.html>
- Scripting, C.-s. (12 de 20 de 2015). *SOAPUI by SMART BEAR*. Obtenido de <https://www.soapui.org/security-testing/security-scans/cross-site-scripting.html>
- Security, P. G. (18 de 05 de 2015). *You Do Not Have To Be A Genius To*. Obtenido de Master Open Source Testing: <http://www.pushtotest.com/products-comparison>
- SMART, S. B. (25 de 01 de 2016). *Build Better | Test Smarter*. Obtenido de <https://www.soapui.org/>
- SmartBear. (20 de 10 de 2015). *Tipos no válidos*. Obtenido de Tipos no válidos: <https://www.soapui.org/security-testing/security-scans/invalid-types.html#1-Introduction>
- SmartBear, B. X. (12 de 11 de 2015). *Bomba XML*. Obtenido de Bomba XML: <https://www.soapui.org/security-testing/security-scans/xml-bomb.html#1-Introduction>
- SmartBear, S. (25 de 02 de 2016). *XML con formato incorrecto*. Obtenido de XML con formato incorrecto: <https://www.soapui.org/security-testing/security-scans/malformed-xml.html#1-Introduction>
- Sosnoski, S. S. (03 de 08 de 2012). *Servicios web Java: El alto costo de (WS-)Security*. Obtenido de <http://www.ibm.com/developerworks/ssa/library/j-jws6/>
- Sun, J. (26 de 03 de 2016). *Java Magazine*. Obtenido de <http://java.sun.com>
- Sysmatec, C. (12 de 03 de 2016). *Centro de información de seguridad de sitios web, SSL y TLS*. Obtenido de <https://goo.gl/Q14zVO>
- TechNet, M. (01 de 02 de 2016). *Inyección de código SQL*. Obtenido de Inyección de código SQL: [https://technet.microsoft.com/es-es/library/ms161953\(v=sql.105\).aspx](https://technet.microsoft.com/es-es/library/ms161953(v=sql.105).aspx)

- technology, w. (22 de 05 de 2016). *FULLY BRANDED E-COMM WEBSITES*. Obtenido de <http://www.webinjected.com/>
- Tecnicomio. (21 de 02 de 2013). *Generando clientes de servicios web JAX-WS desde Java*. Obtenido de <https://eduvitoriatecnicomio.wordpress.com/2013/02/21/generando-clientes-de-servicios-web-jax-ws-desde-java/>
- Types, I. (10 de 08 de 2015). *SOAPUI by SMART BEAR*. Obtenido de <https://www.soapui.org/security-testing/security-scans/invalid-types.html>
- W3C. (23 de 10 de 2012). *Servicios Web de normalización conceptual con SOAP*. Obtenido de Servicios Web de normalización conceptual con SOAP: <https://www.w3.org/2002/ws/>
- webservice, S. W. (20 de 09 de 2011). *Introduccion a los servicios web*. Obtenido de Introduccion a los servicios web: <http://bit.ly/1T88n9v>
- Working, w. W. (11 de 02 de 2016). *ECURED*. Obtenido de conocimientos con todos y para todos: https://www.ecured.cu/Servicios_Web
- WSIT, j. (02 de 11 de 2013). *GlassFish » Metro » WSIT*. Obtenido de GlassFish » Metro » WSIT: <https://wsit.java.net/>
- XML, M. (22 de 05 de 2015). *SOAPUI by SMART BEAR*. Obtenido de <https://www.soapui.org/security-testing/security-scans/malformed-xml.html>

ANEXOS

ANEXO 1: CLASE

```
package evaluacion.rnegocio.clases;

public class Carrera {

    private int id;
    private Facultad facultad;
    private String nombre;
    private String descripcion;

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public Facultad getFacultad() {
        return facultad;
    }

    public void setFacultad(Facultad facultad) {
        this.facultad = facultad;
    }

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public String getDescripcion() {
        return descripcion;
    }

    public void setDescripcion(String descripcion) {
        this.descripcion = descripcion;
    }
}
```

ANEXO 2: FUNCION

```
package evaluacion.rnegocio.funciones;

import evaluacion.accesodatos.*;
import evaluacion.rnegocio.clases.Carrera;
import java.sql.SQLException;
import java.util.ArrayList;

public class FCarrera {

    public static boolean insertar(Carrera obj) throws Exception {
        boolean band = false;
        String sql = "insert into public.carrera values (?, ?, ?, ?)";
        ArrayList<Parametro> lstpar = new ArrayList<Parametro>();

//campos con referencias
        lstpar.add(new Parametro(2, obj.getFacultad().getId()));

//campos sin referencias
        lstpar.add(new Parametro(1, obj.getId()));
        lstpar.add(new Parametro(3, obj.getNombre()));
        lstpar.add(new Parametro(4, obj.getDescripcion()));
        try {
            band = AccesoDatos.ejecutaComando1(sql, lstpar);
        } catch (Exception ex) {
            throw ex;
        }
        return band;
    }

    public static boolean modificar(Carrera obj) throws Exception {
        boolean band = false;
        String sql = "update public.carrera set id=?,id_facultad=?,nombre=?,descripcion=?
where id=? ";
        ArrayList<Parametro> lstpar = new ArrayList<Parametro>();

//campos con referencias
        lstpar.add(new Parametro(2, obj.getFacultad().getId()));

//campos sin referencias
        lstpar.add(new Parametro(1, obj.getId()));
        lstpar.add(new Parametro(5, obj.getId()));
        lstpar.add(new Parametro(3, obj.getNombre()));
        lstpar.add(new Parametro(4, obj.getDescripcion()));
        try {
            band = AccesoDatos.ejecutaComando1(sql, lstpar);
        } catch (Exception ex) {
            throw ex;
        }
        return band;
    }
}
```

```

public static boolean eliminar(Carrera obj) throws Exception {
    boolean band = false;
    String sql = "delete from public.carrera where id=? ";
    ArrayList<Parametro> lstpar = new ArrayList<Parametro>();

//campos con referencias
//campos sin referencias
    lstpar.add(new Parametro(1, obj.getId()));
    try {
        band = AccesoDatos.ejecutaComando1(sql, lstpar);
    } catch (Exception ex) {
        throw ex;
    }
    return band;
}

public static Carrera obtener(int pid) throws Exception {
    Carrera miCarrera = null;
    try {
        String sql = "select id,id_facultad,nombre,descripcion from public.carrera where id=?
";
        ArrayList<Parametro> lstpar = new ArrayList<Parametro>();
        lstpar.add(new Parametro(1, pid));
        ConjuntoResultado rs = AccesoDatos.ejecutaQuery(sql, lstpar);
        ArrayList<Carrera> lst = llenarCarreras(rs);
        for (Carrera c : lst) {
            miCarrera = c;
        }

    } catch (Exception ex) {
        throw ex;
    }
    return miCarrera;
}

public static ArrayList<Carrera> obtener() throws Exception {
    ArrayList<Carrera> lst = new ArrayList<>();
    try {
        String sql = "select id,id_facultad,nombre,descripcion from public.carrera; ";
        ConjuntoResultado rs = AccesoDatos.ejecutaQuery(sql);
        lst = llenarCarreras(rs);

    } catch (Exception ex) {
        throw ex;
    }
    return lst;
}

private static ArrayList<Carrera> llenarCarreras(ConjuntoResultado cr) throws Exception
{

```

```

        ArrayList<Carrera> lst = new ArrayList<Carrera>();
        Carrera obj = null;
        try {
            while (cr.next()) {
                obj = new Carrera();

//campos con referencias
                obj.setFacultad(FFacultad.obtener(cr.getInt(2)));

//campos sin referencias
                obj.setId(cr.getInt(1));
                obj.setNombre(cr.getString(3));
                obj.setDescripcion(cr.getString(4));
                lst.add(obj);

            }
        } catch (Exception ex) {
            throw ex;
        }
        return lst;
    }

    public static ArrayList<Carrera> obtenerCarreraDadoCodigoFacultad(int id_facultad)
    throws Exception {
        ArrayList<Carrera> lst = new ArrayList<Carrera>();
        try {
            String sql = "select * from carrera where id_facultad = " + id_facultad + "";
            ConjuntoResultado rs = AccesoDatos.ejecutaQuery(sql);
            lst = llenarCarreras(rs);
            rs = null;

        } catch (SQLException exConec) {
            throw new Exception(exConec.getMessage());
        }
        return lst;
    }
}

```

ANEXO 3: CONTROLADOR

```
package evaluacion.controladores;

import evaluacion.rnegocio.clases.Carrera;
import servicios_sicoa.ArrayOfFacultad;
import servicios_sicoa.GetCarrerasFacultad;
import servicios_sicoa.GetPeriodoVigenteCarrera;
import evaluacion.migracion.FConsumoSW;
import evaluacion.rnegocio.clases.Dictado_asignatura;
import evaluacion.rnegocio.clases.Facultad;
import evaluacion.rnegocio.clases.Periodo;
import evaluacion.rnegocio.funciones.FCarrera;
import evaluacion.rnegocio.funciones.FDictado_asignatura;
import evaluacion.rnegocio.funciones.FFacultad;
import evaluacion.rnegocio.funciones.FPeriodo;
import evaluacion.migracion.FMigracion;
import evaluacion.utilitarios.Util;
import static evaluacion.utilitarios.Util.addMensajeProperties;
import java.util.ArrayList;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.faces.application.FacesMessage;
import javax.faces.bean.ManagedBean;
import javax.faces.bean.ViewScoped;
import servicios_sicoa.ArrayOfCarrera;

@ManagedBean
@ViewScoped

public class migracionCarreraControlador {

    private String codModalidad = "";
    private String codFacu = "";
    private String codCarre = "";
    private ArrayList<Facultad> listaFac;
    private ArrayList<Carrera> listaCarr;
    private ArrayList<Periodo> listaPeri;
    private ArrayList<Dictado_asignatura> listaDictadosLocal;
    private int dictadoLocal[];
    private List<String[]> listaprogramasLocal;
    private String resultado;
    private int facultadSeleccionada;
    private int carreraSeleccionada;
    private int periodoSeleccionado;

    //vista migracion dictado
    private List<Dictado_asignatura> listaDictadosLocales;

    public List<Dictado_asignatura> getListaDictadosLocales() {
        return listaDictadosLocales;
    }
}
```

```

}

public void setListaDictadosLocales(List<Dictado_asignatura> listaDictadosLocales) {
    this.listaDictadosLocales = listaDictadosLocales;
}

public ArrayList<Dictado_asignatura> getListaDictadosLocal() {
    return listaDictadosLocal;
}

public void setListaDictadosLocal(ArrayList<Dictado_asignatura> listaDictadosLocal) {
    this.listaDictadosLocal = listaDictadosLocal;
}

public ArrayList<Periodo> getListaPeri() {
    return listaPeri;
}

public void setListaPeri(ArrayList<Periodo> listaPeri) {
    this.listaPeri = listaPeri;
}

public String getCodModalidad() {
    return codModalidad;
}

public void setCodModalidad(String codModalidad) {
    this.codModalidad = codModalidad;
}

public String getCodFacu() {
    return codFacu;
}

public void setCodFacu(String codFacu) {
    this.codFacu = codFacu;
}

public String getCodCarre() {
    return codCarre;
}

public void setCodCarre(String codCarre) {
    this.codCarre = codCarre;
}

public List<String[]> getListaprogramasLocal() {
    return listaprogramasLocal;
}

```

```

public void setListaprogramasLocal(List<String[]> listaprogramasLocal) {
    this.listaprogramasLocal = listaprogramasLocal;
}

public String getResultado() {
    return resultado;
}

public void setResultado(String resultado) {
    this.resultado = resultado;
}

public int getFacultadSeleccionada() {
    return facultadSeleccionada;
}

public void setFacultadSeleccionada(int facultadSeleccionada) {
    this.facultadSeleccionada = facultadSeleccionada;
}

public int getCarreraSeleccionada() {
    return carreraSeleccionada;
}

public void setCarreraSeleccionada(int carreraSeleccionada) {
    this.carreraSeleccionada = carreraSeleccionada;
}

public ArrayList<Facultad> getListaFac() {
    return listaFac;
}

public void setListaFac(ArrayList<Facultad> listaFac) {
    this.listaFac = listaFac;
}

public ArrayList<Carrera> getListaCarr() {
    return listaCarr;
}

public void setListaCarr(ArrayList<Carrera> listaCarr) {
    this.listaCarr = listaCarr;
}

public int getPeriodoSeleccionado() {
    return periodoSeleccionado;
}

public void setPeriodoSeleccionado(int periodoSeleccionado) {
    this.periodoSeleccionado = periodoSeleccionado;
}

```

```

public int[] getDictadoLocal() {
    return dictadoLocal;
}

public void setDictadoLocal(int[] dictadoLocal) {
    this.dictadoLocal = dictadoLocal;
}

public migracionCarreraControlador() {
    this.reinit();
}

private void reinit() {
    listaFac = new ArrayList<>();
    listaCarr = new ArrayList<>();
    listaPeri = new ArrayList<>();
    listaDictadosLocal = new ArrayList<>();
    //cargarProgramasMigrados();
    this.cargarFacultades();
    this.cargarCarreras();
    this.cargarPeriodos();
    this.cargarDictadoAsignatura();
    resultado = "";
}

public void cargarFacultades() {
    try {
        listaFac = FFacultad.obtener();
        System.out.println(listaFac.get(0).getNombre());
    } catch (Exception e) {
        Util.addErrorMessage("private void obtenerTodo dice:" + e.getMessage());
        Logger.getLogger(facultadControlador.class.getName()).log(Level.SEVERE, null,
e);
    }
}

public void cargarCarreras() {
    try {
        listaCarr = FCarrera.obtenerCarreraDadoCodigoFacultad(facultadSeleccionada);
    } catch (Exception e) {
        Logger.getLogger(facultadControlador.class.getName()).log(Level.SEVERE, null,
e);
    }
}

public void cargarPeriodos() {
    try {

```

```

        listaPeri = FPeriodo.obtenerxEstado();
    } catch (Exception e) {
        Logger.getLogger(facultadControlador.class.getName()).log(Level.SEVERE, null,
e);
    }
}

public int cargarDictadoAsignatura() {
    try {
        dictadoLocal = FMigracion.migrarDictadosAsignatura(carreraSeleccionada,
periodoSeleccionado);
    } catch (Exception e) {
        Logger.getLogger(facultadControlador.class.getName()).log(Level.SEVERE, null,
e);
    }
    return 0;
}
}

```

ANEXO 4: VISTA

```
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
    xmlns:ui="http://xmlns.jcp.org/jsf/facelets"
    xmlns:p="http://primefaces.org/ui"
    xmlns:h="http://xmlns.jcp.org/jsf/html"
    xmlns:f="http://xmlns.jcp.org/jsf/core"
    template="../../plantillas/principal.xhtml">
    <ui:define name="topMenu">
        <ui:include src="menu.xhtml" />
    </ui:define>

    <ui:define name="content">
        <p:panel class="ui-widget ui-widget-header" style="border: 0px"></p:panel>
        <p:panel id="pnlEvaluacion" style="padding: 0px; margin: 0px;">
            <h:form id="frmMigracion">
                <p:panelGrid id="pnlTrans" columns="2" >
                    <h:outputLabel for="idfacultad" value="FACULTAD" />

                    <p:selectOneMenu id="idfacultad"
value="#{migracionCarreraControlador.facultadSeleccionada}" >
                        <f:selectItem itemLabel="Seleccione la Facultad" itemValue="#{null}" />
                        <f:selectItems value="#{migracionCarreraControlador.listaFac}"
var="facultades" itemLabel="#{facultades.nombre}"
itemValue="#{facultades.id}"/>

                        <p:ajax event="change"
listener="#{migracionCarreraControlador.cargarCarreras()}"
update="idcarrera"/>
                    </p:selectOneMenu>

                    <h:outputLabel for="idcarrera" value="CARRERA" />
                    <p:selectOneMenu id="idcarrera"
value="#{migracionCarreraControlador.carreraSeleccionada}"
>
                        <f:selectItem itemLabel="Seleccione un Carrera" itemValue="" />
                        <f:selectItems value="#{migracionCarreraControlador.listaCarr}"
var="carreras"
itemLabel="#{carreras.nombre}" itemValue="#{carreras.id}"/>
                    </p:selectOneMenu>

                    <h:outputLabel for="idperiodo" value="PERIODO" />
                    <p:selectOneMenu id="idperiodo"
value="#{migracionCarreraControlador.periodoSeleccionado}"
>
                        <f:selectItem itemLabel="Seleccione un Periodo" itemValue="" />
                        <f:selectItems value="#{migracionCarreraControlador.listaPeri}"
var="periodo"
itemLabel="#{periodo.nombre}" itemValue="#{periodo.id}"/>
                    </p:selectOneMenu>
                </h:outputLabel for="idfacultad" value="FACULTAD" />
            </h:form id="frmMigracion">
        </p:panel id="pnlEvaluacion" style="padding: 0px; margin: 0px;">
    </ui:define name="content">
</ui:composition>
```

```

</p:selectOneMenu>

<f:facet name="footer">

    <p:commandButton id="btnMostrar" value="Migracion Carrera"

action="#{migracionCarreraControlador.cargarDictadoAsignatura()}"
        process="@form,@this"          update="pnlProgramasMigrados"
style="center">
    </p:commandButton>
    <!-- <p:commandButton id="btnGuardar" value="Migrar"
        icon="ui-icon-disk"          update="pnlResultado
pnlProgramasMigrados"
        onstart="PF('statusDialog').show()"
onsuccess="PF('statusDialog').hide()"
        onclick="return confirm('Asegúrese que la información en el
servidor IPEC se encuentre completa, este proceso puede durar varios segundos..') ">
    </p:commandButton>

    <p:commandButton
        icon="ui-icon-cancel"
        immediate="true"
        action="index.xhtml?faces-redirect=true"
        value="Cancelar" /> -->
</f:facet>
</p:panelGrid>

    <p:panelGrid id="pnlProgramasMigrados"          style="margin-top:10px"
columns="10">
    <f:facet name="header">
        Resultado
    </f:facet>
    <h:outputText          value="Resultado:
#{migracionCarreraControlador.dictadoLocal[0]==1?'Correcto':'Error'}" />
    <h:outputText          value="Estudiantes          Migrados:
#{migracionCarreraControlador.dictadoLocal[1]}" />
    <h:outputText          value="Docentes          Migrados:
#{migracionCarreraControlador.dictadoLocal[2]}" />
    <h:outputText          value="Asignaturas          Migradas:
#{migracionCarreraControlador.dictadoLocal[3]}" />
    <h:outputText          value="Niveles          Migrados:
#{migracionCarreraControlador.dictadoLocal[4]}" />
    <h:outputText          value="Paralelos          Migrados:
#{migracionCarreraControlador.dictadoLocal[5]}" />
    <h:outputText          value="Dictados          Migrados:
#{migracionCarreraControlador.dictadoLocal[6]}" />
    <h:outputText          value="AutoEvaluaciones          Migradas:
#{migracionCarreraControlador.dictadoLocal[7]}" />
    <h:outputText          value="CoEvaluaciones          Migradas:
#{migracionCarreraControlador.dictadoLocal[8]}" />

```

```
        <h:outputText                value="HeteroEvaluaciones
#{migracionCarreraControlador.dictadoLocal[9]}"/>
```

Migradas:

```
    </p:panelGrid>
  </h:form>
</p:panel>

</ui:define>
<ui:define name="dialogo">
  <!-- PROCESO DE CARGA -->
  <p:dialog widgetVar="statusDialog" modal="true" draggable="false" closable="false"
resizable="false" showHeader="false">
    <h:panelGrid columns="1" style="text-align: center">
      <p:outputLabel value="" />
      <p:graphicImage value="../../resources/imgs/ajaxloadingbar.gif"/>
    </h:panelGrid>
  </p:dialog>
</ui:define>
</ui:composition>
```

ANEXO 6: ACTA ENTREGA-RECEPCIÓN DE LOS SERVICIOS WEB



UNIVERSIDAD NACIONAL DE CHIMBORAZO

UNIDAD TÉCNICA DE CONTROL ACADÉMICO

Ext.: 1040 • 1041 • 1042

Libres por la Ciencia y el Saber

Riobamba, 14 de julio de 2016

Oficio No. 384-UTECA-UNACH-2016

Ingeniero
Danny Velasco
DIRECTOR DE LA CARRERA DE INGENIERÍA EN SISTEMAS Y COMPUTACIÓN
Presente.-

De mi consideración:

Reciba un respetuoso saludo, en atención al oficio No.274-CISYC-2016, me permito indicar que los requerimientos solicitados para el servicio web del sistema de Evaluación Docente de la Universidad Nacional de Chimborazo, han sido procesados y publicados.

Lo que comunico para los fines pertinentes.

Cordialmente;

Ing. Pedro Orozco Quiroz, Mg.
FUNCIONARIO UTECA – ADMINISTRADOR DE SISTEMAS

Adj. Lo indicado.
cc. Archivo.

Elaborado por: Ing. Carlos Padilla.



15/07/16
10:45

Campus Norte "Edison Riera R."
Avda. Antonio José de Sucre, Km. 1.5 Vía a Guano
Teléfonos: (593-3) 37 30 880- ext. 3000

Campus "La Dolorosa"
Avda. Eloy Alfaro y 10 de Agosto.
Teléfonos: (593-3) 37 30 910 - ext. 3001

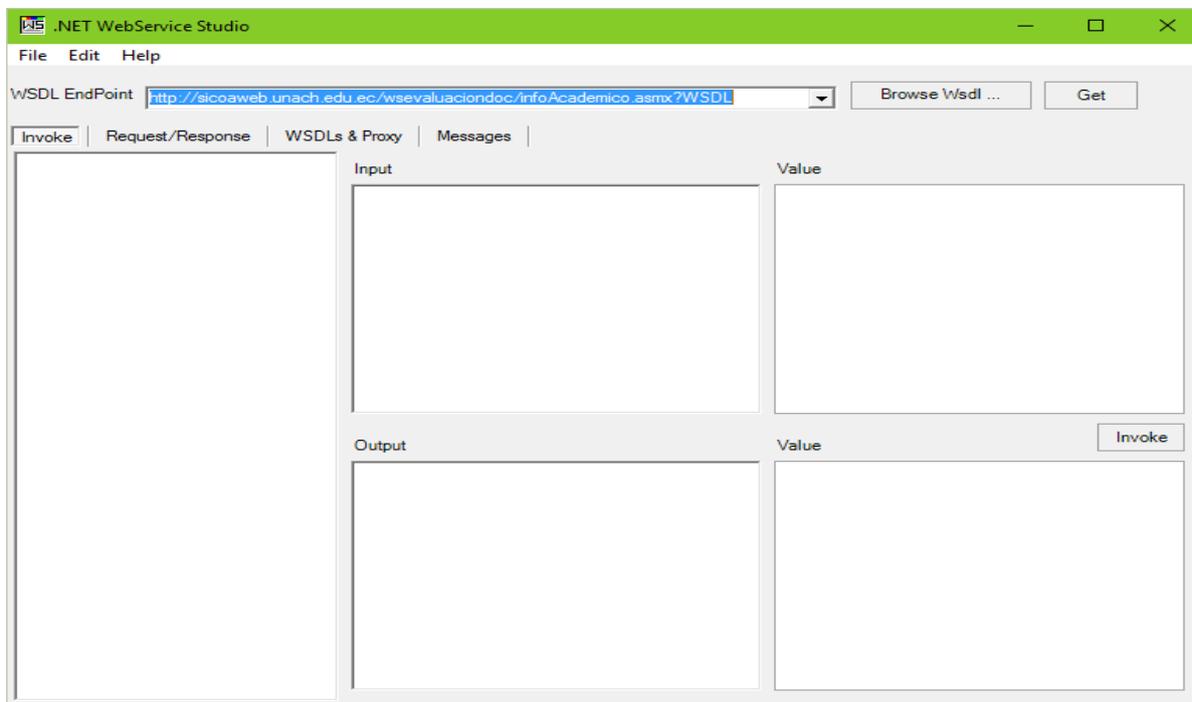
Campus Centro
Duchicela 17-75 y Princesa Toa
Teléfonos: (593-3) 37 30 880- ext. 3500

Campus Guano
Parroquia La Matriz, Barrio San Roque
vía a Asaco

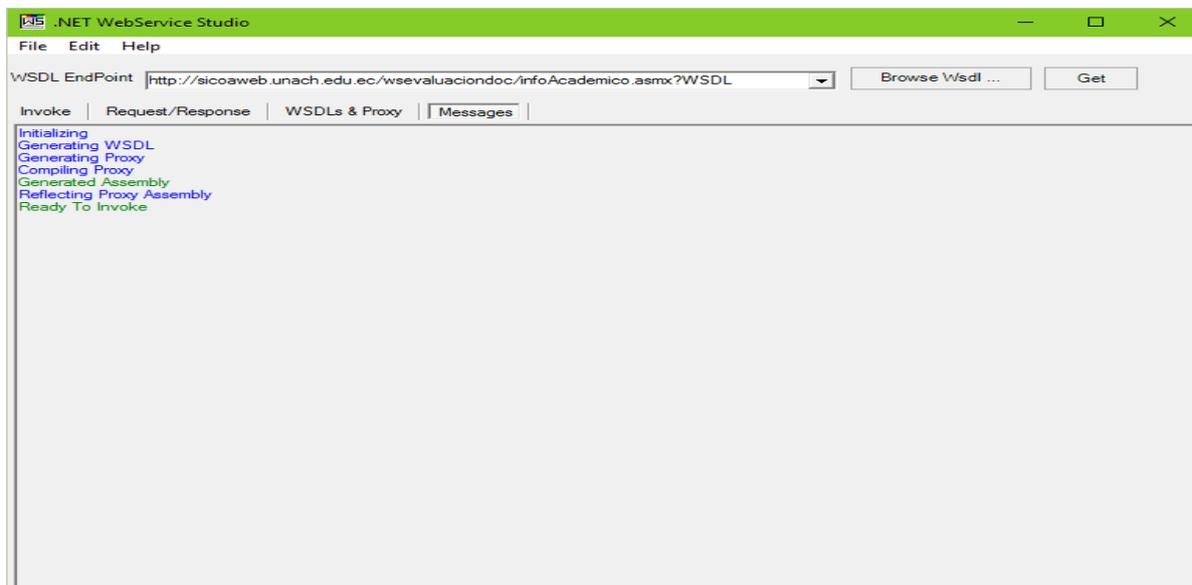
ANEXO 7: HERRAMIENTA .NET WEB SERVICE STUDIO

La siguiente herramienta se utilizó para realizar la comprobación de los servicios web entregados por el departamento de UTECA.

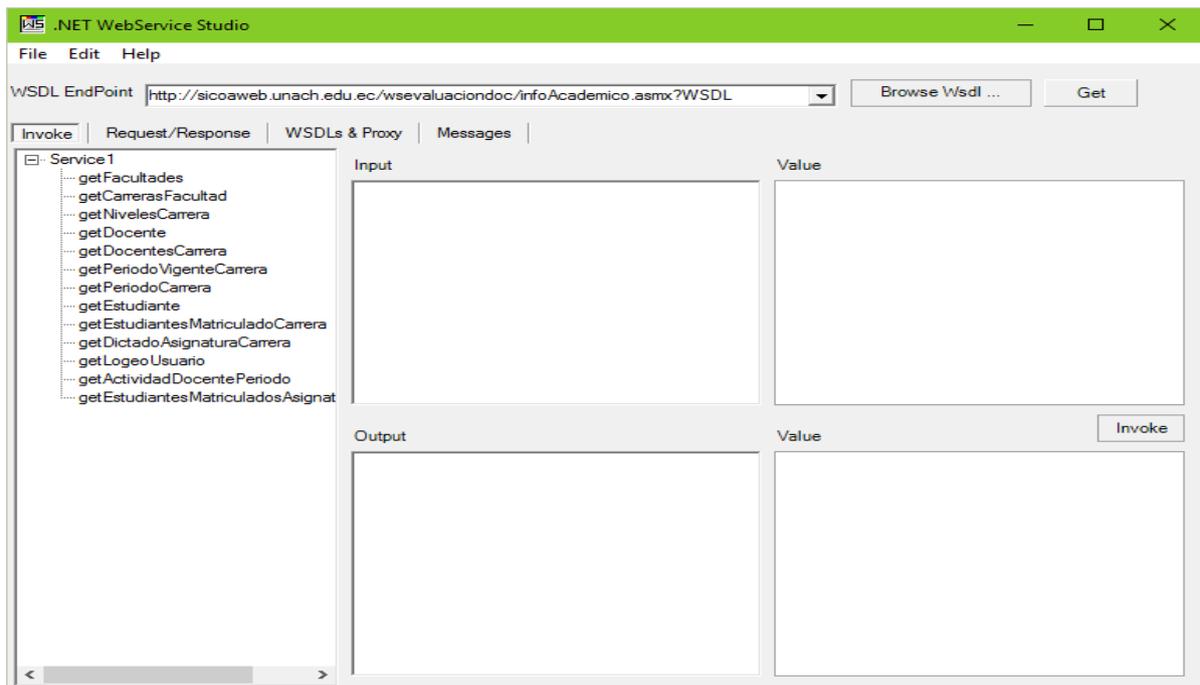
Paso 1: Cargamos el enlace donde están alojados nuestros servicios web como muestra la siguiente ilustración a continuación



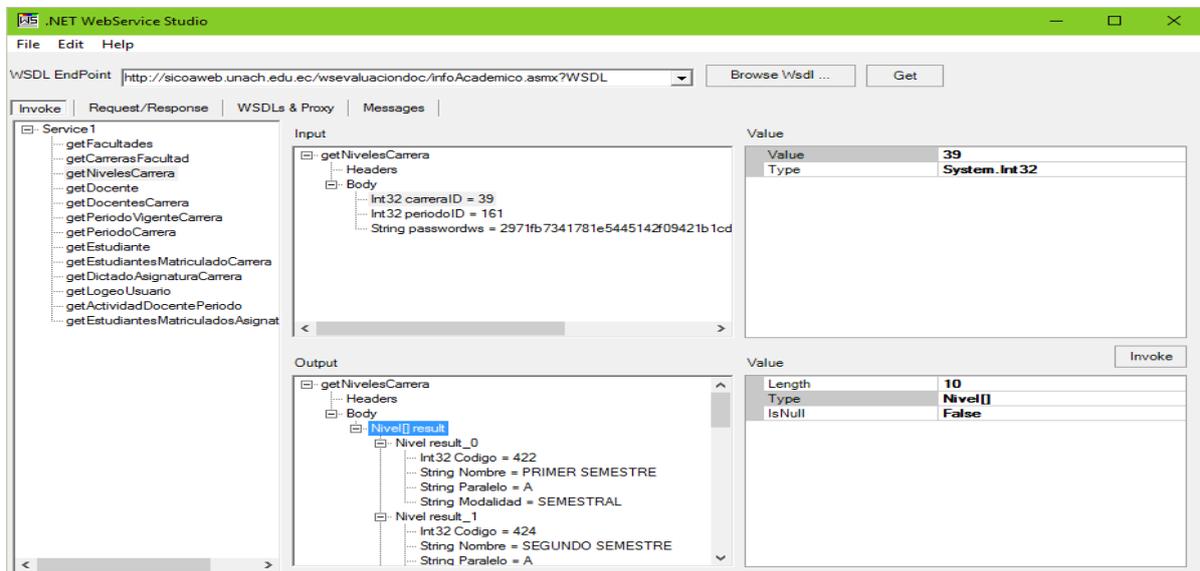
Paso 2: Procedemos a dar clic en el botón GET y visualizamos lo siguiente como muestra la siguiente ilustración.



Paso 3: Visualizamos nuestros métodos del servicio web **wsevaluacióndoc** y procedemos a comprobar cada una de nuestros métodos.



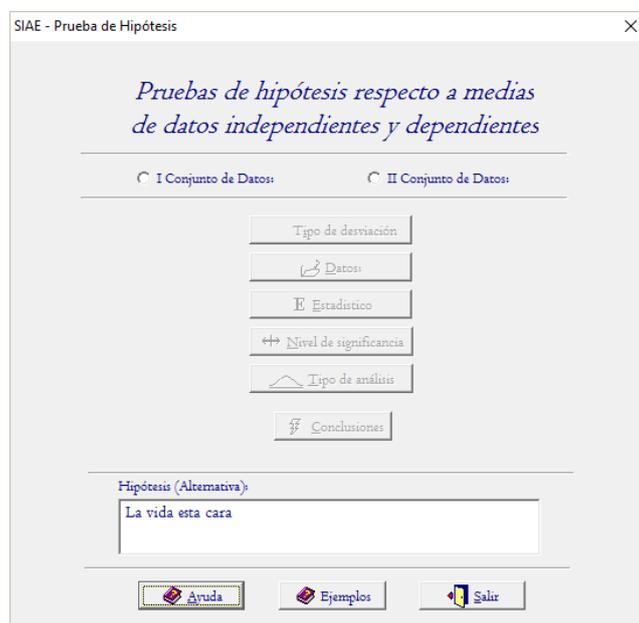
Paso 4: A continuación, se detalla la comprobación de un método de los servicios web que se va utilizar para la Integración del sistema de evaluación docente con el sistema SICOA.



ANEXO 8: HERRAMIENTA DE ANÁLISIS ESTADÍSTICO SIAE

La herramienta SIAE se utilizó para comprobar la hipótesis utilizando la prueba Z

Paso 1: Se procede a seleccionar de acuerdo al tipo de estudio a realizar, comparación de una variable o dos variables.



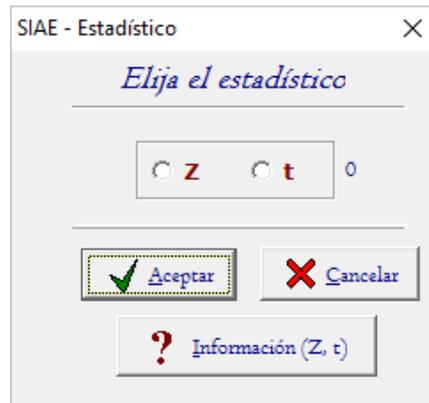
Paso 2: A continuación, se seleccionó el tipo de desviación a utilizar (típica o estándar).



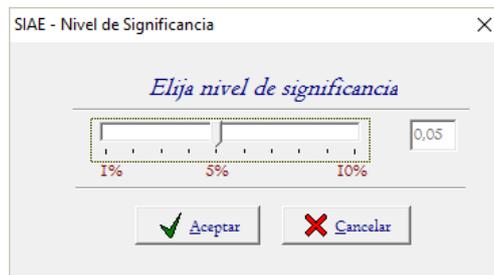
Paso 3: Procedemos a cargar los datos de nuestras dos variables.



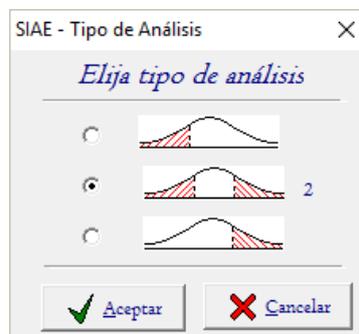
Paso 4: Seleccionamos el tipo de prueba a realizar (Prueba Z o Prueba T).



Paso 5: Seleccionamos el nivel de significancia.



Paso 6: Seleccionamos el tipo de análisis a estudiar de acuerdo al sistema de hipótesis planteado.



Paso 7: Finalmente tendremos nuestras conclusiones.

ANEXO 9: CREACIÓN DE SERVICIO WEB CON SEGURIDAD

```
package evaluacion.service;
import classes.Evaluacion;
import classes.Facultad;
import funciones.FEvaluacion;
import funciones.FFacultad;
import java.util.ArrayList;
public class FacultadWS {
    public Facultad buscar(int id) throws Exception{
        return FFacultad.obtener(id);
    }
    public ArrayList<Facultad> listado() throws Exception{
        ArrayList<Facultad> listfa = FFacultad.obtener();
        return listfa;
    }
    public Boolean ingresar(Facultad obj) throws Exception{
        return FFacultad.insertar(obj);
    }
    public ArrayList<Evaluacion> listadoEstudiantes() throws Exception{
        ArrayList<Evaluacion> listadoE = FEvaluacion.listaEvaluacion();
        return listadoE;
    }
}
```

ANEXO 10: CREACIÓN DE MÉTODO DE SEGURIDAD EN JAVA

```
package evaluacion.service.security;
import org.apache.ws.security.WSPasswordCallback;
import javax.security.auth.callback.Callback;
import javax.security.auth.callback.CallbackHandler;
import javax.security.auth.callback.UnsupportedCallbackException;
import java.io.IOException;
public class PWCBHandler implements CallbackHandler
{
    public void handle(Callback[] callbacks) throws IOException,
UnsupportedCallbackException {
        for (int i = 0; i < callbacks.length; i++) {
            WSPasswordCallback pwcb = (WSPasswordCallback)callbacks[i];
            String id = pwcb.getIdentifier();
            switch (pwcb.getUsage()) {
                case WSPasswordCallback.USERNAME_TOKEN_UNKNOWN:
                    // used when plaintext password in message
                    if (!"earthinfo".equals(id) || !"quakes".equals(pwcb.getPassword())) {
                        throw new UnsupportedCallbackException(callbacks[i], "check failed");
                    }
                    break;
                case WSPasswordCallback.USERNAME_TOKEN:
                    // used when hashed password in message
                    if ("earthinfo".equals(id)) {
                        pwcb.setPassword("quakes");
                    }
                    break;

                case WSPasswordCallback.DECRYPT:
                case WSPasswordCallback.SIGNATURE:

                    // used to retrieve password for private key
                    if ("serverkey".equals(id)) {
                        pwcb.setPassword("serverpass");
                    }
                    break;
            }
        }
    }
}
```

ANEXO 11: CREACIÓN DE POLÍTICAS DE SEGURIDAD

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Server policy for Username Token with plaintext password, sent from client
to server only. This differs from the client version only in that it includes
Rampart extensions for the the password callback class which are not needed for
the client. -->
<wsp:Policy wsu:Id="UsernameToken" xmlns:wsu=
  "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
  xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
  <wsp:ExactlyOne>
    <wsp>All>
      <sp:SupportingTokens
        xmlns:sp="http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702">
        <wsp:Policy>
          <sp:UsernameToken sp:IncludeToken=
            "http://docs.oasis-open.org/ws-sx/ws-
securitypolicy/200702/IncludeToken/AlwaysToRecipient"/>
          </wsp:Policy>
        </sp:SupportingTokens>
        <ramp:RampartConfig xmlns:ramp="http://ws.apache.org/rampart/policy">
        <ramp:passwordCallbackClass>
evaluation.service.security.PWCBHandler</ramp:passwordCallbackClass>
        </ramp:RampartConfig>
        </wsp>All>
      </wsp:ExactlyOne>
    </wsp:Policy>
  </wsp:Policy>
</wsp:Policy>
```

ANEXO 12: GUÍA DE INTEGRACIÓN DEL MÓDULO DE MIGRACIÓN CON EL SISTEMA SICOA.

Sistema SICOA de la UNACH

A continuación, se plantea los 2 escenarios a utilizar para realizar la integración del sistema de evaluación docente con el sistema SICOA, para lo cual se procede a explicar cada uno de los sistemas antes mencionado.

SISTEMA SICOA. - es un conjunto de servicios académicos dirigido a docentes y estudiantes de la Universidad Nacional de Chimborazo.

Podrás tener acceso de una manera fácil, rápida y segura a los siguientes servicios:

- Consultas de registros académicos
- Consultas de calificaciones
- Horarios de clases
- Generación de actas de calificaciones
- Ingreso de actas de calificaciones
- y otros servicios.

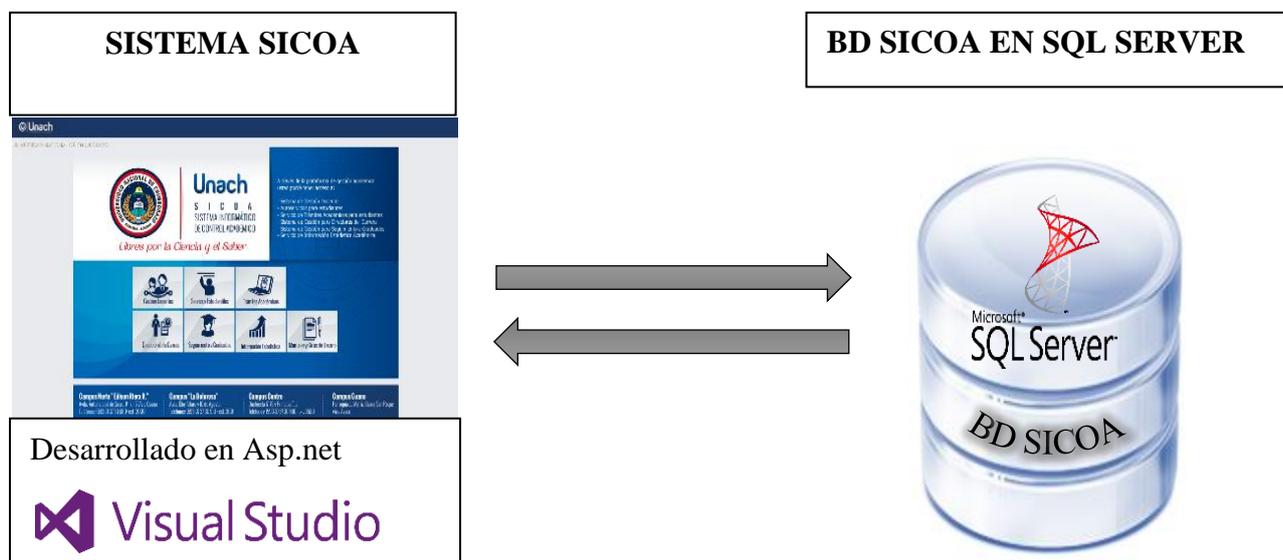


Ilustración 44: Sistema SICOA de la UNACH
Fuente: Kleber Bustan/Jorge Álvarez

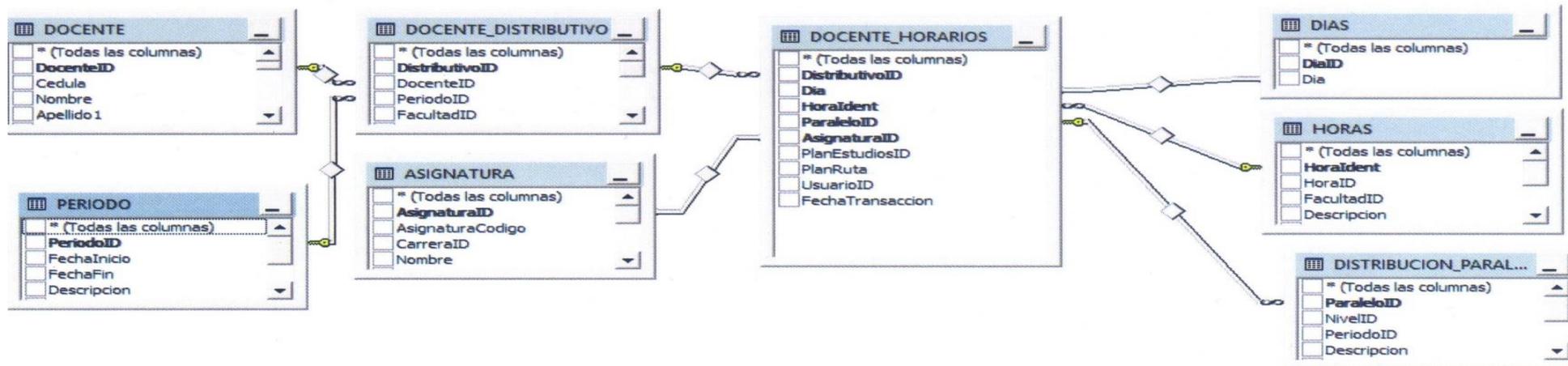


Ilustración 45: Base de datos SICOA Modelo Distributivos docentes
Fuente: Kléber Bustán/Jorge Álvarez

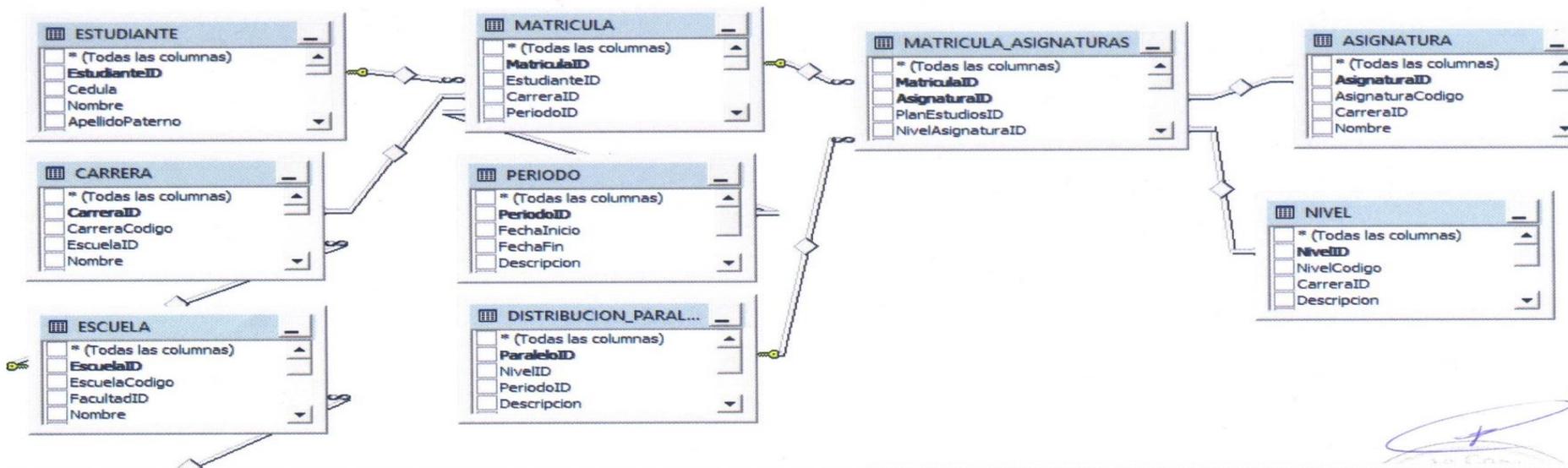


Ilustración 46: Base de datos SICOA Modelo Matriculas
Fuente: Kléber Bustán/Jorge Álvarez

Sistema de Evaluación Docente de la UNACH

SISTEMA DE EVALUACION DOCENTE. – es un conjunto de servicios de evaluaciones dirigido a Administrativos, Directores, Docentes y Estudiantes de la Universidad Nacional de Chimborazo.

Podrás tener acceso de una manera fácil, rápida y segura a los siguientes servicios:

- Realizar las evaluaciones correspondientes a periodo actual.
- Evaluaciones de Administrativos, Directores, Docentes y Estudiantes.
- Consultas de evaluaciones.
- Generación de reportes de evaluaciones.

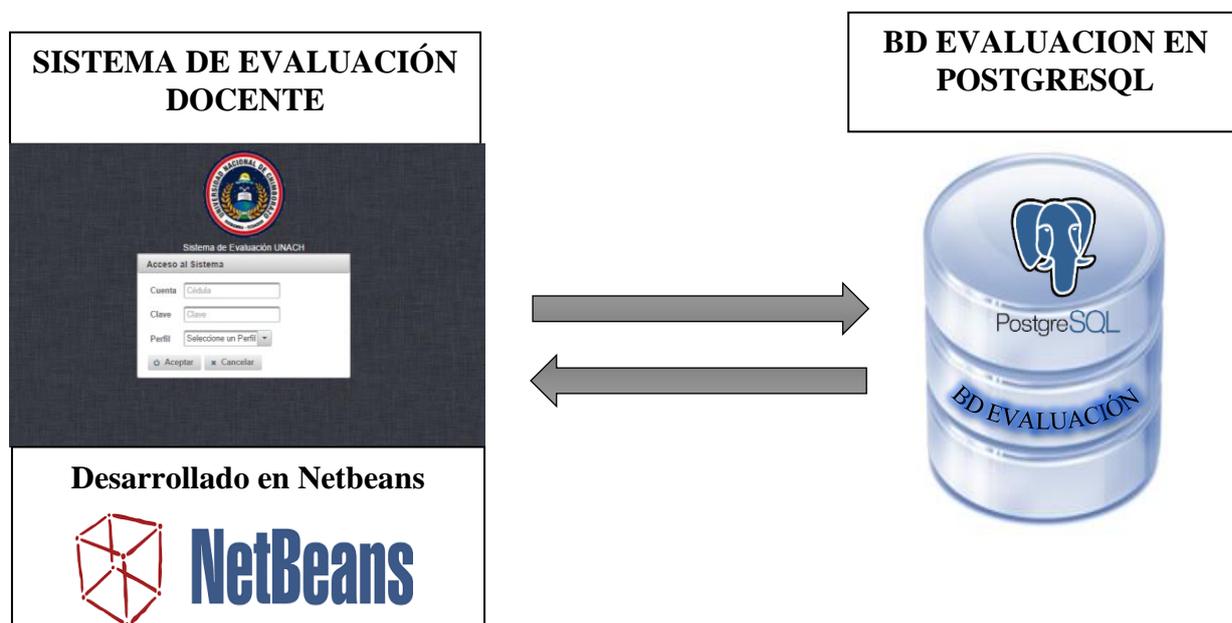


Ilustración 47: Sistema de Evaluación Docente UNACH
Fuente: Kléber Bustán/Jorge Álvarez

Integración de los sistemas SICOA y Sistema de Evaluación Docente de la UNACH.

A continuación, en la presente ilustración se muestra cómo se utiliza los servicios web para realizar la integración del sistema de evaluación docente con el sistema SICOA.

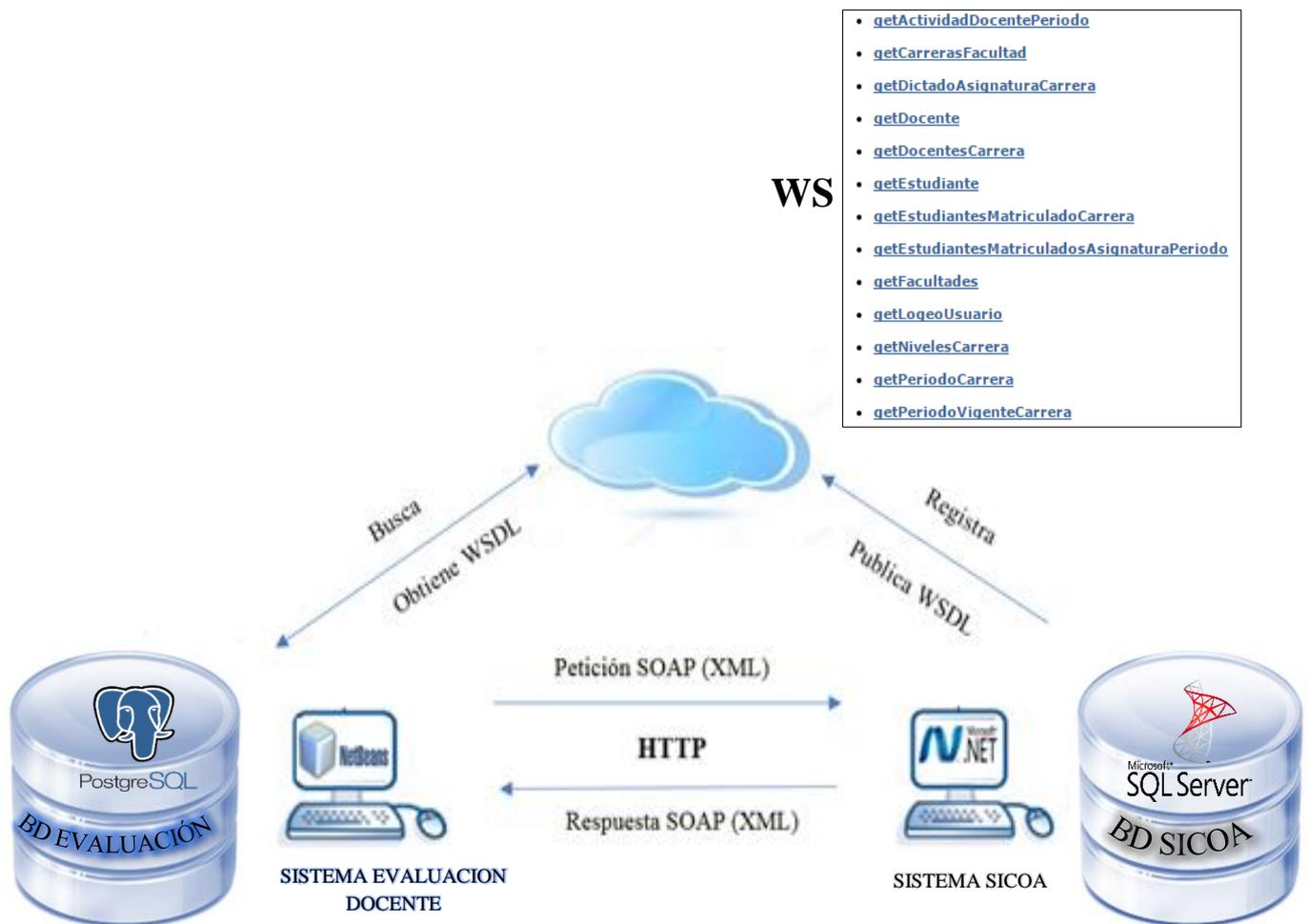


Ilustración 49: Integración del Sistema de Evaluación Docente con el Sistema SICOA-UNACH.

Fuente: Kleber Bustan/Jorge Álvarez

Servicios web utilizados para realizar la integración del sistema de evaluación docente con el sistema SICOA.

Los siguientes servicios web son utilizados para realizar la integración como muestra la ilustración 57, obtenido la url <http://sicoaweb.unach.edu.ec/wsevaluaciondoc/infoAcademico.asmx>.

The screenshot shows a web browser window displaying the service interface for 'Service1'. The page title is 'Service1' and the URL is 'sicoaweb.unach.edu.ec/wsevaluaciondoc/infoAcademico.asmx'. The main content area lists the following supported operations:

- [getActividadDocentePeriodo](#)
- [getCarrerasFacultad](#)
- [getDictadoAsignaturaCarrera](#)
- [getDocente](#)
- [getDocentesCarrera](#)
- [getEstudiante](#)
- [getEstudiantesMatriculadoCarrera](#)
- [getEstudiantesMatriculadosAsignaturaPeriodo](#)
- [getFacultades](#)
- [getLogeoUsuario](#)
- [getNivelesCarrera](#)
- [getPeriodoCarrera](#)
- [getPeriodoVigenteCarrera](#)

Below the list, there is a note: "This web service is using <http://tempuri.org/> as its default namespace. Recommendation: Change the default namespace before the XML Web service is made public." This is followed by explanatory text about namespaces and a code example in C#:

```
C#  
[WebService(Namespace="http://microsoft.com/webservices/")]  
public class MyWebService {  
    // implementation  
}
```

Ilustración 50: Servicio Web utilizado para la Integración
Fuente: Kléber Bustán/ Jorge Álvarez